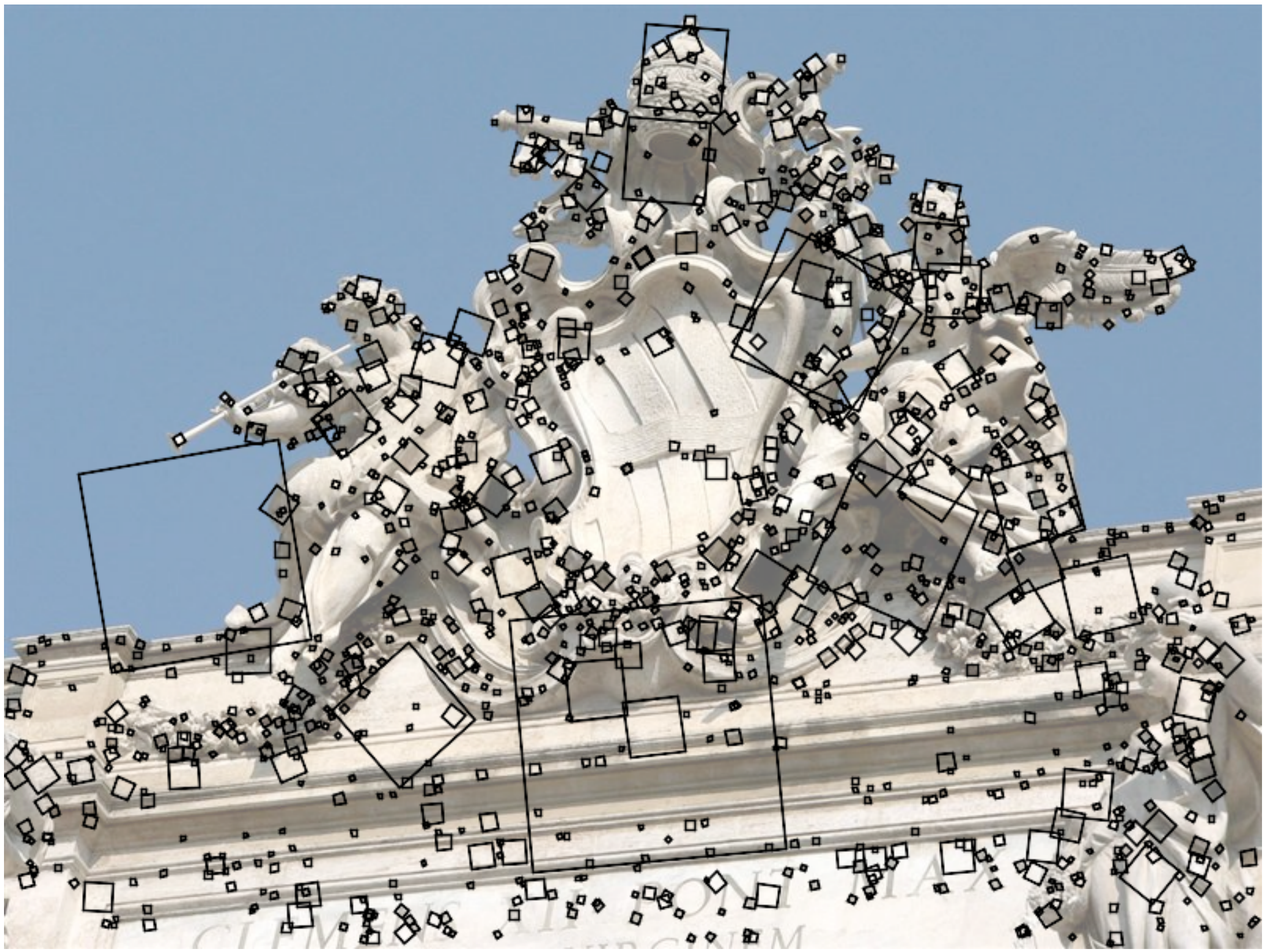
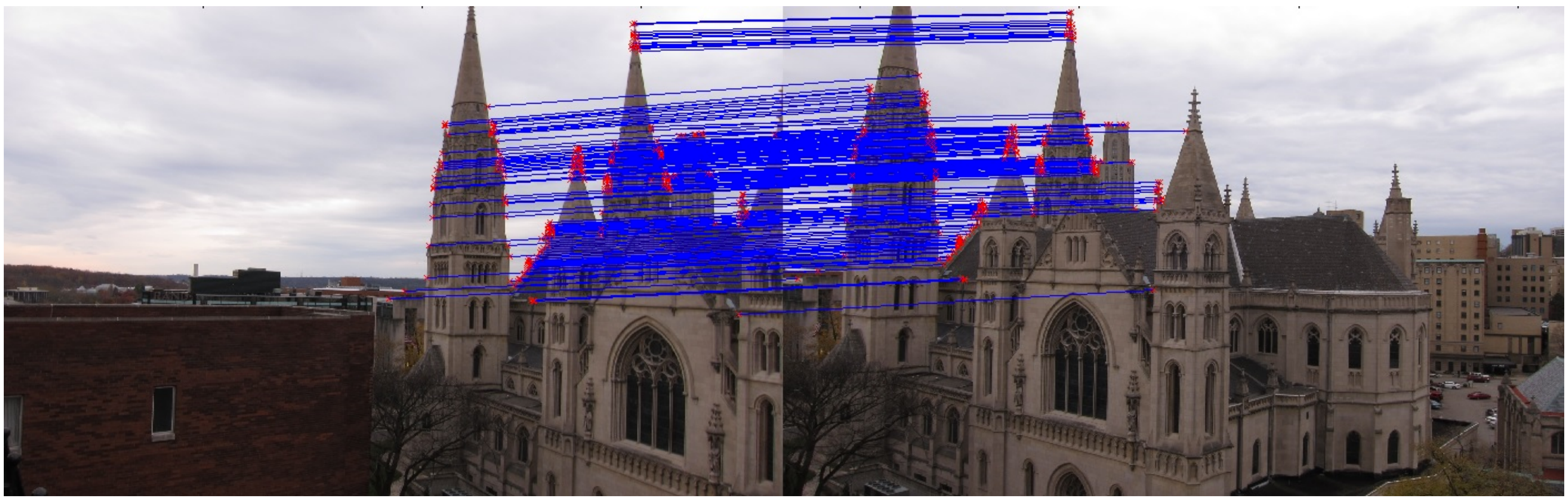
Descriptors

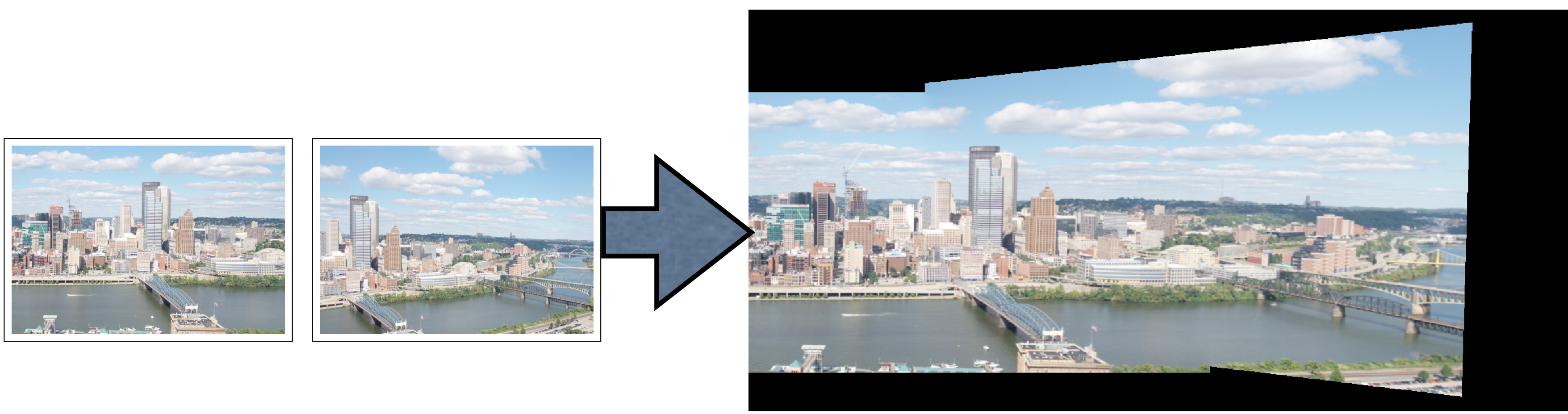


Outline

- Interest point detection (Harris Corners)
- **Descriptors (SIFT, BRIEF, Filter Banks)**
- RANSAC

Core visual understanding task: finding correspondences between images





Overall pipeline for finding correspondences in 2 images

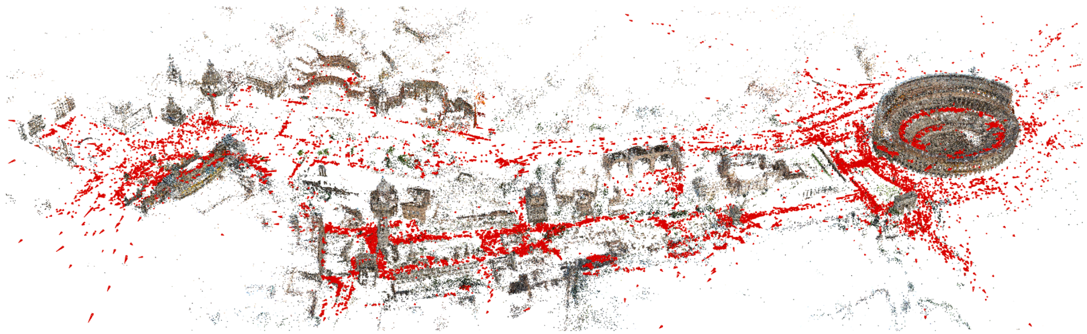
- Find "easy-to-match" patches in both images

