Computer Vision – Lecture 9

Local Features II

20.05.2019

Bastian Leibe
Visual Computing Institute
RWTH Aachen University
http://www.vision.rwth-aachen.de/

leibe@vision.rwth-aachen.de

Course Outline

- Image Processing Basics
- Segmentation & Grouping
- Object Recognition & Categorization
  - Sliding Window based Object Detection
- Local Features & Matching
  - Local Features – Detection and Description
  - Recognition with Local Features
- Deep Learning
- 3D Reconstruction

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

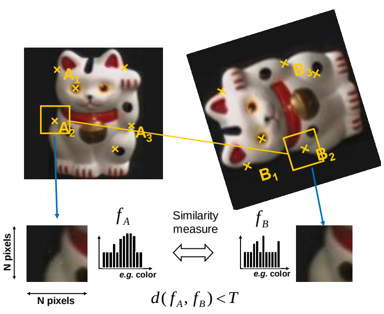
- Chapter 3: Local Feature Extraction (Last lecture)
- Chapter 5: Geometric Verification (Today)

– Available on moodle –

Topics of This Lecture

- Recap: Local Feature Extraction
- Local Descriptors
  - SIFT
  - Applications
- Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation
- Dealing with Outliers
  - RANSAC
  - Generalized Hough Transform

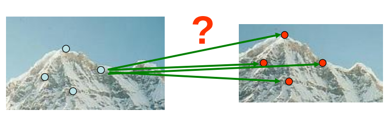
Recap: Local Feature Matching Outline



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: Requirements for Local Features

- Problem 1:
  - Detect the same point *independently* in both images
- Problem 2:
  - For each point correctly recognize the corresponding one



We need a repeatable detector!

We need a reliable and distinctive descriptor!

Slide credit: Darya Frolova, Denis Simakov

Recap: Harris Detector [Harris88]

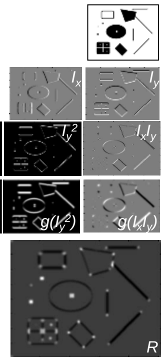
- Compute second moment matrix (autocorrelation matrix)

$$M(\sigma_I,\sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

1. Image derivatives
2. Square of derivatives
3. Gaussian filter $g(\sigma_I)$
4. Cornerness function – two strong eigenvalues

$$R = \det[M(\sigma_I,\sigma_D)] - \alpha[\operatorname{trace}(M(\sigma_I,\sigma_D))]^2$$
$$= g(I_x^2)g(I_y^2) - [g(I_x I_y)]^2 - \alpha[g(I_x^2) + g(I_y^2)]^2$$

5. Perform non-maximum suppression



Slide credit: Krystian Mikolajczyk

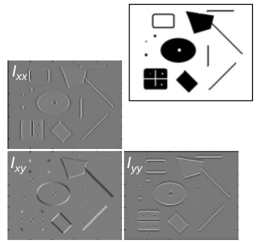
Hessian Detector [Beaudet78]

- Hessian determinant

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

Note: these are 2nd derivatives!

Intuition: Search for strong derivatives in two orthogonal directions



Slide credit: Krystian Mikolajczyk

Hessian Detector [Beaudet78]

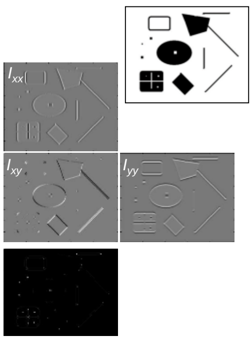
- Hessian determinant

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$

$$\det(Hessian(I)) = I_{xx} I_{yy} - I_{xy}^2$$

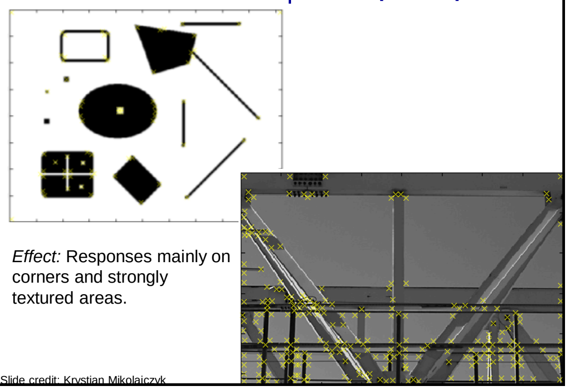
In Matlab:

$$I_{xx} .* I_{yy} - (I_{xy})\char`^2$$



Slide credit: Krystian Mikolajczyk

Hessian Detector – Responses [Beaudet78]



Effect: Responses mainly on corners and strongly textured areas.

Slide credit: Krystian Mikolajczyk

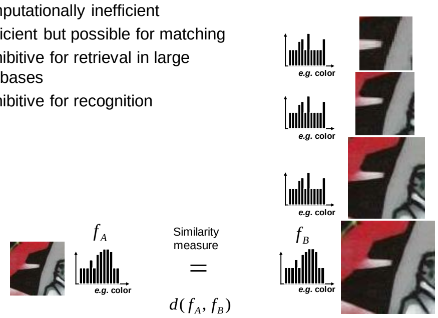
From Points to Regions…

- The Harris and Hessian operators define interest points.
  - Precise localization
  - High repeatability

- In order to compare those points, we need to compute a descriptor over a region.
  - How can we define such a region in a scale invariant manner?
- *I.e. how can we detect scale invariant interest regions?*

