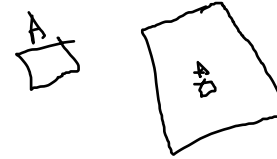


האיות האנחנו מחפשים



בעבר עד הרשתות שבאה והחליפו את מרבית
feature detectors
השתמשו בהם

בהנתן נקודה בתמונה, אני רוצה לבדוק האם הנקודה הספציפית היא מתאימה לנקודה אחרת בתמונה אחרת
בגדול, אם יש לי תמונה אני רוצה למצוא את הצורה התואמת בתמונה אחרת

Feature detectors

Lihi Zelnik-Manor, Computer Vision

Stereo-view geometry

- ▶ **Correspondence:**

Given a point in one image, how can I find the corresponding point in another image?

- ▶ **Camera geometry:**

Given corresponding points in two images, find camera intrinsic and extrinsic parameters

- ▶ **Scene geometry:**

Find coordinates of 3D point from its projection into two or multiple images.

Today

- ▶ **Local invariant features**
 - ▶ Motivation
 - ▶ Requirements, invariances
- ▶ **Keypoint localization**
 - ▶ Harris corner detector
 - ▶ Hessian detector
- ▶ **Scale invariant region selection**
 - ▶ Automatic scale selection
 - ▶ Laplacian-of-Gaussian detector
 - ▶ Difference-of-Gaussian detector

Today

- ▶ **Local invariant features**
 - ▶ **Motivation**
 - ▶ **Requirements, invariances**
- ▶ Keypoint localization
 - ▶ Harris corner detector
 - ▶ Hessian detector
- ▶ Scale invariant region selection
 - ▶ Automatic scale selection
 - ▶ Laplacian-of-Gaussian detector
 - ▶ Difference-of-Gaussian detector

Motivation

- ▶ Global representations have major limitations
- ▶ Instead, describe and match only local regions
- ▶ Increased robustness to

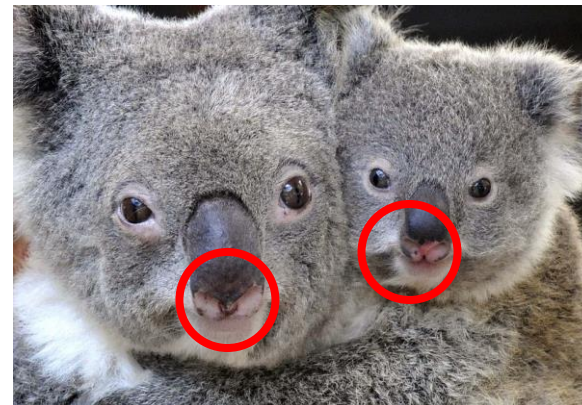
- ▶ Occlusions



- ▶ Articulation



- ▶ Intra-category variations



Application: Image matching



by [Diva Sian](#)



by [swashford](#)

Harder case

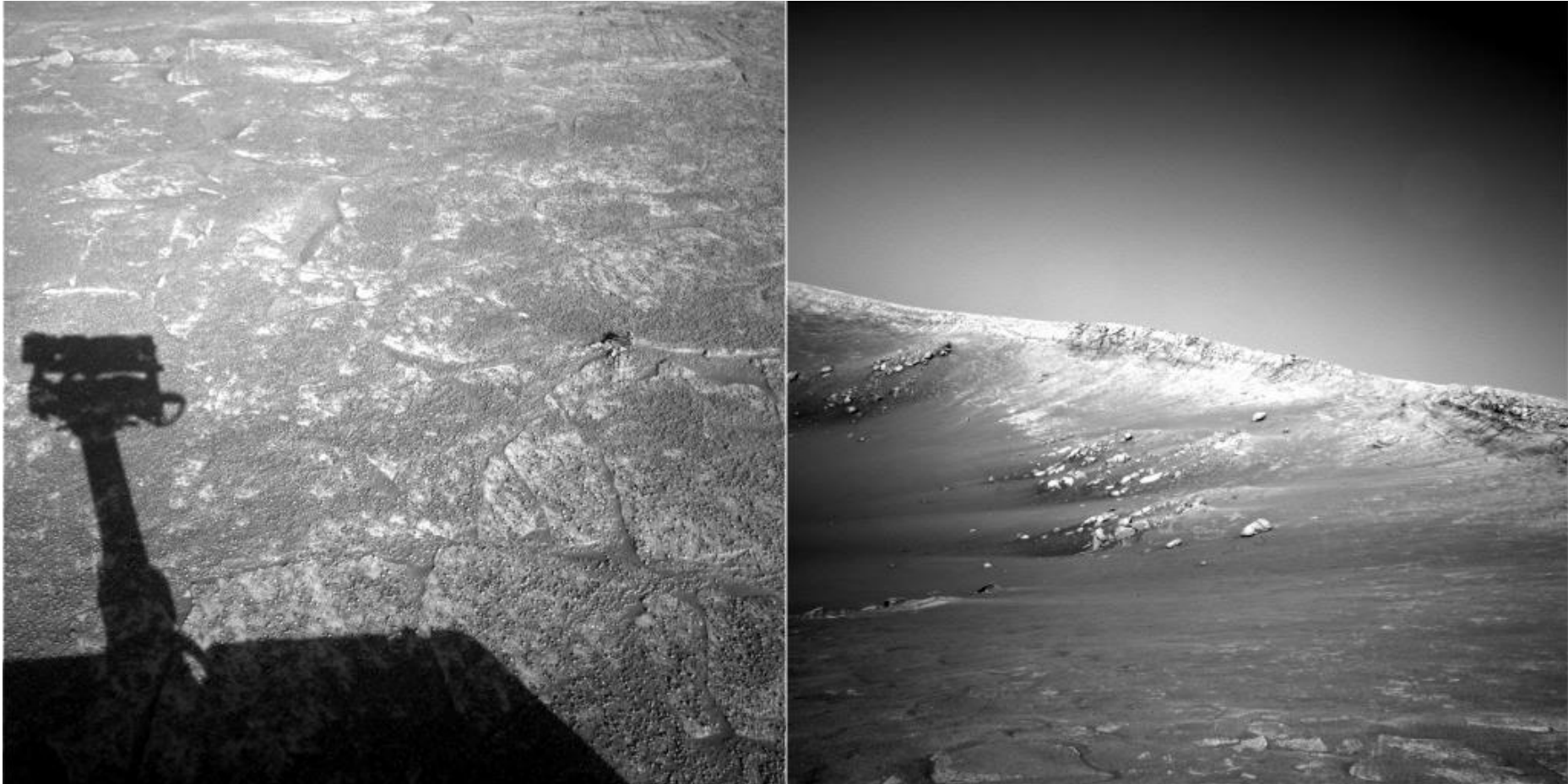


by [Diva Sian](#)



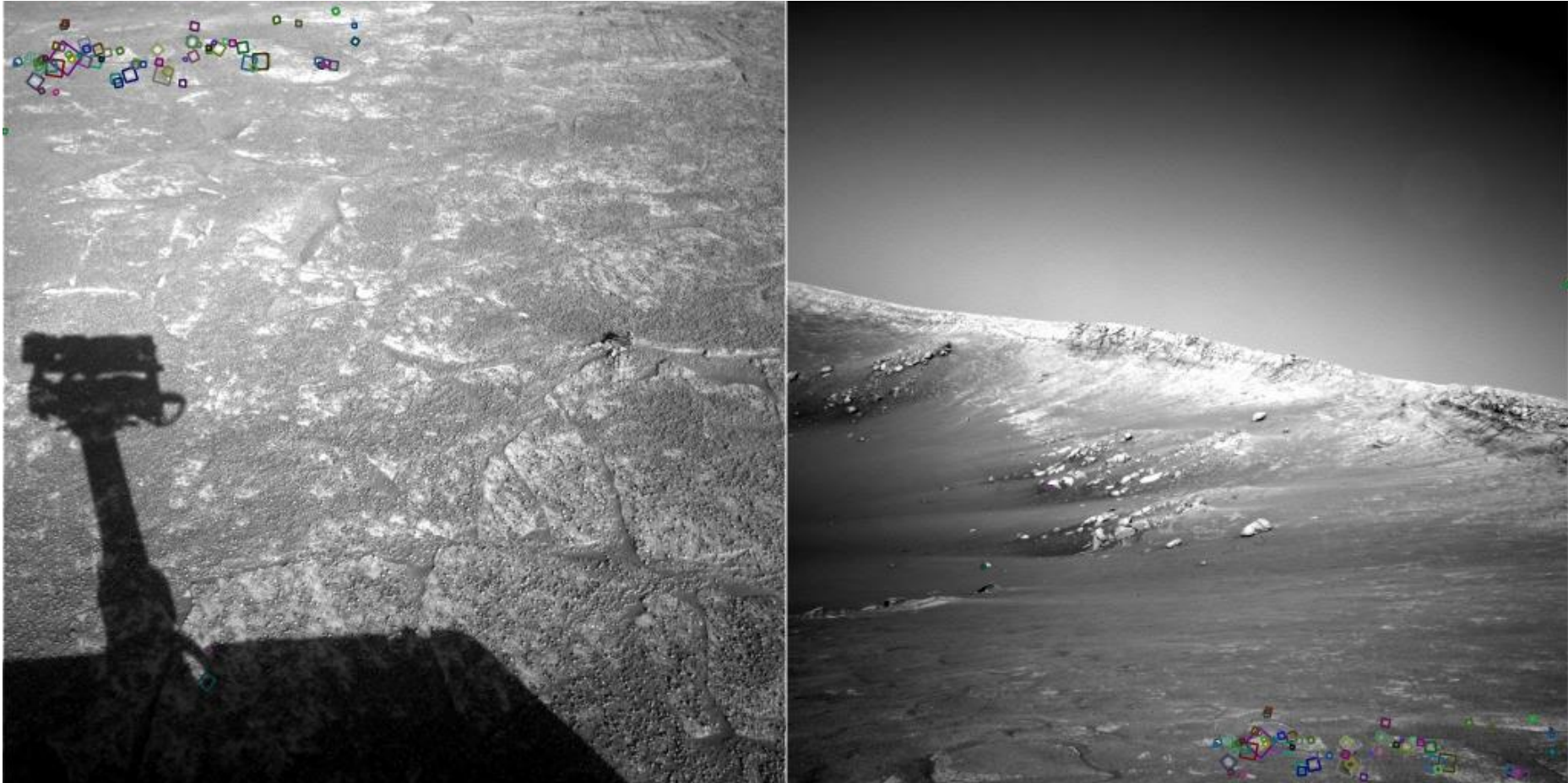
by [scgbt](#)

Harder still?