Harris Corners

- Compute patch "similarities" across images
- For each patch in the left image, find the best match in the right image

Descriptors

Seminal references



Distinctive Image Features from Scale-Invariant Keypoints

IJCV 04

David G. Lowe
Computer Science Department
University of British Columbia
Vancouver, B.C., Canada
lowe@cs.ubc.ca

Structure-from-Motion Revisited

CVPR2016

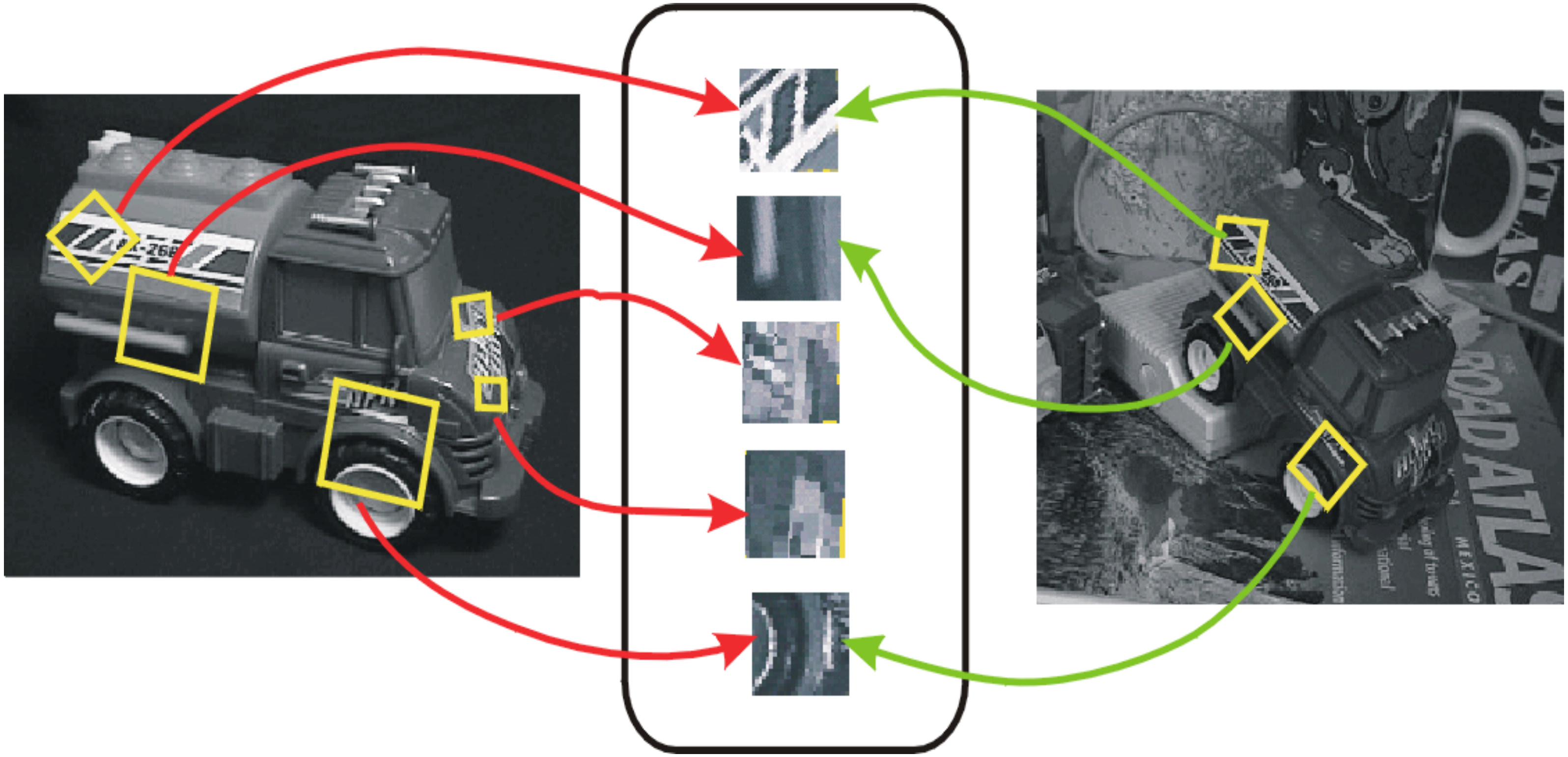
Johannes L. Schönberger[1,2*], Jan-Michael Frahm[1]
[1]University of North Carolina at Chapel Hill
[2]Eidgenössische Technische Hochschule Zürich
jsch@inf.ethz.ch, jmf@cs.unc.edu

Contemporary codebase: https://colmap.github.io/

Still used in state-of-the-art neural methods (e.g., NERF)

1) Find interest points on each image (easy-to-match patches)

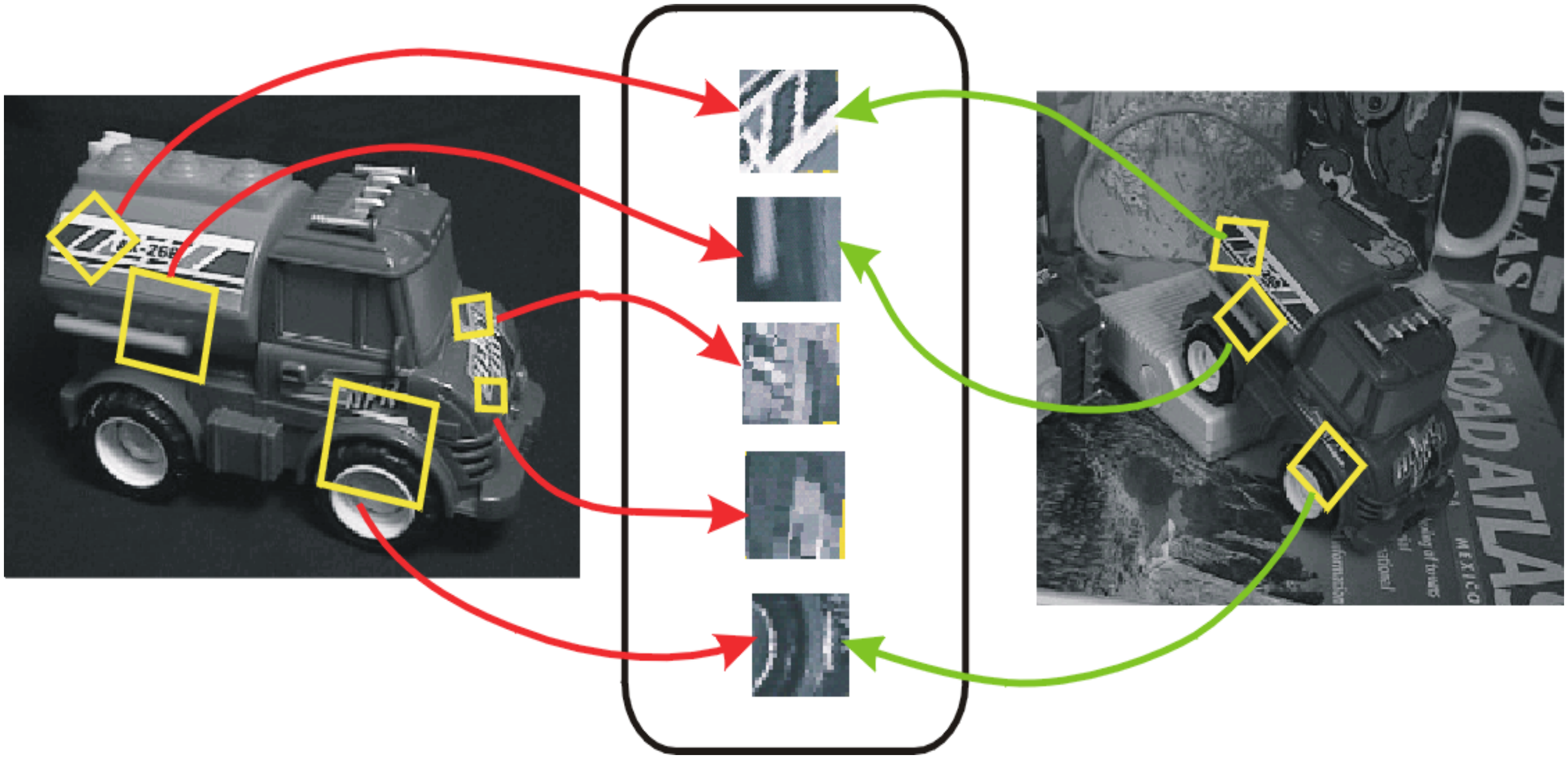
2) Transform each patch into “local coordinates”



3) Compare the transformed patches

How do we match the patches?

