Computer Vision - Lecture 13

Recognition with Local Features

15.12.2015

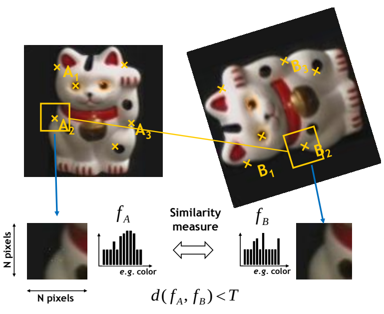
Bastian Leibe
RWTH Aachen
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

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

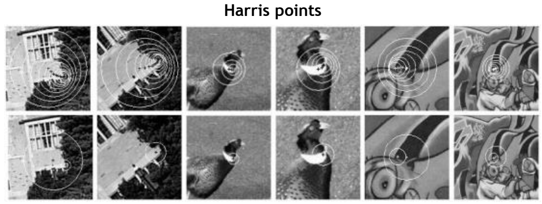


1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

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

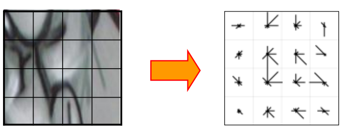


Harris-Laplace points

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



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

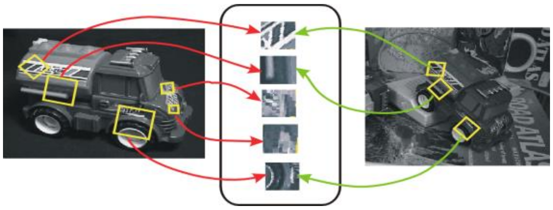
Slide credit: Svetlana Lazebnik

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

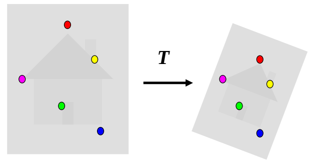
Recognition with Local Features

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

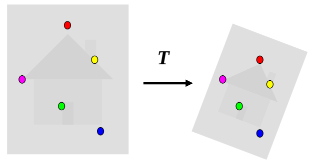


Local Features,
e.g. SIFT

Concepts: Warping vs. Alignment

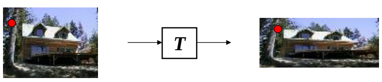


Warping: Given a source image and a transformation, what does the transformed output look like?



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

Parametric (Global) Warping



p = (x,y) p' = (x',y')

- 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:

$$p' = \mathbf{M}p\,, \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{M}\begin{bmatrix} x \\ y \end{bmatrix}$$

What Can be Represented by a 2×2 Matrix?

- 2D Scaling?

$$x' = s_x * x \qquad y' = s_y * y \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Rotation around (0,0)?

$$x' = \cos\theta * x - \sin\theta * y \qquad y' = \sin\theta * x + \cos\theta * y \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Shearing?

$$x' = x + sh_x * y \qquad y' = sh_y * x + y \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & sh_x \\ sh_y & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

What Can be Represented by a 2×2 Matrix?

- 2D Mirror about y axis?

$$x' = -x \qquad y' = y \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Mirror over (0,0)?

$$x' = -x \qquad y' = -y \qquad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

- 2D Translation?

$$x' = x + t_x \qquad y' = y + t_y$$

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

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:

$$\text{Translation} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Basic 2D Transformations

- Basic 2D transformations as 3x3 matrices

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

Translation

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

Scaling

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

Rotation

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & sh_x & 0 \\ sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Shearing

2D Affine Transformations

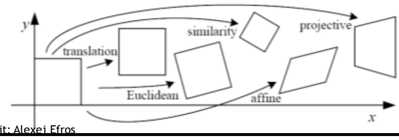
$$\begin{bmatrix} x' \\ y' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- Affine transformations are combinations of …
  - Linear transformations, and
  - Translations
- *Parallel lines remain parallel*

Projective Transformations

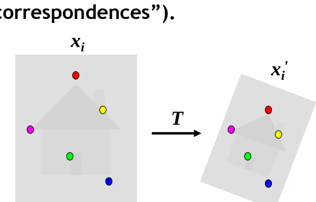
$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

- Projective transformations:
  - Affine transformations, and
  - Projective warps
- *Parallel lines do not necessarily remain parallel*



