Computer Vision - Lecture 14

Recognition with Local Features

16.12.2013

Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/
leibe@vision.rwth-aachen.de
Announcements: You Wanted A Script...

- We’ve created a script... for the part of the lecture on object recognition & categorization
  - K. Grauman, B. Leibe
    Visual Object Recognition
    Morgan & Claypool publishers, 2011

- Chapter 3: Local Feature Extraction (Last 2 lectures)
- Chapter 4: Matching (Tuesday’s topic)
- Chapter 5: Geometric Verification (Today’s topic)

- Available on the L2P -
Announcements(2)

- Lecture evaluation
  - Please fill out the forms...
Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition
- Object Categorization I
  - Sliding Window based Object Detection
- Local Features & Matching
  - Local Features - Detection and Description
  - Recognition with Local Features
  - Indexing & Visual Vocabularies
- Object Categorization II
- 3D Reconstruction
- Motion and Tracking
Recap: Local Feature Matching Outline

1. Find a set of distinctive keypoints
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

\[ d(f_A, f_B) < T \]
Recap: Automatic Scale Selection

- Function responses for increasing scale (scale signature)

\[ f(I_{i_0, i_m}(x, \sigma)) \]

\[ f(I_{i_0, i_m}(x', \sigma')) \]

Slide credit: Krystian Mikolajczyk
Recap: Laplacian-of-Gaussian (LoG)

- Interest points:
  - Local maxima in scale space of Laplacian-of-Gaussian

\[ L_{xx}(\sigma) + L_{yy}(\sigma) \Rightarrow \text{List of } (x, y, \sigma) \]
Recap: LoG Detector Responses
Recap: Key point localization with DoG

- Efficient implementation
  - Approximate LoG with a difference of Gaussians (DoG)

- Approach: DoG Detector
  - Detect maxima of difference-of-Gaussian in scale space
  - Reject points with low contrast (threshold)
  - Eliminate edge responses
Recap: Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian (same procedure with Hessian ⇒ Hessian-Laplace)

Harris points

Harris-Laplace points

Slide adapted from Krystian Mikolajczyk
Recap: SIFT Feature Descriptor

- **Scale Invariant Feature Transform**
- **Descriptor computation:**
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions


Slide credit: Svetlana Lazebnik
Topics of This Lecture

• Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation

• Dealing with Outliers
  - RANSAC
  - Generalized Hough Transform

• Indexing with Local Features
  - Inverted file index
  - Visual Words
  - Visual Vocabulary construction
  - tf-idf weighting
Recognition with Local Features

- Image content is transformed into local features that are invariant to translation, rotation, and scale
- Goal: Verify if they belong to a consistent configuration

Local Features, e.g. SIFT

Slide credit: David Lowe
Concepts: Warping vs. Alignment

**Warping:** Given a source image and a transformation, what does the transformed output look like?

**Alignment:** Given two images with corresponding features, what is the transformation between them?

Slide credit: Kristen Grauman
Parametric (Global) Warping

- Transformation $T$ is a coordinate-changing machine:
  $$p' = T(p)$$

- What does it mean that $T$ is global?
  - It’s the same for any point $p$
  - It can be described by just a few numbers (parameters)

- Let’s represent $T$ as a matrix:
  $$p' = Mp$$
  $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} M$$

Slide credit: Alexej Efros
What Can be Represented by a $2\times2$ Matrix?

- **2D Scaling?**
  \[
  x' = s_x \cdot x \\
  y' = s_y \cdot y
  \]
  \[
  \begin{bmatrix}
  x' \\
  y'
  \end{bmatrix} =
  \begin{bmatrix}
  s_x & 0 \\
  0 & s_y
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y
  \end{bmatrix}
  \]

- **2D Rotation around (0,0)?**
  \[
  x' = \cos \theta \cdot x - \sin \theta \cdot y \\
  y' = \sin \theta \cdot x + \cos \theta \cdot y
  \]
  \[
  \begin{bmatrix}
  x' \\
  y'
  \end{bmatrix} =
  \begin{bmatrix}
  \cos \theta & -\sin \theta \\
  \sin \theta & \cos \theta
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y
  \end{bmatrix}
  \]

- **2D Shearing?**
  \[
  x' = x + sh_x \cdot y \\
  y' = sh_y \cdot x + y
  \]
  \[
  \begin{bmatrix}
  x' \\
  y'
  \end{bmatrix} =
  \begin{bmatrix}
  1 & sh_x \\
  sh_y & 1
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y
  \end{bmatrix}
  \]

Slide credit: Alexej Efros
What Can be Represented by a $2 \times 2$ Matrix?

