Feature Detection, Matching and its Applications

Alex Lin

Outline

- ***Introduction***
- ***What is feature***
- ***Where is feature***
- ***Feature Matching***

Building a Panorama

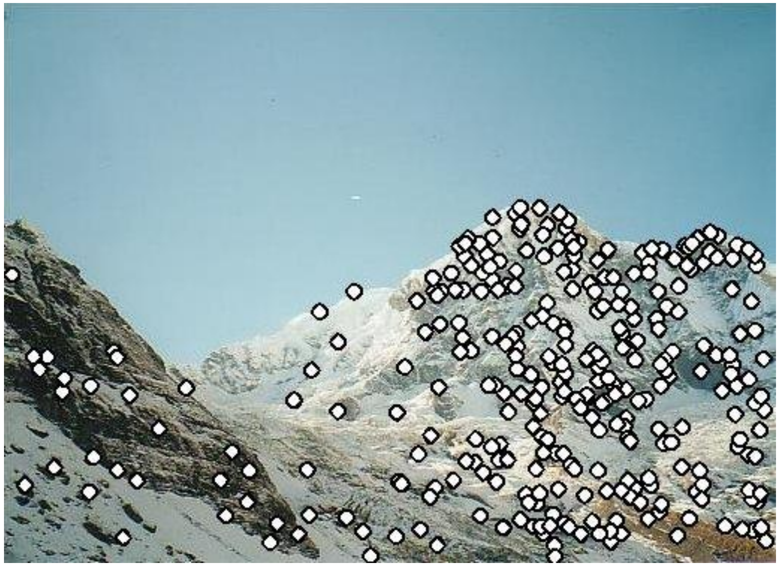
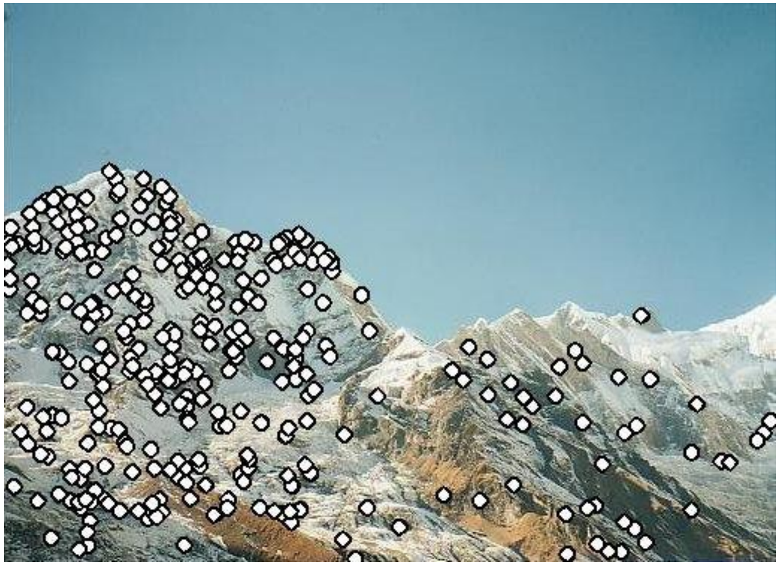
M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

How do we build a panorama?

- We need to match (align) images
- Global methods sensitive to occlusion, lighting, parallax effects. So look for local features that match well.
- How would you do it by eye?

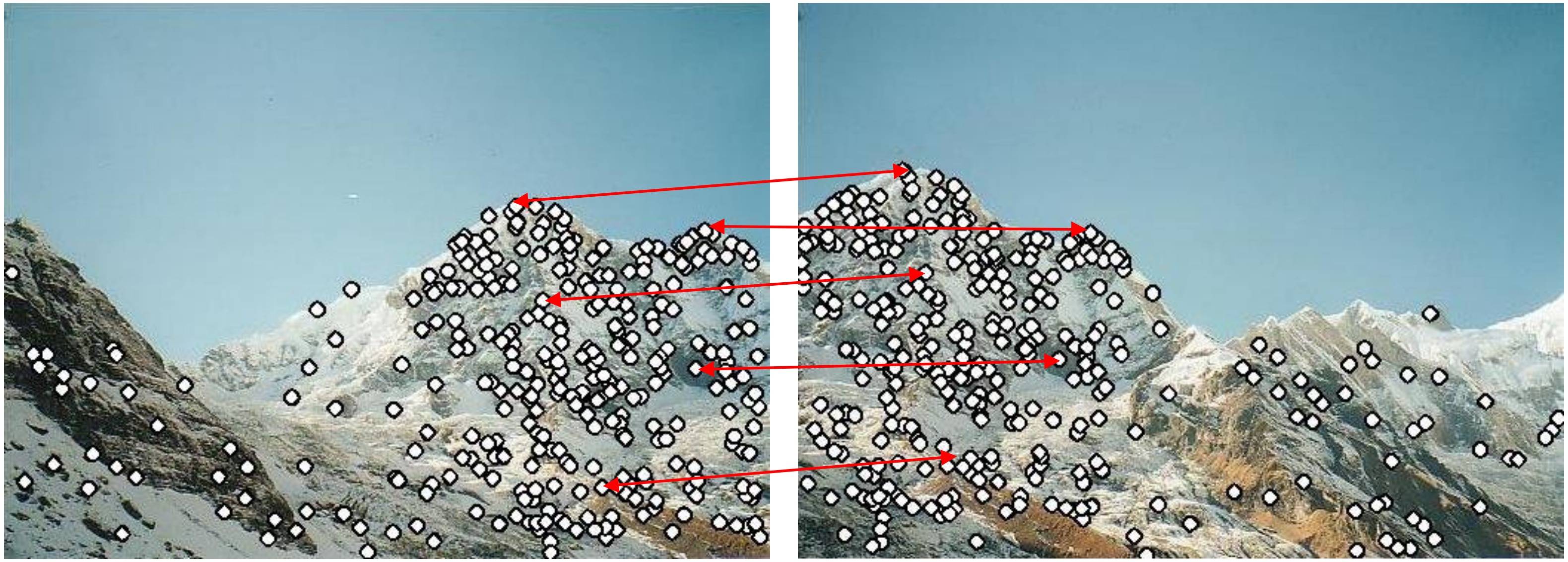
Matching with Features

•Detect feature points in both images

Matching with Features

- Detect feature points in both images
- Find corresponding pairs



Matching with Features

- Detect feature points in both images
- Find corresponding pairs
- Use these pairs to align images (homography)

Matching with Features

- Problem 1:
  - Detect the *same* point *independently* in both images

no chance to match!

We need a repeatable detector

Matching with Features

- Problem 2:
  - For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor

More motivation…

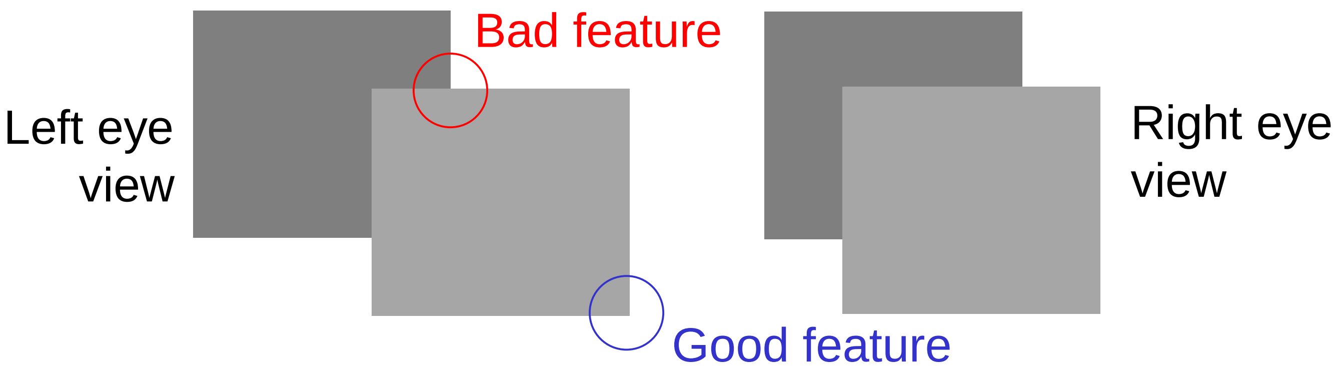
- Feature points are used also for:
  - Image alignment (homography, fundamental matrix)
  - 3D reconstruction
  - Motion tracking
  - Object recognition
  - Indexing and database retrieval
  - Robot navigation
  - … other

Query book

Book identification

Selecting Good Features

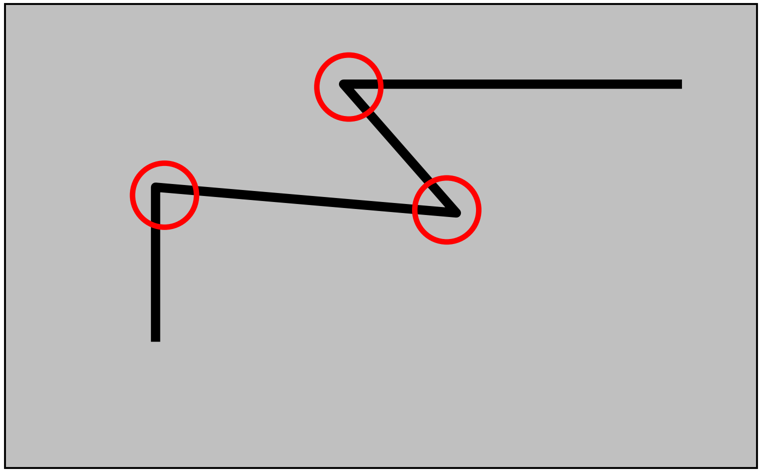
- What's a "good feature"?
  - Satisfies brightness constancy—looks the same in both images
  - Has sufficient texture variation
  - Does not have too much texture variation
  - Corresponds to a "real" surface patch—see below:



  - Does not deform too much over time

An introductory example:

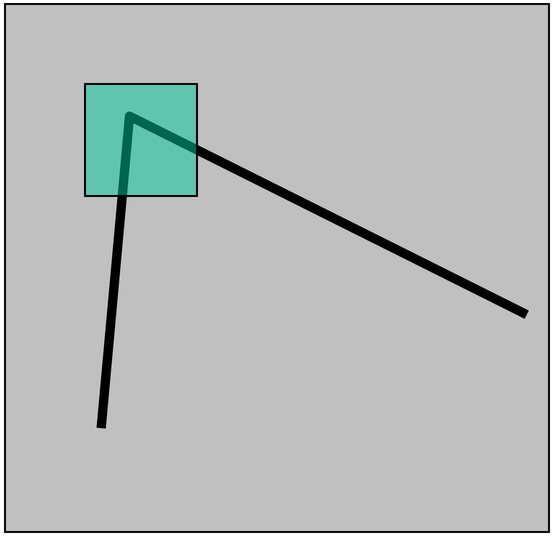
Harris corner detector



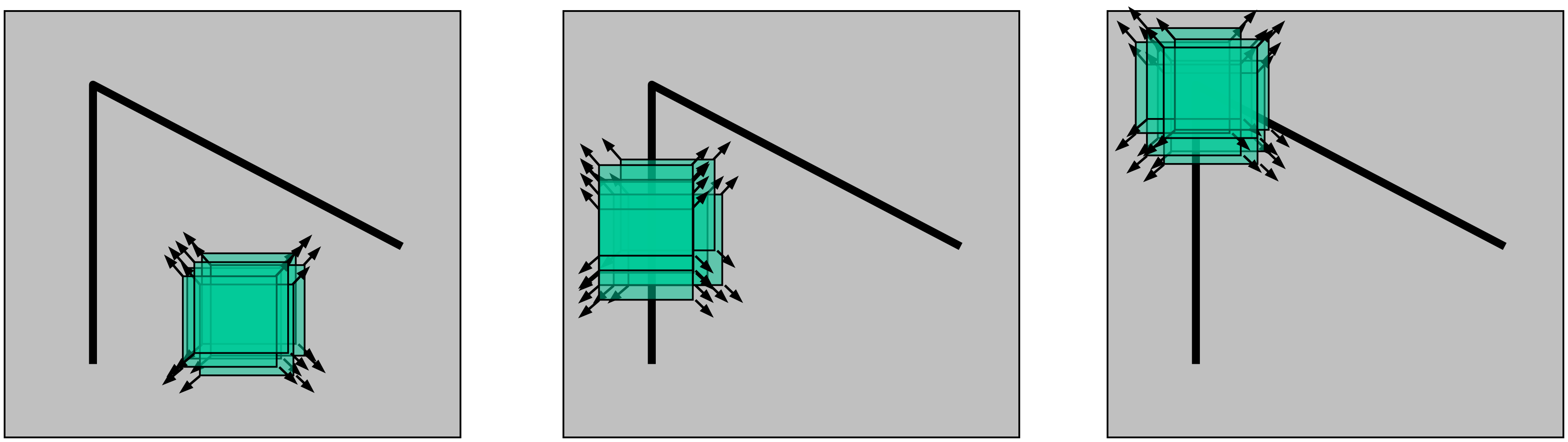
C.Harris, M.Stephens. “A Combined Corner and Edge Detector”. 1988

The Basic Idea

- We should easily localize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



Harris Detector: Basic Idea



"flat" region:
no change as shift window in all direc tions

"edge":
no change as shift window along the edge direction

"corner":
significant change as shift window in all dire ctions

Harris Detector: Mathematics

Window-averaged change of intensity induced by shifting the image data by [u,v]:

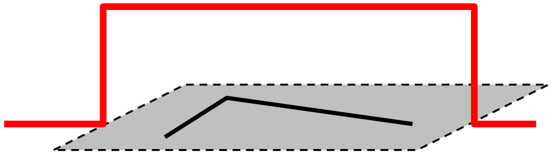
$$E(u,v)=\sum_{x,y} w(x,y)\left[I(x+u,y+v)-I(x,y)\right]^2$$

Window function

Shifted intensity

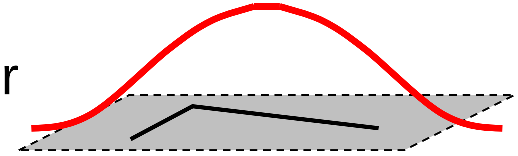
Intensity

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Taylor series approx to shifted image

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

$$= \sum_{x,y} w(x,y)[uI_x + vI_y]^2$$

$$= \sum_{x,y} w(x,y)(u \quad v)\begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}\begin{pmatrix} u \\ v \end{pmatrix}$$

Harris Detector: Mathematics

Expanding I(x,y) in a Taylor series expansion, we have, for small shifts $[u,v]$, a *bilinear* approximation:

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

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 is also called "structure tensor"

Harris Detector: Mathematics

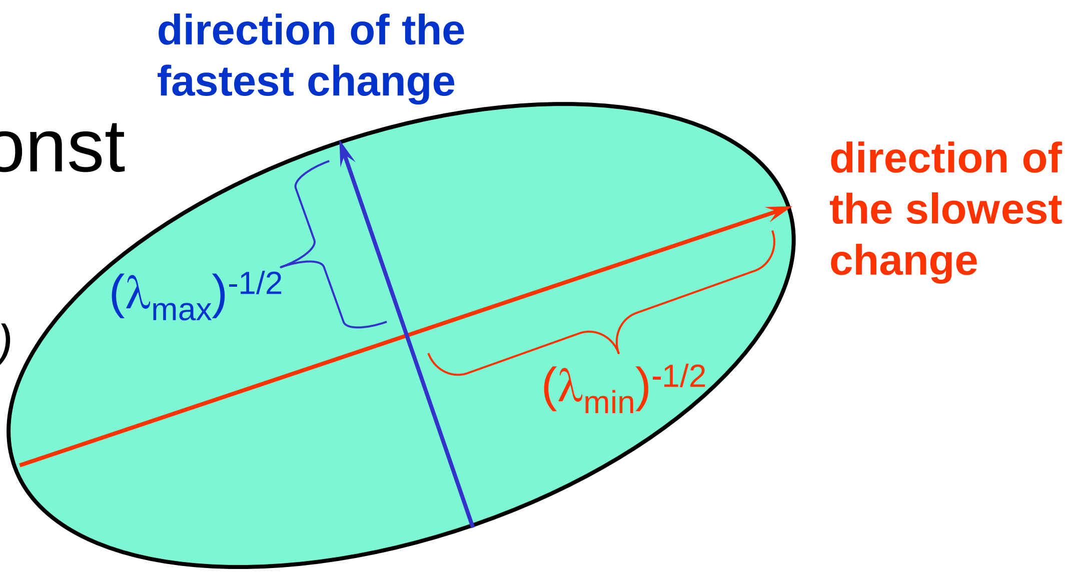
Intensity change in shifting window: eigenvalue analysis