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

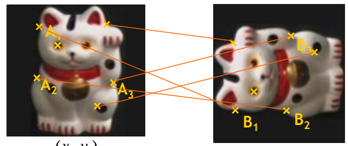
- Simple fitting procedure (linear least squares)
- Approximates viewpoint changes for roughly planar objects and roughly orthographic cameras
- Can be used to initialize fitting for more complex models


Fitting an Affine Transformation

- Affine model approximates perspective projection of planar objects

Fitting an Affine Transformation

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



(x_i, y_i) (x'_i, y'_i)

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

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 Xs:

  $Xa + b = X^{'}$
- We want to find a and b.
- How many $(X, X^{'})$ pairs do we need?

  $X_1 a + b = X_1^{'}$

  $X_2 a + b = X_2^{'}$

  $$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1^{'} \\ X_2^{'} \end{bmatrix} \qquad Ax = B$$
- What if the data is noisy?

  $$\begin{bmatrix} X_1 & 1 \\ X_2 & 1 \\ X_3 & 1 \\ \dots & \dots \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1^{'} \\ X_2^{'} \\ X_3^{'} \\ \dots \end{bmatrix}$$

  Overconstrained problem

  $\min \|Ax - B\|^2$

  ⇒ Least-squares minimization

  Solution:

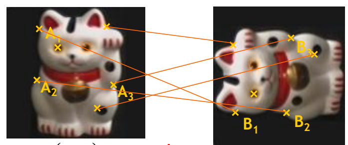
  $x = A^{+}B$

  Matlab:

  $x = A \backslash B$

Fitting an Affine Transformation

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



(x_i, y_i) (x'_i, y'_i)

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

Fitting an Affine Transformation

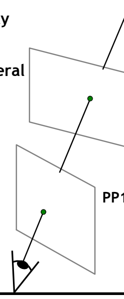
$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x'_i \\ y'_i \\ \cdots \end{bmatrix}$$

- 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_{new}, y_{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**

$$\underset{p'}{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}} = \underset{H}{\begin{bmatrix} * & * & * \\ * & * & * \\ * & * & * \end{bmatrix}} \underset{p}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$



RWTH AACHEN UNIVERSITY

Computer Vision WS 15/16

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

$$\underbrace{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}}_{p'} = \underbrace{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}_{p}$$

Set scale factor to 1
⇒ 8 parameters left.

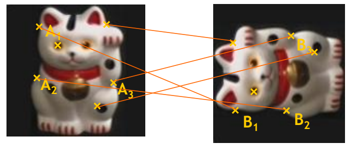
Slide adapted from Alexei Efros
B. Leibe

RWTH AACHEN UNIVERSITY

Computer Vision WS 15/16

Fitting a Homography

- Estimating the transformation



Homogenous coordinates | Image coordinates | Matrix notation

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$x' = Hx \qquad x'' = \tfrac{1}{z'} x'$$

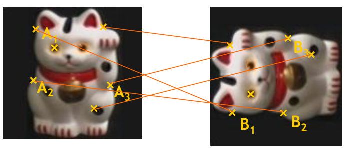
Slide credit: Krystian Mikolajczyk
B. Leibe
29

RWTH AACHEN UNIVERSITY

Computer Vision WS 15/16

Fitting a Homography

- Estimating the transformation



Homogenous coordinates | Image coordinates | Matrix notation

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$x' = Hx \qquad x'' = \tfrac{1}{z'} x'$$

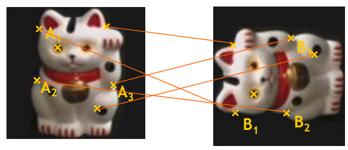
Slide credit: Krystian Mikolajczyk
B. Leibe
30

RWTH AACHEN UNIVERSITY

Computer Vision WS 15/16

Fitting a Homography

- Estimating the transformation



Homogenous coordinates | Image coordinates | Matrix notation

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$x' = Hx \qquad x'' = \tfrac{1}{z'} x'$$

$$x_{A_1} = \frac{h_{11}\, x_{B_1} + h_{12} y_{B_1} + h_{13}}{h_{31}\, x_{B_1} + h_{32} y_{B_1} + 1}$$