- **2D Mirror about y axis?**
  \[
  x' = -x \\
  y' = y
  \]
  \[
  \begin{bmatrix}
  x' \\
  y'
  \end{bmatrix} =
  \begin{bmatrix}
  -1 & 0 \\
  0 & 1
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y
  \end{bmatrix}
  \]

- **2D Mirror over (0,0)?**
  \[
  x' = -x \\
  y' = -y
  \]
  \[
  \begin{bmatrix}
  x' \\
  y'
  \end{bmatrix} =
  \begin{bmatrix}
  -1 & 0 \\
  0 & -1
  \end{bmatrix}
  \begin{bmatrix}
  x \\
  y
  \end{bmatrix}
  \]

- **2D Translation?**
  \[
  x' = x + t_x \\
  y' = y + t_y
  \]
  NO!
2D Linear Transforms

\[
\begin{bmatrix}
  x' \\
  y'
\end{bmatrix} = \begin{bmatrix}
  a & b \\
  c & d
\end{bmatrix}\begin{bmatrix}
  x \\
  y
\end{bmatrix}
\]

- Only linear 2D transformations can be represented with a 2×2 matrix.
- Linear transformations are combinations of ...
  - Scale,
  - Rotation,
  - Shear, and
  - Mirror

Slide credit: Alexej Efros
Homogeneous Coordinates

- Q: How can we represent translation as a 3x3 matrix using homogeneous coordinates?

\[ x' = x + t_x \]
\[ y' = y + t_y \]

- A: Using the rightmost column:

\[
\begin{bmatrix}
1 & 0 & t_x \\
0 & 1 & t_y \\
0 & 0 & 1
\end{bmatrix}
\]

Translation
Basic 2D Transformations

- Basic 2D transformations as 3x3 matrices

Translation

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} = \begin{bmatrix}
    1 & 0 & t_x \\
    0 & 1 & t_y \\
    0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]

Scaling

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} = \begin{bmatrix}
    s_x & 0 & 0 \\
    0 & s_y & 0 \\
    0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]

Rotation

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} = \begin{bmatrix}
    \cos \theta & -\sin \theta & 0 \\
    \sin \theta & \cos \theta & 0 \\
    0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]

Shearing

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} = \begin{bmatrix}
    1 & sh_x & 0 \\
    sh_y & 1 & 0 \\
    0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]
2D Affine Transformations

\[
\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}
\]

- **Affine transformations** are combinations of ...
  - Linear transformations, and
  - Translations
- **Parallel lines remain parallel**
Projective Transformations

\[
\begin{bmatrix}
    x' \\
    y' \\
    w'
\end{bmatrix} =
\begin{bmatrix}
    a & b & c \\
    d & e & f \\
    g & h & i
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    w
\end{bmatrix}
\]

- **Projective transformations:**
  - Affine transformations, and
  - Projective warps

- **Parallel lines do not necessarily remain parallel**

Slide credit: Alexej Efros
Alignment Problem

• We have previously considered how to fit a model to image evidence
  ➢ e.g., a line to edge points

• In alignment, we will fit the parameters of some transformation according to a set of matching feature pairs (“correspondences”).
Let’s Start with Affine Transformations

- Simple fitting procedure (linear least squares)
- Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
- Can be used to initialize fitting for more complex models
Fitting an Affine Transformation

- Affine model approximates perspective projection of planar objects

Slide credit: Kristen Grauman

Image source: David Lowe
Fitting an Affine Transformation

• Assuming we know the correspondences, how do we get the transformation?

\[
\begin{pmatrix}
  x_i' \\
  y_i'
\end{pmatrix}
= \begin{pmatrix}
  m_1 & m_2 \\
  m_3 & m_4
\end{pmatrix}
\begin{pmatrix}
  x_i \\
  y_i
\end{pmatrix}
+ \begin{pmatrix}
  t_1 \\
  t_2
\end{pmatrix}
\]
Recall: Least Squares Estimation

