Matching features

Computational Photography, 6.815/6.86

Frédo Durand

Image and shape descriptors:
  Harris corner detectors and SIFT features.
Ransac
Photo Tourism/Photosynth
Slides from a lot of people
Matching with Invariant Features

Darya Frolova, Denis Simakov
The Weizmann Institute of Science
March 2004

http://www.wisdom.weizmann.ac.il/~deniss/vision_spring04/files/InvariantFeatures.ppt
Contents

• Harris Corner Detector
  – Description
  – Analysis
• Detectors
  – Rotation invariant
  – Scale invariant
  – Affine invariant  SKIP!
• Descriptors
  – Rotation invariant
  – Scale invariant
  – Affine invariant
Recall: Matching with Features

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

We need a repeatable detector:
**DONE**
Recall: Matching with Features

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

We need a reliable and distinctive descriptor
Recognition and Matching Based on Local Invariant Features

David Lowe
Computer Science Department
University of British Columbia
Assume we've found a Rotation/Scale/Blah Invariant Point

- **Harris-Laplacian**¹
  
  Find local maximum of:
  - Harris corner detector in space (image coordinates)
  - Laplacian in scale

---

- **SIFT (Lowe)**²
  
  Find local maximum of:
  - Difference of Gaussians in space and scale

---

² D.Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. Accepted to IJCV 2004
Select canonical orientation

- One option: use maximum eigenvector in Harris

Alternative

- Create histogram of local gradient directions computed at selected scale
- Assign canonical orientation at peak of smoothed histogram
- Each key specifies stable 2D coordinates (x, y, scale, orientation)
Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters.
Recap

- We have scale/rotation/blah invariant distinctive features
- We have a characteristic scale
- We have a characteristic orientation

- Now we need to find a descriptor of this point
  - For matching with other images
Naive option

• Use nxn pixels and sum of square difference
  – Not so bad if image are similar enough
  – Can also use correlation

• Problems
  – Depends on lighting, white balance
  – Needs big number of pixels to be discriminative
  – Could be sensitive to resampling
    (because scale, rotation)
But see Peter Sand’s trick in video matching
http://rvsn.csail.mit.edu/vid-match/
SIFT vector formation

- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions

Image gradients

Keypoint descriptor
(should be 4x4)
Feature stability to noise

- Match features after random change in image scale & orientation, with differing levels of image noise
- Find nearest neighbor in database of 30,000 features
Feature stability to affine change

- Match features after random change in image scale & orientation, with 2% image noise, and affine distortion
- Find nearest neighbor in database of 30,000 features
Distinctiveness of features

- Vary size of database of features, with 30 degree affine change, 2% image noise
- Measure % correct for single nearest neighbor match
Figure 12: The training images for two objects are shown on the left. These can be recognized in a cluttered image with extensive occlusion, shown in the middle. The results of recognition are shown on the right. A parallelogram is drawn around each recognized object showing the boundaries of the original training image under the affine transformation solved for during recognition. Smaller squares indicate the keypoints that were used for recognition.
Figure 13: This example shows location recognition within a complex scene. The training images for locations are shown at the upper left and the 640x315 pixel test image taken from a different viewpoint is on the upper right. The recognized regions are shown on the lower image, with keypoints shown as squares and an outer parallelogram showing the boundaries of the training images under the affine transform used for recognition.
A good SIFT features tutorial


By Estrada, Jepson, and Fleet.
Recap

• Stable (repeatable) feature points can be detected regardless of image changes
  – **Scale**: search for correct scale as *maximum* of appropriate function
  – **Affine**: approximate regions with *ellipses* (this operation is affine invariant)

• Invariant and distinctive descriptors can be computed
  – Invariant *moments*
  – *Normalizing* with respect to scale and affine transformation
Now we have

• Well-localized feature points
• Distinctive descriptor

• We need to
  – match pairs of feature points in different images
  – Robustly compute homographies (in the presence of errors/outliers)
Feature Matching and RANSAC

© Krister Parmstrand

with a lot of slides stolen from Steve Seitz and Rick Szeliski

15-463: Computational Photography
Alexei Efros, CMU, Fall 2005
Feature matching
Strategy to reject bad matches