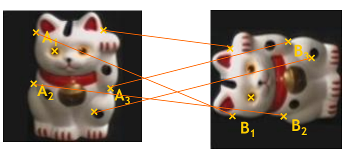
Slide credit: Krystian Mikolajczyk
B. Leibe
31

RWTH AACHEN UNIVERSITY

Computer Vision WS 15/16

Fitting a Homography

- Estimating the transformation



Homogenous coordinates | Image coordinates | Matrix notation

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \frac{1}{z'} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$$

$$x' = Hx \qquad x'' = \tfrac{1}{z'} x'$$

$$x_{A_1} = \frac{h_{11}\, x_{B_1} + h_{12} y_{B_1} + h_{13}}{h_{31}\, x_{B_1} + h_{32} y_{B_1} + 1} \qquad y_{A_1} = \frac{h_{21}\, x_{B_1} + h_{22} y_{B_1} + h_{23}}{h_{31}\, x_{B_1} + h_{32} y_{B_1} + 1}$$

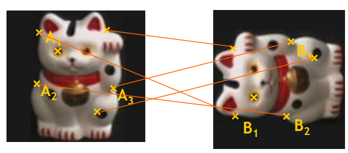
Slide credit: Krystian Mikolajczyk
B. Leibe
32

RWTH AACHEN UNIVERSITY

Computer Vision WS 15/16

Fitting a Homography

- Estimating the transformation



Homogenous coordinates | Image coordinates

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

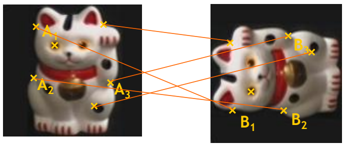
$$x_{A_1} = \frac{h_{11}\, x_{B_1} + h_{12} y_{B_1} + h_{13}}{h_{31}\, x_{B_1} + h_{32} y_{B_1} + 1} \qquad y_{A_1} = \frac{h_{21}\, x_{B_1} + h_{22} y_{B_1} + h_{23}}{h_{31}\, x_{B_1} + h_{32} y_{B_1} + 1}$$

$$x_{A_1} h_{31}\, x_{B_1} + x_{A_1} h_{32} y_{B_1} + x_{A_1} = h_{11}\, x_{B_1} + h_{12} y_{B_1} + h_{13}$$

Slide credit: Krystian Mikolajczyk
B. Leibe
33

Fitting a Homography

- Estimating the transformation



Homogenous coordinates | Image coordinates

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

$$x_{A_1} = \frac{h_{11}x_{B_1}+h_{12}y_{B_1}+h_{13}}{h_{31}x_{B_1}+h_{32}y_{B_1}+1} \qquad y_{A_1} = \frac{h_{21}x_{B_1}+h_{22}y_{B_1}+h_{23}}{h_{31}x_{B_1}+h_{32}y_{B_1}+1}$$

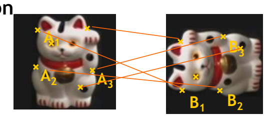
$$x_{A_1}h_{31}x_{B_1}+x_{A_1}h_{32}y_{B_1}+x_{A_1} = h_{11}x_{B_1}+h_{12}y_{B_1}+h_{13}$$

$$\boxed{\begin{aligned} h_{11}x_{B_1}+h_{12}y_{B_1}+h_{13}-x_{A_1}h_{31}x_{B_1}-x_{A_1}h_{32}y_{B_1}-x_{A_1} &= 0 \\ h_{21}x_{B_1}+h_{22}y_{B_1}+h_{23}-y_{A_1}h_{31}x_{B_1}-y_{A_1}h_{32}y_{B_1}-y_{A_1} &= 0 \end{aligned}}$$

Fitting a Homography

- Estimating the transformation

$$\boxed{\begin{aligned} h_{11}x_{B_1}+h_{12}y_{B_1}+h_{13}-x_{A_1}h_{31}x_{B_1}-x_{A_1}h_{32}y_{B_1}-x_{A_1} &= 0 \\ h_{21}x_{B_1}+h_{22}y_{B_1}+h_{23}-y_{A_1}h_{31}x_{B_1}-y_{A_1}h_{32}y_{B_1}-y_{A_1} &= 0 \end{aligned}}$$