For each patch, we compute a “descriptor” p



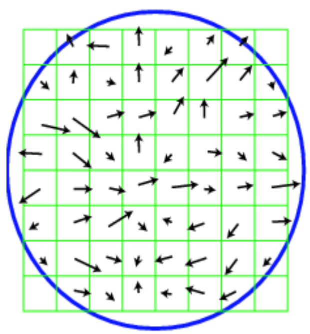
Compare 2 patches by computing a distance between the descriptors:

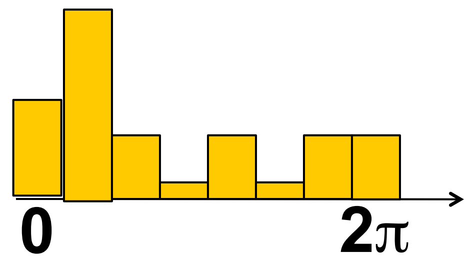
$$d(p_1, p_2) = \left\| \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} - \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} \right\|$$

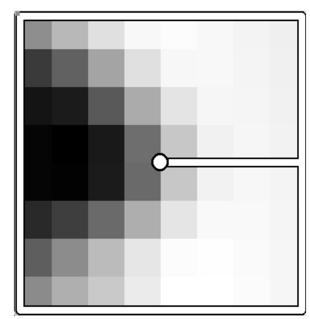
SIFT:

1. Compute gradients for all pixels in patch.
2. Histogram (bin) gradients by orientation.
3. Rotate patch so that dominant orientation is bin "0" **(why?)**

To make it rotation invariant!

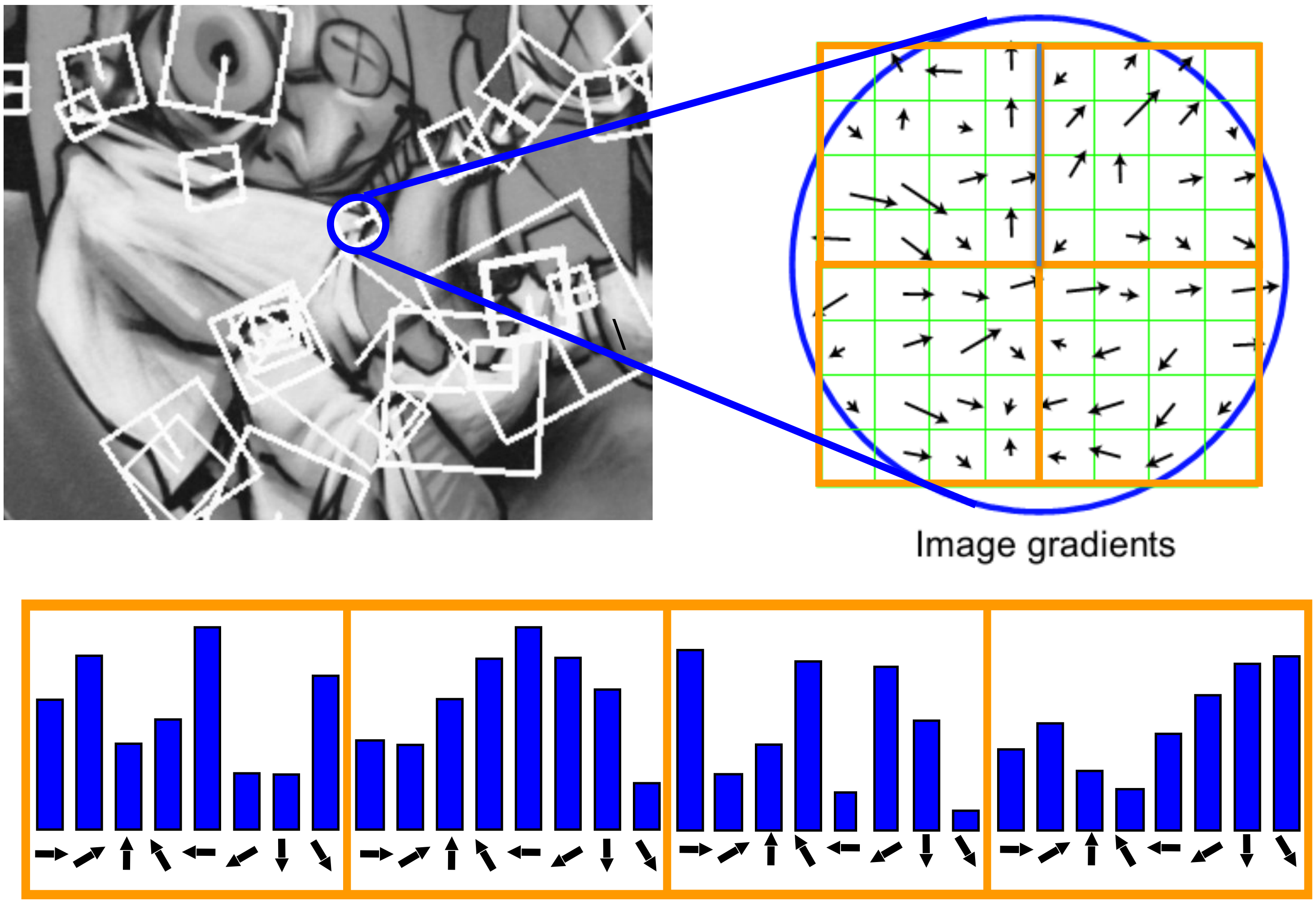






Computing the SIFT Descriptor

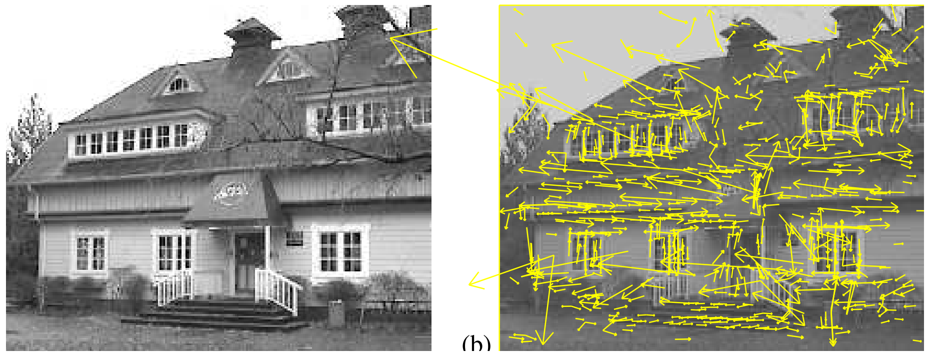
Histograms of gradient directions over spatial regions

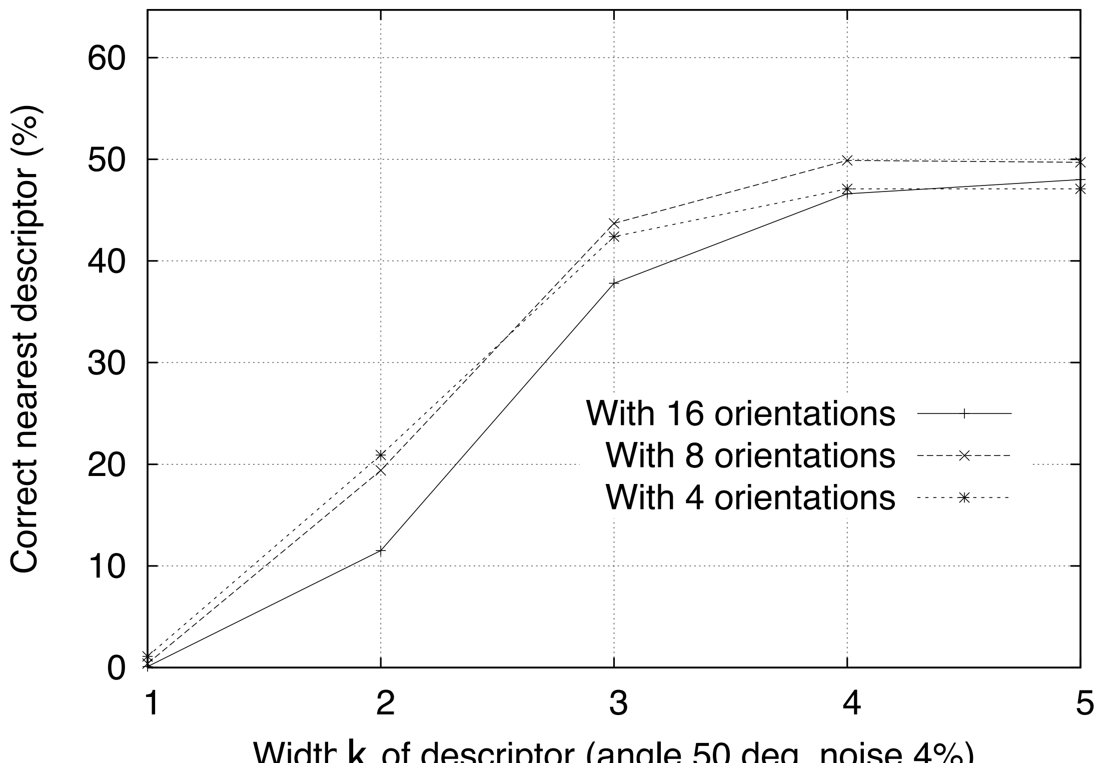


Total: 8 orientation bins * 4 quadrants = 32 dimensional descriptor

SIFT

What made this work? Exhaustive evaluation of hyper-parameters on annotated dataset





Alternate family of approaches: rank-based representations

28	50	70
5	10	80
3	1	30

Order:
6,3,2,9,1,5,4,7,8

125	154	176
87	98	189
92	85	140

Order:
6,3,2,9,1,5,$\boxed{7,4}$,8

Positive: rank ordering is invariant to monotonic transformations of intensity (not just linear!)

Negative: comparing two different ranks is expensive

Dinkar N. Bhat and Shree K. Nayar.
Ordinal Measures for Image Correspondence. PAMI Vol. 20, No. 4, April 1998

Alternate approach: binary patterns

Convert rank-order vector into a N^2 binary matrix of relative comparisons

Is pixel i > pixel j ?