- Set of data points: \((X_1, X'_1), (X_2, X'_2), (X_3, X'_3)\)
- Goal: a linear function to predict \(X'\)’s from \(X\)’s:
  \[ Xa + b = X' \]
- We want to find \(a\) and \(b\).
- How many \((X, X')\) pairs do we need?
  \[ X_1a + b = X'_1 \]
  \[ X_2a + b = X'_2 \]
- What if the data is noisy?
  \[
  \begin{bmatrix}
  X_1 & 1 \\
  X_2 & 1 \\
  X_3 & 1 \\
  \vdots & \vdots \\
  \end{bmatrix}
  \begin{bmatrix}
  a \\
  b \\
  \end{bmatrix}
  =
  \begin{bmatrix}
  X'_1 \\
  X'_2 \\
  X'_3 \\
  \vdots \\
  \end{bmatrix}
  \]
  Overconstrained problem
  \[
  \min \|Ax - B\|^2
  \]
  \(\Rightarrow\) Least-squares minimization

Solution:
\[ x = A^+B \]
Matlab:
\[ x = A\backslash B \]
Fitting an Affine Transformation

- Assuming we know the correspondences, how do we get the transformation?

\[
\begin{bmatrix}
  x'_i \\
  y'_i \\
\end{bmatrix} = \begin{bmatrix}
  m_1 & m_2 \\
  m_3 & m_4 \\
\end{bmatrix} \begin{bmatrix}
  x_i \\
  y_i \\
\end{bmatrix} + \begin{bmatrix}
  t_1 \\
  t_2 \\
\end{bmatrix}
\]

B. Leibe
Fitting an Affine Transformation

\[
\begin{bmatrix}
  x_i & y_i & 0 & 0 & 1 & 0 & \cdots \\
  0 & 0 & x_i & y_i & 0 & 1 & \cdots \\
  \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\end{bmatrix}
\begin{bmatrix}
  m_1 \\
  m_2 \\
  m_3 \\
  m_4 \\
  t_1 \\
  t_2 \\
\end{bmatrix}
=
\begin{bmatrix}
  x_i' \\
  y_i' \\
  \cdots \\
\end{bmatrix}
\]

- How many matches (correspondence pairs) do we need to solve for the transformation parameters?
- Once we have solved for the parameters, how do we compute the coordinates of the corresponding point for \((x_{\text{new}}, y_{\text{new}})\)?
Homography

- A projective transform is a mapping between any two perspective projections with the same center of projection.
  - I.e. two planes in 3D along the same sight ray
- Properties
  - Rectangle should map to arbitrary quadrilateral
  - Parallel lines aren’t
  - But must preserve straight lines
- This is called a homography

\[
\begin{bmatrix} wx' \\ wy' \\ w \\ p' \end{bmatrix} = \begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \\ \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}
\]

\( p' \)

Slide adapted from Alexej Efros
Homography

• A projective transform is a mapping between any two perspective projections with the same center of projection.
  - i.e. two planes in 3D along the same sight ray

• Properties
  - Rectangle should map to arbitrary quadrilateral
  - Parallel lines aren’t
  - but must preserve straight lines

• This is called a homography

\[
\begin{bmatrix}
wx' \\
w'y' \\
w \\
p'
\end{bmatrix} =
\begin{bmatrix}
h_{11} & h_{12} & h_{13} \\
h_{21} & h_{22} & h_{23} \\
h_{31} & h_{32} & h \\
1 & 1 & 1
\end{bmatrix}
\begin{bmatrix}
x \\
y \\
p
\end{bmatrix}
\]

Set scale factor to 1 ⇒ 8 parameters left.

Slide adapted from Alexej Efros
Fitting a Homography

- Estimating the transformation

\[
\begin{bmatrix}
    x' \\
    y' \\
    z'
\end{bmatrix}
= \begin{bmatrix}
    h_{11} & h_{12} & h_{13} \\
    h_{21} & h_{22} & h_{23} \\
    h_{31} & h_{32} & 1
\end{bmatrix}
\begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}
\]

Image coordinates

\[
\begin{bmatrix}
    x'' \\
    y'' \\
    1
\end{bmatrix}
= \frac{1}{z'}
\begin{bmatrix}
    x' \\
    y' \\
    z'
\end{bmatrix}
\]

Matrix notation

\[
x' = Hx
\]

\[
x'' = \frac{1}{z'} x'
\]
Fitting a Homography

- Estimating the transformation

\[
\begin{align*}
A_1 & \leftrightarrow B_1 \\
A_2 & \leftrightarrow B_2 \\
A_3 & \leftrightarrow B_3 \\
\vdots
\end{align*}
\]