Naïve Approach: Exhaustive Search

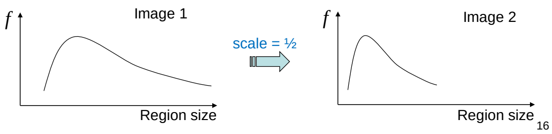
- Comparing descriptors while varying the patch size
  - Computationally inefficient
  - Inefficient but possible for matching
  - Prohibitive for retrieval in large databases
  - Prohibitive for recognition



Slide credit: Krystian Mikolajczyk

Automatic Scale Selection

- Solution:
  - Design a signature function on the region that is "scale invariant" (*the same for corresponding regions, even if they are at different scales*)
  - For a point in one image, we can consider it as a function of region size (patch width)



Slide credit: Kristen Grauman

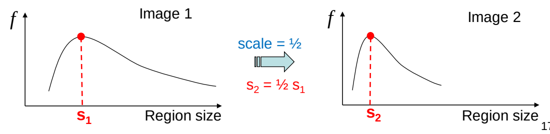
Computer Vision Summer'19

16

Automatic Scale Selection

- 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!



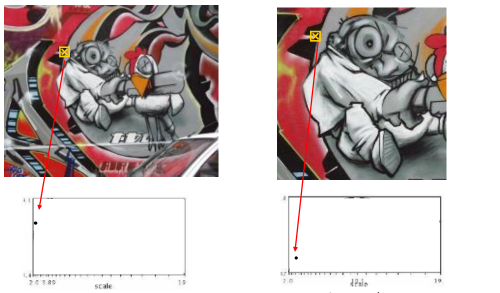
Slide credit: Kristen Grauman

Computer Vision Summer'19

17

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

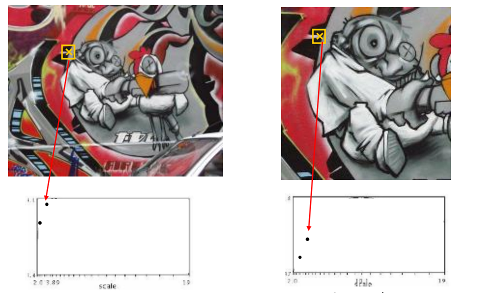
B. Leibe

Computer Vision Summer'19

18

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

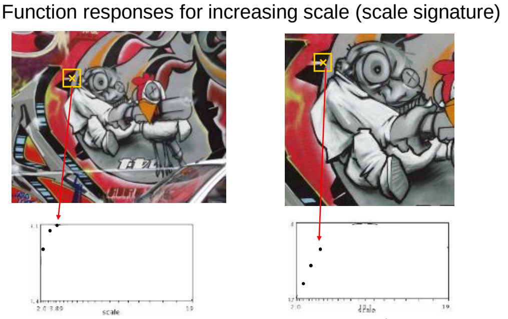
B. Leibe

Computer Vision Summer'19

19

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

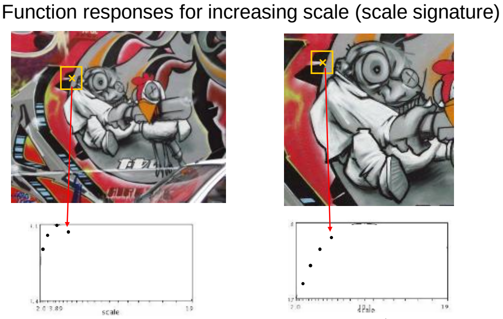
B. Leibe

Computer Vision Summer'19

20

Automatic Scale Selection

- Function responses for increasing scale (scale signature)



$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

Slide credit: Krystian Mikolajczyk

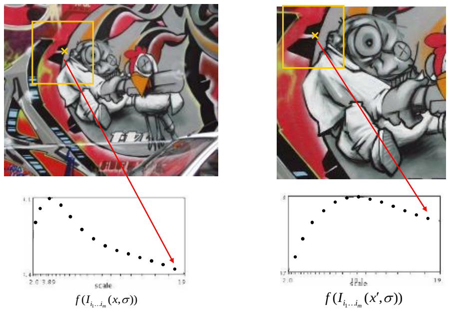
B. Leibe

Computer Vision Summer'19

21

Automatic Scale Selection

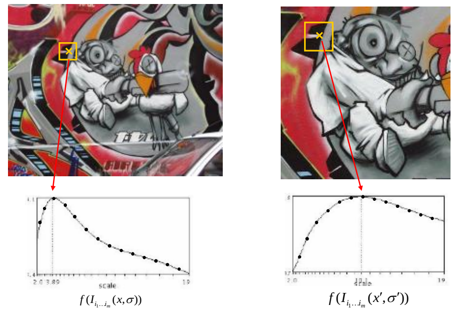
- Function responses for increasing scale (scale signature)



Computer Vision Summer'19

Automatic Scale Selection

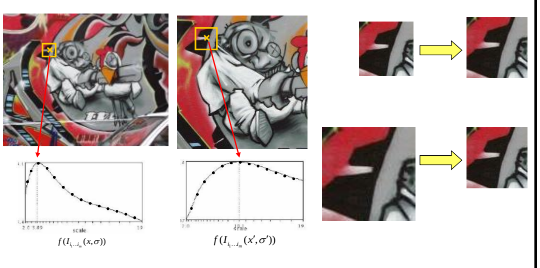
- Function responses for increasing scale (scale signature)



Computer Vision Summer'19

Automatic Scale Selection

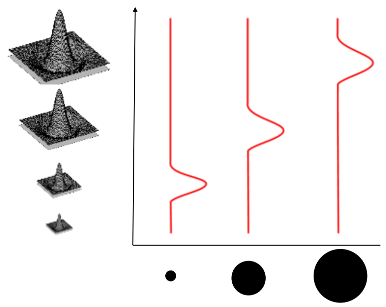
- Normalize: Rescale to fixed size



Computer Vision Summer'19

What Is A Useful Signature Function?

