Local Features

CS 554 – Computer Vision Pinar Duygulu Bilkent University

Today's lecture: Feature Detection and Matching

- Local features
- Pyramids for invariant feature detection
- Invariant descriptors
- Matching

Image matching



by <u>Diva Sian</u>



by swashford

Harder case





by <u>Diva Sian</u>

by <u>scgbt</u>

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



- 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?

• Detect feature points in both images



- Detect feature points in both images
- Find corresponding pairs



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



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





no chance to match!

We need a repeatable detector

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



We need a reliable and distinctive descriptor

Geometric transformations







Photometric transformations



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



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. <u>"A Combined Corner and Edge Detector."</u> *Proceedings of the 4th Alvey Vision Conference*: pages 147--151. 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





"corner": significant change in all directions

Source: A. Efros

Harris Detector formulation Change of intensity for the shift [*u*,*v*]:





















Harris Detector formulation This measure of change can be approximated by:

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

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

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

Sum over image region – area
we are checking for corner
$$M = \begin{bmatrix} \sum_{x} I_x I_x & \sum_{x} I_x I_y \\ \sum_{x} I_x & \sum_{x} I_x I_y \end{bmatrix} = \sum_{x} \begin{bmatrix} I_x \\ I_x \end{bmatrix} \begin{bmatrix} I_x \\ I_x \end{bmatrix}$$

Gradient with espect to x, times radient with espect to y

$$M = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x I_y]$$



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

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

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

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

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



Slide adapted form Darya Frolova, Denis Simakov.

Interpreting the eigenvalues Classification of image points using eigenvalues of *M*:



 λ_1

Corner response function $R = \det(M) - \alpha \operatorname{trace}(M)^2 = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2$

 α : 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 > threshold)
 - Take the points of local maxima of R



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

Compute corner response R



Find points with large corner response: *R*>threshold



Take only the points of local maxima of R

• • ·

.

.



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

• Multi-scale approach







Slide from T. Tuytelaars ECCV 2006 tutorial

• Multi-scale approach







• Multi-scale approach







• Multi-scale approach







Invariance

• Extract patch from each image individually







- 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)



• 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!















Scale selection

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









What Is A Useful Signature Function?



Characteristic scale

• We define the *characteristic scale* as the scale that produces neak of <u>Laplacian response</u>



T. Lindeberg (1998). <u>"Feature detection with automatic scale selection."</u> International Journal of Computer Vision **30** (2): pp 77--116. Source: Lana Lazebnik

Laplacian-of-Gaussian (LoG)

• Interest points: Local maxima in scale space of Laplacian-of- σ^4



 $L_{xx}(\sigma) + L_{yy}(\sigma) \rightarrow \sigma^3$



Scale \Rightarrow List of (x, y, σ)

 σ^2

 σ

Original image at ¾ the size





Kristen Grauman

Original image at ¾ the size

















Kristen Grauman













Scale invariant interest points

Interest points are local maxima in both position and scale.





Squared filter response maps



Source: Lana Lazebnik



sigma = 11.9912



Source: Lana Lazebnik



Image credit: Lana Lazebnik

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



Key point localization with DoG

- Detect maxima of differenceof-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses



Candidate keypoints: list of (x,y,σ)

Example of keypoint detection



- (a) 233x189 image
- (b) 832 DOG extrema
- (c) 729 left after peak value threshold
- (d) 536 left after testing ratio of principle curvatures (removing edge responses)

Scale Invariant Detection: Summary

- Given: two images of the same scene with a large scale difference between them
- Goal: find *the same* interest points *independently* in each image
- Solution: search for maxima of suitable functions in scale and in space (over the image)

Maximally Stable Extremal Regions [Matas '02]

- Based on Watershed segmentation algorithm
- Select regions that stay stable over a large

para




Example Results: MSER







Harris

Comparis







LoG

MSER

Comparison of Keypoint Detectors

Table 7.1 Overview of feature detectors.

