Image Processing

Cosimo Distanete

Lecture 12: Local Features
Last Time: Image Pyramids

- Image Pyramids
- Blending
Today: Feature Detection and Matching

• Local features
• Pyramids for invariant feature detection
• Invariant descriptors
• Matching
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
Local features and alignment

- 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?
Local features and alignment

- Detect feature points in both images
Local features and alignment

- Detect feature points in both images
- Find corresponding pairs

[Darya Frolova and Denis Simakov]
Local features and alignment

• Detect feature points in both images
• Find corresponding pairs
• Use these pairs to align images
Local features and alignment

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

We need a repeatable detector

[Darya Frolova and Denis Simakov]
Local features and alignment

• Problem 2:
  – For each point correctly recognize the corresponding one

We need a reliable and distinctive descriptor

[Darya Frolova and Denis Simakov]
Geometric transformations
Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial
And other nuisances…

- Noise
- Blur
- Compression artifacts
- …
Invariant local features

Subset of local feature types designed to be invariant to common geometric and photometric transformations.

Basic steps:
1) Detect distinctive interest points
2) Extract invariant descriptors

Figure: David Lowe
Main questions

• Where will the interest points come from?
  – What are salient features that we’ll detect in multiple views?

• How to describe a local region?

• How to establish correspondences, i.e., compute matches?
Figure 4.3: Image pairs with extracted patches below. Notice how some patches can be localized or matched with higher accuracy than others.
Finding Corners

Key property: in the region around a corner, image gradient has two or more dominant directions

Corners are repeatable and distinctive

C.Harris and M.Stephens. "A Combined Corner and Edge Detector."
Corners as distinctive interest points

We should easily recognize the point by looking through a small window.

Shifting a window in *any direction* should give a *large change* in intensity.

"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions
Harris Detector formulation

Change of intensity for the shift \([u,v]\):

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

Window function \(w(x,y)\) =

1 in window, 0 outside

or

Gaussian

Source: R. Szeliski
Consider shifting the window $W$ by $(u,v)$

- how do the pixels in $W$ change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD “error” of $E(u,v)$:

$$E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2$$
Small motion assumption

Taylor Series expansion of $I$:

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

If the motion $(u,v)$ is small, then first order approx is good

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

$$\approx I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

shorthand: $I_x = \frac{\partial I}{\partial x}$

Plugging this into the formula on the previous slide…
Consider shifting the window $W$ by $(u,v)$

- how do the pixels in $W$ change?
- compare each pixel before and after by summing up the squared differences
- this defines an “error” of $E(u,v)$:

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

$$\approx \sum_{(x,y)\in W} [I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x, y)]^2$$

$$\approx \sum_{(x,y)\in W} [[I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix}]^2$$
Feature detection: the math

This can be rewritten:

$$E(u, v) = \sum_{(x, y) \in W} [u \ v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} [u \ v]$$

For the example above

- You can move the center of the green window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest $E$ values?
- We can find these directions by looking at the eigenvectors of $M$
Harris Detector formulation

This measure of change can be approximated by:

\[ E(u, v) \approx [u \ v] M [u \ v] \]

where \( M \) is a 2x2 matrix computed from image derivatives:

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

Sum over image region – area we are checking for corner

Gradient with respect to \( x \), times gradient with respect to \( y \)
Harris Detector formulation

where $M$ is a $2 \times 2$ matrix computed from image derivatives:

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

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

Sum over image region – area we are checking for corner

$$ M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \ I_y] $$
Quick eigenvalue/eigenvector review

The **eigenvectors** of a matrix \( A \) are the vectors \( x \) that satisfy:

\[
Ax = \lambda x
\]

The scalar \( \lambda \) is the **eigenvalue** corresponding to \( x \)

- The eigenvalues are found by solving:

\[
det(A - \lambda I) = 0
\]

- In our case, \( A = M \) is a 2x2 matrix, so we have

\[
\det \begin{bmatrix}
m_{11} - \lambda & m_{12} \\
m_{21} & m_{22} - \lambda
\end{bmatrix} = 0
\]

- The solution:

\[
\lambda_{\pm} = \frac{1}{2} \left[ (m_{11} + m_{22}) \pm \sqrt{(m_{11} + m_{22})^2 - 4m_{12}m_{21} + (m_{11} - m_{22})^2} \right]
\]

