


Feature Detection

Reference: Szeliski, Chapter 4

Today's lecture

- Motivation for feature detection
- Harris corner detector

Motivation

Why do we need feature detection?

A typical application is **panorama building:**

[Brown & Lowe, 2003]

Other example applications are

- object recognition
- 3D reconstruction
- image alignment
- etc.

Brown, M. and Lowe, D., "Recognizing Panoramas", ICCV 2003

Motivation

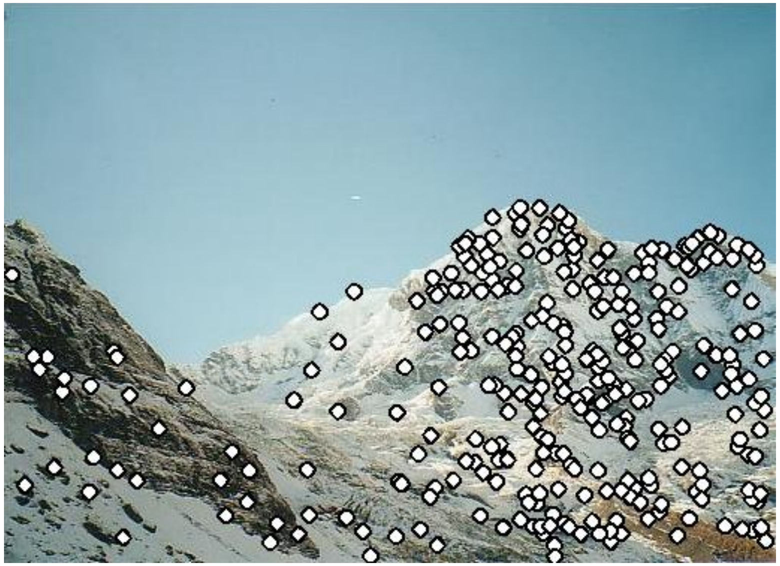
Why do we need feature detection?

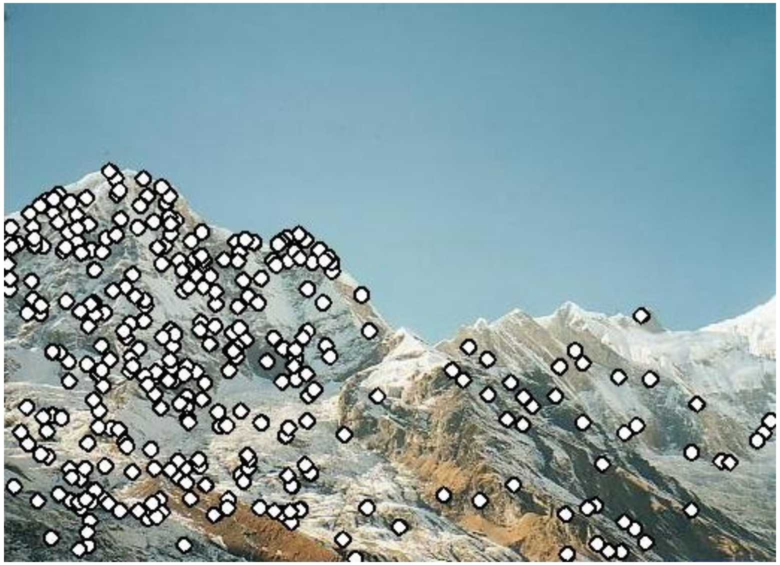
- We need to **match (align)** images first
- Tricky task if done “globally”, due to **occlusion, lighting changes**, etc.
- The idea is to find **local features** in both images **which match well**

Motivation

Why do we need feature detection?