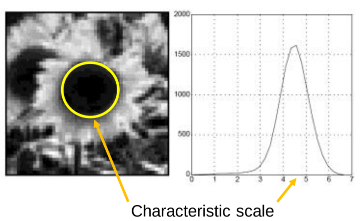
- Laplacian-of-Gaussian = "blob" detector



Computer Vision Summer'19

Characteristic Scale

- We define the *characteristic scale* as the scale that produces peak of Laplacian response

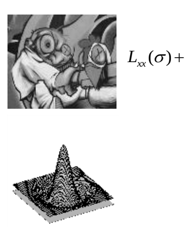


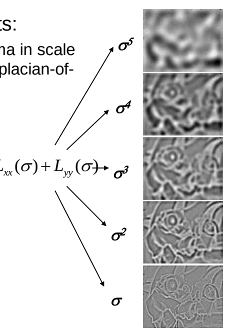
T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* 30 (2): pp 77--116.

Computer Vision Summer'19

Laplacian-of-Gaussian (LoG)

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

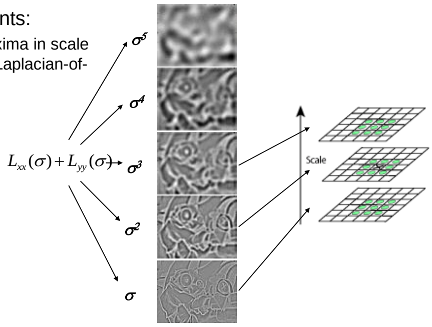




Computer Vision Summer'19

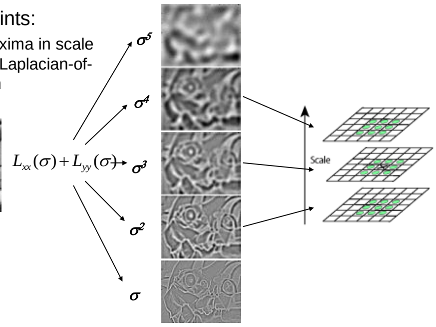
Laplacian-of-Gaussian (LoG)

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

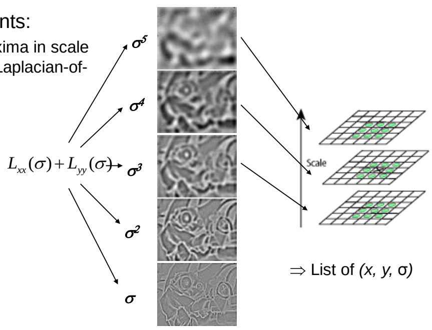
Laplacian-of-Gaussian (LoG)

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

Laplacian-of-Gaussian (LoG)

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

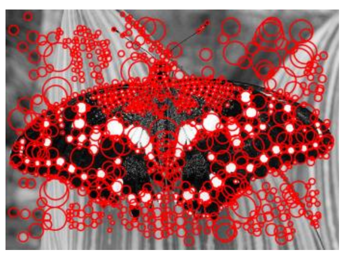
⇒ List of *(x, y, σ)*

LoG Detector: Workflow

LoG Detector: Workflow

sigma = 11.9912

LoG Detector: Workflow



Technical Detail

RWTH AACHEN UNIVERSITY

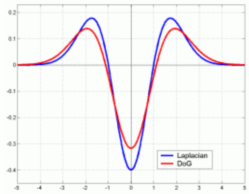
- We can efficiently approximate the Laplacian with a difference of Gaussians:

$$L = \sigma^2 \left(G_{xx}(x,y,\sigma) + G_{yy}(x,y,\sigma) \right)$$

(Laplacian)

$$DoG = G(x,y,k\sigma) - G(x,y,\sigma)$$

(Difference of Gaussians)

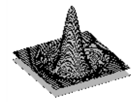


Computer Vision Summer'19

Difference-of-Gaussian (DoG)

RWTH AACHEN UNIVERSITY

- Difference of Gaussians as approximation of the LoG
  - This is used e.g. in Lowe's SIFT pipeline for feature detection.
- Advantages
  - No need to compute 2nd derivatives
  - Gaussians are computed anyway, e.g. in a Gaussian pyramid.

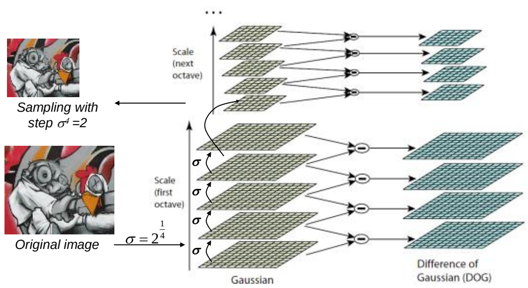


Computer Vision Summer'19

DoG – Efficient Computation

RWTH AACHEN UNIVERSITY

- Computation in Gaussian scale pyramid



Computer Vision Summer'19

Slide adapted from Krystian Mikolajczyk

Results: Lowe's DoG

RWTH AACHEN UNIVERSITY

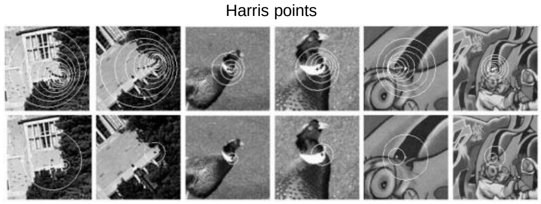
Computer Vision Summer'19

Harris-Laplace [Mikolajczyk '01]