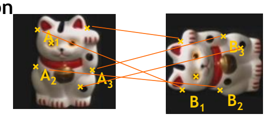
$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

$$\begin{bmatrix} x_{B_1} & y_{B_1} & 1 & 0 & 0 & 0 & -x_{A_1}x_{B_1} & -x_{A_1}y_{B_1} & -x_{A_1} \\ 0 & 0 & 0 & x_{B_1} & y_{B_1} & 1 & -y_{A_1}x_{B_1} & -y_{A_1}y_{B_1} & -y_{A_1} \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \end{bmatrix} \cdot \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ . \end{bmatrix}$$

$$Ah = 0$$

Fitting a Homography

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



$$Ah = 0$$

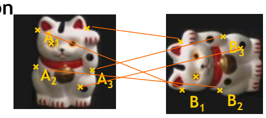
$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

SVD

$$\mathbf{A} = \mathbf{?}$$

Fitting a Homography

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



$$Ah = 0$$

$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

SVD

$$\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{V}^T = \mathbf{U}\begin{bmatrix} d_{11} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{99} \end{bmatrix}\begin{bmatrix} v_{11} & \cdots & \boxed{v_{19}} \\ \vdots & \ddots & \vdots \\ v_{91} & \cdots & v_{99} \end{bmatrix}^T$$

$$\boxed{\mathbf{h} = \frac{[v_{19}, \cdots, v_{99}]}{v_{99}}}$$

Minimizes least square error

Image Warping with Homographies

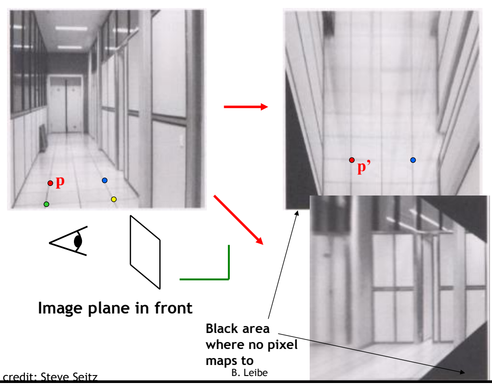
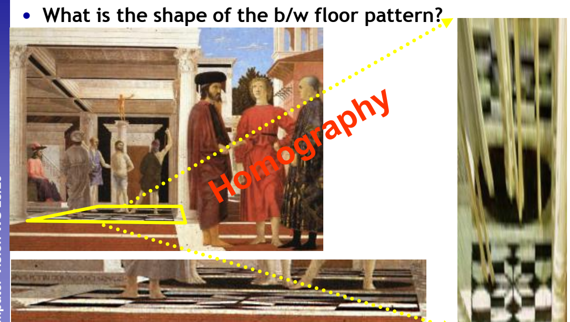


Image plane in front

Black area where no pixel maps to

Uses: Analyzing Patterns and Shapes

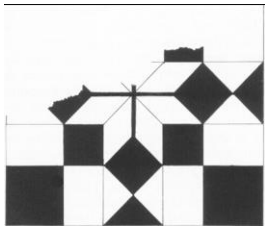
- What is the shape of the b/w floor pattern?



The floor (enlarged)

Analyzing Patterns and Shapes

Automatic rectification



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

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

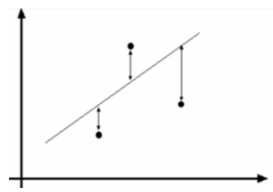
Problem: Outliers

- 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.