- **Find features points in both images**



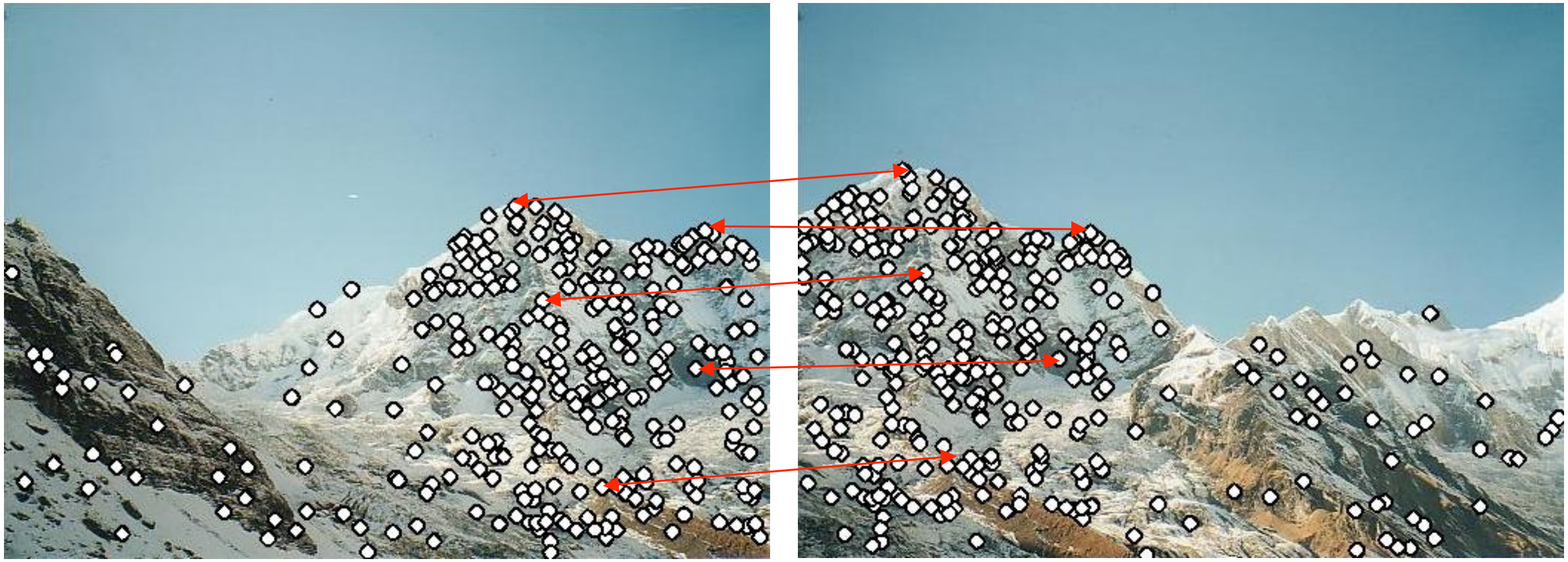


Slide adapted from Antonio Torralba

Motivation

Why do we need feature detection?

- Find features points in both images
- **Match corresponding pairs**



Slide adapted from Antonio Torralba

Motivation

Why do we need feature detection?

- Find features points in both images
- Match corresponding pairs
- **Use the matched points to merge images together**

Slide adapted from Antonio Torralba

Matching with features

Important properties

When matching images using features, two important properties are:

(1) Repeatability

We need to be able to detect the same keypoint in both images independently!

(2) Reliability (or "distinctiveness")

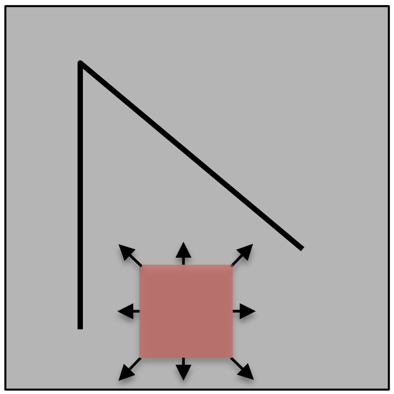
We need a "description" of the image content at each point which reliably identifies the point (in other words, it needs to be distinct).

Note: Property (2) is more a property of the keypoint descriptor and less a property of the keypoint detector!