28	50	70
5	10	80
3	1	30

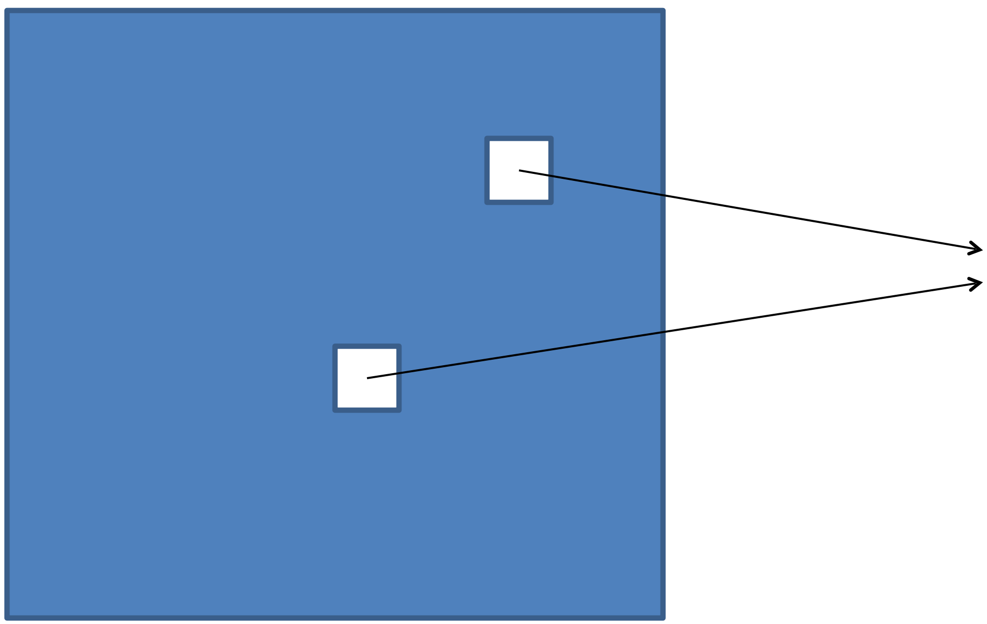
0	1	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0	0
0	1	1	1	0	0	1	0	1
0	1	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0	0
0	1	0	0	0	1	1	1	0
0	0	0	1	0	1	0	0	0
0	1	1	1	0	1	0	0	0
0	1	0	0	0	0	0	1	0

Compare two descriptors by # of matching bits
(Hamming distance with bitwise operations)

Ho, Tin (1995) "Random Decision Forests" Int'l Conf. Document Analysis and Recognition
Breiman, Leo (2001) "Random Forests" Machine Learning
Amit, Y. & Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation, 1997

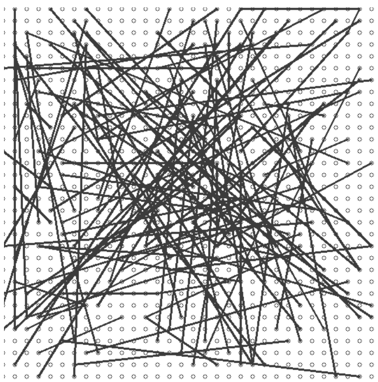
BRIEF:
Binary Robust Independent Elementary Features

- Randomly select pairs p and p’ for comparison
- Design choice: How to sample p,p’?

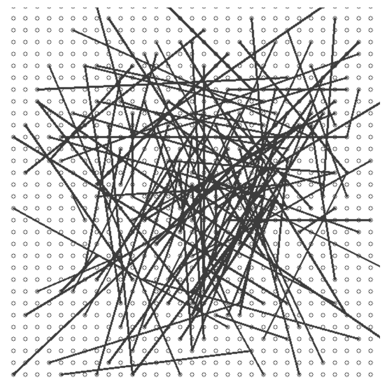


$$b = \begin{cases} 1 \; if \; I(p) > I(p') \\ 0 \; otherwise \end{cases}$$

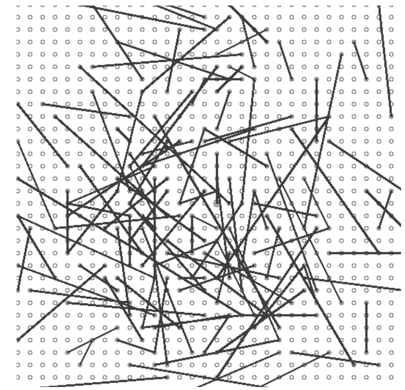
Sampling strategies



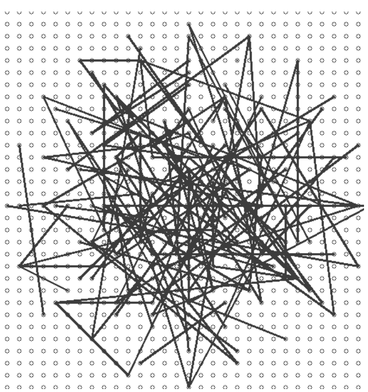
p1,p2 ~ uniform(x,y)



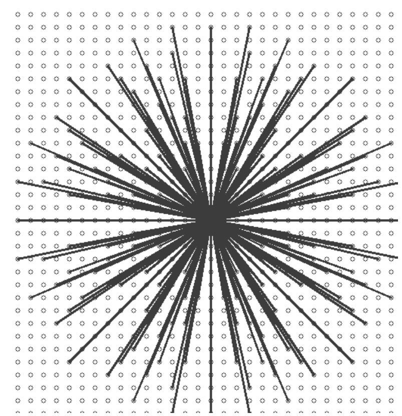
p1,p2 ~ N(0,sigma)



p1 ~ N(0,sigma)
p2 ~N(p1,$sigma_2$)



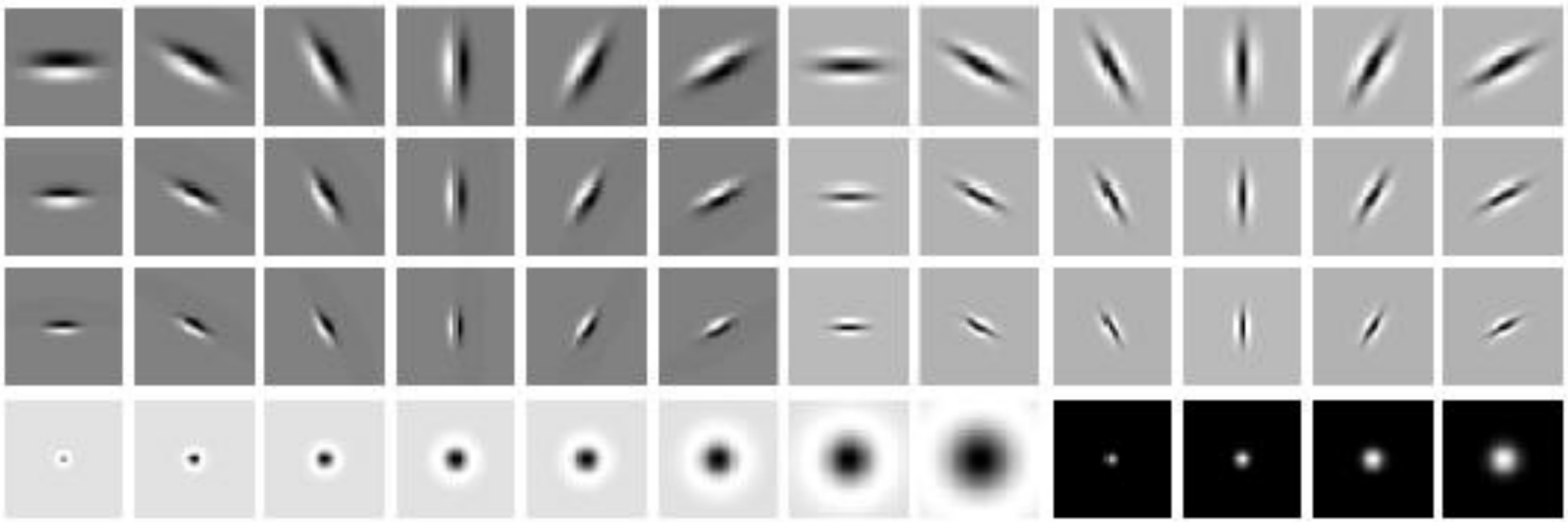
p1,p2 ~ uniform(r,theta)



p1 = 0
p2 = grid(r,theta)

BRIEF, BRISK, LBP, etc

Filter banks



The LeungMalik filter bank has a mix of edge, bar and spot filters at multiple scales and orientations. It has a total of 48 filters - 2 Gaussian derivative filters at 6 orientations and 3 scales, 8 Laplacian of Gaussian filters and 4 Gaussian filters.