- For each feature point, find most similar point in other image (SIFT distance)
  - Reject bad matches
  - i.e. ambiguous matches where there are too many similar points
- Given good matches, look for possible homographies
  - Reject bad homographies
  - i.e. homographies that don’t match many points
Feature matching

• Exhaustive search
  – for each feature in one image, look at all the other features in the other image(s)

• Hashing
  – compute a short descriptor from each feature vector, or hash longer descriptors (randomly)

• Nearest neighbor techniques
  – $k$-trees and their variants
Feature-space outlier rejection

• Let’s not match all features, but only these that have “similar enough” matches?
• How can we do it?
  – SSD(patch1,patch2) < threshold
  – How to set threshold?
Feature-space outlier rejection

- A better way [Lowe, 1999]:
  - 1-NN: SSD of the closest match
  - 2-NN: SSD of the second-closest match
  - Look at how much better 1-NN is than 2-NN, e.g. 1-NN/2-NN
  - That is, is our best match so much better than the rest?
• Can we now compute H from the blue points?
  – No! Still too many outliers…
  – What can we do?
Matching features

What do we do about the “bad” matches?
Random SAMple Consensus

Select one match, count inliers
RAndom SAmple Consensus

Select one match, count inliers
Least squares fit

Find “average” translation vector
RANSAC for estimating homography

- RANSAC loop:
  1. Select four feature pairs (at random)
  2. Compute homography H (exact)
  3. Compute inliers where $\|p_i', Hp_i\| < \varepsilon$
  4. Keep largest set of inliers
  5. Re-compute least-squares H estimate on all of the inliers
Simple example: fit a line

- Rather than homography $H$ (8 numbers), fit $y=ax+b$ (2 numbers $a$, $b$) to 2D pairs
Simple example: fit a line

- Pick 2 points
- Fit line
- Count inliers

3 inliers
Simple example: fit a line

- Pick 2 points
- Fit line
- Count inliers

4 inlier
Simple example: fit a line

- Pick 2 points
- Fit line
- Count inliers

9 inlier
Simple example: fit a line

• Pick 2 points
• Fit line
• Count inliers

8 inlier
Simple example: fit a line

- Use biggest set of inliers
- Do least-square fit
RANSAC

red:
  rejected by 2nd nearest neighbor criterion
blue:
  Ransac outliers
yellow:
  inliers
Robustness

• Proportion of inliers in our pairs is $X$
• Our model needs $P$ pairs
  – $P=4$ for homography
• Probability that we pick $P$ inliers?
  – $X^P$
• Probability that after $N$ RANSAC iterations we have not picked a set of inliers?
  – $(1-X^P)^N$
Robustness: example

- Proportion of inliers $X=0.5$
- Probability that we pick $P=4$ inliers?
  - $0.5^4=0.0625$ (6% chances)
- Probability that we have not picked a set of inliers?
  - $N=100$ iterations:
    $$(1-0.5^4)^{100}=0.00157$$(1 chance in 600)
  - $N=1000$ iterations:
    1 chance in 1e28
Robustness: example

- Proportion of inliers $X=0.3$
- Probability that we pick $P=4$ inliers?
  - $0.3^4=0.0081$ (0.8% chances)
- Probability that we have not picked a set of inliers?
  - $N=100$ iterations:
    - $(1-0.3^4)^{100}=0.44$ (1 chance in 2)
  - $N=1000$ iterations:
    - 1 chance in 3400
Robustness: example

- Proportion of inliers $X=0.1$
- Probability that we pick $P=4$ inliers?
  - $0.1^4=0.0001$ (0.01% chances, 1 in 10,000)
- Probability that we have not picked a set of inliers?
  - $N=100$ iterations: $(1-0.1^4)^{100}=0.99$
  - $N=1000$ iterations: 90%
  - $N=10,000$: 36%
  - $N=100,000$: 1 in 22,000
Robustness: conclusions

- Effect of number of parameters of model/number of necessary pairs
  - Bad exponential
- Effect of percentage of inliers
  - Base of the exponential
- Effect of number of iterations
  - Good exponential
Example: Recognising Panoramas