NASA Mars Rover images

Answer below (look for tiny colored squares...)



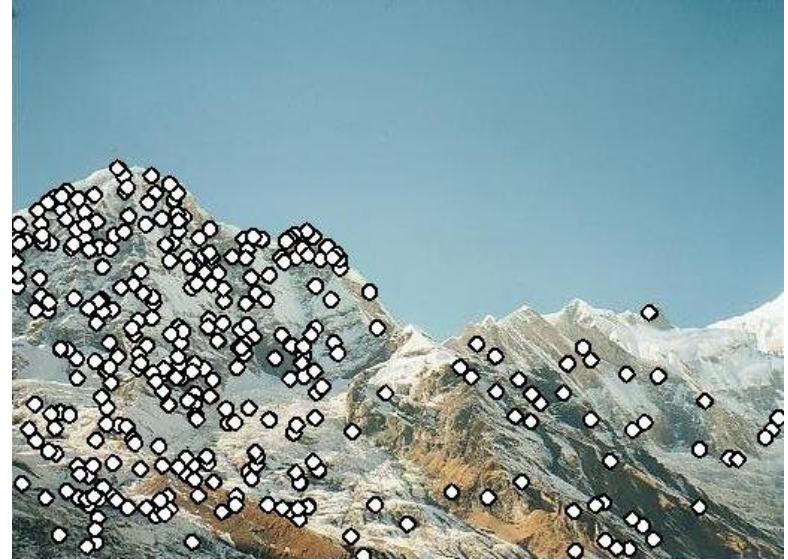
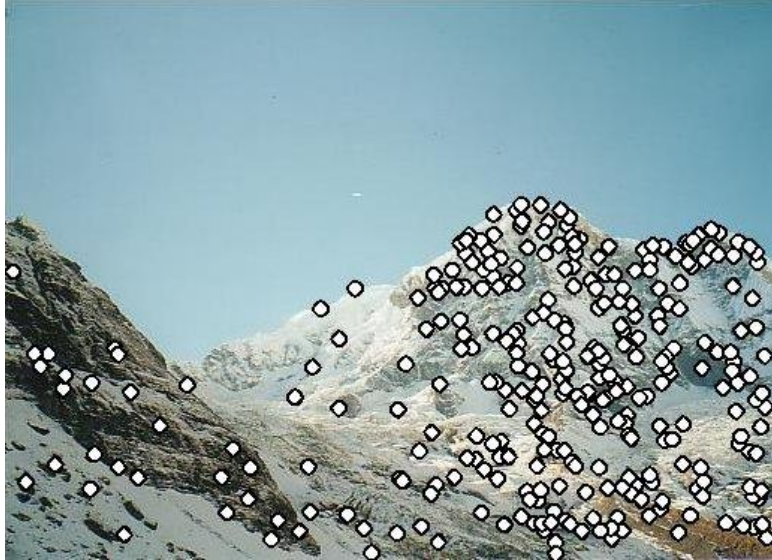
NASA Mars Rover images with SIFT feature matches. Figure by Noah Snavely

Application: Image stitching



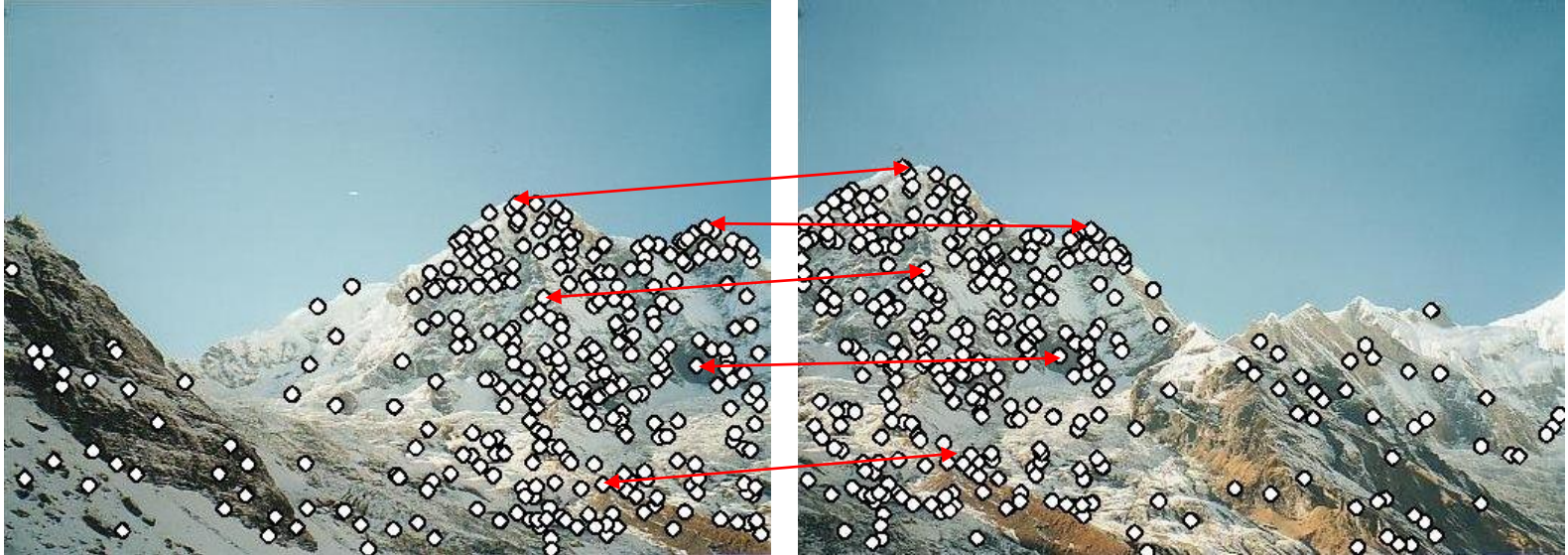
בניית תמונה אחת משתי תמונות לפי מאפיין מושתפים

Application: Image stitching



- ▶ Procedure:
 - ▶ Detect feature points in both images

Application: Image stitching



- ▶ Procedure:
 - ▶ Detect feature points in both images
 - ▶ Find corresponding pairs

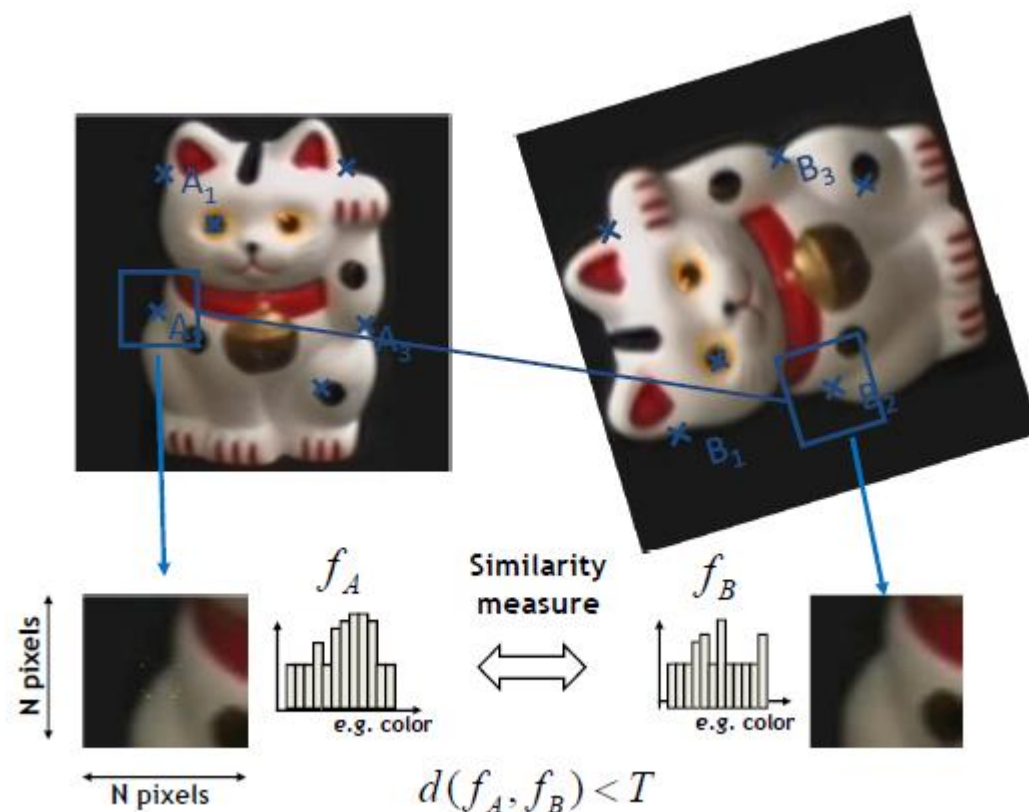
Application: Image stitching



- ▶ Procedure:
 - ▶ Detect feature points in both images
 - ▶ Find corresponding pairs
 - ▶ Use these pairs to align images

General approach

- ▶ Find a set of distinctive key-points
- ▶ Define a region around each keypoint
- ▶ Extract and normalize the region content
- ▶ Compute a local descriptor from the normalized regions
- ▶ Match local descriptors



Requirements

► Problem 1:

- Detect the *same point independently* in both images

no chance to match!