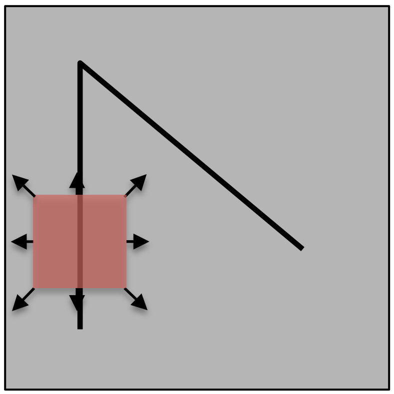
Harris corner detector

[Harris & Stephens, "A Combined Corner and Edge Detector", 1988]

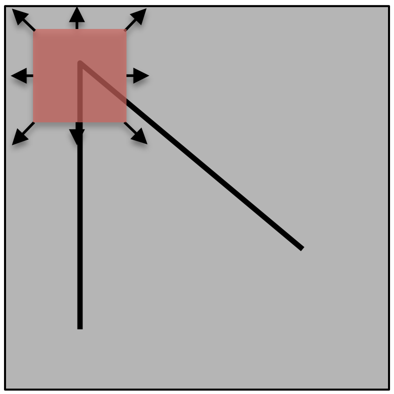
Harris and Stephens observed the following*:



Flat region: no change as we move freely



Edge: no change as we move along the edge



Corner: a lot of change around corner

→ **What is of interest?** Large differences in the shifted image!

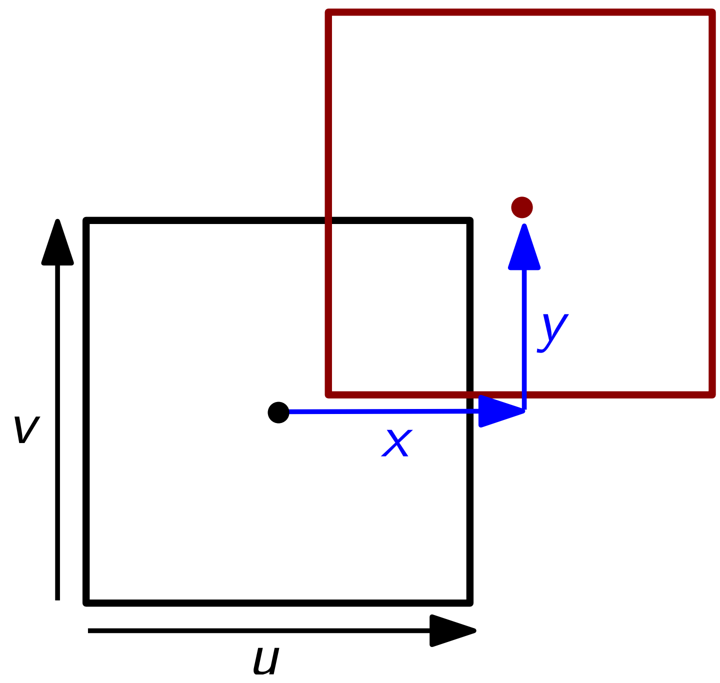
$$E(x,y) = \sum_u \sum_v \underbrace{\boxed{w(u,v)}}_{\substack{\textbf{Window function} \\ \text{(typically Gaussian)}}} \left[\underbrace{\boxed{I(x+u,y+v)}}_{\substack{\textbf{Shifted patch} \\ \text{shifted by (x,y)}}} - I(u,v) \right]^2$$

*Motivated by work of H. Moravec

Harris corner detector

[Harris & Stephens, “A Combined Corner and Edge Detector”, 1988]

Illustration of the **basic principle**:



patch shifted by (x, y)

Harris corner detector

Derivation details

Lets take a look at the **Taylor-series expansion** (1st order) of

$$I(x+u, y+v) = I(u,v) + \frac{\partial I(u,v)}{\partial u}x + \frac{\partial I(u,v)}{\partial v}y$$