General approach: define continuous filter and then discretize

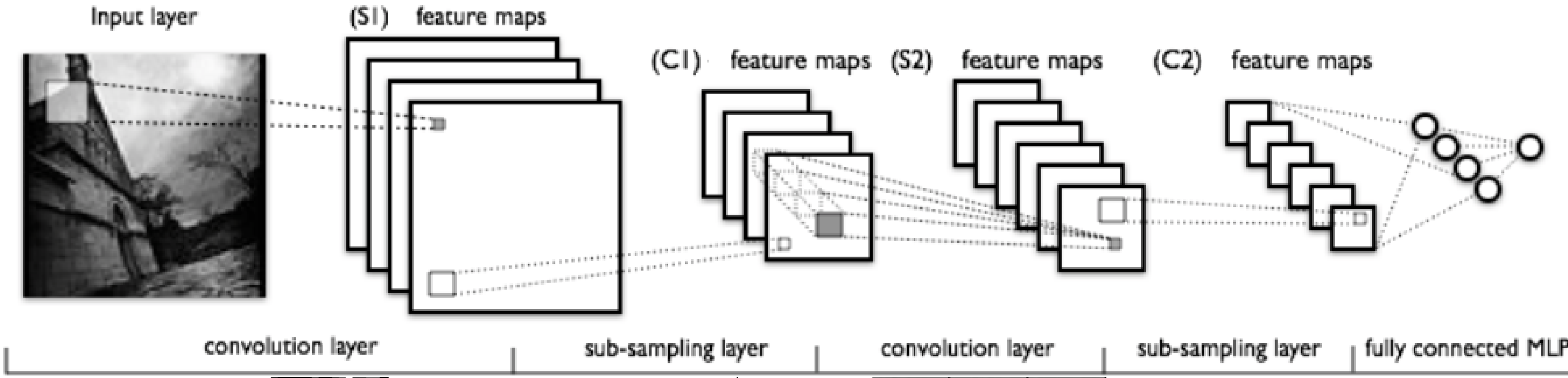
$$G[i,j] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[i+u,j+v]$$

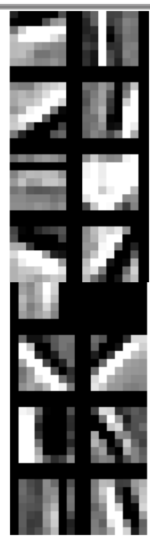
$$= H^T F_{ij} = ||H|| ||F_{ij}|| \cos\theta, \quad H, F_{ij} \in R^{(2K+1)^2}$$

Modern filter banks

Convolutional Neural Nets (CNNs) Lecun et al 98

Learn filters from training data to look for low, mid, and high-level features





5x5 low-level filters
(16 of them)

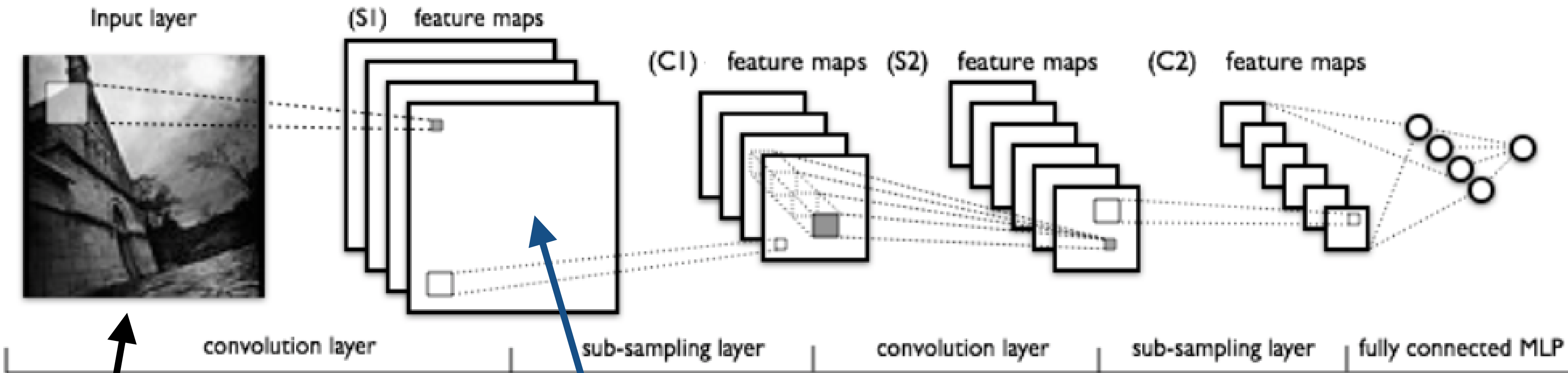
mid-level filters

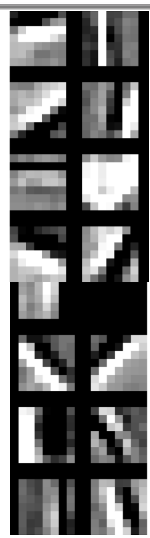
high-level filters

Modern filter banks

Convolutional Neural Nets (CNNs) Lecun et al 98

Learn filters from training data to look for low, mid, and high-level features





5x5 low-level filters
(16 of them)

Input is HxWx3

What are the dimensions here?

HxWx16

One "feature map" for each filter

For every pixel in the input, we have a 16-dimensional "filter response"

This is the "descriptor" of that pixel

- **BRIEF, (D-BRIEF, binboost, LBGM, …):**
  - Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua . BRIEF: Binary Robust Independent Elementary Features, Proc. ECCV 2010.
  - http://cvlabwww.epfl.ch/~lepetit/
- **BRISK:**
  - Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. Proc. ICCV 2011.
- **GIST:**
  - Aude Oliva, Antonio Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision, Vol. 42(3): 145-175, 2001.
  - http://people.csail.mit.edu/torralba/code/spatialenvelope/

- K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, T. Kadir and L. Van Gool, A Comparison of Affine Region Detectors; International Journal of Computer Vision (IJCV), Volume 65, Number 1, 2005.
- Major resource for all the features (Vision Lab Features Library (VLFeat)):

http://www.vlfeat.org/api/index.html

- **SIFT:**
  - David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. IJCV (International Journal of Computer Vision), 2004.
  - http://www.cs.ubc.ca/~lowe/keypoints/
- **MSER:**
  - P-E. Forssen, P-E. and D. Lowe, Shape Descriptors for Maximally Stable Extremal Regions International Conference on Computer Vision (ICCV), 2007.
- **SURF:**
  - Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, 2008.
  - http://www.vision.ee.ethz.ch/~surf/
  - **DAISY:** Engin Tola, Vincent Lepetit, Pascal Fua, DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2010.
- **LBP:**
  - Timo Ojala, Matti Pietikainen, and Topi Maenpa. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 24, No. 7, July 2002
  - http://www.cse.oulu.fi/CMV/Research/LBP

Outline

- Motivation
- Interest point detection
- **Descriptors (SIFT, BRIEF, Filter banks)**
- RANSAC

Outline

- Motivation
- Interest point detection
- Descriptors (SIFT, BRIEF, Filter banks)
- **RANSAC**

Overall pipeline for finding correspondences in 2 images

- Find “easy-to-match” patches in both images

Harris Corners

- Compute patch “similarities” across images

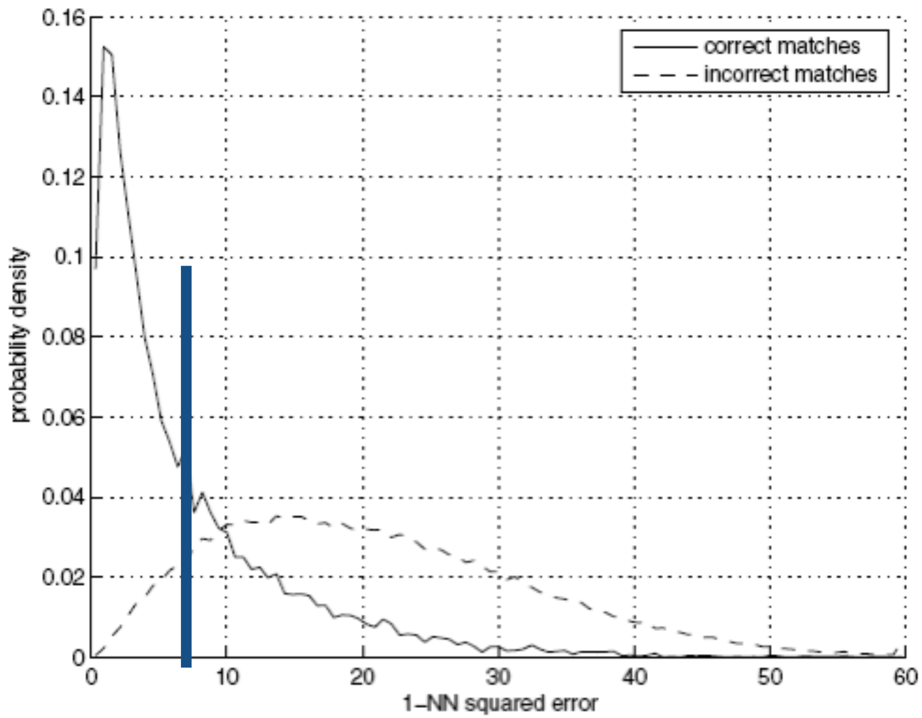
Descriptors

- For each patch in the left image, find the best match in the right image

- **Filter out bad matches**

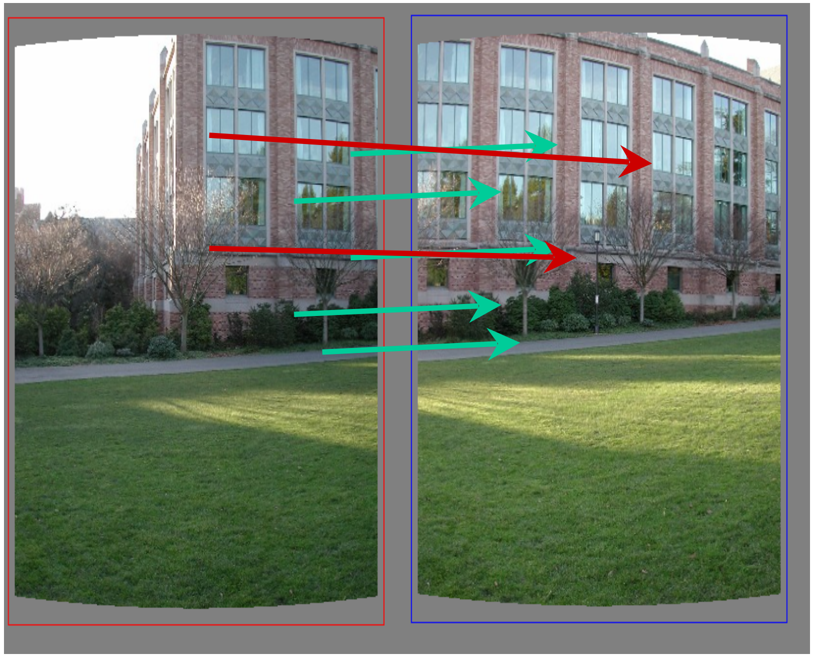
V0: Outlier rejection

Only keep if distance(SIFT(patch1), SIFT(patch2)) < threshold



Feature matching

- What can cause bad matches?