We need a repeatable detector

אם אני מוצא נוקדות מסויימות בתמונה א, אני רוצה למצוא נקודות מקבילות בתמונה באנחנו רוצים נקודות ספיציפיות כך שיהיה בניהם קשר

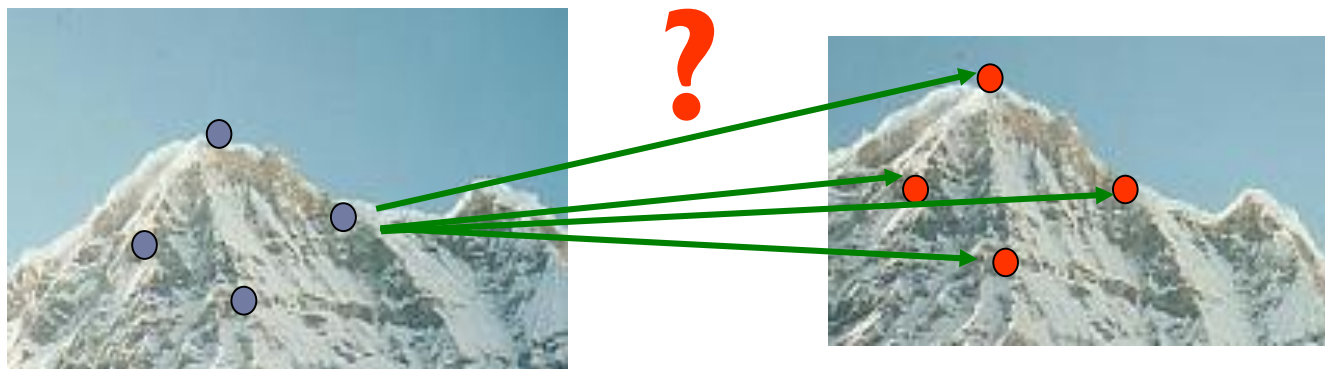
Requirements

► Problem 1:

- Detect the *same* point *independently* in both images

► Problem 2:

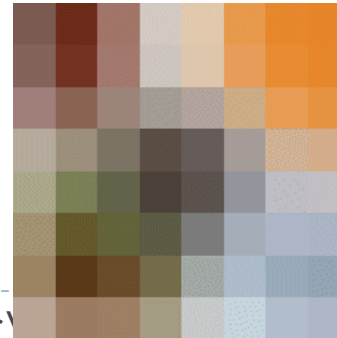
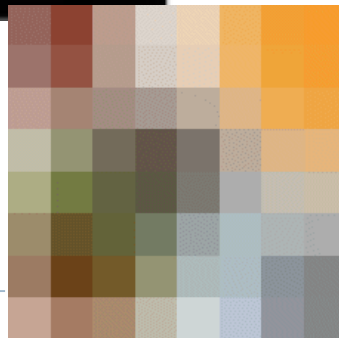
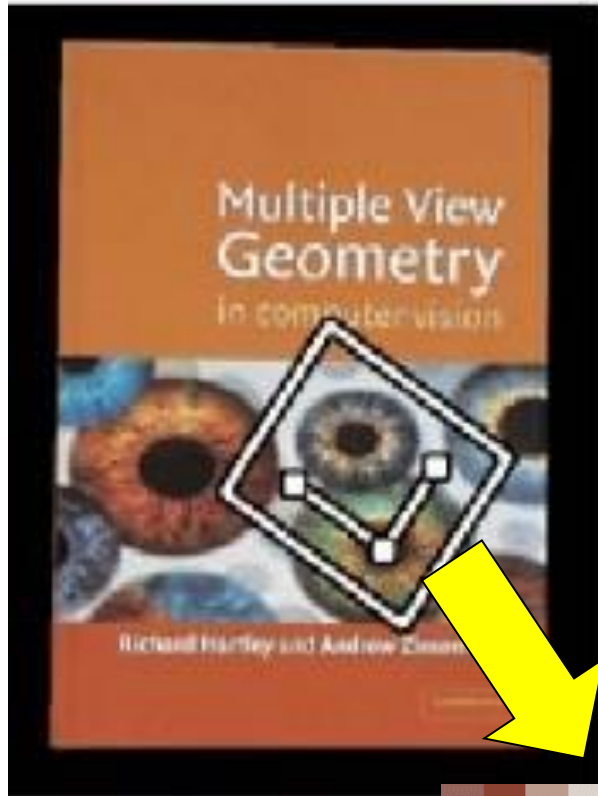
- For each point correctly recognize the corresponding one



We need a repeatable and distinctive detector

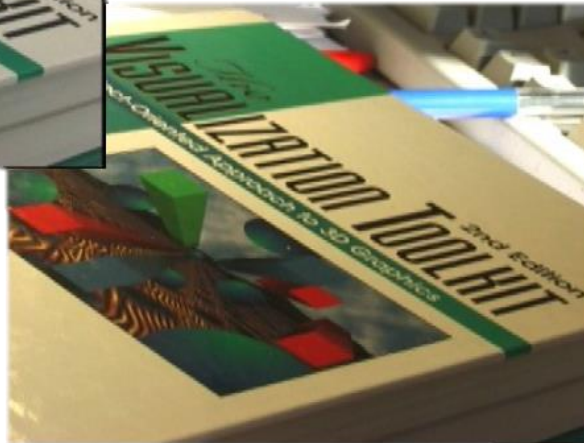
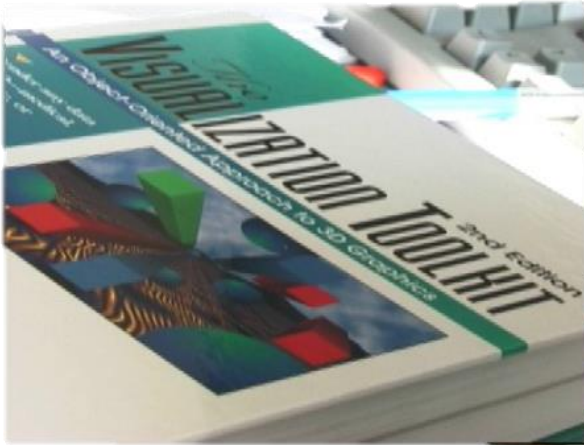
הנקודות יתאימו לנקודות אחרת באופן מיוחד, כלומר כל נקודה תתאים לנקודה אחת
איזה מין פונקציה חד ערכית ועל, אבל לא באופן מושלם כזה

Invariance 1: Geometric transformations



Invariance 2: Photometric transformations

חסין לפילטרים, רעשים, שנויים בצבעים, ארטיפקטים של דחיסה ועוד



Often modeled as a linear transformation:
scaling + offset of colors

And other nuisances...

- ▶ Noise
- ▶ Blur
- ▶ Compression artifacts
- ▶ ...

Requirements summary

אותם הנקודות בשתי התמונות

- ▶ Region extraction needs to be **repeatable** and **accurate**
 - ▶ **Invariant** to translation, rotation, scale changes
 - ▶ **Robust** or **covariant** to out-of-plane (\sim affine) transformations
 - ▶ **Robust** to lighting variations, noise, blur, quantization
- ▶ **Locality**: Features are local, therefore robust to occlusion and clutter.
- ▶ **Quantity**: We need a sufficient number of regions to cover the object.
- ▶ **Distinctiveness**: The regions should contain “interesting” structure.
- ▶ **Efficiency**: Close to real-time performance.