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

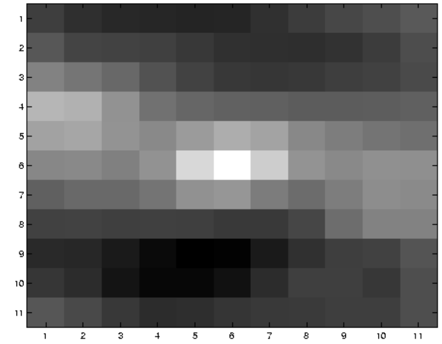
λ_1, λ_2 – eigenvalues of M

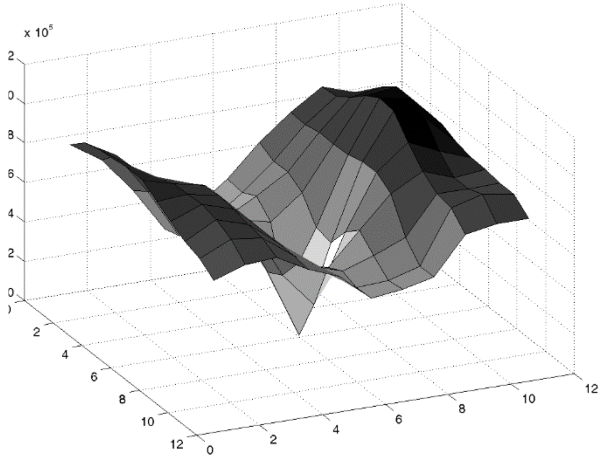
Ellipse $E(u,v)$ = const

Iso-intensity contour of $E(u,v)$



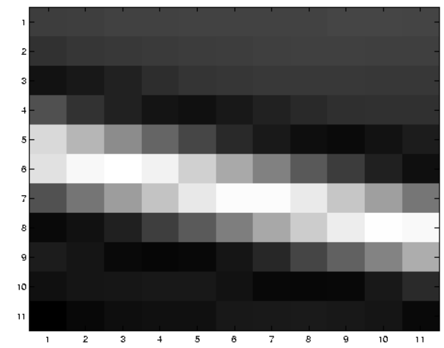
Selecting Good Features

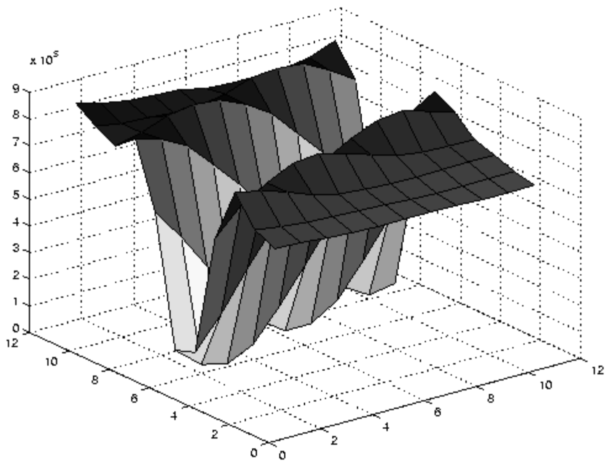




λ_1 and λ_2 are large

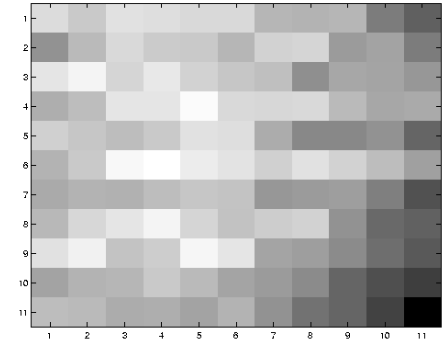
Selecting Good Features

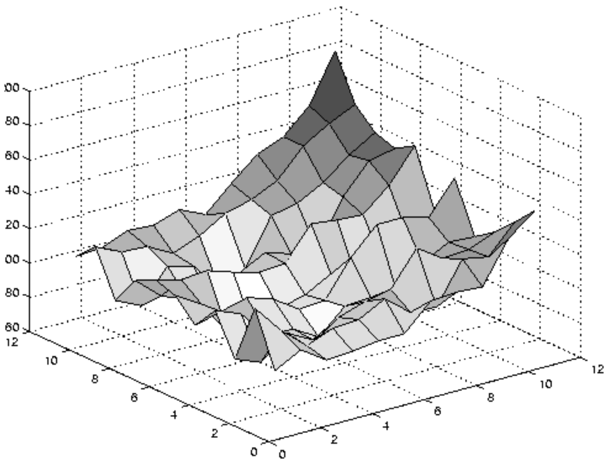




large λ_1, small λ_2

Selecting Good Features

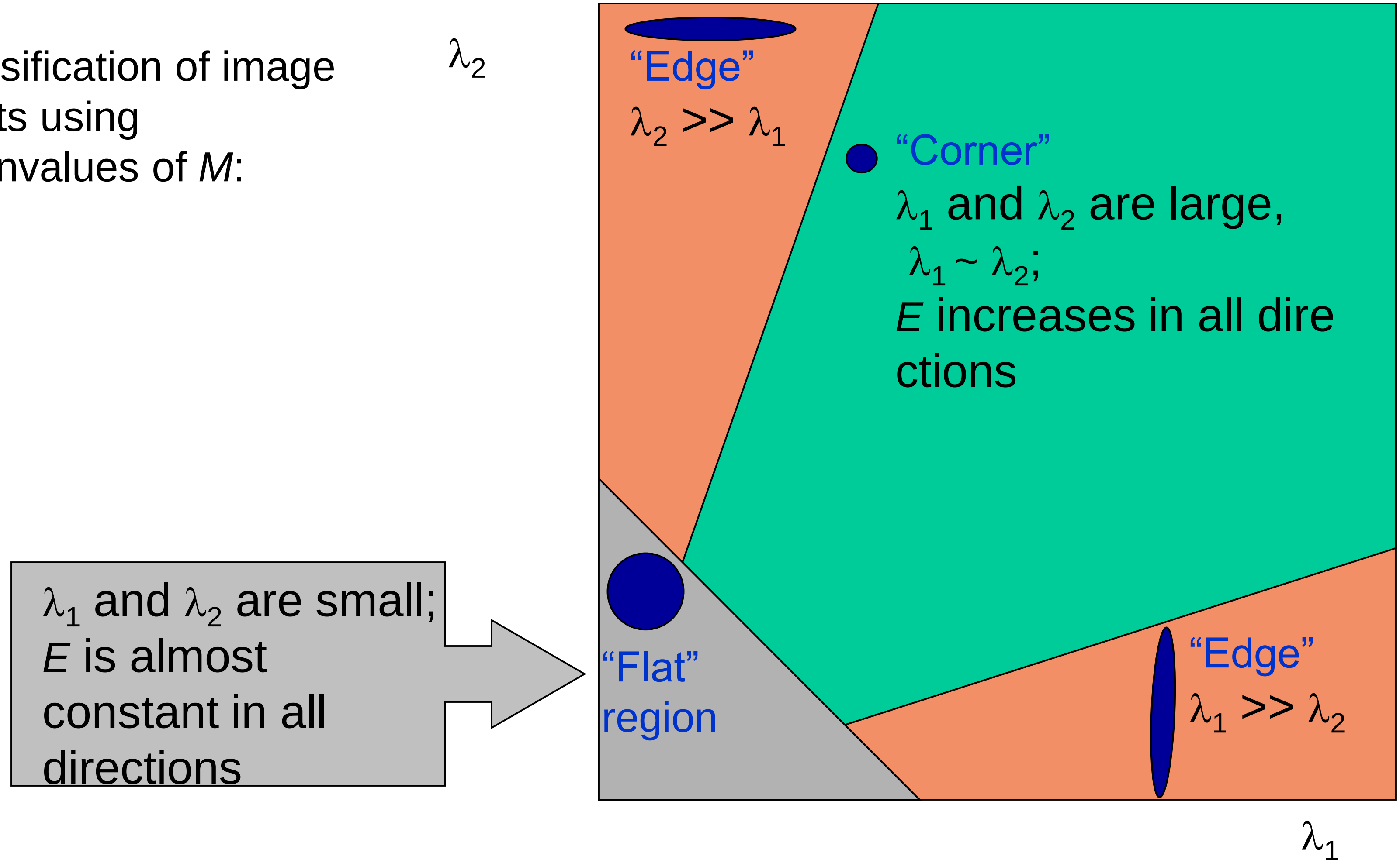




small λ_1, small λ_2

Harris Detector: Mathematics

Classification of image points using eigenvalues of *M*:



Harris Detector: Mathematics

Measure of corner response:

$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

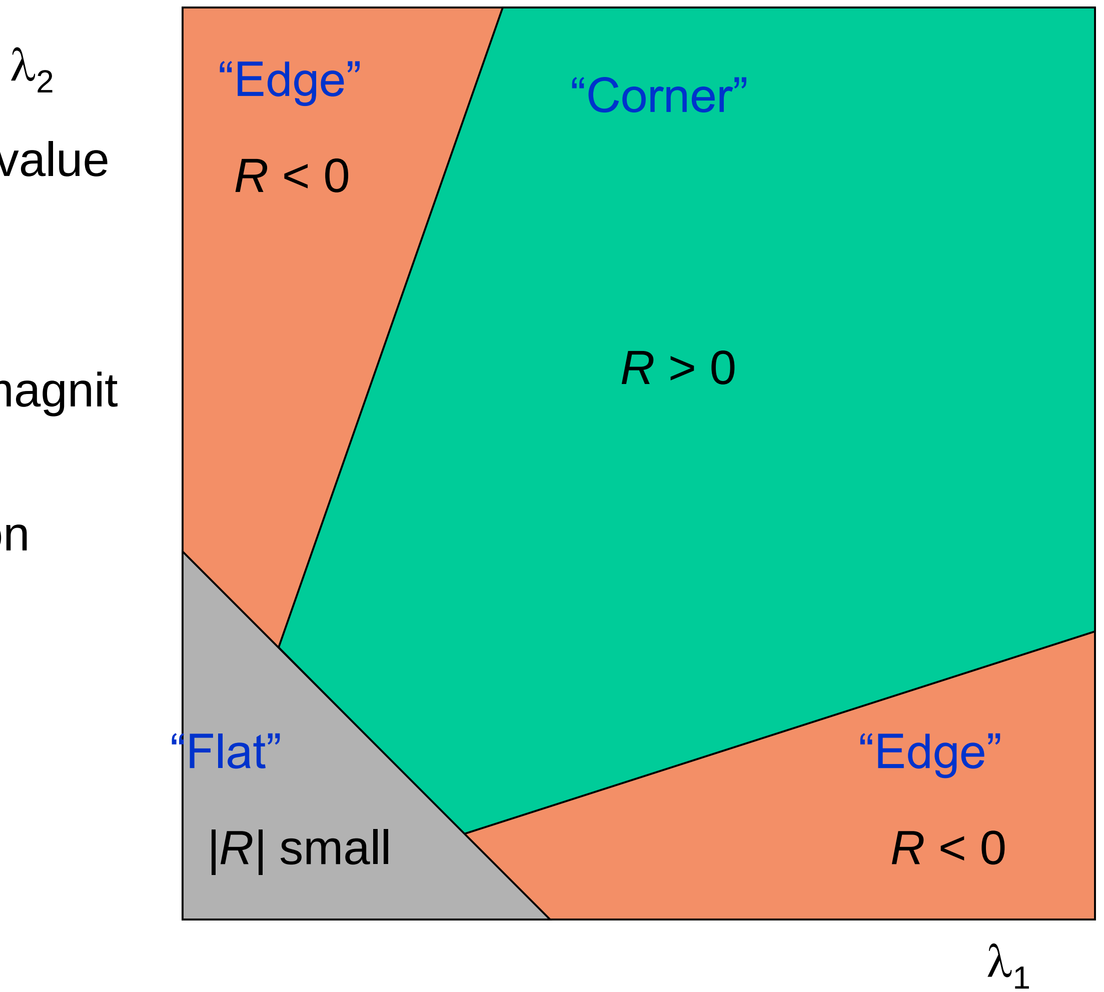
This expression does not requires computing the eigenvalues.