**Homogenous coordinates**

\[
\begin{bmatrix}
x' \\ y' \\ z'
\end{bmatrix} =
\begin{bmatrix}
h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1
\end{bmatrix}
\begin{bmatrix}
x \\ y \\ 1
\end{bmatrix}
\]

**Image coordinates**

\[
\begin{bmatrix}
x'' \\ y'' \\ 1
\end{bmatrix} = \frac{1}{z'}
\begin{bmatrix}
x' \\ y' \\ z'
\end{bmatrix}
\]

**Matrix notation**

\[
x' = Hx
\]

\[
x'' = \frac{1}{z'} x'
\]
Fitting a Homography

- Estimating the transformation

Image coordinates

Homogenous coordinates

Matrix notation

\[
\begin{align*}
\mathbf{x'} &= \mathbf{Hx} \\
\mathbf{x''} &= \frac{1}{z'} \mathbf{x'}
\end{align*}
\]
Fitting a Homography

- Estimating the transformation

\[ x' = Hx \]
\[ x'' = \frac{1}{z'} x' \]

\[ x_{A_i} = \frac{h_{11} x_{B_i} + h_{12} y_{B_i} + h_{13}}{h_{31} x_{B_i} + h_{32} y_{B_i} + 1} \]
\[ y_{A_i} = \frac{h_{21} x_{B_i} + h_{22} y_{B_i} + h_{23}}{h_{31} x_{B_i} + h_{32} y_{B_i} + 1} \]
Fitting a Homography

- Estimating the transformation

\[ x_{A_1} \leftrightarrow x_{B_1} \]
\[ x_{A_2} \leftrightarrow x_{B_2} \]
\[ x_{A_3} \leftrightarrow x_{B_3} \]
\[ \vdots \]

Homogenous coordinates

\[ x_{A_i} = \frac{h_{11} x_{B_i} + h_{12} y_{B_i} + h_{13}}{h_{31} x_{B_i} + h_{32} y_{B_i} + 1} \]

Image coordinates

\[ y_{A_i} = \frac{h_{21} x_{B_i} + h_{22} y_{B_i} + h_{23}}{h_{31} x_{B_i} + h_{32} y_{B_i} + 1} \]

\[ x_{A_i} h_{31} x_{B_i} + x_{A_i} h_{32} y_{B_i} + x_{A_i} = h_{11} x_{B_i} + h_{12} y_{B_i} + h_{13} \]
Fitting a Homography

- Estimating the transformation

\[ x_{A_i} = \frac{h_{11} x_{B_i} + h_{12} y_{B_i} + h_{13}}{h_{31} x_{B_i} + h_{32} y_{B_i} + 1} \]
\[ y_{A_i} = \frac{h_{21} x_{B_i} + h_{22} y_{B_i} + h_{23}}{h_{31} x_{B_i} + h_{32} y_{B_i} + 1} \]

\[ x_{A_i} h_{31} x_{B_i} + x_{A_i} h_{32} y_{B_i} + x_{A_i} = h_{11} x_{B_i} + h_{12} y_{B_i} + h_{13} \]
\[ y_{A_i} h_{31} y_{B_i} + y_{A_i} h_{32} x_{B_i} + y_{A_i} = h_{21} x_{B_i} + h_{22} y_{B_i} + h_{23} \]

Image coordinates

\[ h_{11} x_{B_i} + h_{12} y_{B_i} + h_{13} - x_{A_i} h_{31} x_{B_i} - x_{A_i} h_{32} y_{B_i} - x_{A_i} = 0 \]
\[ h_{21} x_{B_i} + h_{22} y_{B_i} + h_{23} - y_{A_i} h_{31} x_{B_i} - y_{A_i} h_{32} y_{B_i} - y_{A_i} = 0 \]

Homogenous coordinates

Slide credit: Krystian Mikolajczyk
Fitting a Homography

- Estimating the transformation

\[
\begin{align*}
    h_{11} x_B + h_{12} y_B + h_{13} - x_{A1} h_{31} x_B - x_{A1} h_{32} y_B - x_{A1} &= 0 \\
    h_{21} x_B + h_{22} y_B + h_{23} - y_{A1} h_{31} x_B - y_{A1} h_{32} y_B - y_{A1} &= 0
\end{align*}
\]

\[
\begin{pmatrix}
    x_B & y_B & 1 & 0 & 0 & 0 & -x_{A1} x_B & -x_{A1} y_B & -x_{A1} \\
    0 & 0 & 0 & x_B & y_B & 1 & -y_{A1} x_B & -y_{A1} y_B & -y_{A1}
\end{pmatrix}
\begin{pmatrix}
    h_{11} \\
    h_{12} \\
    h_{13} \\
    h_{21} \\
    h_{22} \\
    h_{23} \\
    h_{31} \\
    h_{32} \\
    1
\end{pmatrix} = \begin{pmatrix}
    0 \\
    0 \\
    .
\end{pmatrix}
\]

\[
Ah = 0
\]
Fitting a Homography

- Estimating the transformation
- Solution:
  - Null-space vector of A

\[ Ah = 0 \]

