Image features

Aykut Erdem
Dept. of Computer Engineering
Hacettepe University
Uses for feature point detectors and descriptors in computer vision and graphics.

- Image alignment and building panoramas
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- ... other

Slide credit: B. Freeman and A. Torralba
Example 1: Wide baseline matching

- Establish correspondence between two (or more) images
- Useful in visual geometry: Camera calibration, 3D reconstruction, Structure and motion estimation, ...

Local transf: scale/affine – Detector: affine-Harris Descriptor: SIFT
Example 2: Panoramic mosaic

Slide credit: Jiri Matas & Andrew Zisserman
Example 2: Image Stitching: Building a Panorama

Example 3: 3D reconstruction

- Photo Tourism overview

Input photographs

Scene reconstruction

Photo Explorer

Relative camera positions and orientations
Point cloud
Sparse correspondence
Example 3: 3D reconstruction

The Old City of Dubrovnik

57,845 downloaded images, 11,868 registered images. This video: 4,619 images.

Building Rome in a Day, Agarwal, Snavely, Simon, Seitz, Szeliski, ICCV 2009
See also [Havlena, Torrii, Knop pand Pajdla, CVPR 2009].

Slide credit: Jiri Matas & Andrew Zisserman
Example 4: Query by example search in large scale image datasets

- Search the web with a visual query ...

Find these objects...in these images and 1M more
Example 5: Google goggles
Example 6: Where am I?

- Place recognition - retrieval in a structured (on a map) database

Query ▸ Image database ▸ Best match

Query Expansion (Panoramio, Flickr, ... )

Confuser Suppression
Only negative training data (from geotags)

Image indexing with spatial verification

Slide credit: Jiri Matas & Andrew Zisserman
Development of Low-Level Image Features

1999

Classical features
- Raw pixel
- Histogram feature
  - Color histogram
  - Edge histogram
- Frequency analysis
- Image filters
- Texture features
  - LBP
- Scene features
  - GIST
- Shape descriptors
- Edge detection
- Corner detection

Local Descriptors
- SIFT
- HOG
- SURF
- DAISY
- BRIEF
- ...
- DoG
- Hessian detector
- Laplacian of Harris
- FAST
- ORB
- ...

Slide credit: Liangliang Cao
Concatenating Raw Pixels As 1D Vector

Credit: The Face Research Lab
Concatenated Raw Pixels

Famous applications (widely used in ML field)
• Face recognition

• Hand-written digits

Pictures courtesy to Sam Roweis

Slide credit: Liangliang Cao
Tiny Images

Antonio Torralba et al. proposed to resize images to 32x32 color thumbnails, which are called “tiny images”

- Related applications
  - Scene recognition
  - Object recognition

Fast speed with limited accuracy

Torralba et al, MIT CSAIL report, 2007
Problem of raw-pixel based representation

- Rely heavily on good alignment
- Assume the images are of similar scale
- Suffer from occlusion
- Recognition from different viewpoint will be difficult

We want more powerful features for real-world problems like the following.
Color Histogram

Each pixel is described by a vector of pixel values

\[
\begin{pmatrix}
    r \\
    g \\
    b 
\end{pmatrix}
\]

Distribution of color vectors is described by a histogram

Note: There are different choices for color space: RGB, HSV, Lab, etc. For gray images, we usually use 256 or fewer bins for histogram.

[Swain and Ballard 91]
Benefits of Histogram Representations

No longer sensitive to alignment, scale transform, or even global rotation

Similar color histograms (after normalization)
Limitation of Global Histogram

Global histogram has no location information at all

They’re equal in terms of global histogram

Example courtesy to Erik Learned-Miller
Histogram with Spatial Layout

- Concatenated histogram for each region
Spatial Pyramid Matching

- Lazebnik, Schmid and Ponce, CVPR’06
General Histogram Representation

• Color histogram

  patch
  patch
  patch

  Extract color descriptor

  →

  histogram accumulation

• General histogram

  patch
  patch
  patch

  Extract other descriptors
  - edge histogram
  - shape context histogram
  - local binary patterns

  →

  histogram accumulation

Click to edit title style

Slide credit: Liangliang Cao
Local Binary Pattern (LBP)

- For each pixel $p$, create an 8-bit number $b_1 b_2 b_3 b_4 b_5 b_6 b_7 b_8$, where $b_i = 0$ if neighbor $i$ has value less than or equal to $p$'s value and 1 otherwise.