החפצים שלי לוקאליות לכן הם צריכים להיות חסינים להסתרות, ועצמים אחרים שנמצאים בתמונה

יעיל

Many existing detectors are available

- ▶ **Hessian & Harris** [Beaudet '78], [Harris '88]
 - ▶ **Laplacian, DoG** [Lindeberg '98], [Lowe '99]
 - ▶ **Harris-/Hessian-Laplace** [Mikolajczyk & Schmid '01]
 - ▶ **Harris-/Hessian-Affine** [Mikolajczyk & Schmid '04]
 - ▶ **EBR and IBR** [Tuytelaars & Van Gool '04]
 - ▶ **MSER** להתאמת אוטיות [Matas '02]
 - ▶ **Salient Regions** [Kadir & Brady '01]
 - ▶ **Others...**
-
- ▶ *Those detectors have become a basic building block for many recent applications in Computer Vision.*

Today

- ▶ Local invariant features
 - ▶ Motivation
 - ▶ Requirements, invariances
- ▶ **Keypoint localization**
 - ▶ **Harris corner detector** 
 - ▶ **Hessian detector**
- ▶ Scale invariant region selection
 - ▶ Automatic scale selection
 - ▶ Laplacian-of-Gaussian detector
 - ▶ Difference-of-Gaussian detector

Keypoint localization

► Goals:

- Repeatable detection
- Precise localization
- Interesting content

➔ Look for image regions that are unusual

- Lead to unambiguous matches in other images

אני לא רוצה שפות כי אני יכול להתאים שפות אחר
לשניה בהמון מקומות אפילו אם זה לא מתאים

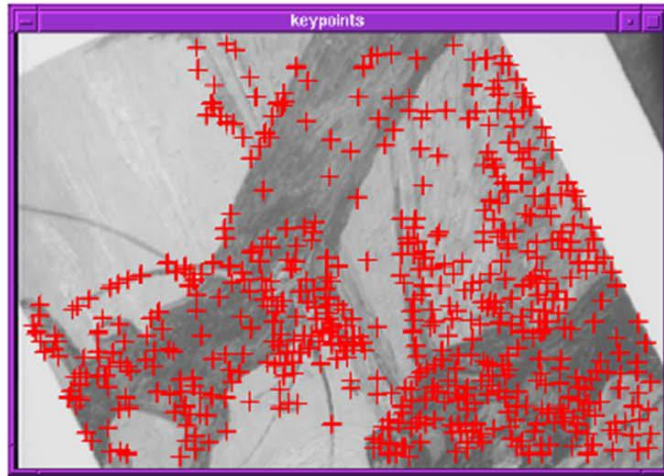
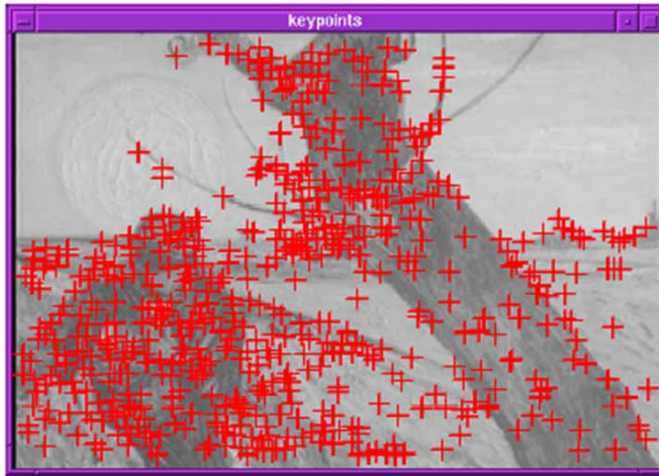


How to define “unusual”?

שתי השפות האלה אני יכול להתאים אותם לכן אני לא רוצה להתאים בעזרת שפות

Finding Corners

אנחנו משתמשים בפינות, כי פינות
פחות חוזרות אחת על השניה
לפינות יש זווית ויש גם שינויים בגרדינטים בשני הצירים
בניגוד לציר אחד



פינה לפי הגדרה זה שנוי
הגרדינטים בשני הצירים

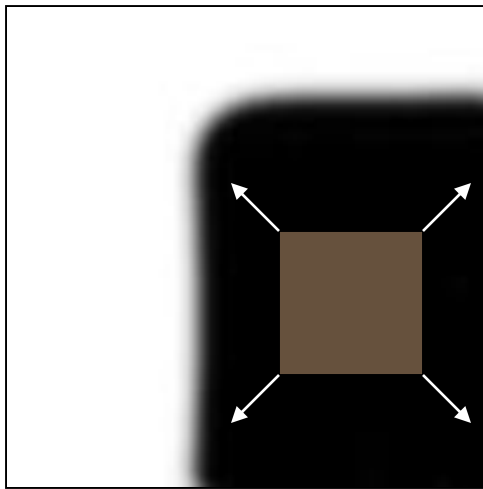
- ▶ Key property:
 - ▶ In the region around a corner, image gradient has two or more dominant directions
- ▶ Corners are repeatable and **distinctive**

[Harris et al. "[A Combined Corner and Edge Detector.](#)" 1988]

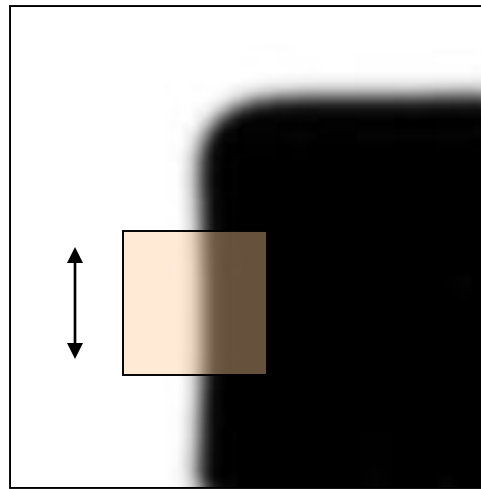
Corners as distinctive interest points

► Design criteria:

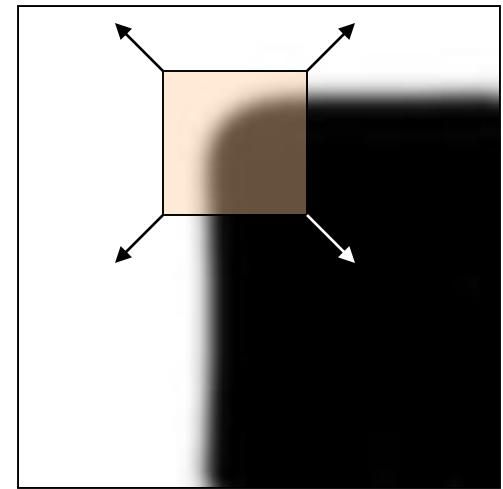
- We should easily recognize the point by looking through a small window (locality)
- Shifting the window in *any direction* should give a large change in intensity (good localization)



“flat” region:
no change in all
directions



“edge”:
no change along the
edge direction



“corner”:
significant change in
all directions

יותר טוב
אבל אין שנוי
אם אני זז
למעלה למטה

בפינה יש שנוי בכל תזוזה
לכל כוון לכן יותר טוב

Harris Detector formulation

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

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

ריקט או גאוסיאן

Window
function

אני רוצה לסכם או לעשות סוג של ממוצע

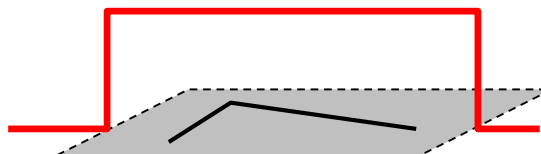
Shifted
intensity

אני רוצה לדעת כמה השיפט השפיע עלי

Intensity

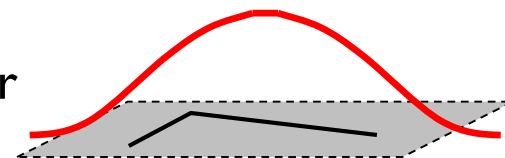
נקבל 0 באזורים חלקים
בשפות באזורים מסויימים יהיה לי 0 ולכן אחר יהיה לי יותר
בפינות יהיה לי אנטיסיטי/שינוי יותר משמעותי

Window function $w(x, y) =$



I in window, 0 outside

or



Gaussian

Harris Detector formulation

This measure of change can be approximated by (Taylor expansion):

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

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

מטריצה שתגיד לי אם אני בפינה או לא בפינה

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

Sum over image region – area we are checking for corner

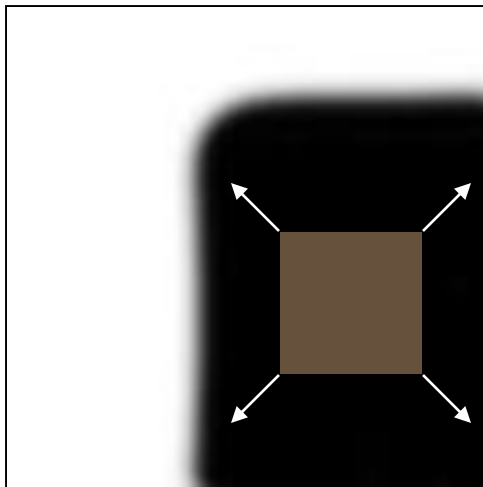
גזירה סדר ראשון בריבוע

אם יש לי שנוי ב X
שלושת האברים במטריצה יהיו אפס
כנל אם יש שנוי רק ב Y
אם אין שנוי לא ב Y
 X, Y
כל המריצה תהיה אפסים

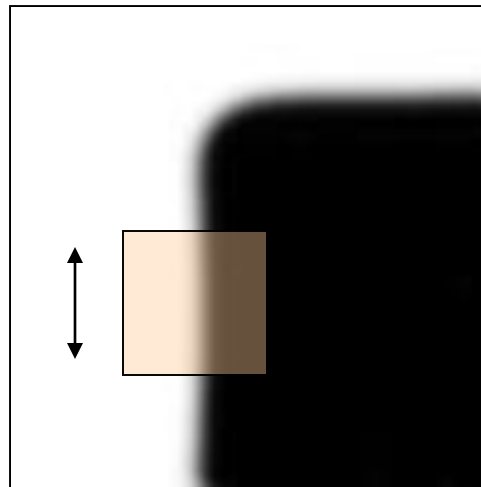
Gradient with respect to x , times gradient with respect to y