Slide credit: Kristen Grauman

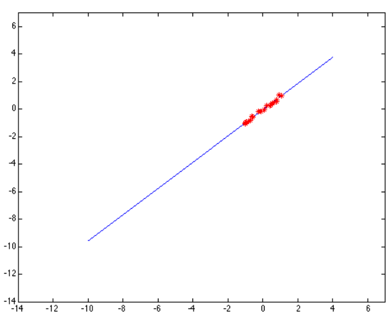
Example: Least-Squares Line Fitting

- Assuming all the points that belong to a particular line are known



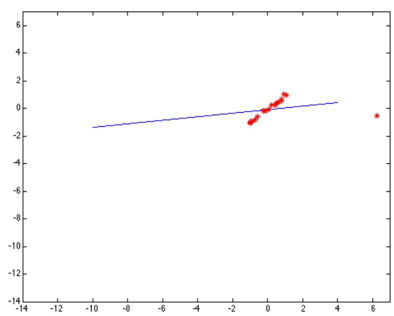
Source: Forsyth & Ponce

Outliers Affect Least-Squares Fit



Source: Forsyth & Ponce

Outliers Affect Least-Squares Fit



Source: Forsyth & Ponce

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

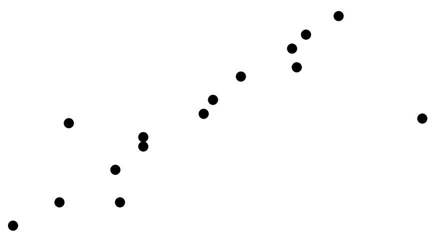
RANSAC loop:

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

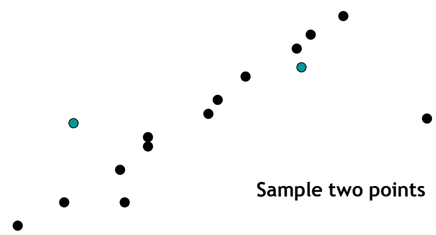
RANSAC Line Fitting Example

- Task: Estimate the best line
  - *How many points do we need to estimate the line?*



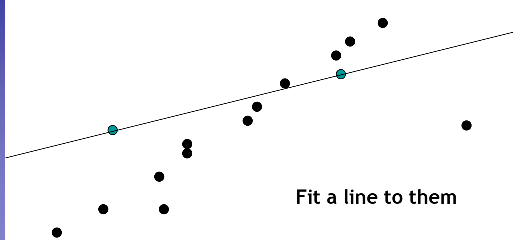
RANSAC Line Fitting Example

- Task: Estimate the best line



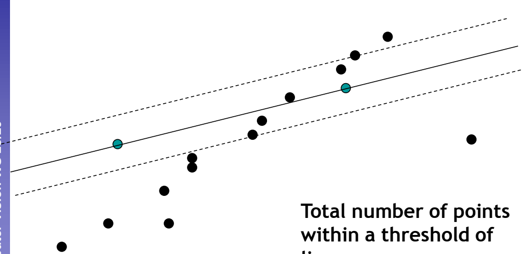
RANSAC Line Fitting Example

- Task: Estimate the best line



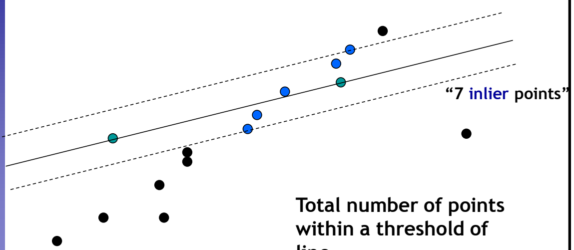
RANSAC Line Fitting Example

- Task: Estimate the best line



RANSAC Line Fitting Example

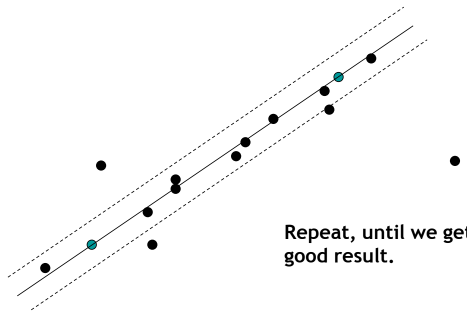
- Task: Estimate the best line



Total number of points within a threshold of line.

RANSAC Line Fitting Example

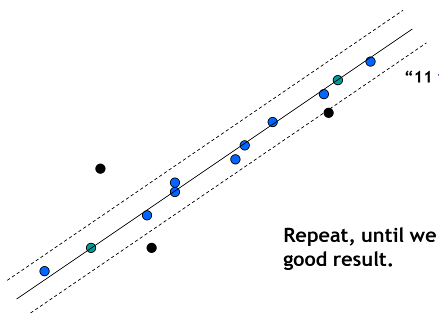
- Task: Estimate the best line



Repeat, until we get a good result.

RANSAC Line Fitting Example

- Task: Estimate the best line



Repeat, until we get a good result.