- Represent the texture in the image (or a region) by the histogram of these numbers.

$$
\begin{array}{ccc}
1 & 2 & 3 \\
100 & 101 & 103 \\
40 & 50 & 80 \\
50 & 60 & 90 \\
7 & 6 & 5 \\
\end{array}
\quad \rightarrow \quad
\begin{array}{c}
111111100
\end{array}
$$

Ojala et al, PR'96, PAMI'02
LBP Histogram

- Divide the examined window to cells (e.g. 16x16 pixels for each cell).
- Compute the histogram, over the cell, of the frequency of each "number" occurring.
- Optionally normalize the histogram.
- Concatenate normalized histograms of all cells.
Why local features?

- LocalFeature=Interest“Point”=Keypoint=Feature “Point”= Distinguished Region = Covariant Region

Local: robust to occlusion/clutter
Invariant: to scale/view/illumination changes

Local feature can be discriminant!
How do we build a panorama?

- We need to match (align) images
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 matching pairs to align images – the required mapping is called a homography.

Slide credit: B. Freeman and A. Torralba
Local features: main components

1) **Detection**: Identify the interest points

2) **Description**: Extract vector feature descriptor surrounding each interest point.

3) **Matching**: Determine correspondence between descriptors in two views

\[ \mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}] \]

\[ \mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}] \]

Slide credit: K. Grauman
Local features: desired properties

• **Repeatability**
  – The same feature can be found in several images despite geometric and photometric transformations

• **Saliency**
  – Each feature has a distinctive description

• **Compactness and efficiency**
  – Many fewer features than image pixels

• **Locality**
  – A feature occupies a relatively small area of the image; robust to clutter and occlusion
Goal: interest operator repeatability

- We want to detect (at least some of) the same points in both images.

No chance to find true matches!

- Yet we have to be able to run the detection procedure independently per image.
Goal: descriptor distinctiveness

• We want to be able to reliably determine which point goes with which.

• Must provide some invariance to geometric and photometric differences between the two views.
• What points would you choose?
Motivation: Patch Matching

- Elements to be matched are image patches of fixed size

Task: find the best (most similar) patch in a second image

Slide credit: R. Collings
Not all Patches are Created Equal!

- Intuition: this would be a good patch for matching, since it is very distinctive (there is only one patch in the second frame that looks similar).
Not all Patches are Created Equal!

- Intuition: this would be a BAD patch for matching, since it is not very distinctive (there are many similar patches in the second frame).
What about Corners?

- Intuitively, junctions of contours
- Generally, more stable features over changes of viewpoint
- Intuitively, large variations in the neighborhood of the point in all directions
- They are GOOD features to match!

Slide credit: R. Collings
Local features: main components

1) **Detection:** Identify the interest points

2) **Description:** Extract vector feature descriptor surrounding each interest point.

3) **Matching:** Determine correspondence between descriptors in two views

\[
x_1 = [x_1^{(1)}, \ldots, x_d^{(1)}]
\]

\[
x_2 = [x_1^{(2)}, \ldots, x_d^{(2)}]
\]
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

Slide credit: K. Grauman, Alyosha Efros, Darya Frolova, Denis Simakov
Harris Detector: Mathematics

Window-averaged squared 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 \(w(x, y)\) =

- 1 in window, 0 outside
- Gaussian

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
Harris Detector: Mathematics

- Taylor series approximation to shifted image gives quadratic form for error as function of image shifts.