If we substitute this into our E(x,y) term we get

$$E(x,y) \approx \sum_{u,v} w(x,y)[I_u^2 x^2 + 2 I_u I_v xy + I_v^2 y^2]$$

$$= \sum_{u,v} w(x,y) \left[(x\ y)^\top \begin{pmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{pmatrix} (x\ y) \right]$$

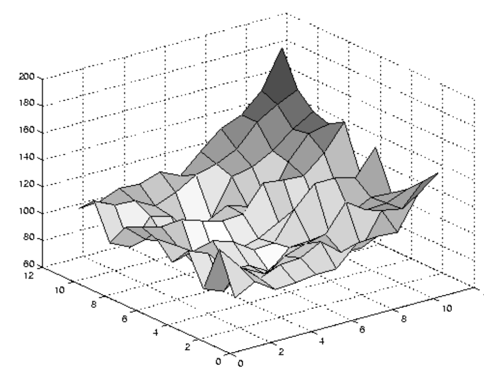
$$= (x\ y)^\top \boxed{\left(\sum_{u,v} w(x,y) \begin{pmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{pmatrix} \right)} (x\ y)$$

This well be our “quantity” (**M**), of interest!

Harris corner detector

Illustration of the key principle – Eigenvalues of **M**

Courtesy of Szeliski

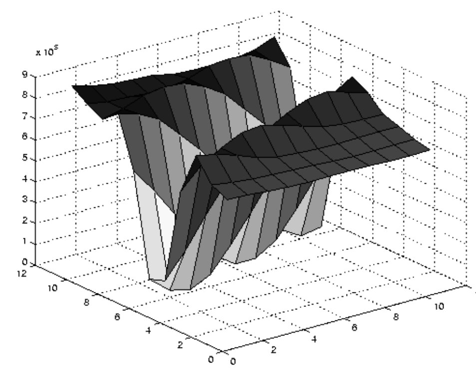


both eigenvalues
are **small**