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: $(1-w^n)^k$

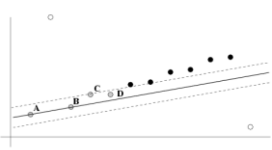
⇒ Choose k high enough to keep this below desired failure rate.

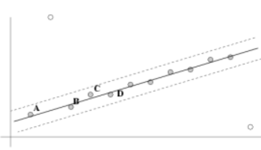
RANSAC: Computed k (p=0.99)

Sample size n	Proportion of outliers 5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

After RANSAC

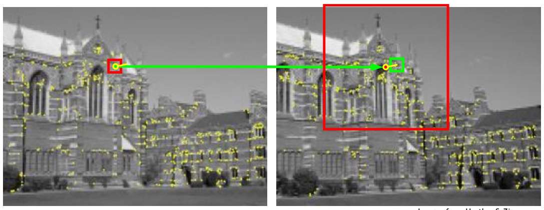
- RANSAC divides data into inliers and outliers and yields estimate computed from minimal set of inliers.
- Improve this initial estimate with estimation over all inliers (e.g. with standard least-squares minimization).
- But this may change inliers, so alternate fitting with re-classification as inlier/outlier.





Example: Finding Feature Matches

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



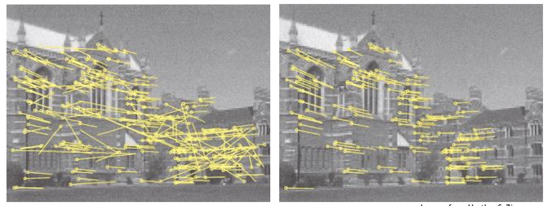
Images from Hartley & Zisserman

Slide credit: David Lowe

Example: Finding Feature Matches

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

before RANSAC after RANSAC



Images from Hartley & Zisserman

Slide credit: David Lowe

Problem with RANSAC

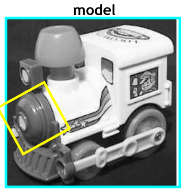
- In many practical situations, the percentage of outliers (incorrect putative matches) is often very high (90% or above).
- Alternative strategy: Generalized Hough Transform

Slide credit: Svetlana Lazebnik

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).

model

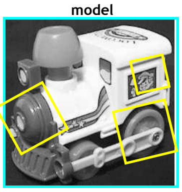


Slide credit: Svetlana Lazebnik

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.

model



Slide credit: Svetlana Lazebnik

Pose Clustering and Verification with SIFT

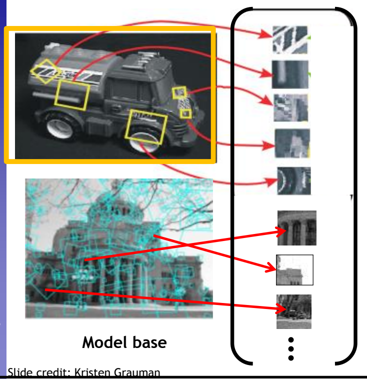
- To detect instances of objects from a model base:

1. Index descriptors
   - Distinctive features narrow down possible matches

Slide credit: Kristen Grauman

Image source: David Lowe

Indexing Local Features



Slide credit: Kristen Grauman

Image source: David Lowe

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

B. Leibe

65

Image source: David Lowe

Object Recognition Results

Background subtract for model boundaries

Objects recognized

Recognition in spite of occlusion

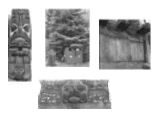
Slide credit: Kristen Grauman

B. Leibe

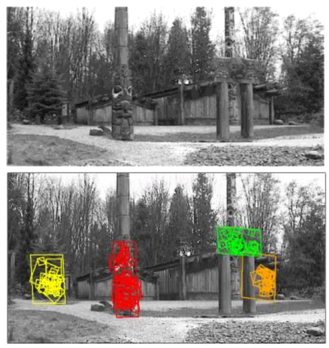
66

Image source: David Lowe

Location Recognition



Training



[Lowe, IJCV'04]

B. Leibe

67

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.

B. Leibe

68

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

B. Leibe

70

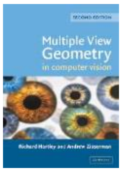
Computer Vision WS 15/16

RWTH AACHEN UNIVERSITY

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.