Harris Detector formulation

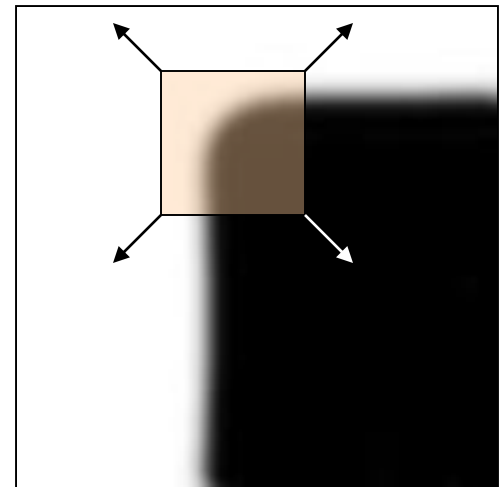
- ▶ What will M look like for each case?
 - ▶ (On the board)



“flat” region:
no change in all
directions



“edge”:
no change along the
edge direction



“corner”:
significant change in
all directions

Interpreting the eigenvalues

Classification of image points using eigenvalues of M :



וקטורים ערכים עצמיים

λ_2

“Edge”

$\lambda_2 \gg \lambda_1$

אחת מה תהיה גדולה מאד ביחס לשניה

“Corner”

λ_1 and λ_2 are large,

$\lambda_1 \sim \lambda_2$; פחות יותר מאותו סדר גודל

E increases in all directions

λ_1 and λ_2 are small;
 E is almost constant
in all directions

“Flat”
region

“Edge”

$\lambda_1 \gg \lambda_2$

λ_1

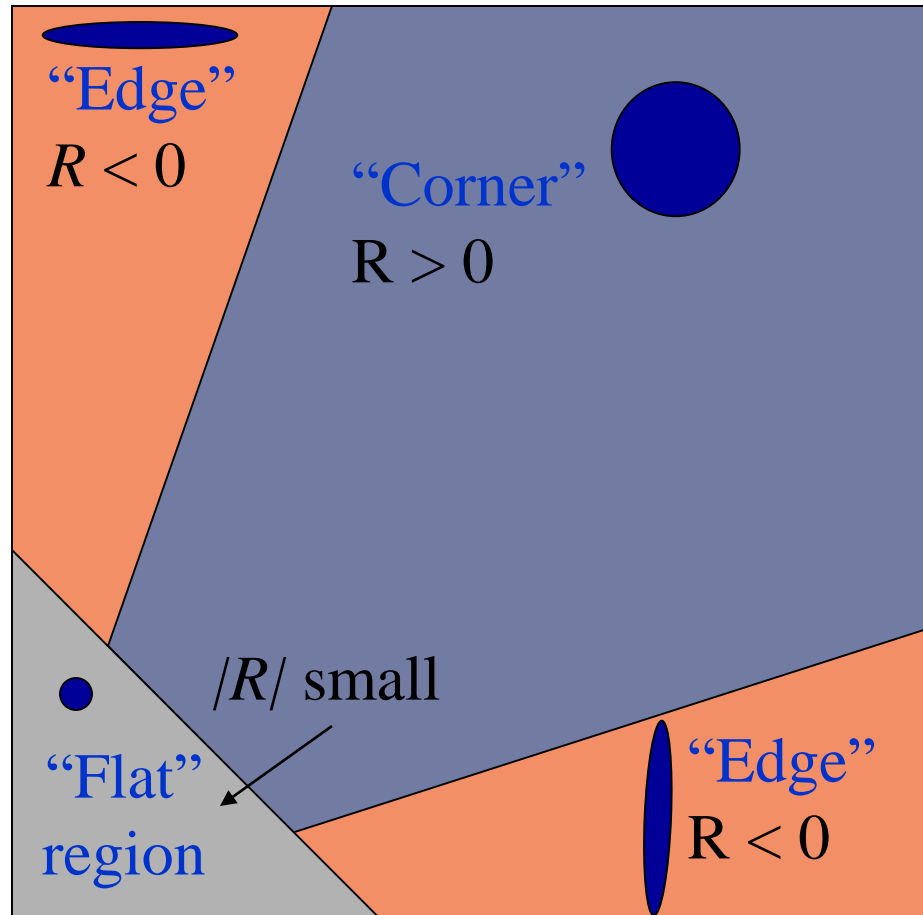
Corner response function

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

אני רוצה להגדיר משוואה בלי לחשב ערכים עצמיים

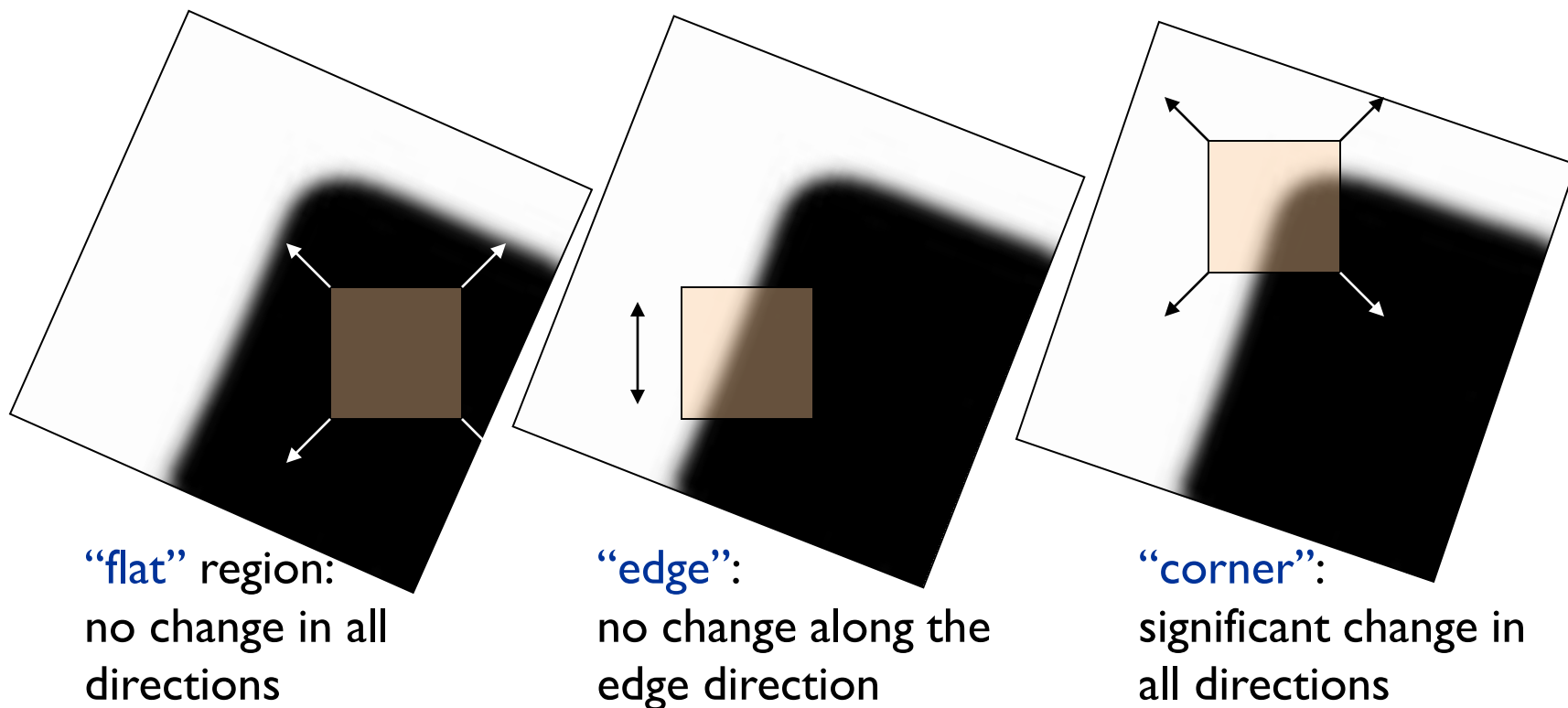
Fast approximation:

- Avoid computing the eigenvalues
- α : constant (0.04 to 0.15)






Rotation invariance

- ▶ Eigenvalues are of the same nature as before



Summary: Harris Corner Detector

► Compute second moment matrix M

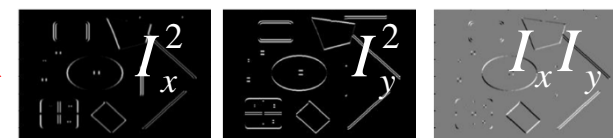
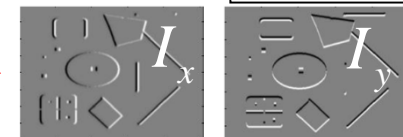
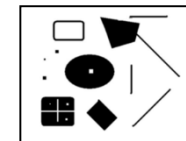
1. Compute derivatives I_x, I_y 
2. Square of derivatives $I_x^2, I_y^2, I_x I_y$ 
3. Gaussian filter $g(\sigma)$ 