Harris corner detector

Illustration of the key principle – Eigenvalues of **M**

Courtesy of Szeliski

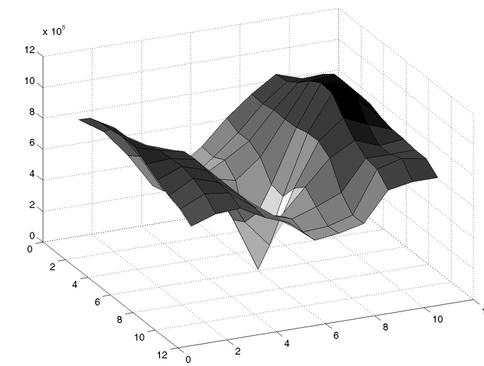


one eigenvalue **large**,
one eigenvalue **small**

Harris corner detector

Illustration of the key principle – Eigenvalues of **M**

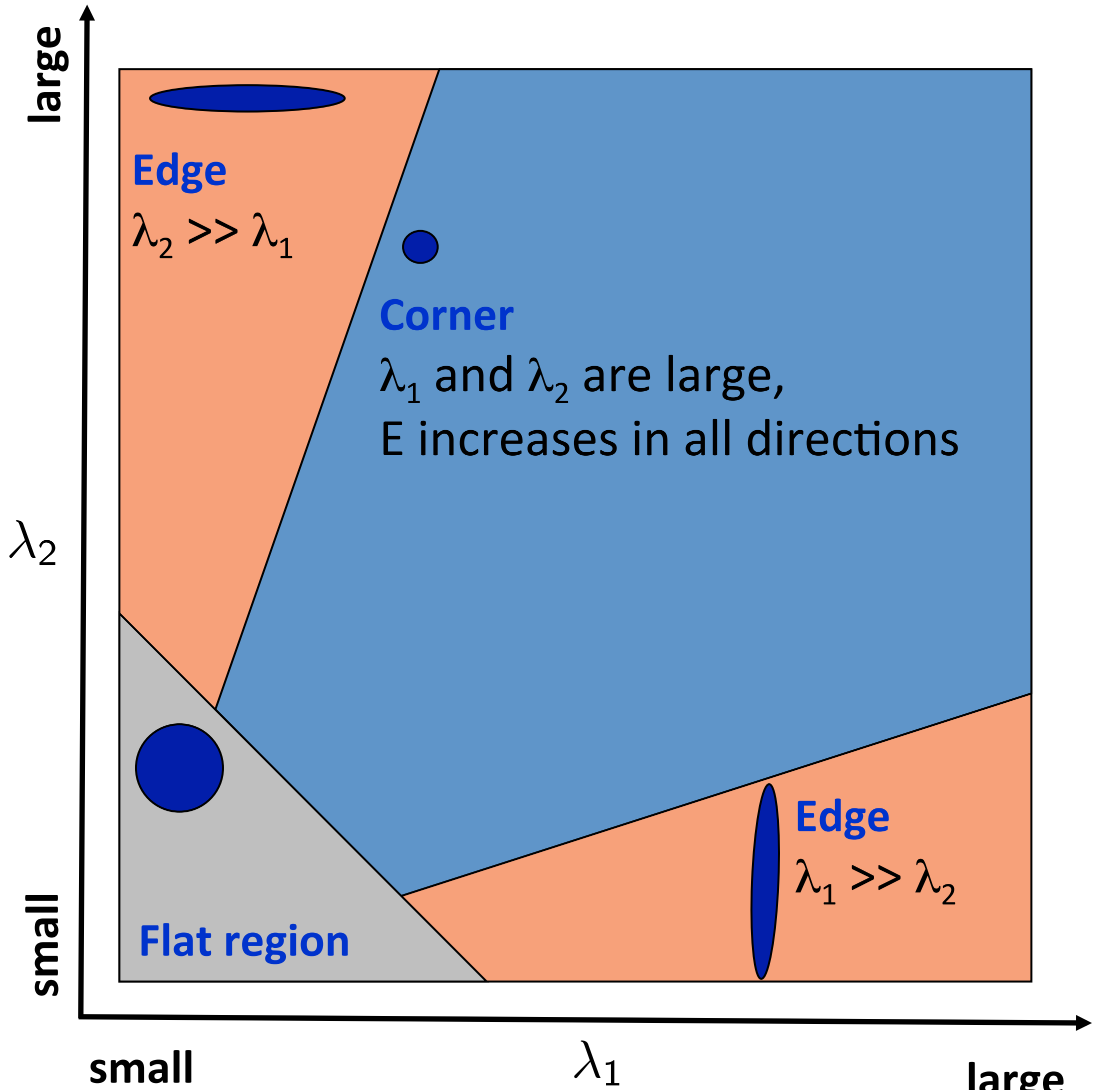
Courtesy of Szeliski



both large

Harris corner detector

Classification of points based on the eigenvalues of **M**



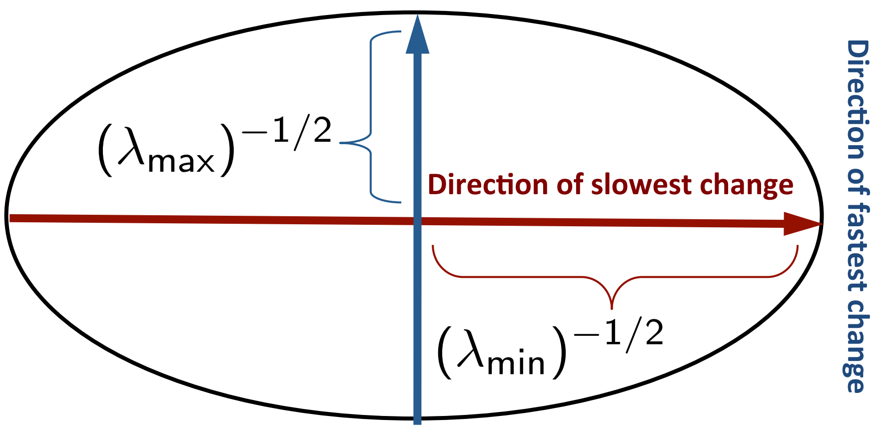
Harris corner detector

Design of a "response" function

We can look at the eigenvalues of **M**. If **both are large**, this is indicative of a corner (*i.e.*, large variation in both directions).