RWTH AACHEN UNIVERSITY

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

Harris points



Harris-Laplace points

Computer Vision Summer'19

Slide adapted from Krystian Mikolajczyk

Summary: Scale Invariant Detection

RWTH AACHEN UNIVERSITY

- 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).
- Two strategies
  - Laplacian-of-Gaussian (LoG)
  - Difference-of-Gaussian (DoG) as a fast approximation
  - *These can be used either on their own, or in combinations with single-scale keypoint detectors (Harris, Hessian).*

Computer Vision Summer'19

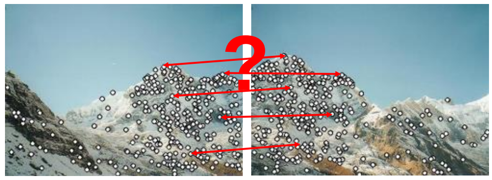
Topics of This Lecture

- Recap: Local Feature Extraction
- Local Descriptors
  - SIFT
  - Applications
- Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation
- Dealing with Outliers
  - RANSAC
  - Generalized Hough Transform

Local Descriptors

- We know how to detect points
- Next question:

How to *describe* them for matching?



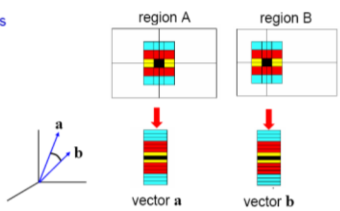
Point descriptor should be:
1. Invariant
2. Distinctive

Local Descriptors

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

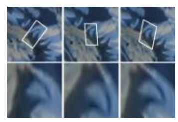
Write regions as vectors

$A \rightarrow \mathbf{a},\ B \rightarrow \mathbf{b}$

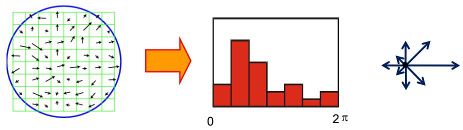


Feature Descriptors

- Disadvantage of patches as descriptors:
  - Small shifts can affect matching score a lot

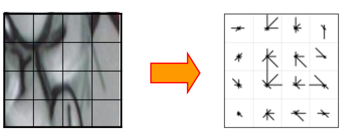


- Solution: histograms



Feature Descriptors: SIFT

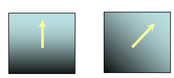
- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions



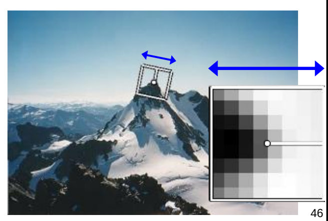
David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Rotation Invariant Descriptors

- Find local orientation
  - Dominant direction of gradient for the image patch



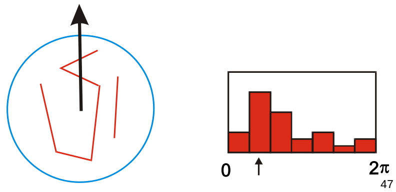
- Rotate patch according to this angle
  - This puts the patches into a canonical orientation.



Orientation Normalization: Computation

[Lowe, 1999]

- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

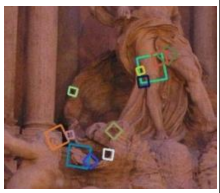


Summary: SIFT

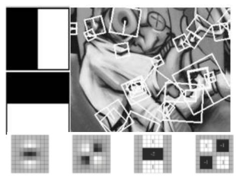
- Extraordinarily robust matching technique
  - Can handle changes in viewpoint up to ~60 deg. out-of-plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
    - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_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]



Local Descriptors: SURF

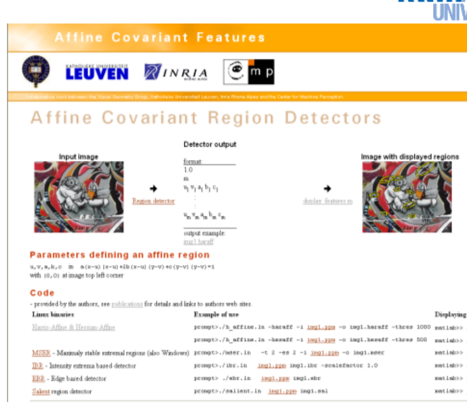


- Fast approximation of SIFT idea
  - Efficient computation by 2D box filters & integral images
    ⇒ 6 times faster than SIFT
  - Equivalent quality for object identification
  - http://www.vision.ee.ethz.ch/~surf
- GPU implementation available
  - Feature extraction @ 200Hz (detector + descriptor, 640×480 img)
  - http://homes.esat.kuleuven.be/~ncorneli/gpusurf/

[Bay, ECCV'06], [Cornelis, CVGPU'08]

You Can Try It At Home…

- For most local feature detectors, executables are available online:
- http://robots.ox.ac.uk/~vgg/research/affine
- http://www.cs.ubc.ca/~lowe/keypoints/
- http://www.vision.ee.ethz.ch/~surf
- http://homes.esat.kuleuven.be/~ncorneli/gpusurf/



http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries

Topics of This Lecture

- Recap: Local Feature Extraction
- Local Descriptors
  - SIFT
  - Applications
- Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation
- Dealing with Outliers
  - RANSAC
  - Generalized Hough Transform