4. Compute R scores

- Find points with large corner response
 $R > threshold$

5. Perform non-maximum suppression

כי יכולות להיות הרבה נקודות סמוכות שכולן עוברות את
 TH
 ואני צריך רק נקודה אחת



כל מה שעובר
 TH
 מוגדר להיות פינה

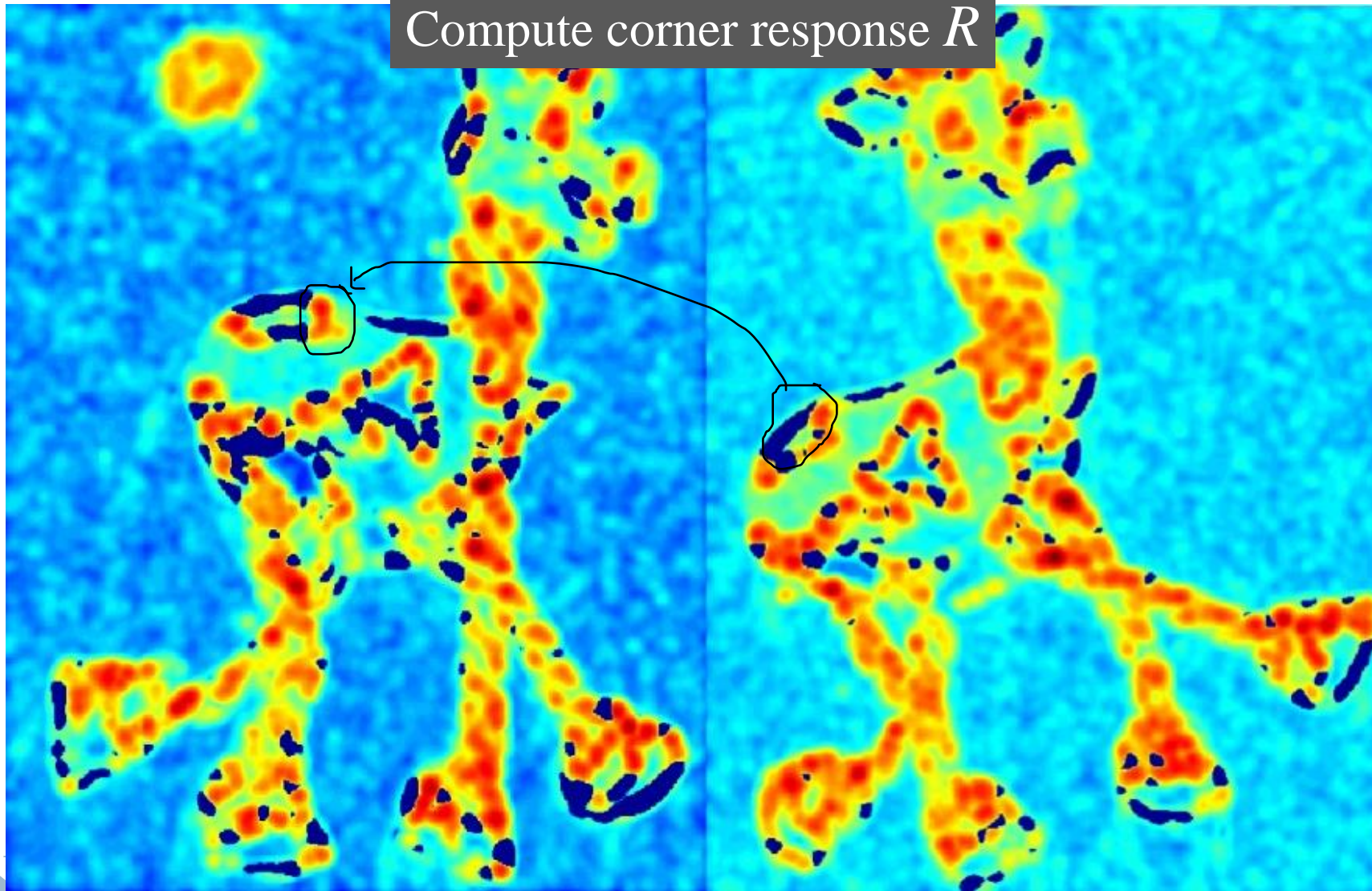
Harris Detector: Workflow



Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

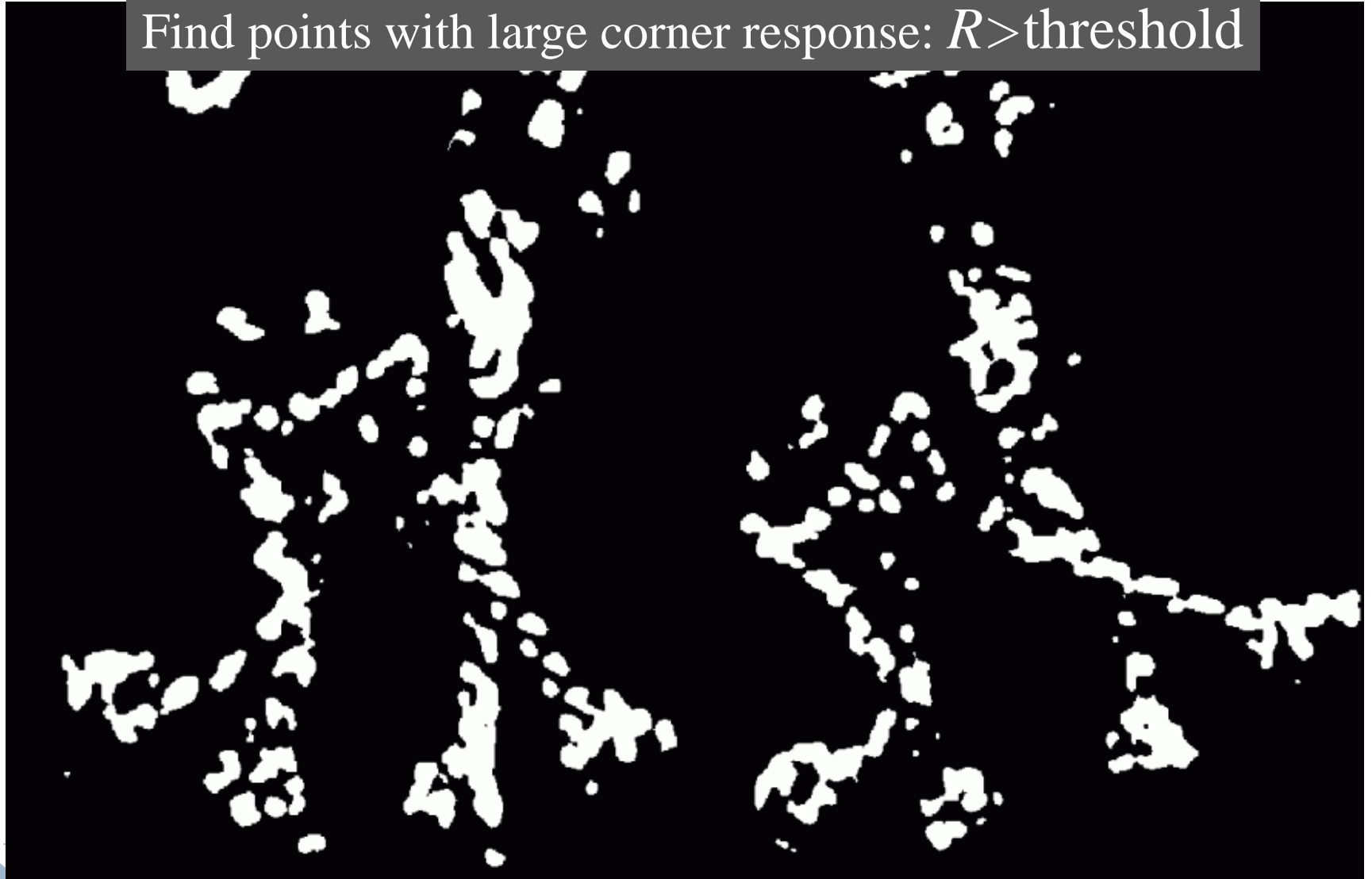
Harris Detector: Workflow

Compute corner response R



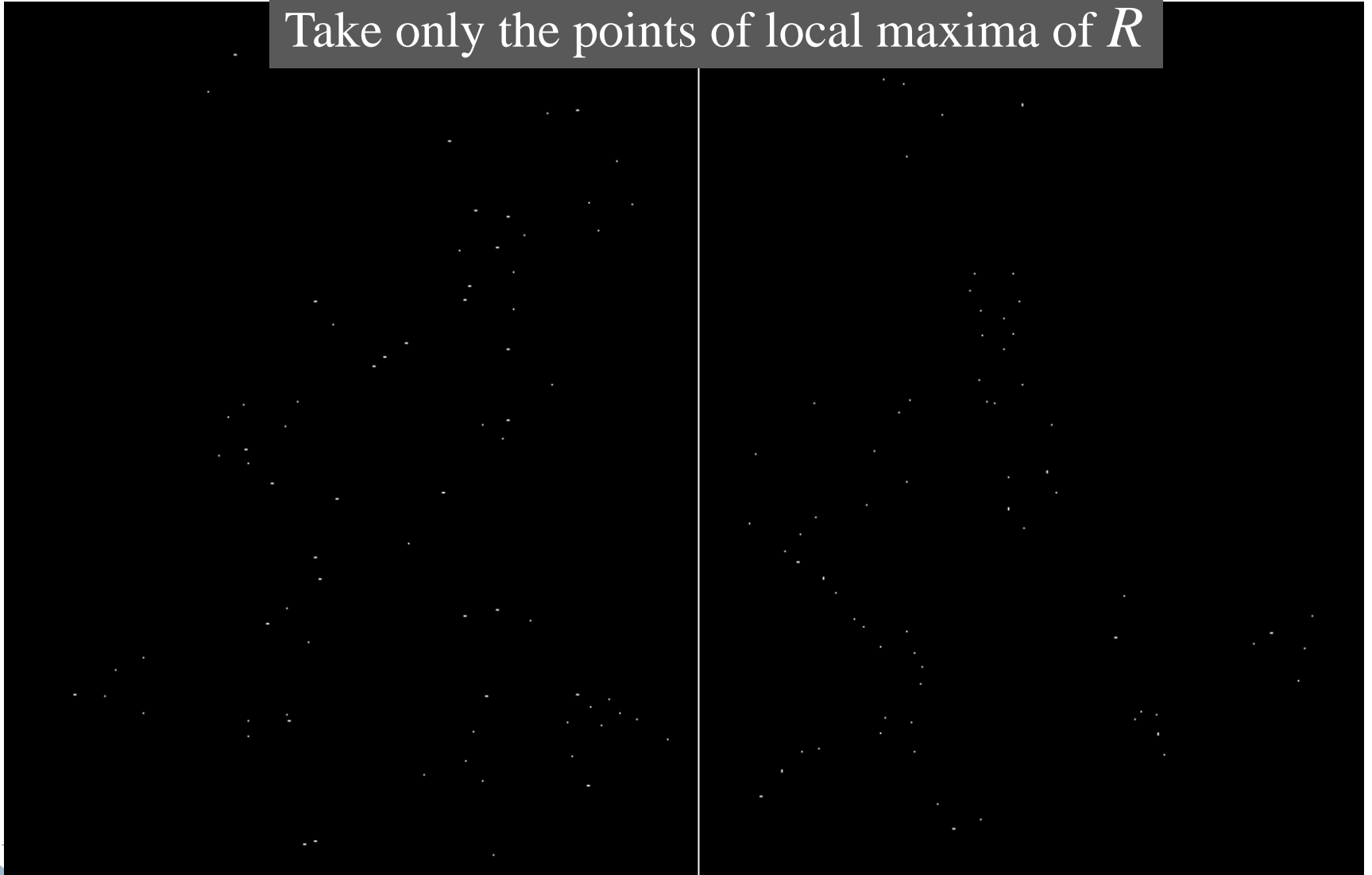
Harris Detector: Workflow

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



Harris Detector: Workflow