Harris & Stephens propose the following **response function**:

$$
\begin{aligned}
R(\mathbf{M}) &= \lambda_1\lambda_2 - \gamma(\lambda_1+\lambda_2)^2 \\
&= det(\mathbf{M}) - \gamma\, trace(\mathbf{M})^2, \quad \gamma > 0
\end{aligned}
$$



Note: Determinant (det) and trace are appealing, since **no** eigenvalues need to be computed!

Harris corner detector

Other response functions

There are many other choices of response functions, *e.g.*,

[Shi & Tomasi, 1994]: take the *minimum* eigenvalue

[Triggs, 2004]: $R(\mathbf{M}) = \lambda_{\text{min}} - \gamma \lambda_{\text{max}}$

(for more examples, see Szeliski's book)

Algorithmically*, we

1) determine locations where **response function value > threshold**
2) find points of local maxima (*e.g.*, using **non-maximum suppression**)

*to see how this is really implemented, look at the source code of the `vl_harris` function of the vlfeat MATLAB toolbox.

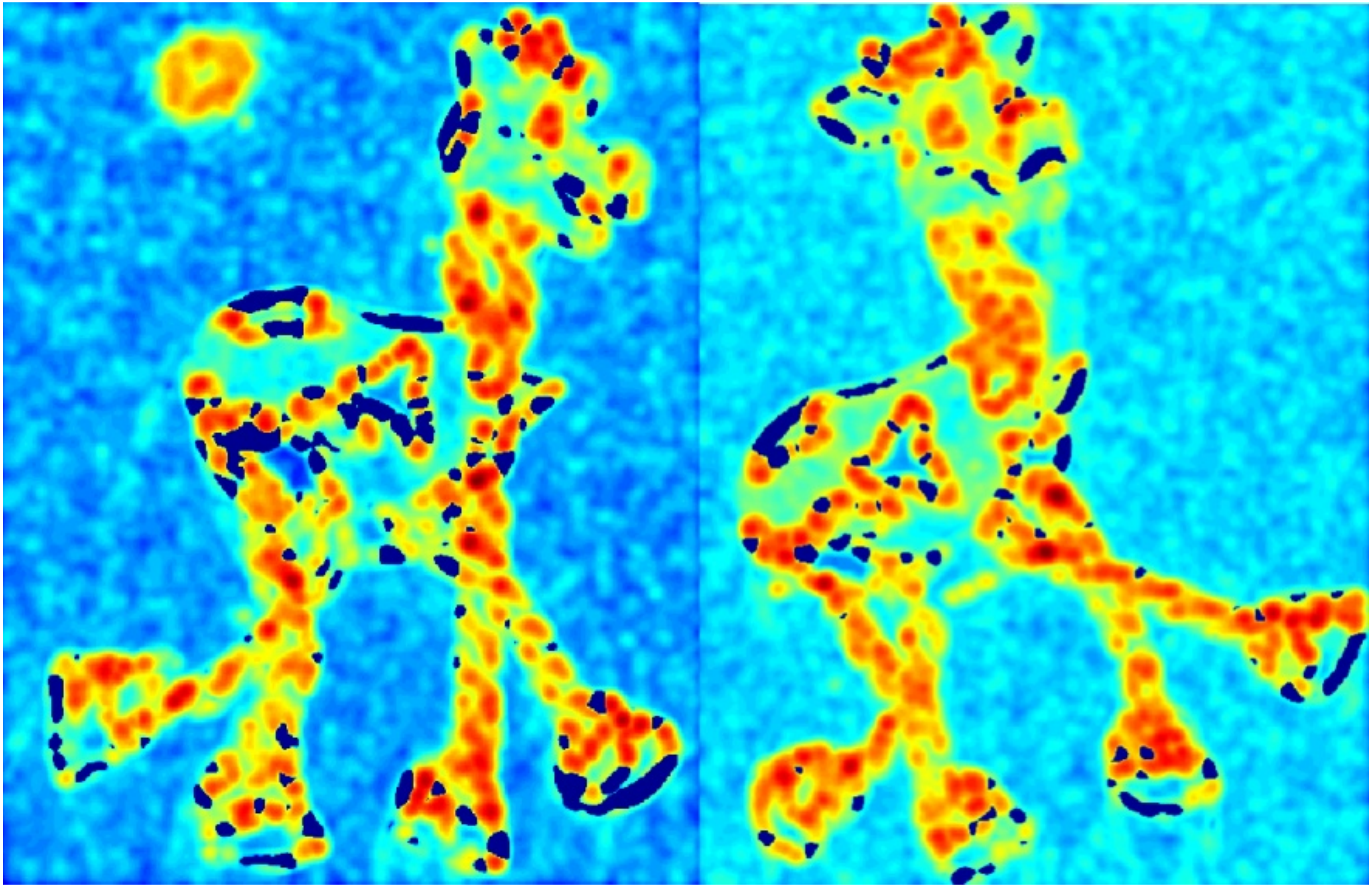
Harris corner detector

Example: Original input image

Slide by Antonio Torralba

Harris corner detector

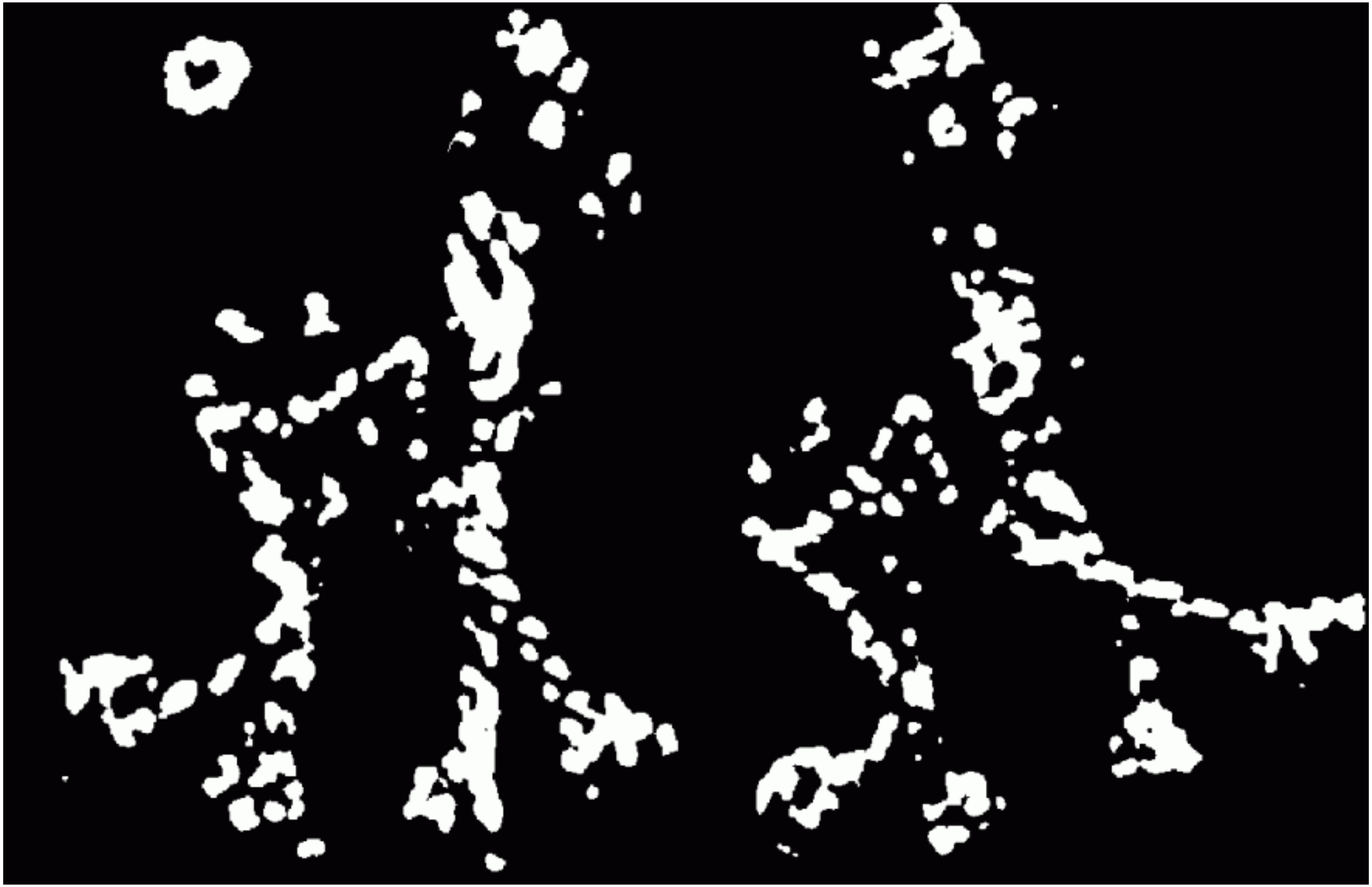
Example: Corner response



Slide by Antonio Torralba

Harris corner detector

Example: Response values > threshold



Slide by Antonio Torralba

Harris corner detector

Example: after non-maxima suppression

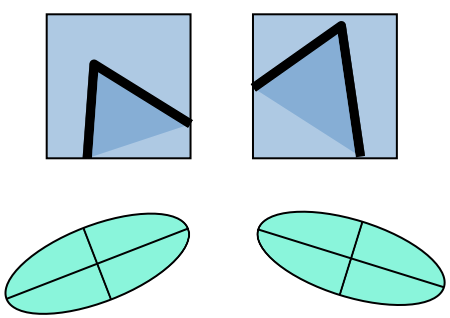
Slide by Antonio Torralba

Harris corner detector

Properties

In the **ideal case** *(i.e.*, assuming we would have a “perfect” detector), we would **always** find the same points regardless of lighting, perspective, scale, etc.