Slide credit: Krystian Mikolajczyk

B. Leibe
Fitting a Homography

- Estimating the transformation
- Solution:
  - Null-space vector of $A$
  - Corresponds to smallest singular vector

\[ A h = 0 \]

\[
\begin{bmatrix}
  d_{11} & \cdots & 0 \\
  \vdots & \ddots & \vdots \\
  0 & \cdots & d_{99}
\end{bmatrix}
\begin{bmatrix}
  v_{11} \\
  \vdots \\
  v_{99}
\end{bmatrix}

\[ h = \begin{bmatrix} v_{19}, \cdots, v_{99} \end{bmatrix} \]

Minimizes least square error

Slide credit: Krystian Mikolajczyk
Image Warping with Homographies

Image plane in front

Black area where no pixel maps to $p'$

Slide credit: Steve Seitz
Uses: Analyzing Patterns and Shapes

• What is the shape of the b/w floor pattern?

The floor (enlarged)

Slide credit: Antonio Criminisi
Analyzing Patterns and Shapes

From Martin Kemp *The Science of Art* (manual reconstruction)

Automatic rectification

Slide credit: Antonio Criminisi
Topics of This Lecture

• Recognition with Local Features
  ➢ Matching local features
  ➢ Finding consistent configurations
  ➢ Alignment: linear transformations
  ➢ Affine estimation
  ➢ Homography estimation

• Dealing with Outliers
  ➢ RANSAC
  ➢ Generalized Hough Transform

• Indexing with Local Features
  ➢ Inverted file index
  ➢ Visual Words
  ➢ Visual Vocabulary construction
  ➢ tf-idf weighting
Problem: Outliers

- Outliers can hurt the quality of our parameter estimates, e.g.,
  - An erroneous pair of matching points from two images
  - A feature point that is noise or doesn’t belong to the transformation we are fitting.
Example: Least-Squares Line Fitting

• Assuming all the points that belong to a particular line are known
Outliers Affect Least-Squares Fit

Source: Forsyth & Ponce
Outliers Affect Least-Squares Fit
Strategy 1: RANSAC [Fischler81]

- **RAN**dom **SA**mple **C**onsensus

- Approach: we want to avoid the impact of outliers, so let’s look for “inliers”, and use only those.

- Intuition: if an outlier is chosen to compute the current fit, then the resulting line won’t have much support from rest of the points.
RANSAC

RANSAC loop:

1. Randomly select a *seed group* of points on which to base transformation estimate (e.g., a group of matches)

2. Compute transformation from seed group

3. Find *inliers* to this transformation

4. If the number of inliers is sufficiently large, re-compute least-squares estimate of transformation on all of the inliers

- Keep the transformation with the largest number of inliers
RANSAC Line Fitting Example

- Task: Estimate the best line
  - How many points do we need to estimate the line?
RANSAC Line Fitting Example

- Task: Estimate the best line

Sample two points

Slide credit: Jinxiang Chai
RANSAC Line Fitting Example

- Task: Estimate the best line

Fit a line to them
RANSAC Line Fitting Example

- Task: Estimate the best line

Total number of points within a threshold of line.

Slide credit: Jinxiang Chai
RANSAC Line Fitting Example

- Task: Estimate the best line

"7 inlier points"

Total number of points within a threshold of line.

Slide credit: Jinxiang Chai
RANSAC Line Fitting Example

- Task: Estimate the best line

Repeat, until we get a good result.

Slide credit: Jinxiang Chai
RANSAC Line Fitting Example

- Task: Estimate the best line

"11 inlier points"

Repeat, until we get a good result.

Slide credit: Jinxiang Chai
RANSAC: How many samples?

• How many samples are needed?
  - Suppose $w$ is fraction of inliers (points from line).
  - $n$ points needed to define hypothesis (2 for lines)
  - $k$ samples chosen.