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

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

\[
= (u \quad v) \sum_{x, y} w(x, y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_y I_x & I_y I_y \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}
\]
Expanding $I(x,y)$ in a Taylor series expansion, we have, for small shifts $[u,v]$, a quadratic approximation to the error surface between a patch and itself, shifted by $[u,v]$:

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

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

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

$M$ is often called *structure tensor*.

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
Harris Detector: Mathematics

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

2 x 2 matrix of image derivatives (averaged in neighborhood of a point).

Notation:

\[ I_x \iff \frac{\partial I}{\partial x} \quad I_y \iff \frac{\partial I}{\partial y} \quad I_x I_y \iff \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \]
What does this matrix reveal?

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

First, consider an axis-aligned corner:

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

This means dominant gradient directions align with x or y axis.

Look for locations where both \( \lambda \)'s are large.

If either \( \lambda \) is close to 0, then this is not corner-like.

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

Slide credit: K. Grauman
What does this matrix reveal?

Since $M$ is symmetric, we have

$$M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T$$

$$Mx_i = \lambda_i x_i$$

The *eigenvalues* of $M$ reveal the amount of intensity change in the two principal orthogonal gradient directions in the window.
Classification of image points using eigenvalues of $M$:

- $\lambda_1$ and $\lambda_2$ are small; $E$ is almost constant in all directions.
- $\lambda_1$ and $\lambda_2$ are large, $\lambda_1 \sim \lambda_2$; $E$ increases in all directions.
- $\lambda_1 \gg \lambda_2$, “Corner”
- $\lambda_2 \gg \lambda_1$, “Edge”

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
Measure of corner response:

\[ R = \det M - k \left( \text{trace } M \right)^2 \]

\[ \det M = \lambda_1 \lambda_2 \]

\[ \text{trace } M = \lambda_1 + \lambda_2 \]

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

(Shi-Tomasi variation: use \( \min(\lambda_1, \lambda_2) \) instead of R)

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
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

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
Harris Detector

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

Slide credit: B. Freeman, A. Torralba
Harris corner detector algorithm

• Compute image gradients $I_x$, $I_y$ for all pixels

• For each pixel
  – Compute $M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$ by looping over neighbors $x,y$
  – Compute $R = \det M - k \left( \text{trace } M \right)^2$

• Find points with large corner response function $R$ ($R > \text{threshold}$)

• Take the points of locally maximum $R$ as the detected feature points (i.e., pixels where $R$ is bigger than for all the 4 or 8 neighbors).
Harris Detector: Workflow

Compute corner response $R$
Harris Detector: Workflow

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

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
Harris Detector: Workflow

Take only the points of local maxima of R

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
Harris Detector: Workflow

Slide credit: B. Freeman, A. Torralba, Darya Frolova, Denis Simakov
Properties of the Harris corner detector

- Rotation invariant? Yes

\[ M = X \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} X^T \]

- Scale invariant?
Properties of the Harris corner detector

• Rotation invariant? Yes

• Scale invariant? No

All points will be classified as edges

Corner!
Scale invariant interest points

How can we independently select interest points in each image, such that the detections are repeatable across different scales?
Automatic scale selection

Intuition:
• Find scale that gives local maxima of some function $f$ in both position and scale.

Slide credit: K. Grauman
• What can be the “signature” function?
Recall: Edge detection

Edge = maximum of derivative

Slide credit: K. Grauman, S. Seitz
Recall: Edge detection

Edge = zero crossing of second derivative

Source: S. Seitz
Slide credit: K. Grauman
From edges to blobs

- Edge = ripple
- Blob = superposition of two ripples

Spatial selection: the magnitude of the Laplacian response will achieve a maximum at the center of the blob, provided the scale of the Laplacian is “matched” to the scale of the blob.
Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Blob detection in 2D: scale selection

- Laplacian-of-Gaussian = “blob” detector

\[ \nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \]
Blob detection in 2D

- We define the characteristic scale as the scale that produces peak of Laplacian response
Example

Original image at \( \frac{3}{4} \) the size
Original image at 3/4 the size

Slide credit: K. Grauman
Scale invariant interest points

Interest points are local maxima in both position and scale.