Once you know \( \lambda \), you find \( x \) by solving

\[
\begin{bmatrix}
m_{11} - \lambda & m_{12} \\
m_{21} & m_{22} - \lambda
\end{bmatrix} \begin{bmatrix}
x \\
y
\end{bmatrix} = 0
\]
Feature detection: the math

\[ E(u, v) = \sum_{(x, y) \in W} [u \ v] \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \]

\[ M \]

Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- \( x_+ \) = direction of largest increase in E.
- \( \lambda_+ \) = amount of increase in direction \( x_+ \)
- \( x_- \) = direction of smallest increase in E.
- \( \lambda_- \) = amount of increase in direction \( x_- \)

\[ Mx_+ = \lambda_+ x_+ \]

\[ Mx_- = \lambda_- x_- \]
What does this matrix reveal?

First, consider an axis-aligned corner:
What does this matrix reveal?

First, consider an axis-aligned corner:

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

This means dominant gradient directions align with x or y axis

If either \( \lambda \) is close to 0, then this is **not** a corner, so look for locations where both are large.

What if we have a corner that is not aligned with the image axes?

Slide credit: David Jacobs
General Case

Since $M$ is symmetric, we have

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

We can visualize $M$ as an ellipse with axis lengths determined by the eigenvalues and orientation determined by $R$.

Slide adapted from Darya Frolova, Denis Simakov.
Interpreting the eigenvalues

Classification of image points using eigenvalues of $M$:

- **“Corner”**
  - $\lambda_1$ and $\lambda_2$ are large,
  - $\lambda_1 \sim \lambda_2$;
  - $E$ increases in all directions

- **“Edge”**
  - $\lambda_1 \gg \lambda_2$ (blue circle)

- **“Flat”** region
  - $\lambda_1$ and $\lambda_2$ are small;
  - $E$ is almost constant in all directions

- **“Edge”**
  - $\lambda_1 \gg \lambda_2$ (blue circle)
Corner response function

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

\( \alpha \): constant (0.04 to 0.06)
Harris Corner Detector

- Algorithm steps:
  - Compute M matrix within all image windows to get their R scores
  - Find points with large corner response \((R > \text{threshold})\)
  - Take the points of local maxima of \(R\)
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
Harris Detector: Properties

• Rotation invariance

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

Corner response $R$ is invariant to image rotation
Harris Detector: Properties

- Not invariant to image scale

All points will be classified as edges

Corner!
• How can we detect **scale invariant** interest points?
How to cope with transformations?

• Exhaustive search
• Invariance
• Robustness
Exhaustive search

• Multi-scale approach

Slide from T. Tuytelaars ECCV 2006 tutorial
Exhaustive search

• Multi-scale approach
Exhaustive search

- Multi-scale approach
Exhaustive search

• Multi-scale approach
Invariance

- Extract patch from each image individually
Automatic scale selection

• Solution:
  – Design a function on the region, which is “scale invariant” (the same for corresponding regions, even if they are at different scales)
    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.
  – For a point in one image, we can consider it as a function of region size (patch width)
Automatic scale selection

- Common approach:

  Take a local maximum of this function

  Observation: region size, for which the maximum is achieved, should be *invariant* to image scale.

**Important:** this scale invariant region size is found in each image *independently*!

![Graphs showing scale selection](image)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

$$f(I_{i_{1..m}}(x, \sigma))$$

$$f(I_{i_{1..m}}(x', \sigma))$$
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Automatic Scale Selection

- Function responses for increasing scale (scale signature)

$$f(I_{k...m}(x, \sigma))$$

$$f(I_{k...m}(x', \sigma))$$
Automatic Scale Selection

- Function responses for increasing scale (scale signature)
Scale selection

- Use the scale determined by detector to compute descriptor in a normalized frame

[Images from T. Tuytelaars]
What Is A Useful Signature Function?

- Laplacian-of-Gaussian = “blob” detector
We define the *characteristic scale* as the scale that produces peak of Laplacian response.

Scale-space blob detection

$L_{xx}(\sigma) + L_{yy}(\sigma) \rightarrow \sigma$,

Squared filter response maps

$\Rightarrow$ List of $(x, y, \sigma)$
Scale-space blob detector: Example

Source: Lana Lazebnik
Scale-space blob detector: Example

sigma = 11.9912

Source: Lana Lazebnik
Scale-space blob detector: Example

Source: Lana Lazebnik