Take only the points of local maxima of R



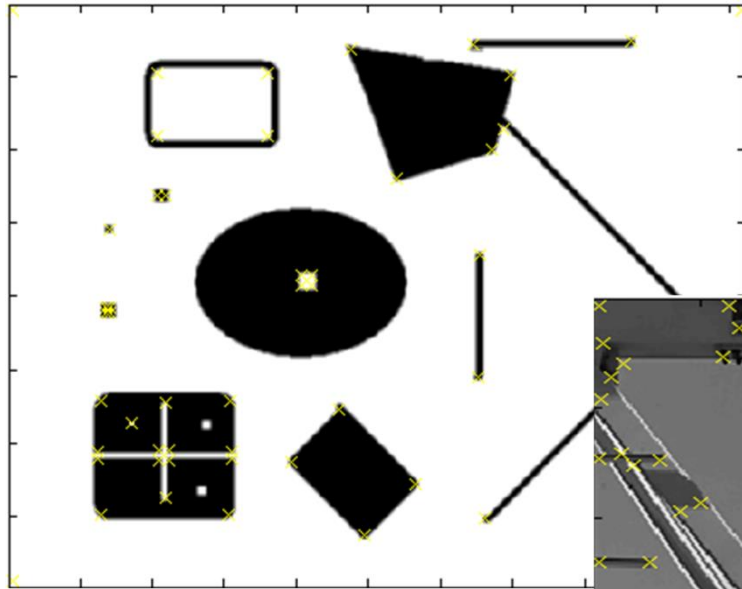
Harris Detector: Workflow

אפשר לראות כאשר הראש לא באותו רוטציה
אנחנו מצליחים לחלץ אותם מאפיינים

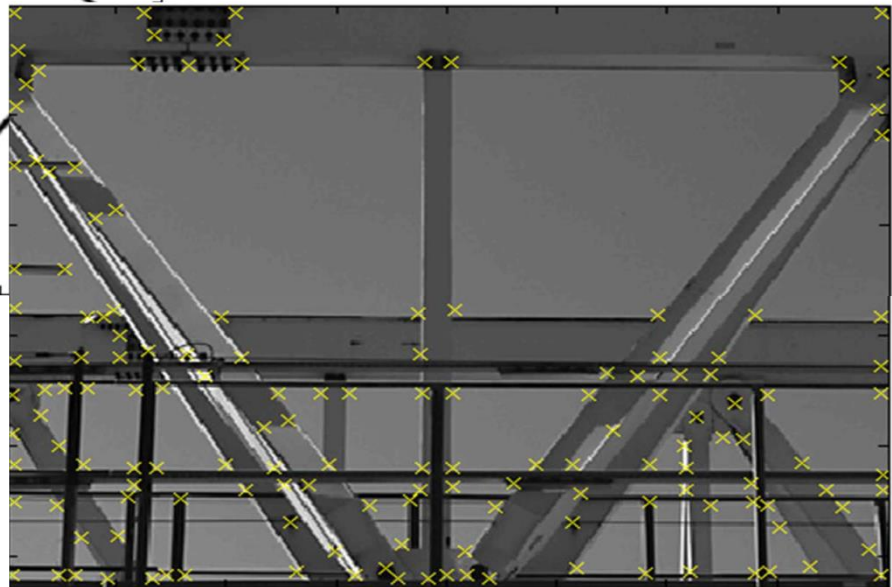
Resulting Harris points



Harris detector – example result



A very accurate
corner detector

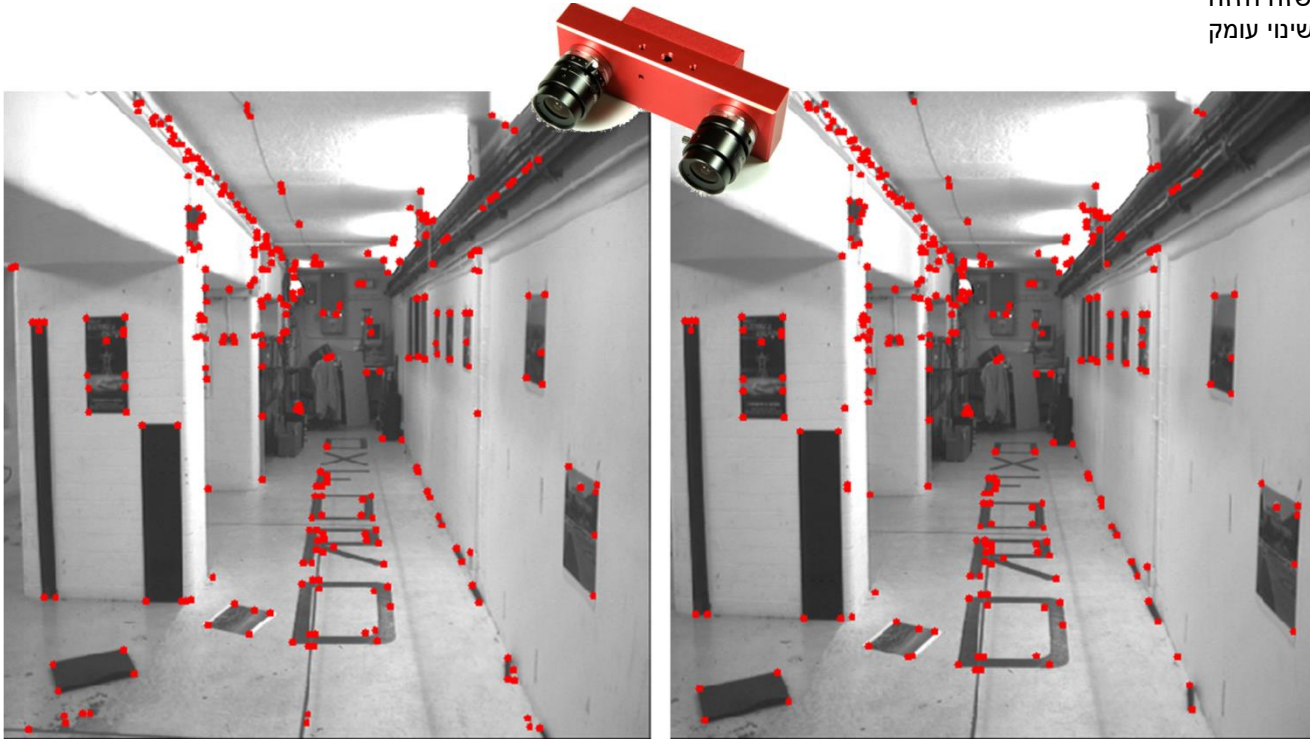


Harris detector – example result



Harris detector – example result

קצת טרנסלציה שזה הזזה
קצת שינוי עומק

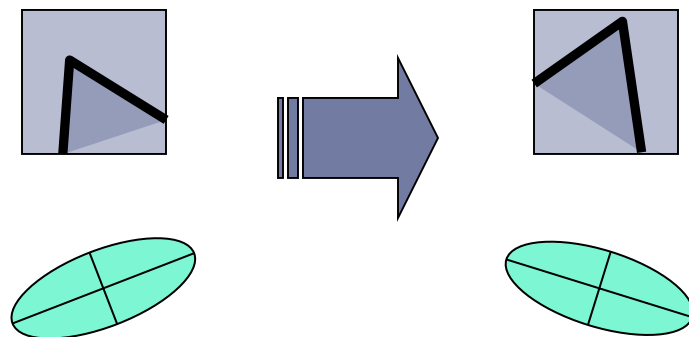


Results are usually good for stereo correspondences

Harris Detector: Properties

► Is it rotation invariant?