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

(*k* – empirical constant, *k* = 0.04-0.06)

Harris Detector: Mathematics

- R depends only on eigenvalues of M
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region



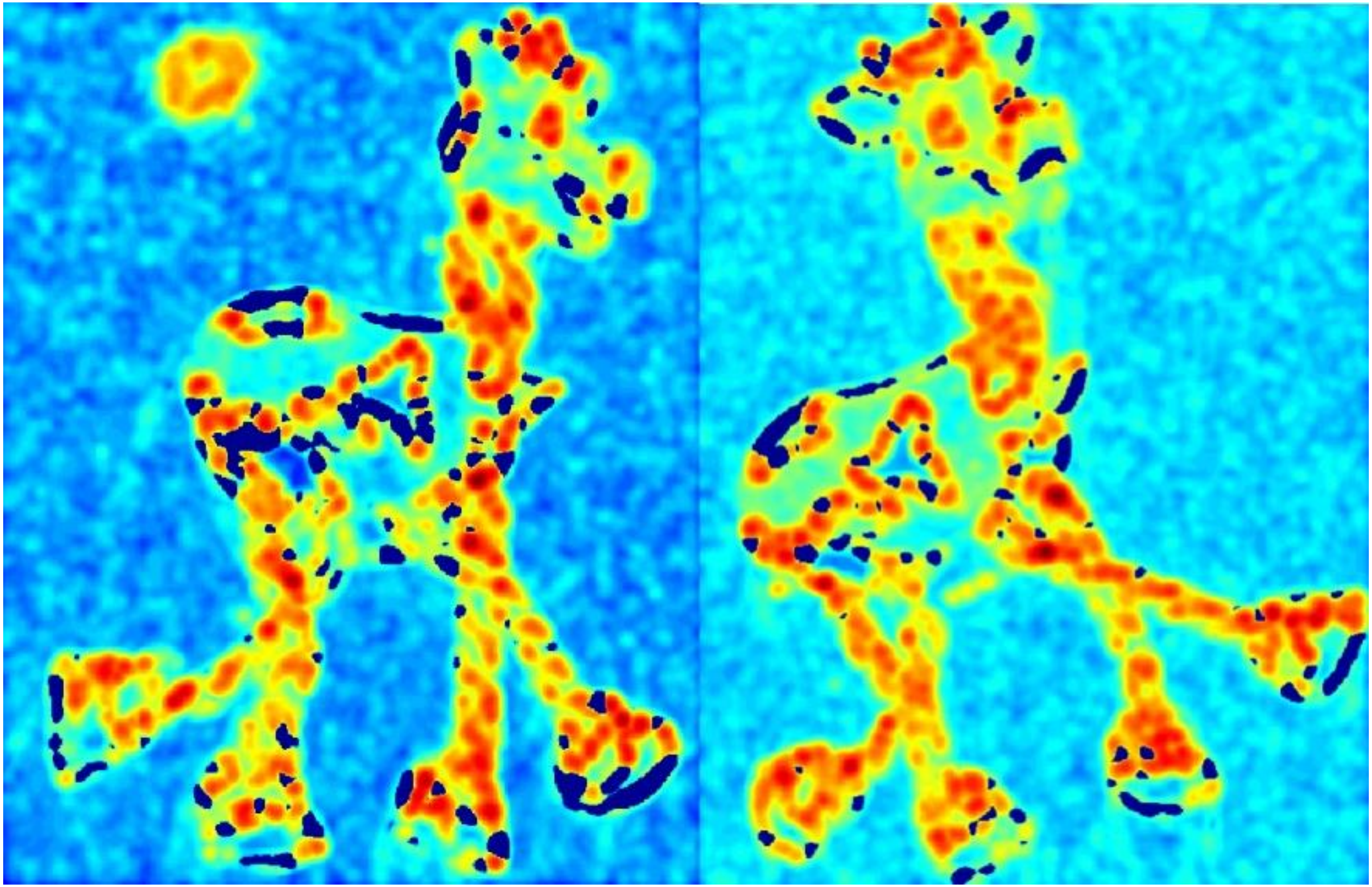
Harris Detector

- The Algorithm:
  - Find points with large corner response function R ($R >$ threshold)
  - Take the points of local maxima of R

Harris Detector: Workflow

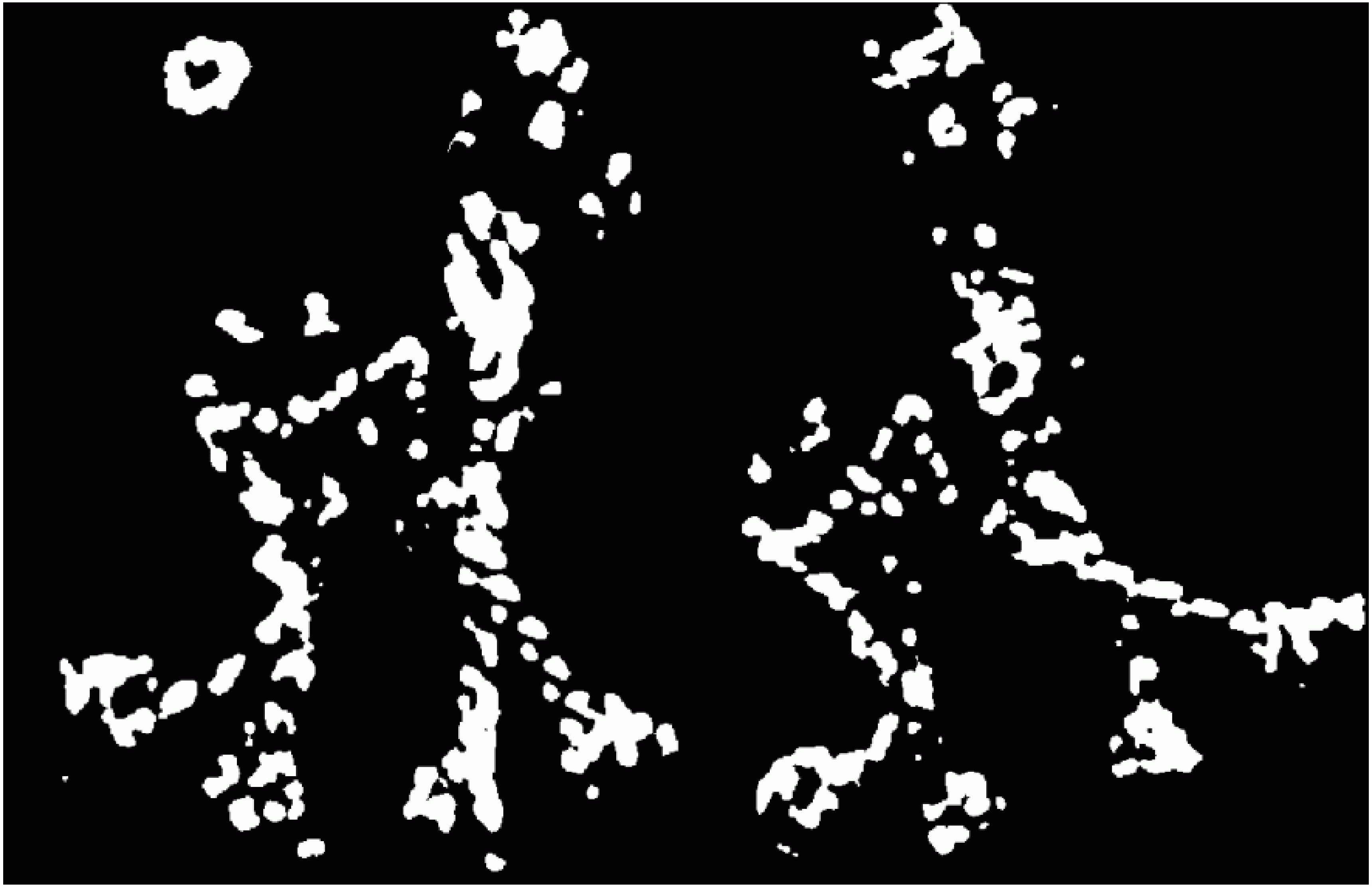
Harris Detector: Workflow

Compute corner response R



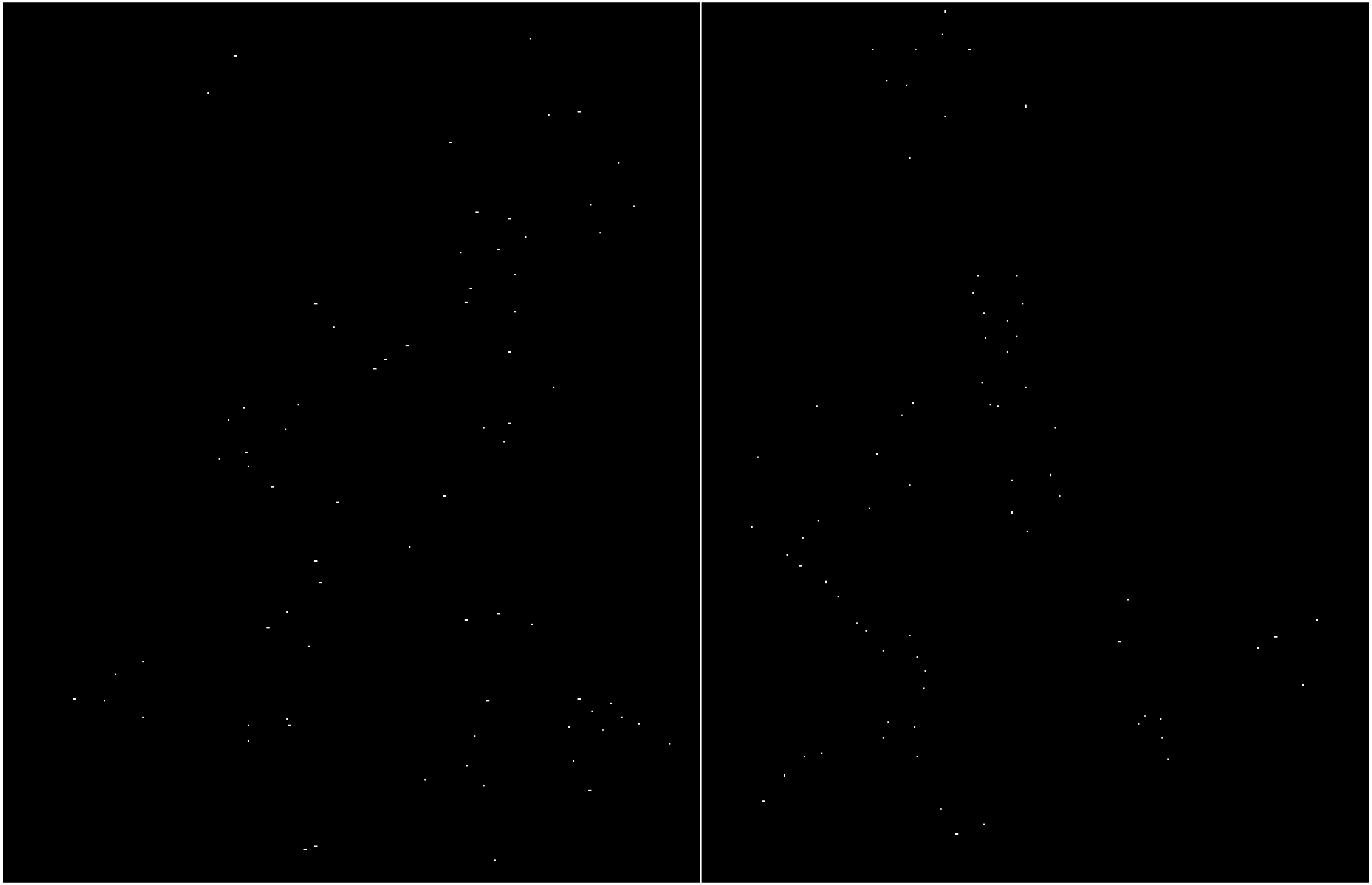
Harris Detector: Workflow

Find points with large corner response: R>threshold



Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector: Summary

- Average intensity change in direction $[u, v]$ can be expressed as a bilinear form:

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

- Describe a point in terms of eigenvalues of M: *measure of corner response*

$$R = \lambda_1 \lambda_2 - k\left(\lambda_1 + \lambda_2\right)^2$$

- A good (corner) point should have a *large intensity change* in *all directions*, i.e. R should be large positive

Ideal feature detector

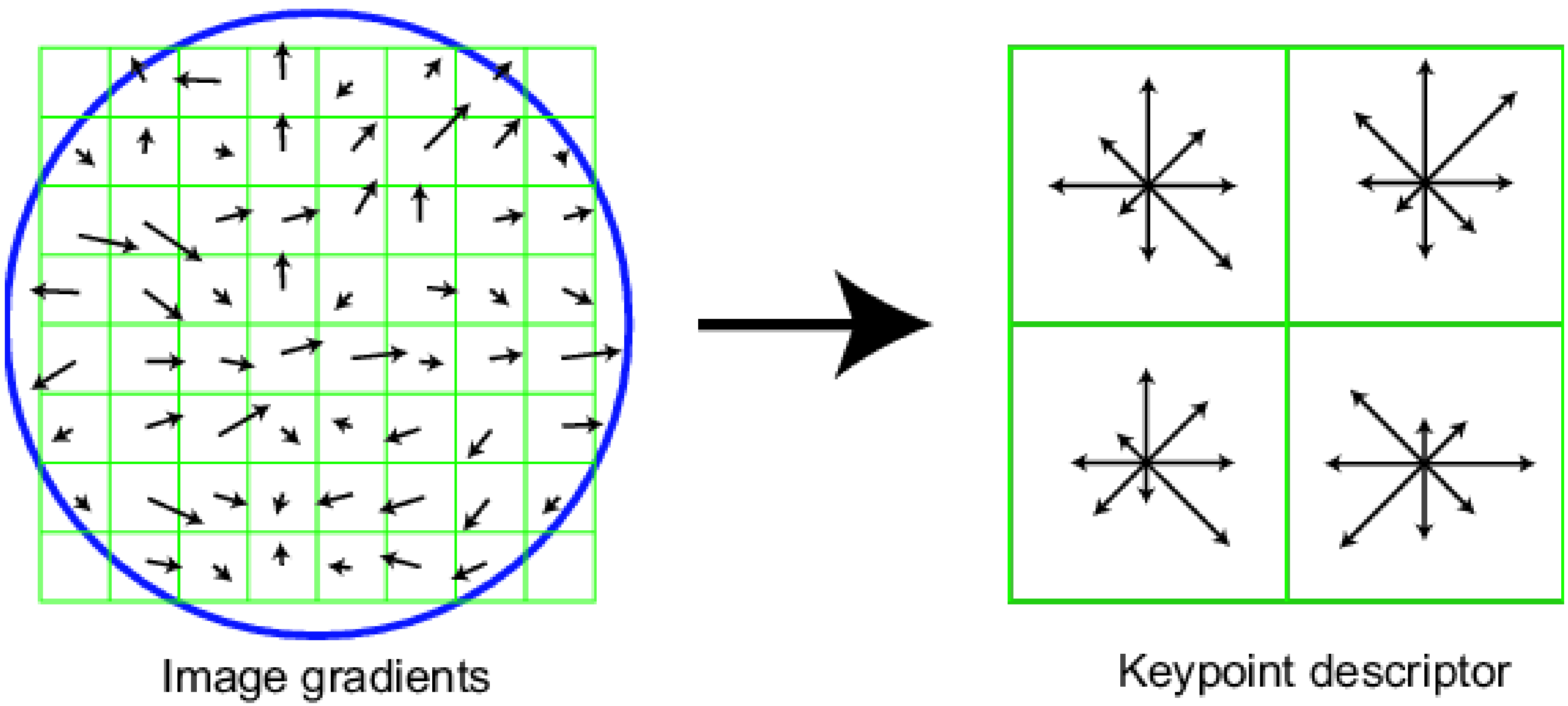
- Would always find the same point on an object, regardless of changes to the image.
- i.e, insensitive to changes in:
  - Scale
  - Lighting
  - Perspective imaging
  - Partial occlusion

We want to:

detect *the same* interest points
regardless of *image changes*

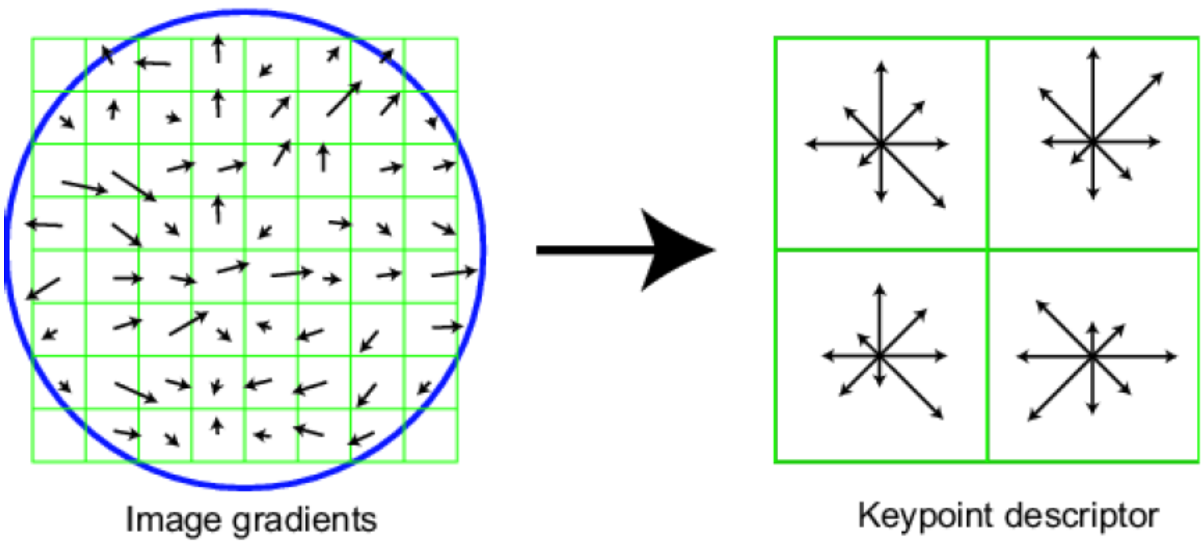
SIFT Descriptor

- Basic idea:
  - Take 16x16 square window around detected feature
  - Compute edge orientation (angle of the gradient -90) for each pixel
  - Throw out weak edges (threshold gradient magnitude)
  - Create histogram of surviving edge orientations



SIFT Descriptor

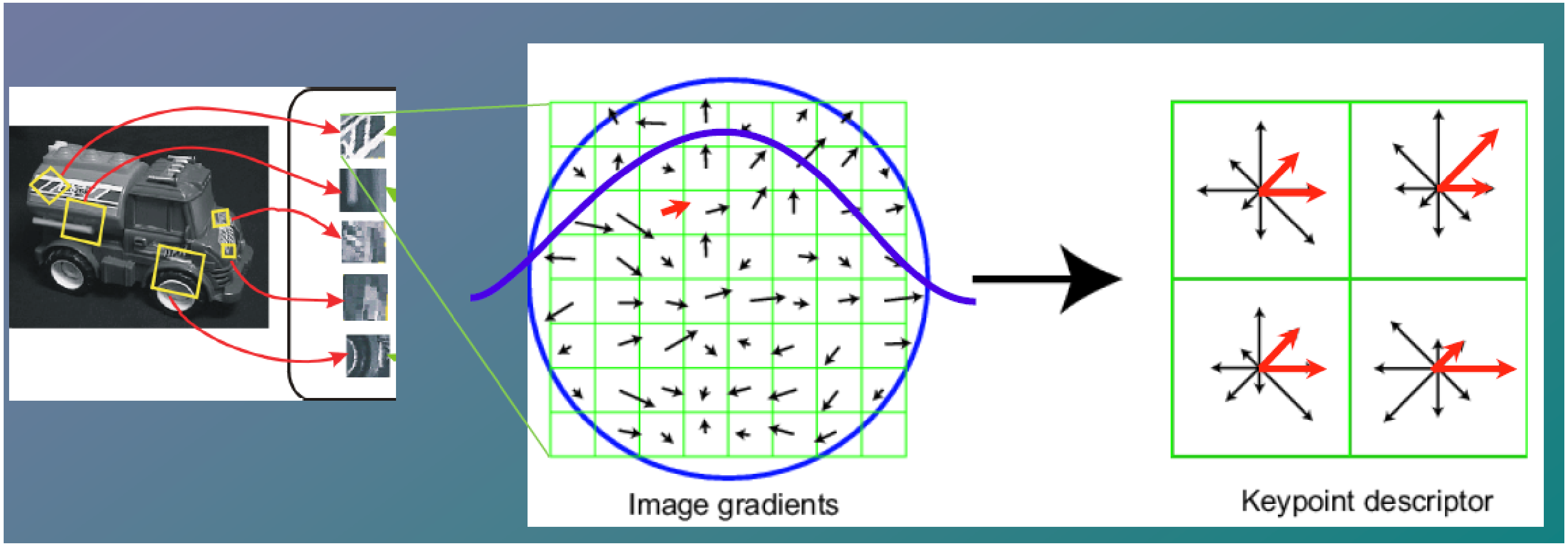
Full version
•Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
•Compute an orientation histogram for each cell
•16 cells * 8 orientations = 128 dimensional descriptor