⇒ List of $(x, y, \sigma)$

Squared filter response maps

Slide credit: K. Grauman
Scale-space blob detector: Example

Slide credit: K. Grauman, L. Lazebnik
Technical detail

- We can approximate the Laplacian with a difference of Gaussians; more efficient to implement.

(Laplacian)

(Difference of Gaussians)

Slide credit: K. Grauman
Local features: main components

1) Detection: Identify the interest points

2) Description: Extract vector feature descriptor surrounding each interest point.

3) Matching: Determine correspondence between descriptors in two views

\[ \mathbf{x}_1 = [x_1^{(1)}, \ldots, x_d^{(1)}] \]

\[ \mathbf{x}_2 = [x_1^{(2)}, \ldots, x_d^{(2)}] \]
Geometric transformations

e.g. scale, translation, rotation

Slide credit: K. Grauman
Photometric transformations

Figure from T. Tuytelaars ECCV 2006 tutorial

Slide credit: K. Grauman
Raw patches as local descriptors

The simplest way to describe the neighborhood around an interest point is to write down the list of intensities to form a feature vector.

But this is very sensitive to even small shifts, rotations.
SIFT descriptor [Lowe 2004]

- Use histograms to bin pixels within sub-patches according to their orientation.

Why subpatches?
Why does SIFT have some illumination invariance?
Making descriptor rotation invariant

- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.
SIFT descriptor [Lowe 2004]

Extraordinarily robust matching technique

Can handle changes in viewpoint
  Up to about 60 degree out of plane rotation
Can handle significant changes in illumination
  Sometimes even day vs. night (below)
Fast and efficient—can run in real time
Lots of code available


Slide credit: K. Grauman, S. Seitz
Example

NASA Mars Rover images

Slide credit: K. Grauman
Example

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

Slide credit: K. Grauman
SIFT properties

• Invariant to
  – Scale
  – Rotation

• Partially invariant to
  – Illumination changes
  – Camera viewpoint
  – Occlusion, clutter
Local features: main components

1) **Detection:** Identify the interest points

2) **Description:** Extract vector feature descriptor surrounding each interest point.

3) **Matching:** Determine correspondence between descriptors in two views
Matching local features

Slide credit: K. Grauman
Matching local features

To generate candidate matches, find patches that have the most similar appearance (e.g., lowest SSD)

Simplest approach: compare them all, take the closest (or closest k, or within a thresholded distance)
Ambiguous matches

At what SSD value do we have a good match?

To add robustness to matching, can consider ratio: distance to best match / distance to second best match

If low, first match looks good.

If high, could be ambiguous match.
Matching SIFT Descriptors

- Nearest neighbor (Euclidean distance)
- Threshold ratio of nearest to 2nd nearest descriptor

Lowe IJCV 2004
Slide credit: K. Grauman
Recap: robust feature-based alignment
Recap: robust feature-based alignment

- Extract features

Slide credit: K. Grauman, L. Lazebnik
Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*
Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
  - *Verify* transformation (search for other matches consistent with $T$)

Slide credit: K. Grauman, L. Lazebnik
Recap: robust feature-based alignment

- Extract features
- Compute *putative matches*
- Loop:
  - *Hypothesize* transformation $T$ (small group of putative matches that are related by $T$)
  - *Verify* transformation (search for other matches consistent with $T$)
Summary

• Interest point detection
  – Harris corner detector
  – Laplacian of Gaussian, automatic scale selection

• Invariant descriptors
  – Rotation according to dominant gradient direction
  – Histograms for robustness to small shifts and translations (SIFT descriptor)