Matching II

Hao Jiang
Computer Science Department
Hough transform for circles

• Circle: center \((a, b)\) and radius \(r\)

\[
(x_i - a)^2 + (y_i - b)^2 = r^2
\]

- For a fixed radius \(r\), unknown gradient direction

Source: K. Grauman
Hough transform for circles

- Circle: center \((a, b)\) and radius \(r\)

\[(x_i - a)^2 + (y_i - b)^2 = r^2\]

- For a fixed radius \(r\), unknown gradient direction

Image space

Hough space

Intersection: most votes for center occur here.

Source: K. Grauman
Hough transform for circles

- Circle: center \((a,b)\) and radius \(r\)
  \[(x_i - a)^2 + (y_i - b)^2 = r^2\]

- For an unknown radius \(r\), unknown gradient direction

Source: K. Grauman
Hough transform for circles

- Circle: center \((a, b)\) and radius \(r\)

\[
(x_i - a)^2 + (y_i - b)^2 = r^2
\]

- For an unknown radius \(r\), unknown gradient direction

Source: K. Grauman
Hough transform for circles

- Circle: center \((a, b)\) and radius \(r\)

\[
(x_i - a)^2 + (y_i - b)^2 = r^2
\]

- For an unknown radius \(r\), **known** gradient direction

Source: K. Grauman
Hough transform for circles

For every edge pixel \((x,y)\):

For each possible radius value \(r\):

For each possible gradient direction \(\theta\): // or use estimated gradient

\[
a = x + r \cos(\theta)
\]

\[
b = y + r \sin(\theta)
\]

\[
H[a,b,r] += 1
\]

end

end

Source: K. Grauman
Example: detecting circles with Hough

Crosshair indicates results of Hough transform, bounding box found via motion differencing.

Source: K. Grauman
Example: detecting circles with Hough

Original

Edges

Votes: Penny

Note: a different Hough transform (with separate accumulators) was used for each circle radius (quarters vs. penny).

Source: K. Grauman
Example: detecting circles with Hough

Original

Edges

Votes: Quarter

Source: K. Grauman
Coin finding sample images from: Vivek Kwatra
General Hough Transform

- Hough transform can also be used to detection arbitrary shaped object.
General Hough Transform

- Hough transform can also be used to detection arbitrary shaped object.

Template  

Target
General Hough Transform

- Hough transform can also be used to detect arbitrary shaped objects.

Template  Target
General Hough Transform

- Hough transform can also be used to detect arbitrary shaped object.
General Hough Transform

- Rotation Invariance
Say we’ve already stored a table of displacement vectors as a function of edge orientation for this model shape.
Example

Now we want to look at some edge points detected in a *new* image, and vote on the position of that shape.

displacement vectors for model points
Example

range of voting locations for test point
Example

range of voting locations for test point
Example

votes for points with $\theta = \uparrow$
Example

displacement vectors for model points
Example

range of voting locations for test point
Example

votes for points with $\theta = \theta$
Application in recognition

- Instead of indexing displacements by gradient orientation, index by “visual codeword”

B. Leibe, A. Leonardis, and B. Schiele, *Combined Object Categorization and Segmentation with an Implicit Shape Model*, ECCV Workshop on Statistical Learning in Computer Vision 2004

Source: L. Lazebnik
Application in recognition

• Instead of indexing displacements by gradient orientation, index by “visual codeword”
RANSAC

- RANSAC – Random Sampling Consensus

- Procedure:
  - Randomly generate a hypothesis
  - Test the hypothesis
  - Repeat for large enough times and choose the best one
Line Detection
Line Detection
Line Detection
Line Detection

Count “inliers”: 7
Line Detection

Inlier number: 3
After many trials, we decide this is the best line.
Probability of Success

- What’s the success rate of RANSAC given a fixed number of validations?
- Given a success rate, how to choose the minimum number of validations?
Object Detection Using RANSAC