showing only 2x2 here but is 4x4

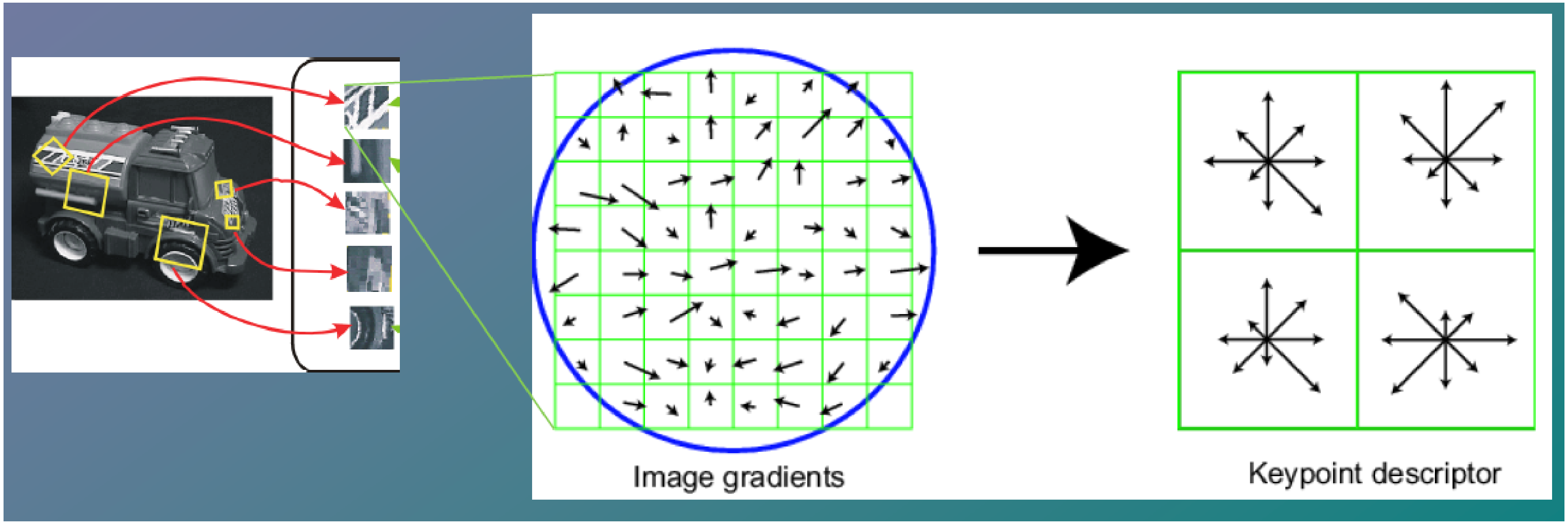
SIFT Descriptor

- Gaussian weight
- Trilinear interpolation
  –a given gradient contributes to 8 bins: 4 in space times 2 in orientation

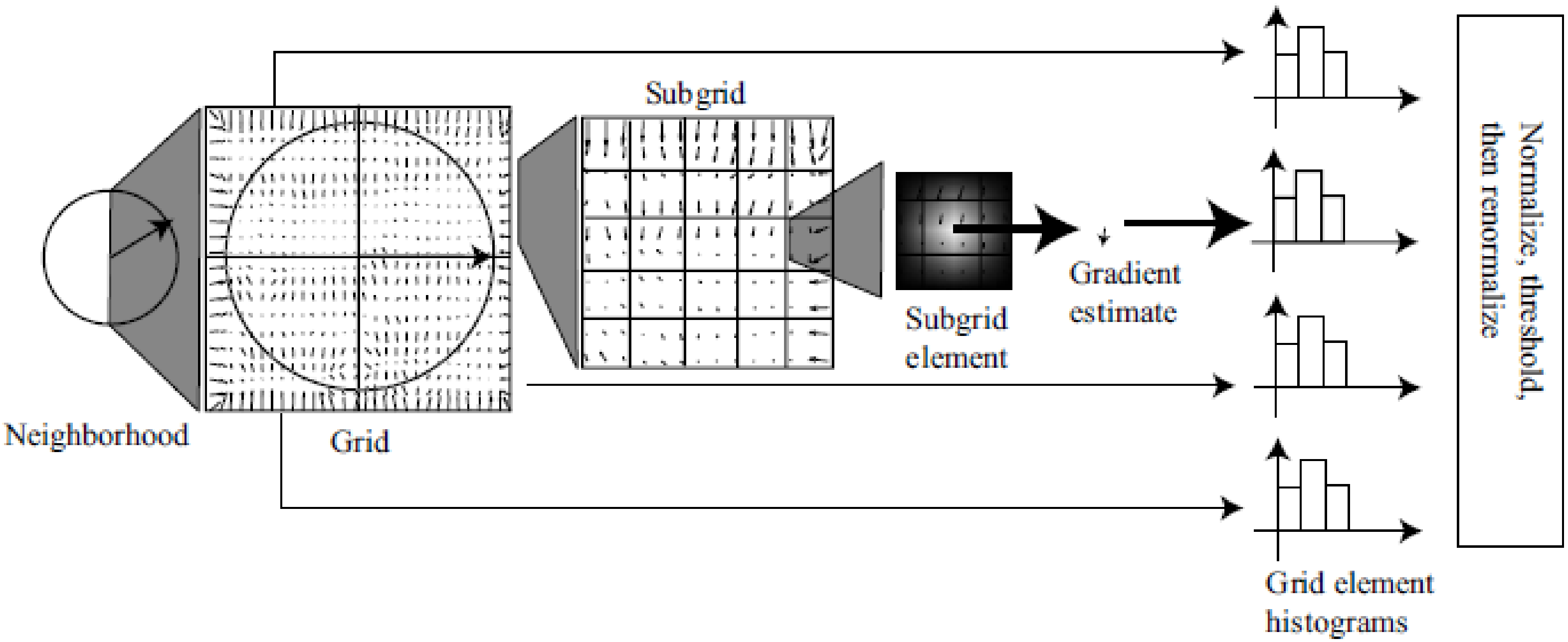


SIFT Descriptor

- 128-dim vector normalized to 1
- Threshold gradient magnitudes to avoid excessive influence of high gradients
  - –after normalization, clamp gradients >0.2
  - –renormalize



SIFT Descriptor

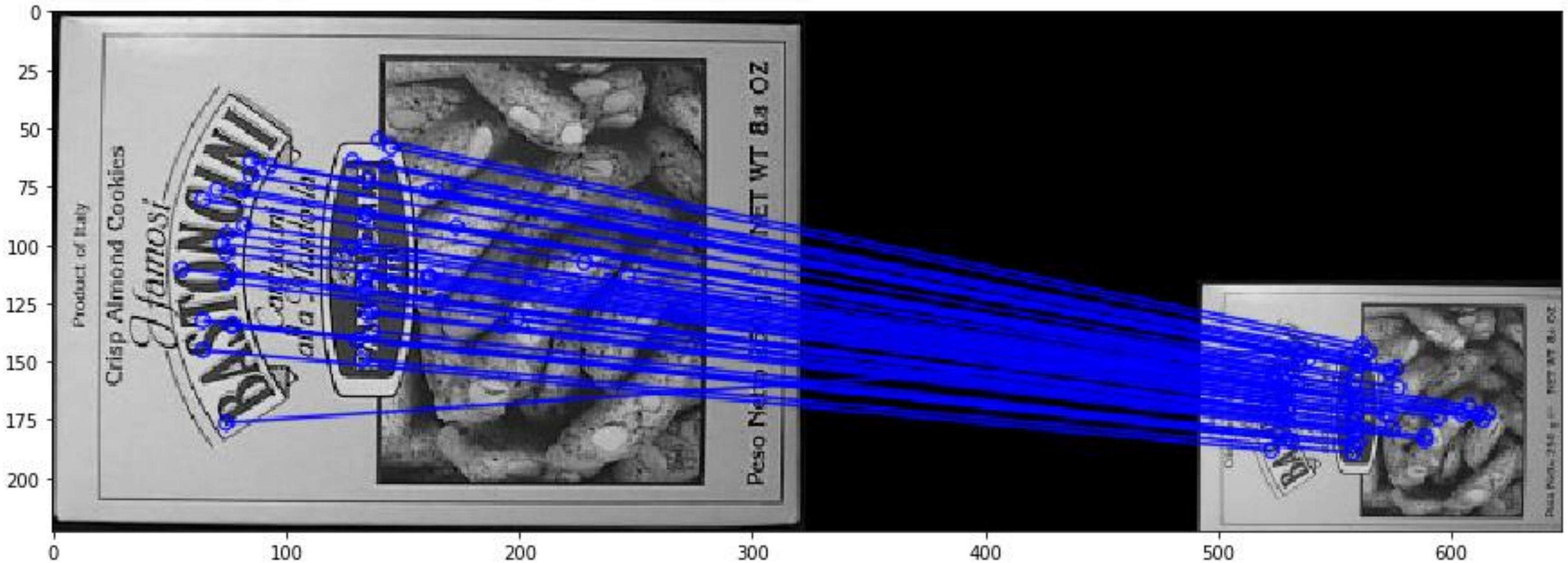


SIFT Descriptor

Given an image $\mathcal{I}$, and a patch with center (x_c, y_c),
radius r, orientation θ, and parameters n, m, q, k and t.
For each element of the $n \times n$ grid centered at (x_c, y_c) with spacing kr
Compute a weighted q element histogram of the averaged
gradient samples at each point of the $m \times m$ subgrid,
as in Algorithm 5.5.
Form an $n \times n \times q$ vector v by concatenating the histograms.
Compute $u = v / \sqrt{v \cdot v}$.
Form w whose i'th element w_i is $\min(u_i, t)$.
The descriptor is $d = w / \sqrt{w \cdot w}$.