				Rotation	Scale	Affine		Localization		
Feature Detector	Corner	Blob	Region	invariant	invariant	invariant	Repeatability	accuracy	Robustness	Efficiency
Harris	\checkmark			\checkmark			+++	+++	+++	++
Hessian				\checkmark			++	++	++	+
SUSAN	\checkmark			\checkmark			++	++	++	+++
Harris-Laplace	\checkmark	(√)		\checkmark	\checkmark		+++	+++	++	+
Hessian-Laplace	(\scrime)	\checkmark		\checkmark	\checkmark		+++	+++	+++	+
DoG	()			\checkmark	\checkmark		++	++	++	++
SURF	(\scrime)			\checkmark	\checkmark		++	++	++	+++
Harris-Affine	\checkmark	(√)		\checkmark	\checkmark	\checkmark	+++	+++	++	++
Hessian-Affine	(\scrime)	\checkmark		\checkmark	\checkmark	\checkmark	+++	+++	+++	++
Salient Regions	(\scrime)			\checkmark	\checkmark	()	+	+	++	+
Edge-based	\checkmark			\checkmark	\checkmark	\checkmark	+++	+++	+	+
MSER			\checkmark	\checkmark	\checkmark	\checkmark	+++	+++	++	+++
Intensity-based			\checkmark	\checkmark	\checkmark	\checkmark	++	++	++	++
Superpixels			\checkmark	\checkmark	(√)	()	+	+	+	+

Tuytelaars Mikolajczyk 2008

Choosing a detector

- What do you want it for?
 - Precise localization in x-y: Harris
 - Good localization in scale: Difference of Gaussian
 - Flexible region shape: MSER
- Best choice often application dependent
 - Harris-/Hessian-Laplace/DoG work well for many natural categories
 - MSER works well for buildings and printed things
- Why choose?
 - Get more points with more detectors
- There have been extensive evaluations/comparisons
 - [Mikolajczyk et al., IJCV'05, PAMI'05]
 - All detectors/descriptors shown here work well

- For most local feature detectors, executables are available online:
 - <u>http://robots.ox.ac.uk/~vgg/research/affine</u>
 - <u>http://www.cs.ubc.ca/~lowe/keypoints/</u>
 - <u>http://www.vision.ee.ethz.ch/~surf</u>

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?

Local descriptors

- We know how to detect points
- Next question:

How to *describe* them for matching?



Point descriptor should be:

- 1. Invariant
- 2. Distinctive

Local descriptors

- Simplest descriptor: list of intensities within a patch.
- What is this going to be invariant to?



Feature descriptors

Disadvantage of set based criptors:
 – Small shifts car





Source: Lana Lazebnik

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?

Feature descriptors: SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
 - Divide patch into 4x4 sub-patches: 16 cells



David G. Lowe. <u>"Distinctive image features from scale-invariant keypoints."</u> *IJCV* 60 (2), pp. 91-110, 2004.

Source: Lana Lazebnik

Rotation Invariant Descriptors

• Find local orientation

Dominant direction of gradient for the image patch



 Rotate patch according to this angle This puts the patches into a canonical orientation.

Rotation Invariant Descriptors



Image from Matthew Brown

Feature descriptors: SIFT

- 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
 - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT



Working with SIFT descriptors

- One image yields:
 - n 128-dimensional descriptors: each one is a histogram of the gradient orientations within a patch
 - [n x 128 matrix]
 - n scale parameters specifying the size of each patch
 - [n x 1 vector]
 - n orientation parameters specifying the angle of the patch
 - [n x 1 vector]
 - n 2d points giving positions of the patches



• [n x 2 matrix]