Applications of Local Invariant Features

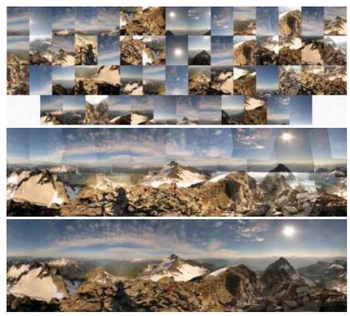
- Wide baseline stereo
- Motion tracking
- Panoramas
- Mobile robot navigation
- 3D reconstruction
- Recognition
  - Specific objects
  - Textures
  - Categories
- …

Slide credit: Kristen Grauman

Wide-Baseline Stereo

Image from T. Tuytelaars ECCV 2006 tutorial

Automatic Mosaicing



[Brown & Lowe, ICCV'03]

Panorama Stitching

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

iPhone version available

[Brown, Szeliski, and Winder, 2005]

Recognition of Specific Objects, Scenes

Schmid and Mohr 1997

Sivic and Zisserman, 2003

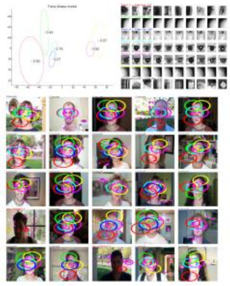
Rothganger et al. 2003

Lowe 2002

Slide credit: Kristen Grauman

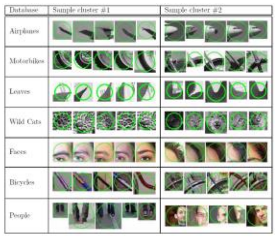
Recognition of Categories

Constellation model



Weber et al. (2000)
Fergus et al. (2003)

Bags of words



Csurka et al. (2004)
Sivic et al. (2005)
Lazebnik et al. (2006), …

Value of Local Features

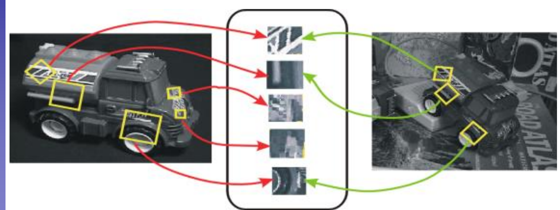
- Advantages
  - Critical to find distinctive and repeatable local regions for multi-view matching.
  - Complexity reduction via selection of distinctive points.
  - Describe images, objects, parts without requiring segmentation; robustness to clutter & occlusion.
  - Robustness: similar descriptors in spite of moderate view changes, noise, blur, etc.
- How can we use local features for such applications?
  - Next: matching and recognition

Topics of This Lecture

- Recap: Local Feature Extraction
- Local Descriptors
  - SIFT
  - Applications
- Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation
- Dealing with Outliers
  - RANSAC
  - Generalized Hough Transform

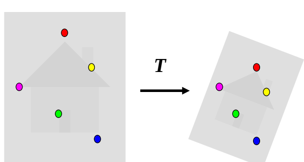
Recognition with Local Features

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

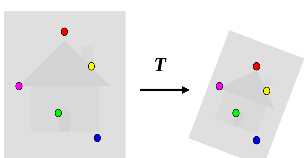


Local Features, e.g. SIFT

Warping vs. Alignment

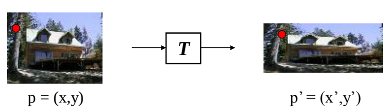


Warping: Given a source image and a transformation, what does the transformed output look like?



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

Parametric (Global) Warping



- 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:

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

What Can be Represented by a 2×2 Matrix?

- 2D Scaling?

$$x' = s_x * x \qquad y' = s_y * y$$

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

- 2D Rotation around (0,0)?

$$x' = \cos\theta * x - \sin\theta * y \qquad y' = \sin\theta * x + \cos\theta * y$$

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

- 2D Shearing?

$$x' = x + sh_x * y \qquad y' = sh_y * x + y$$

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

Computer Vision Summer'19 · Slide credit: Alexei Efros · B. Leibe · 65

What Can be Represented by a 2×2 Matrix?

- 2D Mirror about y axis?

$$x' = -x \qquad y' = y$$

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

- 2D Mirror over (0,0)?

$$x' = -x \qquad y' = -y$$

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

- 2D Translation?

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

NO!

Computer Vision Summer'19 · Slide credit: Alexei Efros · B. Leibe · 66

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 2x2 matrix.
- Linear transformations are combinations of …
  - Scale,
  - Rotation,
  - Shear, and
  - Mirror

Computer Vision Summer'19 · Slide credit: Alexei Efros · B. Leibe · 67

Homogeneous Coordinates

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

$$\mathbf{x}' = \mathbf{x} + \mathbf{t}_x \qquad \mathbf{y}' = \mathbf{y} + \mathbf{t}_y$$

- A: Using the rightmost column:

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

Computer Vision Summer'19 · Slide credit: Alexei Efros · B. Leibe · 68

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

Computer Vision Summer'19 · Slide credit: Alexei Efros · B. Leibe · 69

2D Affine Transformations

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

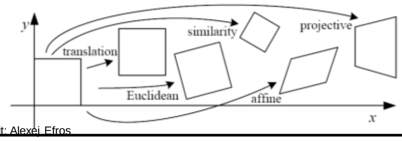
- Affine transformations are combinations of …
  - Linear transformations, and
  - Translations
- Parallel lines remain parallel

Computer Vision Summer'19 · Slide credit: Alexei Efros · B. Leibe · 70

Projective Transformations

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

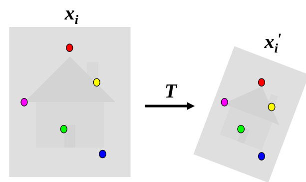
- Projective transformations:
  - Affine transformations, and
  - Projective warps
- Parallel lines do not necessarily remain parallel



Slide credit: Alexei 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").