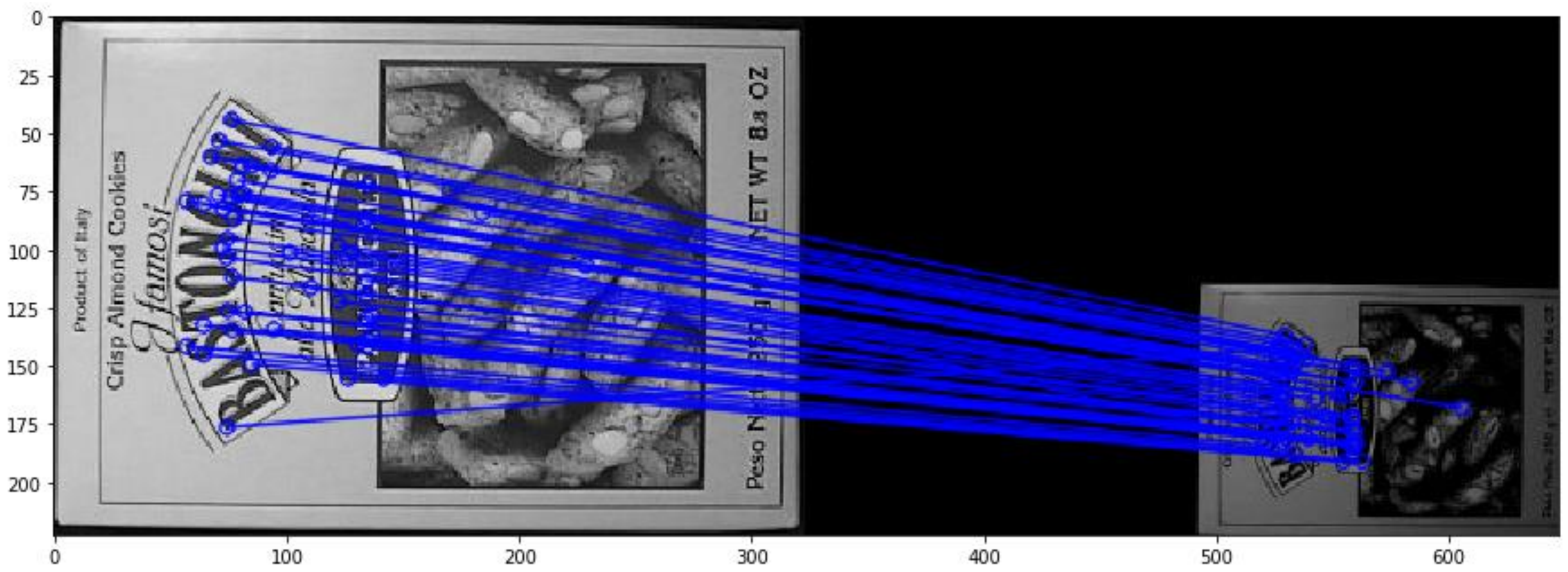
Example for scale invariance:

- Scale variation is just piece of cake!



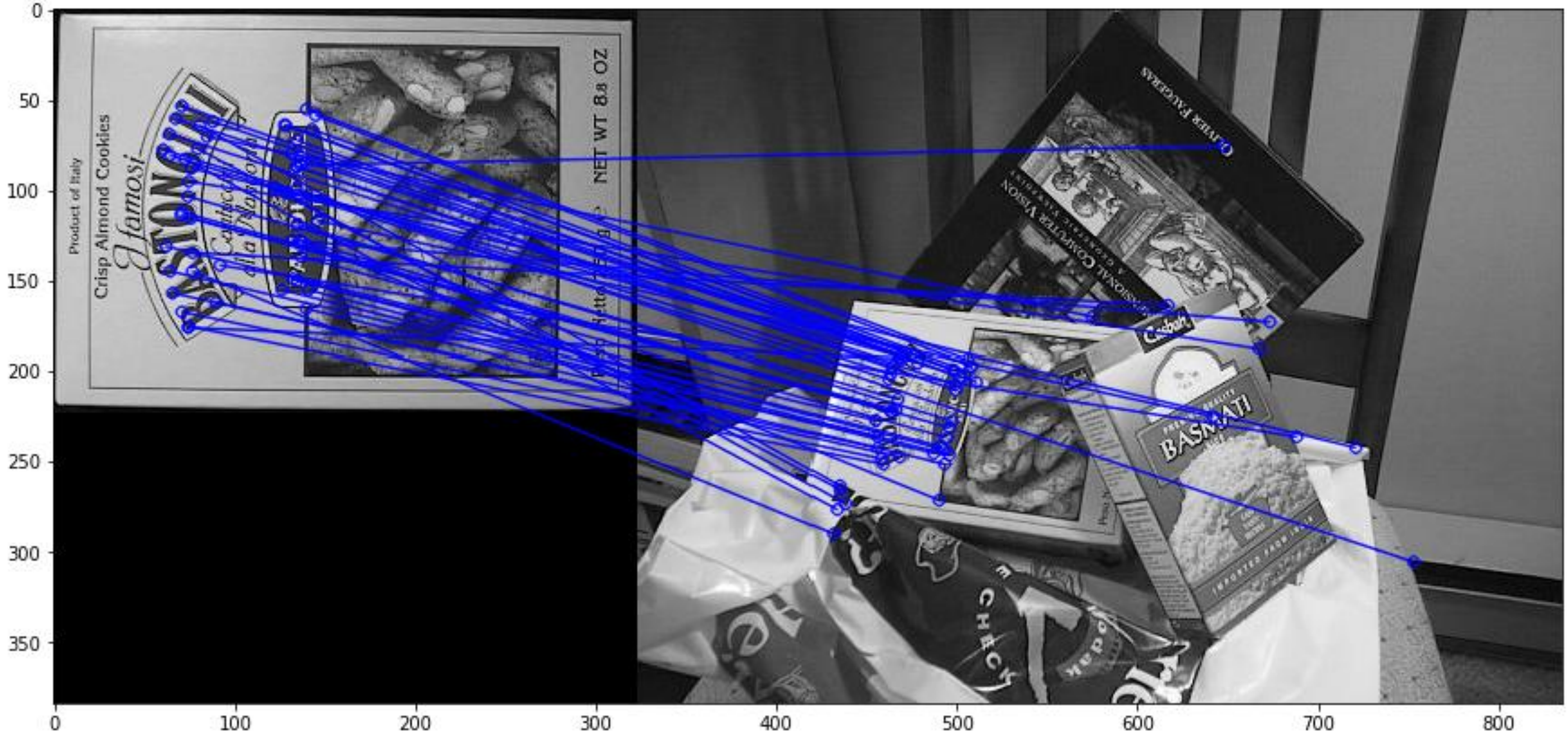
Example for color invariance:

- Scale + color variation is still piece of cake!



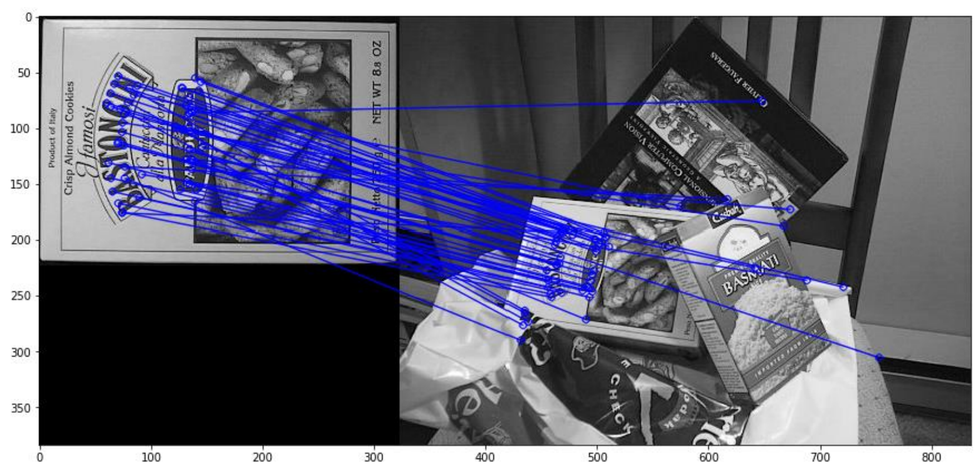
Example for scale/color/geometrical invariance:

- Scale + color + geometrical variation is still piece of cake!



~~SIFT~~ ORB + homography without RANSAC

- No corners need to be manually assigned
- ~~SIFT~~ ORB would lead to perform the transformation between the original image and the target image
- However, the transformation would fail every time. Do you know why?

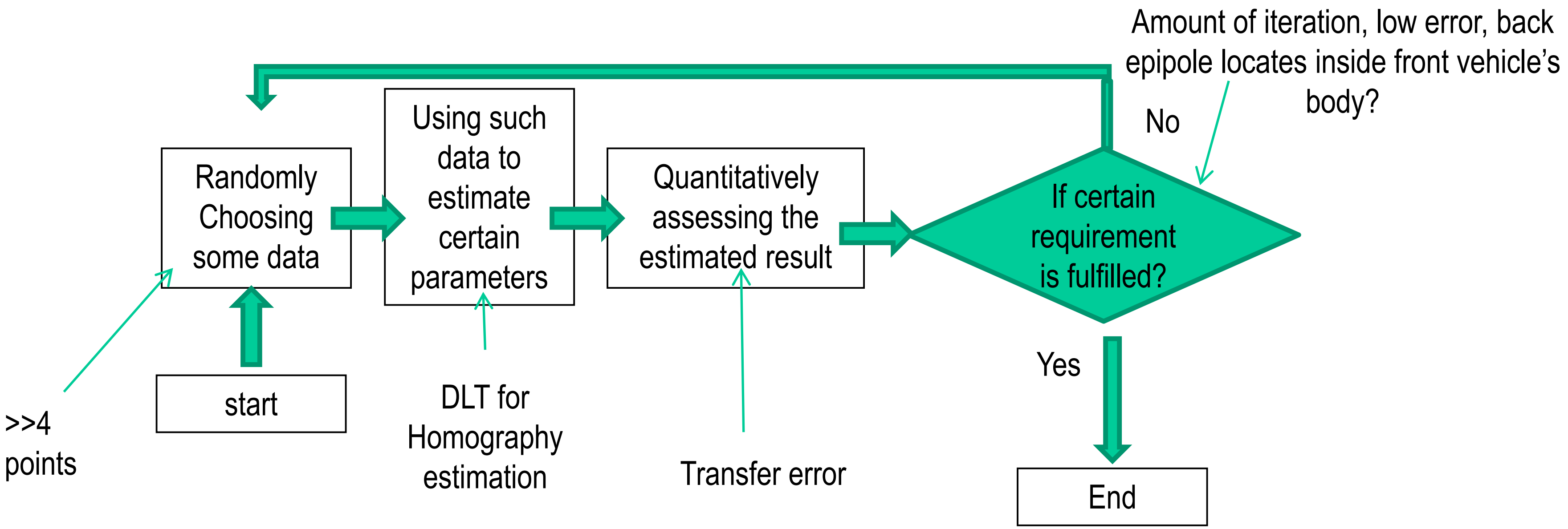


Before transformation

After transformation

How to obtain optimal H?