Template

Target
Scale and Rotation Invariant Feature

- SIFT (D. Lowe, UBC)
Stable Feature
Stable Feature

Local max/min point’s values are stable when the scale changes
Filtering the image using filters at different scales. (for example using Gaussian filter)

\[ L(x, y, \sigma) = G(x, y, \sigma) \ast I(x, y), \]

where \( \ast \) is the convolution operation in \( x \) and \( y \), and

\[ G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}. \]
Difference of Gaussian
SIFT Feature Points

(b) Shows the points at the local max/min of DOG scale space for the image in (a).
Feature descriptors

- We know how to detect points
- Next question: **How to match them?**

Point descriptor should be:

1. Invariant
2. Distinctive
Feature matching
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
  - *kd*-trees and their variants
What about outliers?
Feature-space outlier rejection

- Let’s not match all features, but only these that have “similar enough” matches?
- How can we do it?
  - Best_match_cost/second_best_cost < threshold
  - How to set threshold?
Feature-space outlier rejection

- Can we now compute H from the blue points?
  - No! Still too many outliers…
  - What can we do?
RANSAC for estimating homography

RANSAC loop:
1. Select four feature pairs (at random) in the pre-pruned pair set.
2. Compute homography $H$ (exact)
3. Compute a matching score

Re-compute least-squares $H$ estimate on all of the inliers
RANSAC

Randomly pick up 4 point pairs in pre-pruned pair set.
RANSAC

Compute the Homograph H.
Project all the (inlier) points to the second image.
Compute a matching score using the “good” matches. Repeat the above procedure for a large number of times. Pick up the best guess and recompute H.
RANSAC
Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
Why Mosaic?

- Are you getting the whole picture?
  - Compact Camera FOV = 50 x 35°
  - Human FOV = 200 x 135°
  - Panoramic Mosaic = 360 x 180°

Slide from Brown & Lowe
Mosaics: stitching images together
How to do it?

- Basic Procedure
  - Take a sequence of images from the same position
    - Rotate the camera about its optical center
  - Compute transformation between second image and first
  - Transform the second image to overlap with the first
  - Blend the two together to create a mosaic
  - If there are more images, repeat

- ...but **wait**, why should this work at all?
  - What about the 3D geometry of the scene?
  - Why aren’t we using it?
Aligning images

left on top

right on top

Translations are not enough to align the images
The mosaic has a natural interpretation in 3D
- The images are reprojected onto a common plane
- The mosaic is formed on this plane
- Mosaic is a *synthetic wide-angle camera*
Example: Recognising Panoramas

M. Brown and D. Lowe,
University of British Columbia
Why “Recognising Panoramas”?
Why “Recognising Panoramas”?

- 1D Rotations ($\theta$)
  - Ordering $\Rightarrow$ matching images
Why “Recognising Panoramas”? 

- 1D Rotations ($\theta$)
  - Ordering $\Rightarrow$ matching images
Why “Recognising Panoramas”?

- 1D Rotations ($\theta$)
  - Ordering $\Rightarrow$ matching images
Why “Recognising Panoramas”?

- 1D Rotations ($\theta$)
  - Ordering $\Rightarrow$ matching images

- 2D Rotations ($\theta, \phi$)
  - Ordering $\nRightarrow$ matching images
Why “Recognising Panoramas”?

- 1D Rotations ($\theta$)
  - Ordering $\Rightarrow$ matching images

- 2D Rotations ($\theta$, $\phi$)
  - Ordering $\not\Rightarrow$ matching images
Why “Recognising Panoramas”?

- 1D Rotations ($\theta$)
  - Ordering $\Rightarrow$ matching images

- 2D Rotations ($\theta, \phi$)
  - Ordering $\not\Rightarrow$ matching images
Why “Recognising Panoramas”?
RANSAC for Homography
RANSAC for Homography
RANSAC for Homography
Probabilistic model for verification
Finding the panoramas
Finding the panoramas
Finding the panoramas
Finding the panoramas
Bundle Adjustment

- New images initialised with rotation, focal length of best matching image
Bundle Adjustment

- New images initialised with rotation, focal length of best matching image
Multi-band Blending

- Burt & Adelson 1983
  - Blend frequency bands over range $\propto \lambda$
Results