Slide credit: Kristen Grauman B. Leibe

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

Slide credit: Svetlana Lazebnik B. Leibe

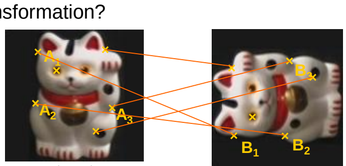
Fitting an Affine Transformation

- Affine model approximates perspective projection of planar objects

Slide credit: Kristen Grauman B. Leibe Image source: David Lowe

Fitting an Affine Transformation

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



(x_i, y_i) (x_i', y_i')

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

B. Leibe

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?

$$X_1 a + b = X_1' \qquad X_2 a + b = X_2' \qquad \begin{bmatrix} X_1 & 1 \\ X_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} X_1' \\ X_2' \end{bmatrix} \qquad Ax = B$$

- 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

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

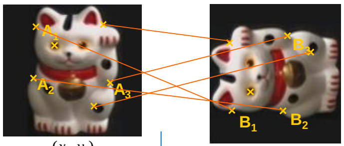
$\Rightarrow$ Least-squares minimization

Matlab: $x = A \backslash B$

Slide credit: Alexei Efros B. Leibe

Fitting an Affine Transformation

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



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

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

Fitting an Affine Transformation

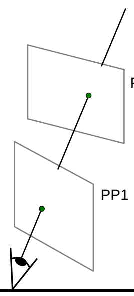
$$\begin{bmatrix} & & \cdots & & & \\ x_i & y_i & 0 & 0 & 1 & 0 \\ 0 & 0 & x_i & y_i & 0 & 1 \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} \cdots \\ x'_i \\ y'_i \\ \cdots \end{bmatrix}$$

- 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})?

Slide credit: Kristen Grauman

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



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

Slide adapted from Alexei 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

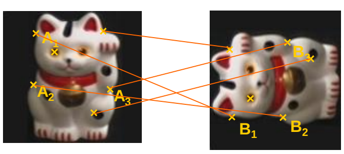
$$\underset{p'}{\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix}} = \underset{H}{\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}} \underset{p}{\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}}$$

Set scale factor to 1
⇒ 8 parameters left.

Slide adapted from Alexei Efros

Fitting a Homography

- Estimating the transformation



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

Homogenous coordinates

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

Image coordinates

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

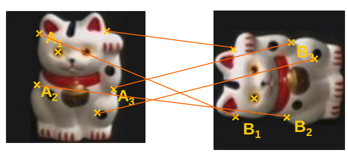
Matrix notation

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

Slide credit: Krystian Mikolajczyk

Fitting a Homography

- Estimating the transformation



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

Homogenous coordinates

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

Image coordinates

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

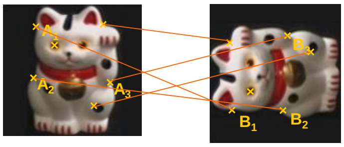
Matrix notation

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

Slide credit: Krystian Mikolajczyk

Fitting a Homography

- Estimating the transformation



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

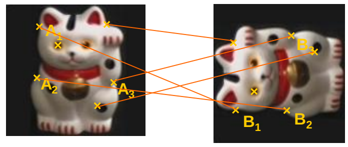
$$x' = Hx \qquad x'' = \tfrac{1}{z'} x'$$

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

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

Fitting a Homography

- Estimating the transformation



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

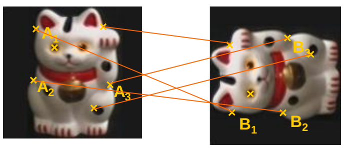
$$x' = Hx \qquad x'' = \tfrac{1}{z'} x'$$

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

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

Fitting a Homography

- Estimating the transformation



Homogenous coordinates

Image coordinates

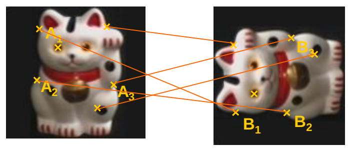
$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

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

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

Fitting a Homography

- Estimating the transformation



Homogenous coordinates

Image coordinates

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

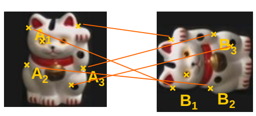
$$x_{A_1} = \frac{h_{11}\, x_{B_1} + h_{12} y_{B_1} + h_{13}}{h_{31}\, x_{B_1} + h_{32} y_{B_1} + 1} \qquad y_{A_1} = \frac{h_{21}\, x_{B_1} + h_{22} y_{B_1} + h_{23}}{h_{31}\, x_{B_1} + h_{32} y_{B_1} + 1}$$

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

$$h_{11}\, x_{B_1} + h_{12} y_{B_1} + h_{13} - x_{A_1} h_{31}\, x_{B_1} - x_{A_1} h_{32} y_{B_1} - x_{A_1} = 0$$
$$h_{21}\, x_{B_1} + h_{22} y_{B_1} + h_{23} - y_{A_1} h_{31}\, x_{B_1} - y_{A_1} h_{32} y_{B_1} - y_{A_1} = 0$$