• Prob. that a single sample of $n$ points is correct: $w^n$

• Prob. that all $k$ samples fail is: $(1-w^n)^k$

⇒ Choose $k$ high enough to keep this below desired failure rate.
## RANSAC: Computed k (p=0.99)

<table>
<thead>
<tr>
<th>Sample size n</th>
<th>5%</th>
<th>10%</th>
<th>20%</th>
<th>25%</th>
<th>30%</th>
<th>40%</th>
<th>50%</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>11</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>7</td>
<td>9</td>
<td>11</td>
<td>19</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>5</td>
<td>9</td>
<td>13</td>
<td>17</td>
<td>34</td>
<td>72</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>6</td>
<td>12</td>
<td>17</td>
<td>26</td>
<td>57</td>
<td>146</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>7</td>
<td>16</td>
<td>24</td>
<td>37</td>
<td>97</td>
<td>293</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>8</td>
<td>20</td>
<td>33</td>
<td>54</td>
<td>163</td>
<td>588</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>9</td>
<td>26</td>
<td>44</td>
<td>78</td>
<td>272</td>
<td>1177</td>
</tr>
</tbody>
</table>
After RANSAC

- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers.
- Improve this initial estimate with estimation over all inliers (e.g. with standard least-squares minimization).
- But this may change inliers, so alternate fitting with re-classification as inlier/outlier.
Example: Finding Feature Matches

- Find best stereo match within a square search window (here 300 pixels²)
- Global transformation model: epipolar geometry

Images from Hartley & Zisserman
Example: Finding Feature Matches

- Find best stereo match within a square search window (here 300 pixels$^2$)
- Global transformation model: epipolar geometry

Images from Hartley & Zisserman

Slide credit: David Lowe
Problem with RANSAC

- In many practical situations, the percentage of outliers (incorrect putative matches) is often very high (90% or above).
- Alternative strategy: Generalized Hough Transform
Strategy 2: Generalized Hough Transform

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).
Strategy 2: Generalized Hough Transform

- Suppose our features are scale- and rotation-invariant
  - Then a single feature match provides an alignment hypothesis (translation, scale, orientation).
  - Of course, a hypothesis from a single match is unreliable.
  - Solution: let each match vote for its hypothesis in a Hough space with very coarse bins.

Slide credit: Svetlana Lazebnik
Pose Clustering and Verification with SIFT

• To detect instances of objects from a model base:

1. Index descriptors
   • Distinctive features narrow down possible matches
Indexing Local Features

Model base

New image
Pose Clustering and Verification with SIFT

- To detect instances of objects from a model base:

1. Index descriptors
   - Distinctive features narrow down possible matches

2. Generalized Hough transform to vote for poses
   - Keypoints have record of parameters relative to model coordinate system

3. Affine fit to check for agreement between model and image features
   - Fit and verify using features from Hough bins with 3+ votes

Slide credit: Kristen Grauman
Image source: David Lowe
Object Recognition Results

Background subtract for model boundaries

Objects recognized

Recognition in spite of occlusion

Slide credit: Kristen Grauman

Image source: David Lowe
Location Recognition

Training

[Low, IJCV’04]

B. Leibe

Slide credit: David Lowe
Recall: Difficulties of Voting

- Noise/clutter can lead to as many votes as true target.
- Bin size for the accumulator array must be chosen carefully.
- (Recall Hough Transform)

- In practice, good idea to make broad bins and spread votes to nearby bins, since verification stage can prune bad vote peaks.
Summary

• Recognition by alignment: looking for object and pose that fits well with image
  - Use good correspondences to designate hypotheses.
  - Invariant local features offer more reliable matches.
  - Find consistent “inlier” configurations in clutter
    - Generalized Hough Transform
    - RANSAC

• Alignment approach to recognition can be effective if we find reliable features within clutter.
  - Application: large-scale image retrieval
  - Application: recognition of specific (mostly planar) objects
    - Movie posters
    - Books
    - CD covers
References and Further Reading

• A detailed description of local feature extraction and recognition can be found in Chapters 3-5 of Grauman & Leibe (available on the L2P).

  ➢ K. Grauman, B. Leibe
    Visual Object Recognition
    Morgan & Claypool publishers, 2011

  ➢ R. Hartley, A. Zisserman
    Multiple View Geometry in Computer Vision
    2nd Ed., Cambridge Univ. Press, 2004

• More details on RANSAC can also be found in Chapter 4.7 of Hartley & Zisserman.