Rotation invariance ✔

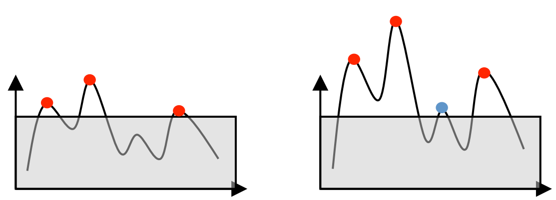


Ellipse rotates, but shape remains

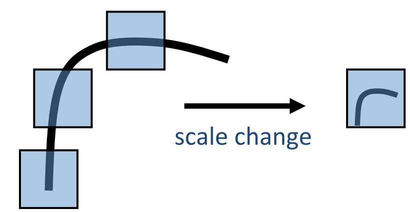
Intensity shift (roughly scale) ✔

Shift: Derivatives are used → no problem

Scale: Intensity threshold fix → no full invariance



Scale ✗



Edge becomes a corner!

Additional resources

- look at the implementation of the Harris corner detector (function `vl_harris`) in the **vlfeat** toolbox
- read **Chapter 4.1.1** of Szeliski's book
- read the **original paper** on the Harris corner detector:

Harris & Stephens, *A Combined Corner and Edge Detector*, 4th Alvey Vision Conference, 1988

(online: http://www.bmva.org/bmvc/1988/avc-88-023.pdf)