Fitting a Homography

- Estimating the transformation



$$h_{11}\, x_{B_1} + h_{12} y_{B_1} + h_{13} - x_{A_1} h_{31}\, x_{B_1} - x_{A_1} h_{32} y_{B_1} - x_{A_1} = 0$$
$$h_{21}\, x_{B_1} + h_{22} y_{B_1} + h_{23} - y_{A_1} h_{31}\, x_{B_1} - y_{A_1} h_{32} y_{B_1} - y_{A_1} = 0$$

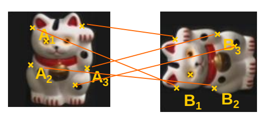
$\mathbf{x}_{A_1} \leftrightarrow \mathbf{x}_{B_1}$
$\mathbf{x}_{A_2} \leftrightarrow \mathbf{x}_{B_2}$
$\mathbf{x}_{A_3} \leftrightarrow \mathbf{x}_{B_3}$
$\vdots$

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

$$Ah = 0$$

Fitting a Homography

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



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

$$Ah = 0$$

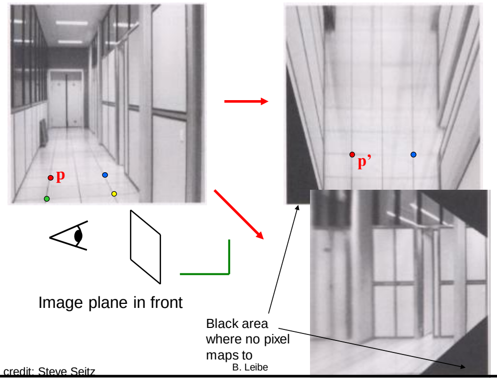
SVD

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

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

Minimizes least square error

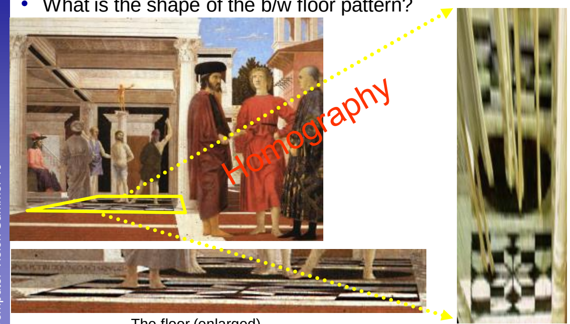
Image Warping with Homographies



Slide credit: Steve Seitz

Uses: Analyzing Patterns and Shapes

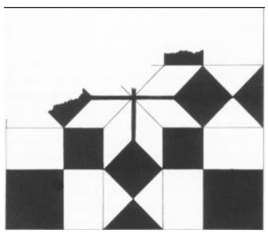
- What is the shape of the b/w floor pattern?



The floor (enlarged)

Slide credit: Antonio Criminisi

Analyzing Patterns and Shapes



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

Slide credit: Antonio Criminisi

Summary: Recognition by Alignment

- Basic matching algorithm
  1. Detect interest points in two images.
  2. Extract patches and compute a descriptor for each one.
  3. Compare one feature from image 1 to every feature in image 2 and select the nearest-neighbor pair.
  4. Repeat the above for each feature from image 1.
  5. Use the list of best pairs to estimate the transformation between images.
- Transformation estimation
  - Affine
  - Homography

Time for a Demo…

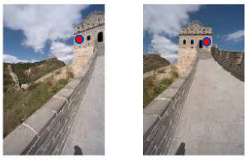
Automatic panorama stitching

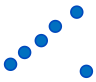
Topics of This Lecture

- Recap: Local Feature Extraction
- Local Descriptors
  - SIFT
  - Applications
- Recognition with Local Features
  - Matching local features
  - Finding consistent configurations
  - Alignment: linear transformations
  - Affine estimation
  - Homography estimation
- Dealing with Outliers
  - RANSAC
  - Generalized Hough Transform

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.

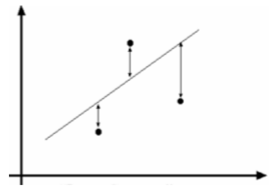




Slide credit: Kristen Grauman

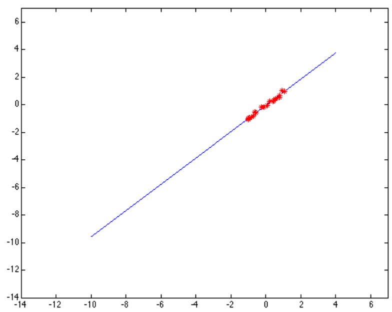
Example: Least-Squares Line Fitting

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



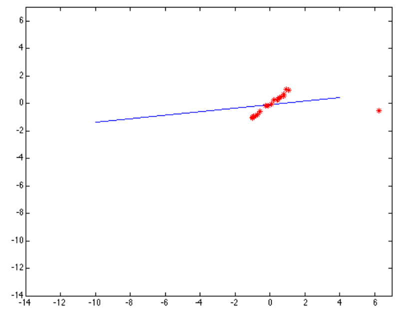
Source: Forsyth & Ponce

Outliers Affect Least-Squares Fit



Source: Forsyth & Ponce

Outliers Affect Least-Squares Fit



Source: Forsyth & Ponce

Strategy 1: RANSAC [Fischler81]

- RANdom SAmple Consensus
- 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.