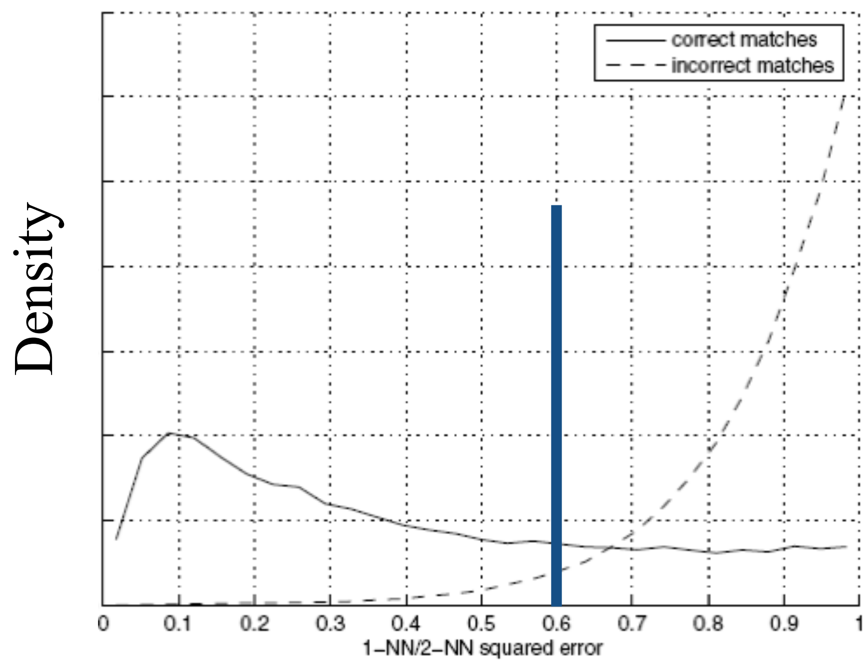
Example: Repeated textures (like windows) are challenging to match!

V1: reject unstable matches

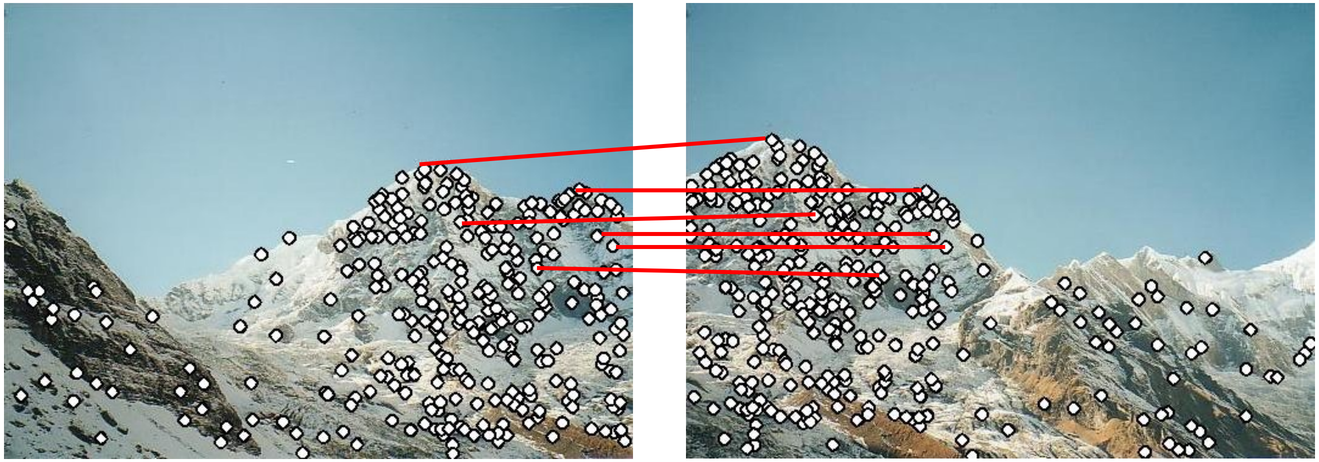
"If there's a good *alternative* (2nd best) match, don't trust it!"

[Lowe, 1999]:

- 1-NN: distance of the closest match
- 2-NN: distance of the second-closest match
- Look at how much better 1-NN is than 2-NN, e.g. 1-NN/2-NN



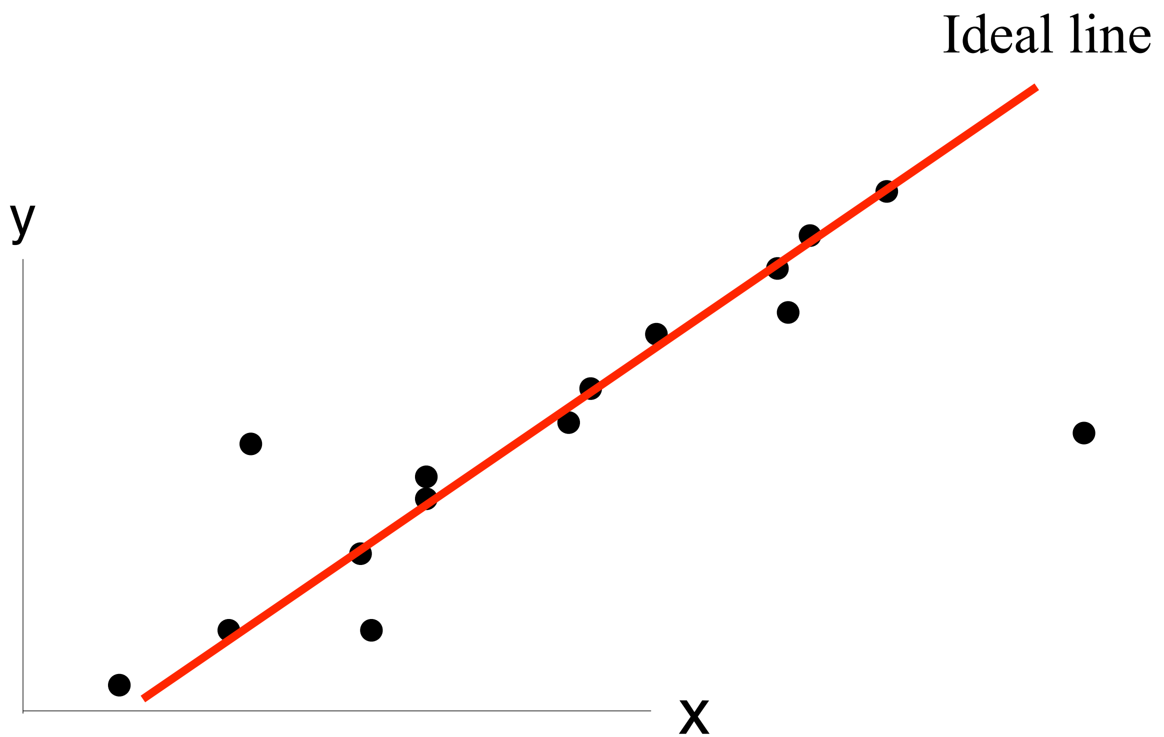
Much more effective: look for matches consistent with a *global motion model*



RANSAC

https://en.wikipedia.org/wiki/Random_sample_consensus

Example: How would you fit a line to a set of points with noise and outliers?



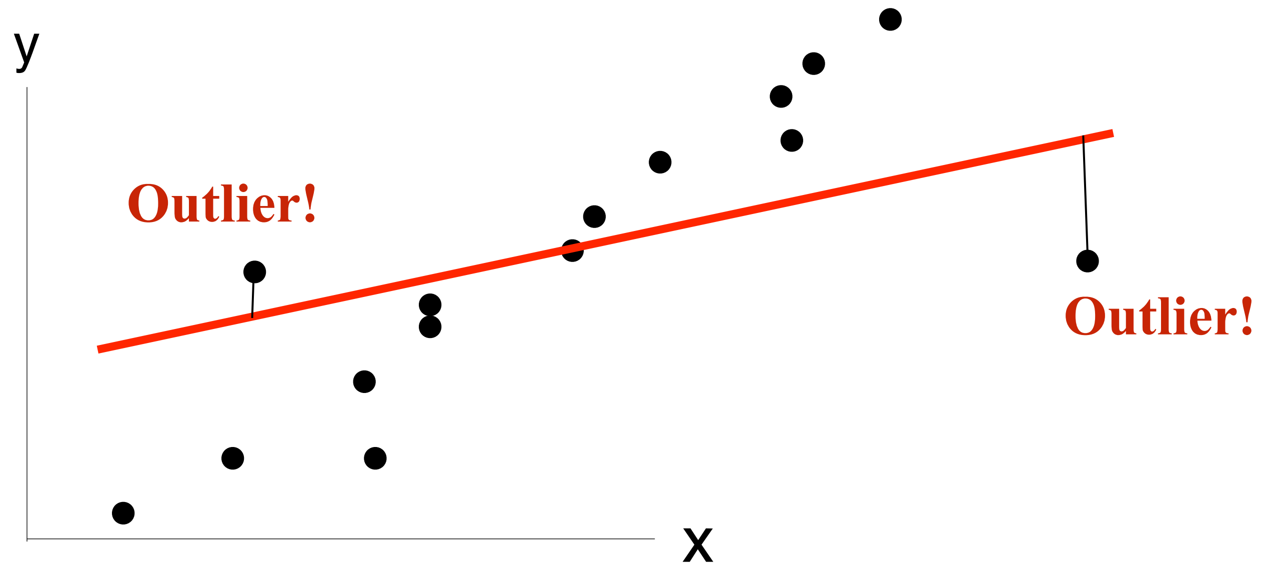
Version 1: Given data $\{x_i, y_i\}$ fit a line that minimizes the squared error

$$\min_{w,b} \sum_i (y_i - f_{w,b}(x_i))^2$$

Sum of Squared Errors

How can we fix this?