- The overall algorithm:
  - Using RANSAC+ Homography estimation
  - Input: feature correspondences,
  - Output: Optimal H



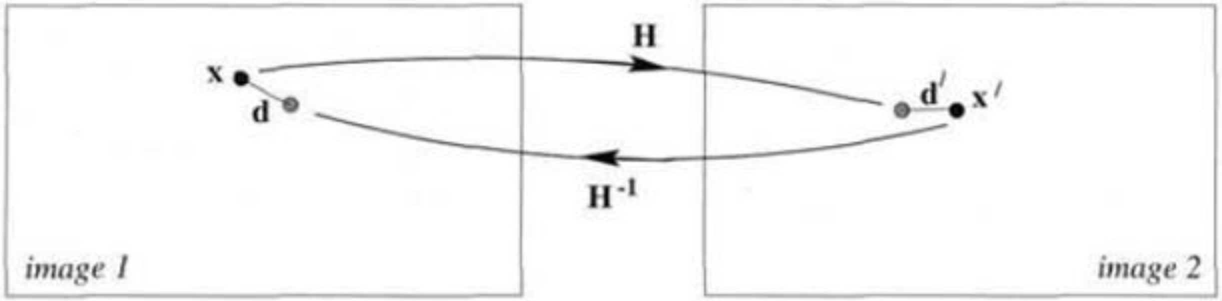
Cost function of homography estimation

- Algebraic error:
  - The standard SVD approach to minimize the norm ||Ah|| under the constraint ||h||=1
- Transfer error:
  - for a given correspondence (x_i , x_i'), iterate to find H that minimizes the following cost function with LM optimization given the initial estimate from algebraic error
- $\sum_i d(x_i', Hx_i)^2$
- Symmetric transfer error:
  - for a given correspondence (x_i , x_i'), iterate to find H that minimizes the following cost function with LM optimization given the initial estimate from algebraic error
- $\sum_i d(x_i, H^{-1}x_i')^2 + d(x_i', Hx_i)^2$

45

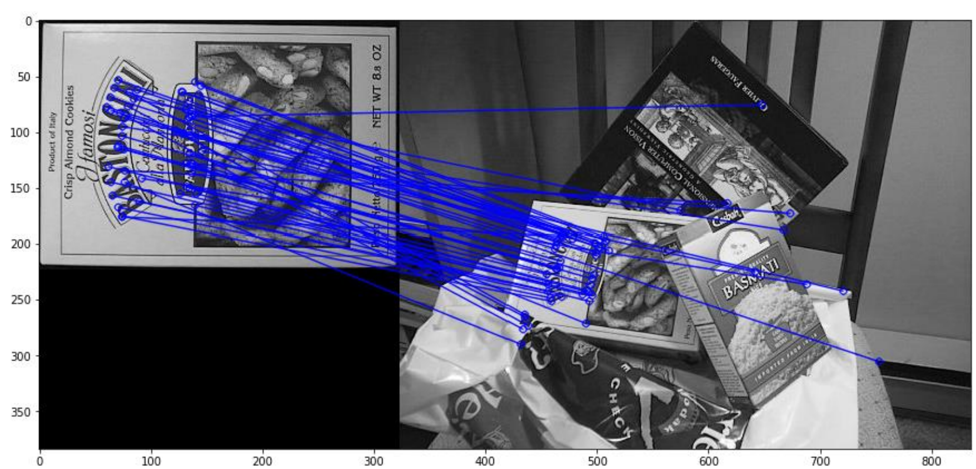
Cost function of homography estimation (contd)

Symmetric transfer error:



~~SIFT~~ ORB + homography with RANSAC = object identification

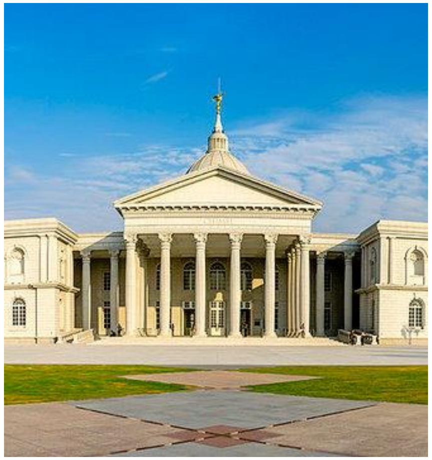
- The number of RANSAC is important but how large should we set?
- Giving more iterations is simply the answer if your application is not necessary to function in real-time.



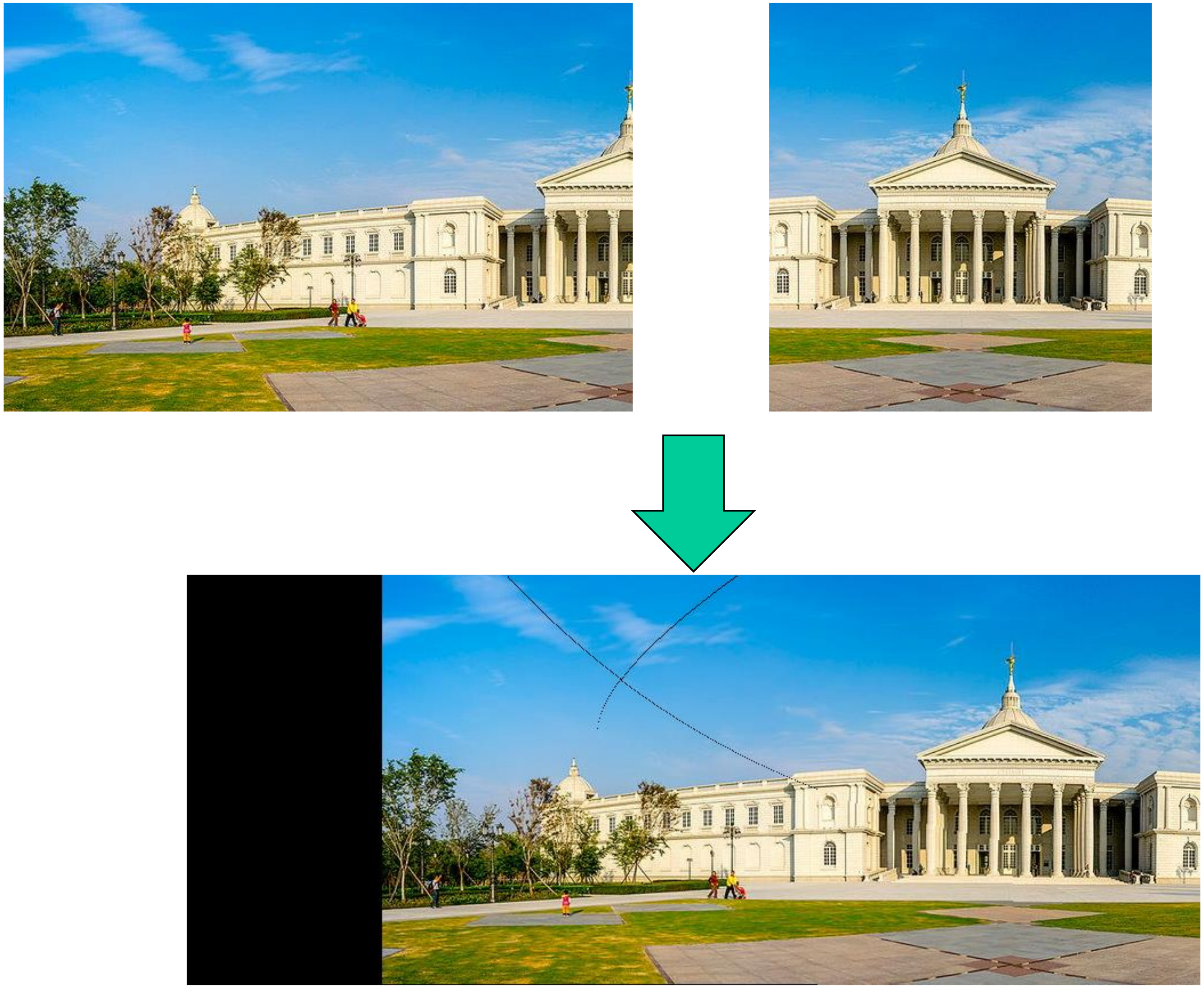
Before transformation

After transformation

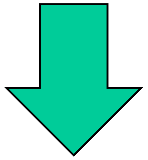
Panorama example: ~~SIFT~~ ORB + homography with RANSAC to produce Panorama



Panorama example: ~~SIFT~~ ORB + homography with RANSAC to produce Panorama



Panorama example: ~~SIFT~~ ORB + homography with RANSAC to produce Panorama



Panorama implementation review

- Could it be more efficient?
- Is it possible to control the seam line?
- Where is the best seam line?
- What if there is a moving object inside?
- Is it possible to alleviate parallax?
- What if the brightness of each image is different to some extent?
- What if the image is captured by a wide angle camera?

Without color blending

With color blending