Slide credit: Kristen Grauman

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

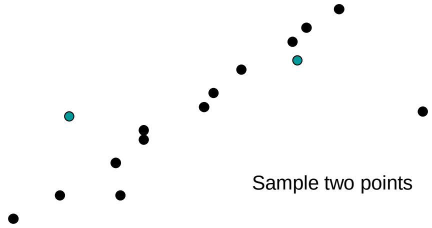
Slide credit: Kristen Grauman

RANSAC Line Fitting Example

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

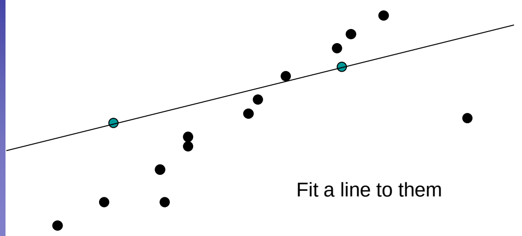
RANSAC Line Fitting Example

- Task: Estimate the best line



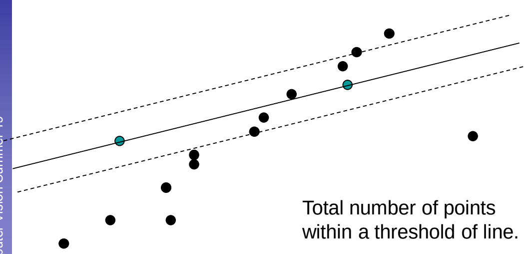
RANSAC Line Fitting Example

- Task: Estimate the best line



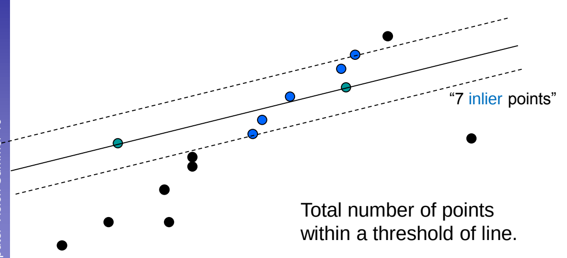
RANSAC Line Fitting Example

- Task: Estimate the best line



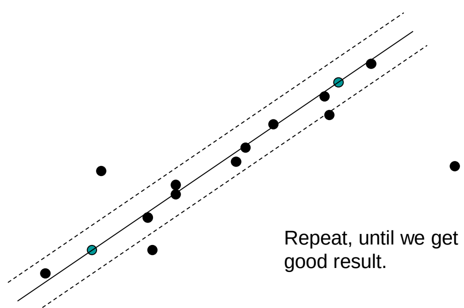
RANSAC Line Fitting Example

- Task: Estimate the best line



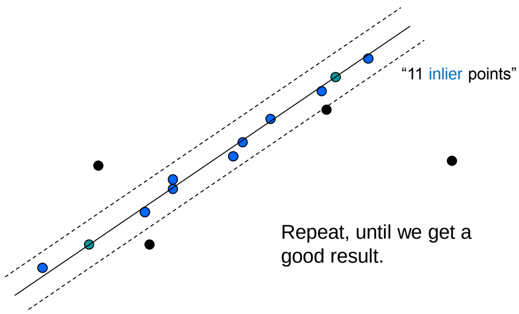
RANSAC Line Fitting Example

- Task: Estimate the best line



RANSAC Line Fitting Example

- Task: Estimate the best line



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

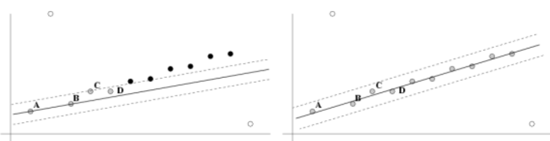
$\Rightarrow$ Choose k high enough to keep this below desired failure rate.

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

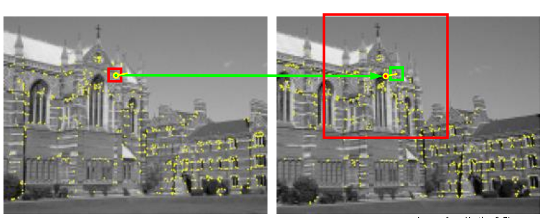
After RANSAC

- 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

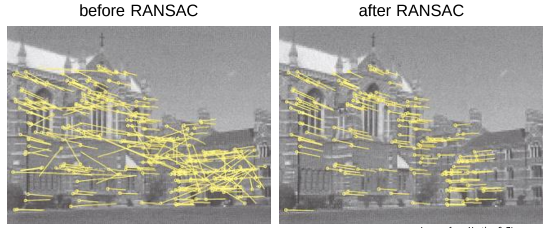


Images from Hartley & Zisserman

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

Computer Vision Summer'19

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

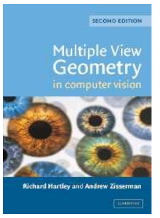
B. Leibe

113

Computer Vision Summer'19

References and Further Reading

- More details on homography estimation can be found in Chapter 4.7 of
  - R. Hartley, A. Zisserman
    Multiple View Geometry in Computer Vision
    2nd Ed., Cambridge Univ. Press, 2004



- Details about the DoG detector and the SIFT descriptor can be found in
  - D. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60(2), pp. 91-110, 2004
- Try the available local feature detectors and descriptors
  - http://www.robots.ox.ac.uk/~vgg/research/affine/detectors.html#binaries