More on feature detection/description



Affine Covariant Regions

Publications

Region detectors

- Harris-Affine & Hessian Affine: <u>K. Mikolajczyk</u> and <u>C. Schmid</u>, Scale and Affine invariant interest point detectors. In IJCV 1(60):63-86, 2004. <u>PDF</u>
- MSER: J.Matas, O. Chum, M. Urban, and T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions. In BMVC p. 384-393, 2002. PDF
- IBR & EBR: T.Tuytelaars and L. Van Gool, Matching widely separated views based onaffine invariant regions. In IJCV 1 (59):61-85, 2004. <u>PDF</u>
- Salient regions: <u>T. Kadir</u>, <u>A. Zisserman</u>, and <u>M. Brady</u>, An affine invariant salient region detector. In ECCV p. 404-416, 2004. <u>PDF</u>
- Region descriptors . SIFT: D. Lowe, Distinctive image features from scale invariant keypoints. In LICV 2(60):91-110, 2004. PDF

Performance evaluation

- <u>K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir</u> and <u>L. Van Gool</u>, A comparison of affine region detectors. Technical Report, accepted to IJCV. <u>PDF</u>
 - K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors. Technical Report, accepted to PAMI. PDF

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?

Feature descriptors We know how to detect and describe good points



Feature matching

Given a feature in I₁, how to find the best match in I₂?

- Define distance function that compares two descriptors
- 2. Test all the features in I_2 , find the one with min distance

Feature distance How to define the difference between two features f₁, f₂?

- Simple approach is SSD(f_1, f_2)
 - sum of square differences between entries of the two descriptors



Feature distance How to define the difference between two features f₁, f₂?

Better approach: ratio distance = SSD(f₁, f₂) / SSD(f₁, f₂')



Evaluating the results



feature distance

True/false positives



feature distance

The distance threshold affects performance

- True positives = # of detected matches that are correct
 - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
 - Suppose we want to minimize these—how to choose threshold?

Evaluating the results How can we measure the performance of a feature matcher?



Evaluating the results How can we measure the performance of a feature matcher?



ROC Curves

- Generated by counting # current/incorrect matches, for different threholds
- Want to maximize area under the curve (AUC)
- Useful for comparing different feature matching methods
- For more info: <u>http://en.wikipedia.org/wiki/Receiver_operating_characteristic</u>









• Extract features



- Extract features
- Compute *putative matches*



- Extract features
- Compute *putative matches*
- Loop:
 - Hypothesize transformation T (small group of putative matches that are related by T)



- 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)



- 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)
 Source: L. Lazebnik

Applications of local invariant features

- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition

Automatic mosaicing



http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

Wide baseline stereo



[Image from T. Tuytelaars ECCV 2006 tutorial]

Recognition of specific objects, scenes



Schmid and Mohr 1997



Sivic and Zisserman, 2003



Rothganger et al. 2003



Lowe 2002

Object recognition (David Lowe)


Multi-view matching



Matching two given views for depth

Search for a matching view for recognition

Kristen Grauman

How to quickly find images in a large database that match a given image region?





Video Google Sy

- Collect all words within query region
- 2. Inverted file index to find relevant frames
- 3. Compare word counts
- 4. Spatial verification

Sivic & Zisserman, ICCV 2003

• Demo online at :

http://www.robots.ox.ac.uk/~vgg/research/vgoo gle/index.html



Query region



Retrieved frames

Kristen Grauman

Example Applications





Mobile tourist guide

- Self-localization
- Object/building recognition
- Photo/video augmentation

[Quack, Leibe, Van Gool, CIVR'08]

Application: Large-Scale Retrieval



Results from 5k Flickr images (demo available for 100k set)

Query

[Philbin CVPR'07]

Application: Image Auto-Annotation

















Left: Wikipedia image Right: closest match from Flickr





[Quack CIVR'08]

K. Grauman, B. Leibe





Photo Tourism



Slide Credits

- Trevor Darrell
- Bill Freeman
- Kristen Grauman
- Steve Seitz
- Ivan Laptev
- Tinne Tuytelaars
- James Hays
- Svetlana Lazebnik