\mathcal{D}

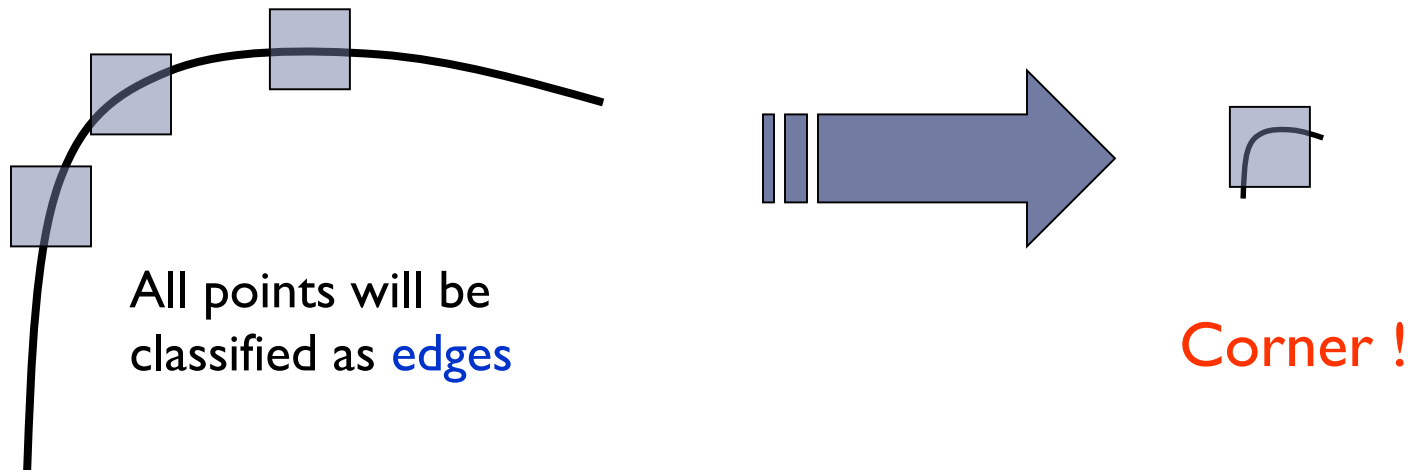


Ellipse rotates but its shape (i.e. eigenvalues) remains the same

Yes: Corner response R is invariant to image rotation

Harris Detector: Properties

- Is it invariant to image scale?



No: Not invariant to image scale!

Today

- ▶ Local invariant features
 - ▶ Motivation
 - ▶ Requirements, invariances
- ▶ **Keypoint localization**
 - ▶ **Harris corner detector**
 - ▶ **Hessian detector**
- ▶ Scale invariant region selection
 - ▶ Automatic scale selection
 - ▶ Laplacian-of-Gaussian detector
 - ▶ Difference-of-Gaussian detector

Hessian detector [Beaudet 78]

▶ Hessian determinant

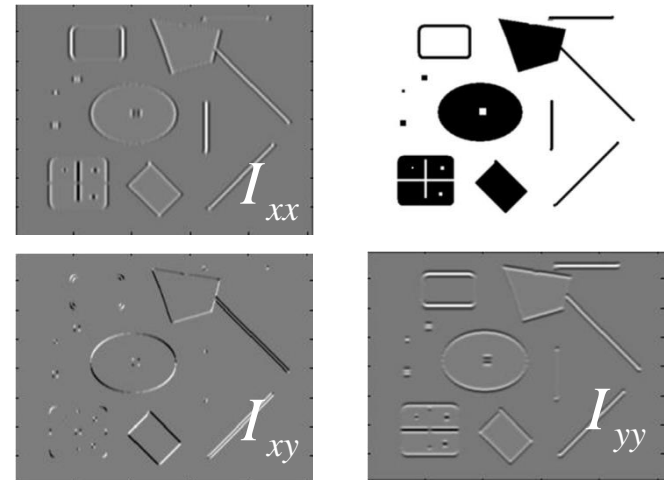
▶ Determined by second derivatives

נגזרת שניה רק ב
X

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

נגזרות משולבות

פחות מוצלח מהאריס, פחות מוכר, פחות שימושי



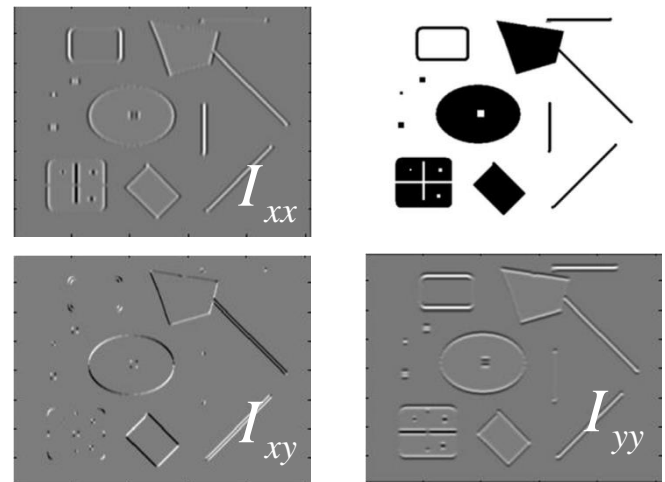
▶ Key idea:

▶ Search for strong derivatives in two orthogonal directions

Hessian detector [Beaudet 78]

- ▶ Hessian determinant
 - ▶ Determined by second derivatives

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

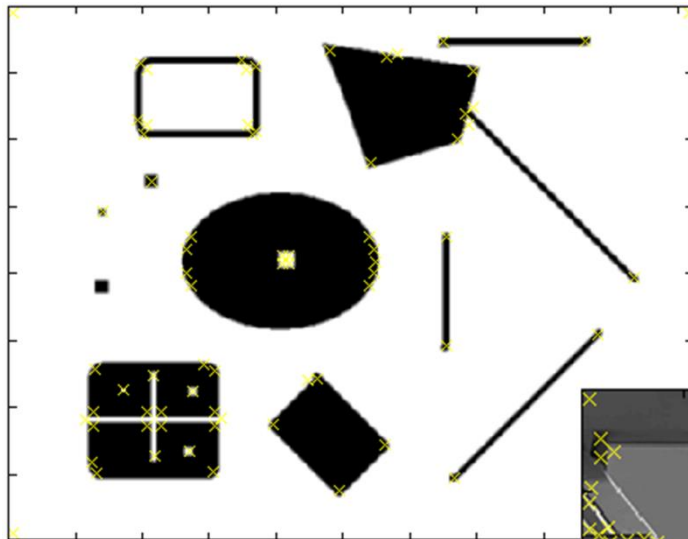


- ▶ Key idea:
 - ▶ Search for strong derivatives in two orthogonal directions

$$\det(Hessian(I)) = I_{xx}I_{yy} - I_{xy}^2$$

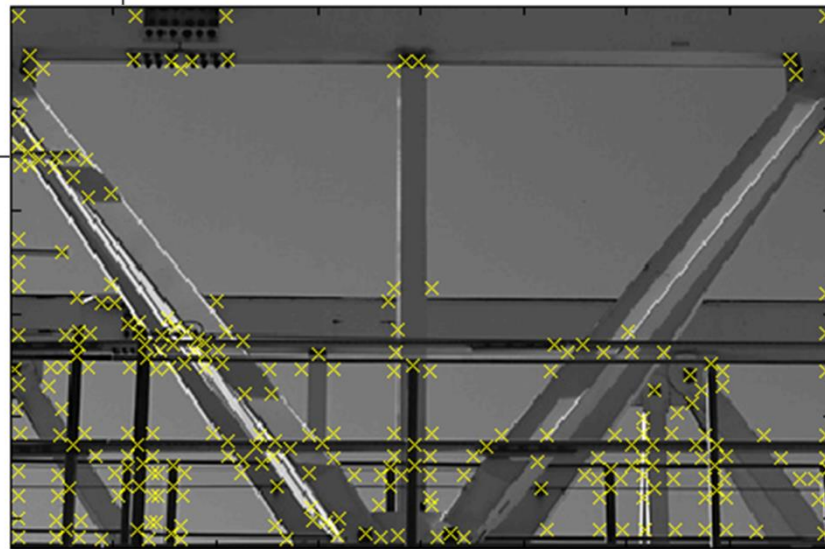


Hessian detector – example result



Responses mainly
on corners and
strongly textured
areas

אזורים בעלי מרקם חזק
בעניין הזה יותר טוב מהארים



Hessian detector – example result



End – Feature detectors

Now you know how it works