M. Brown and D. Lowe, University of British Columbia
“Recognising Panoramas”?
RANSAC for Homography
RANSAC for Homography
Probabilistic model for verification
Finding the panoramas
Finding the panoramas
Finding the panoramas
Finding the panoramas
Results
Photo Tourism: Exploring Photo Collections in 3D

Noah Snavely
Steven M. Seitz
   University of Washington
Richard Szeliski
   Microsoft Research

© 2006 Noah Snavely
Photo Tourism overview

Input photographs

Scene reconstruction

Relative camera positions and orientations
Point cloud
Sparse correspondence

Photo Explorer

© 2006 Noah Snavely
Related work

- Image-based modeling
  - Debevec, *et al.*, SIGGRAPH 1996
  - Schaffalitzky and Zisserman, ECCV 2002
  - Brown and Lowe, 3DIM 2005

- Image-based rendering

Photorealistic IBR:
  - Levoy and Hanrahan, SIGGRAPH 1996
  - Gortler, *et al.*, SIGGRAPH 1996
  - Seitz and Dyer, SIGGRAPH 1996
  - Aliaga, *et al.*, SIGGRAPH 2001
  - and many others
Related work

- Image browsing

McCurdy and Griswold, Mobisys 2003
Sivic and Zisserman, ICCV 2003
Photo Tourism overview

Input photographs → Scene reconstruction → Photo Explorer
Scene reconstruction

- Automatically estimate
  - position, orientation, and focal length of cameras
  - 3D positions of feature points

Feature detection

Pairwise feature matching

Correspondence estimation

Incremental structure from motion
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]
Feature detection

Detect features using SIFT [Lowe, IJCV 2004]
Feature matching

Match features between each pair of images
Feature matching

Refine matching using RANSAC [Fischler & Bolles 1987] to estimate fundamental matrices between pairs

See 6.801/6.866 for essential matrix
Structure from motion

\[
\begin{align*}
\text{minimize} & \quad f (R, T, P) \\
\end{align*}
\]
Incremental structure from motion
Photo Tourism overview

- Input photographs
- Scene reconstruction
- Photo Explorer
  - Navigation
  - Rendering
  - Annotations
Object-based browsing
Object-based browsing

- Visibility
- Resolution
- Head-on view
Relation-based browsing

Find all details
Move left
Move right
Zoom in
Zoom out
Find all similar images
Find all zoom outs

Name: 55668857@N00_11...
Added by: 55668857@N00
Date: March 3, 2006, ...
Relation-based browsing

Image A

Image B
Relation-based browsing

Image A

Image B
Relation-based browsing

Image A

Image B

to the right of

Image C
Relation-based browsing

Image A

Image B

Image C
Relation-based browsing

Image A

Image B

Image C

Image D

is detail of

to the right of
Relation-based browsing

Image A

Image B

to the right of

Image C

is detail of

Image D
Relation-based browsing

Image A

is zoom-out of

is detail of

Image B

is detail of

is zoom-out of

to the left of

to the right of

Image C

is detail of

is zoom-out of

Image D

is zoom-out of

is detail of
Rendering
Rendering transitions
Rendering transitions
Rendering transitions
Rendering transitions

Camera A

Camera B
Photo Tourism
Exploring photo collections in 3D

Noah Snavely  Steven M. Seitz  Richard Szeliski
University of Washington  Microsoft Research

SIGGRAPH 2006
Live demo

• http://phototour.cs.washington.edu/applet/index.html
Limitations / Future work

- Not all photos can be reliably matched
  - Better feature detection / matching
  - Integrating GPS & other localization info.
- Structure from motion scalability
  - More efficient (sparse) algorithms
- Plane-based transitions lack parallax
  - Richer transitions
- Photo explorer scalability…
Conclusion

Indexing everyone’s photos provides a new way to share and experience our world

To find out more:
- http://phototour.cs.washington.edu
- http://research.microsoft.com/IVM/PhotoTourism
- http://labs.live.com/photosynth
- Exhibition booth #2619