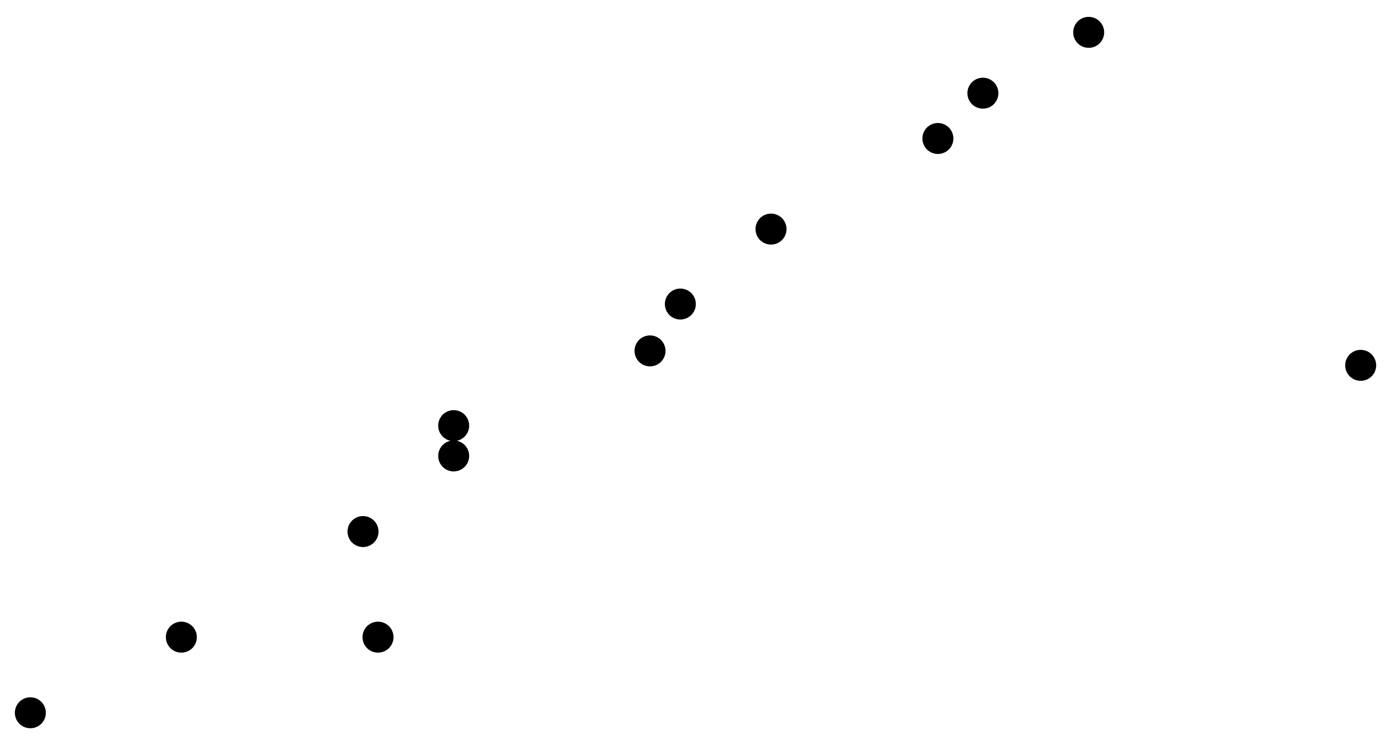
Crucial limitation of least squares: affected by outliers



RANSAC algorithm

Key idea:

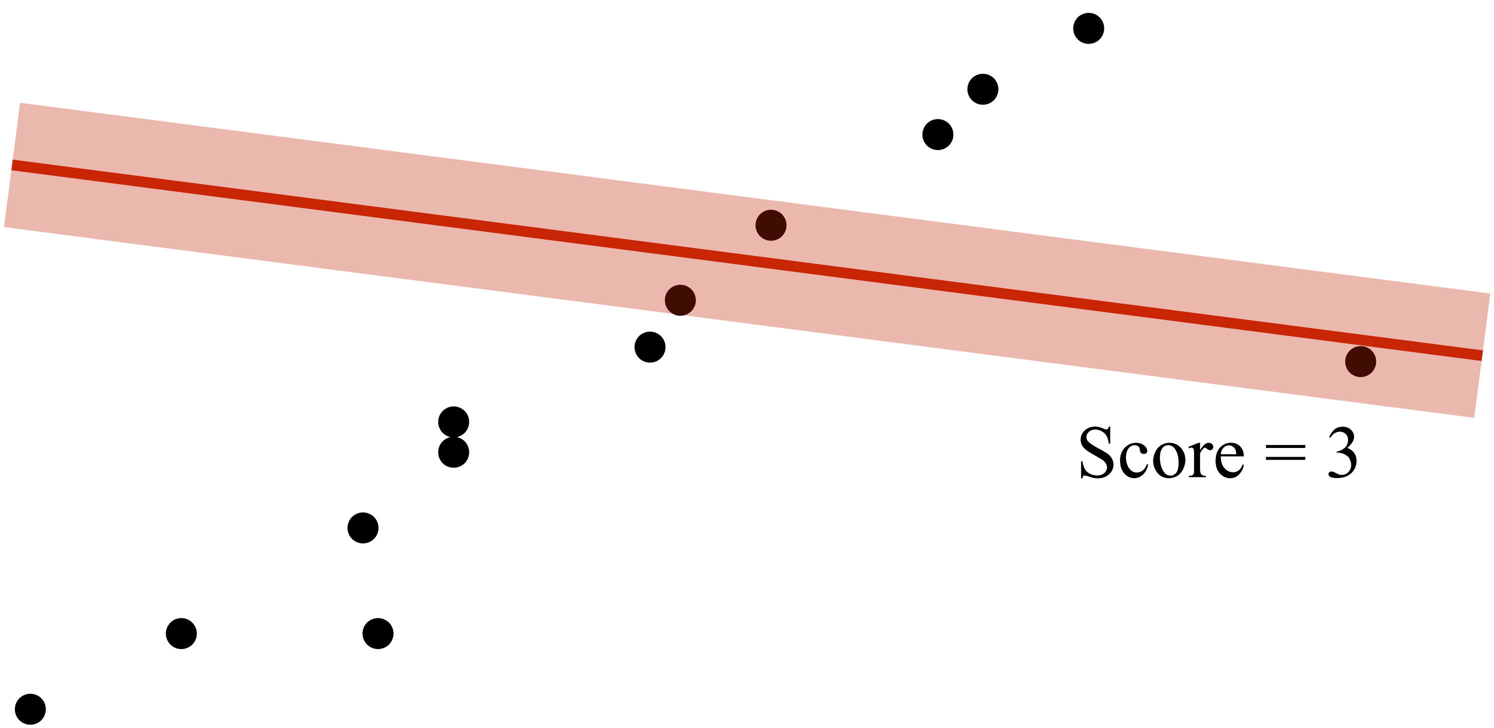
- Sample many lines
- For each line, compute a “score”: count the number of points that lie within a threshold of that line (“inliers”)
- Choose the line with the highest score



RANSAC algorithm

Key idea:

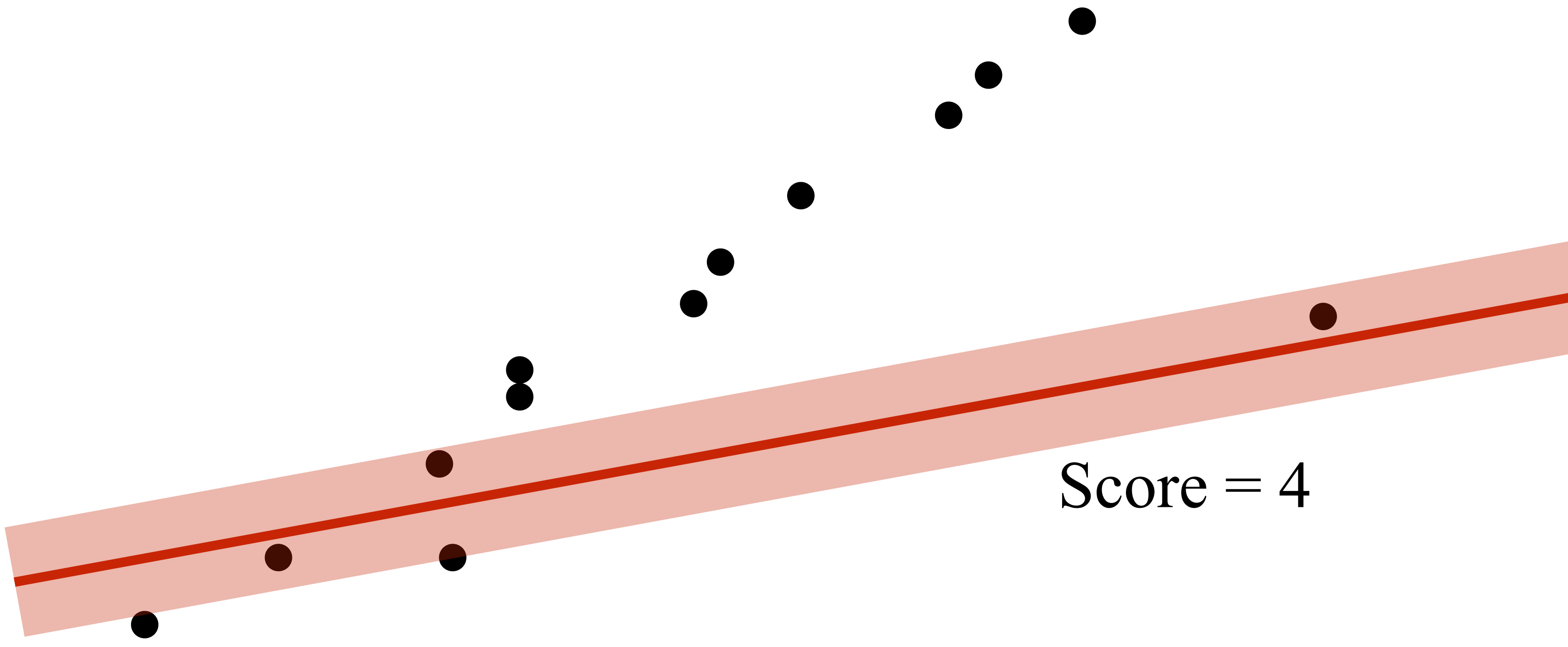
- Sample many lines
- For each line, compute a “score”: count the number of points that lie within a threshold of that line (“inliers”)
- Choose the line with the highest score



RANSAC algorithm

Key idea:

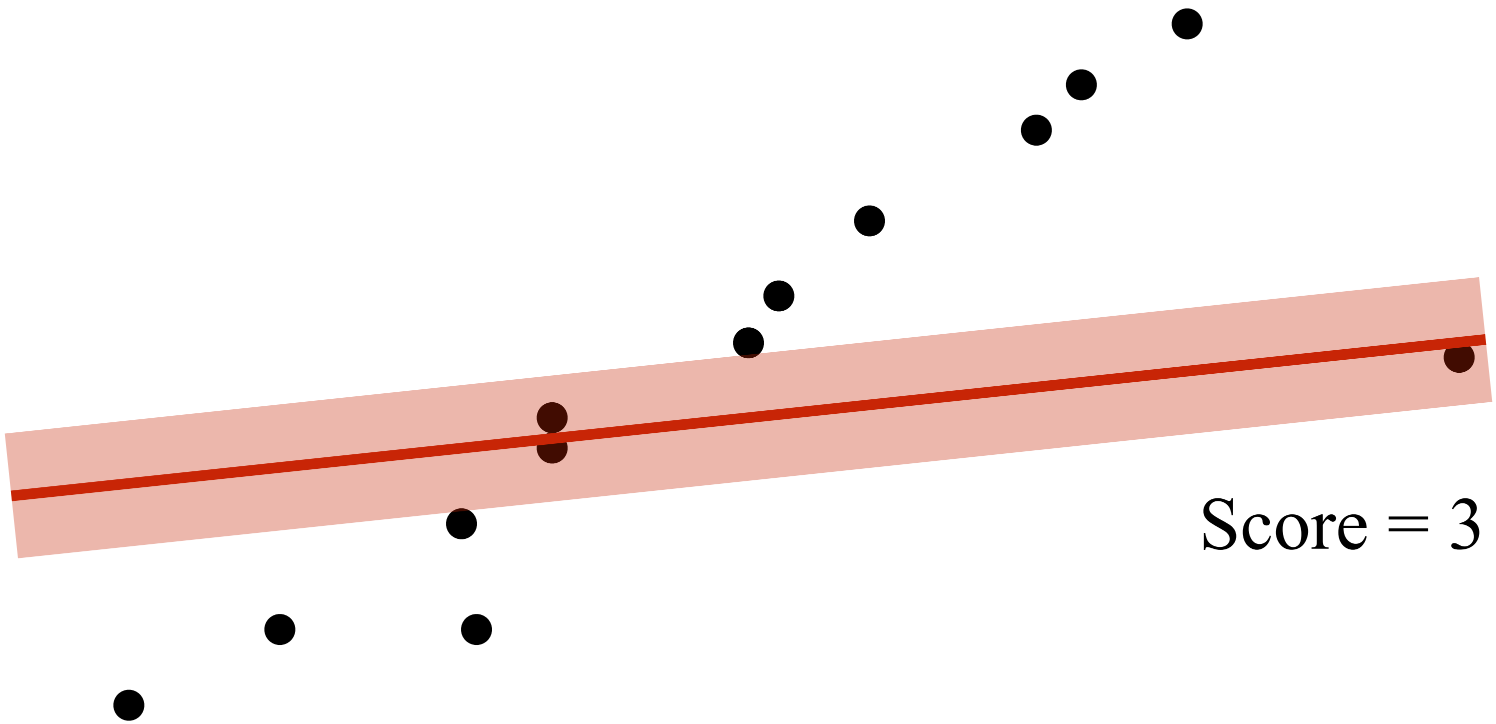
- Sample many lines
- For each line, compute a “score”: count the number of points that lie within a threshold of that line (“inliers”)
- Choose the line with the highest score



RANSAC algorithm

Key idea:

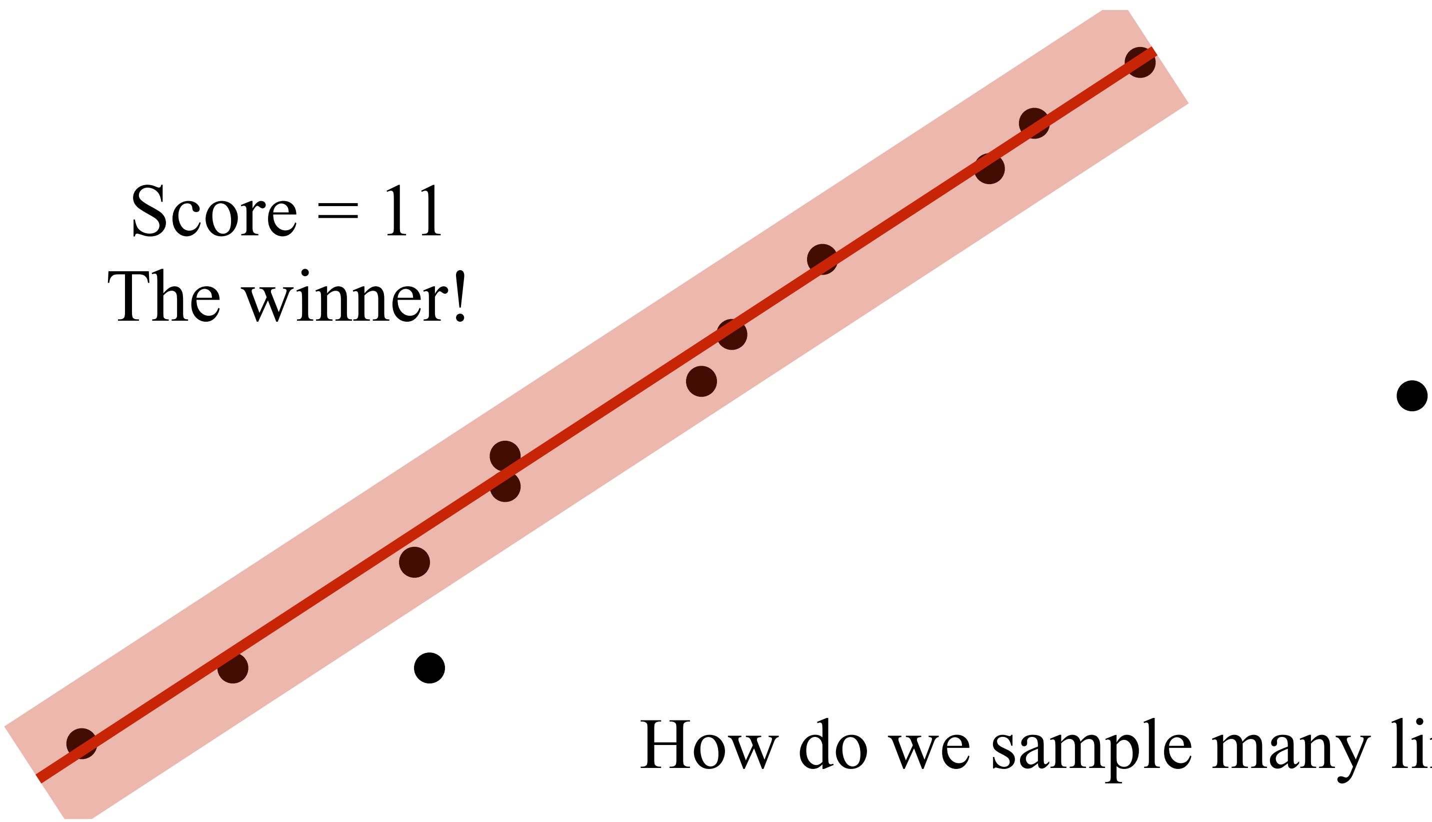
- Sample many lines
- For each line, compute a “score”: count the number of points that lie within a threshold of that line (“inliers”)
- Choose the line with the highest score



RANSAC algorithm

Key idea:

- Sample many lines
- For each line, compute a “score”: count the number of points that lie within a threshold of that line (“inliers”)
- Choose the line with the highest score

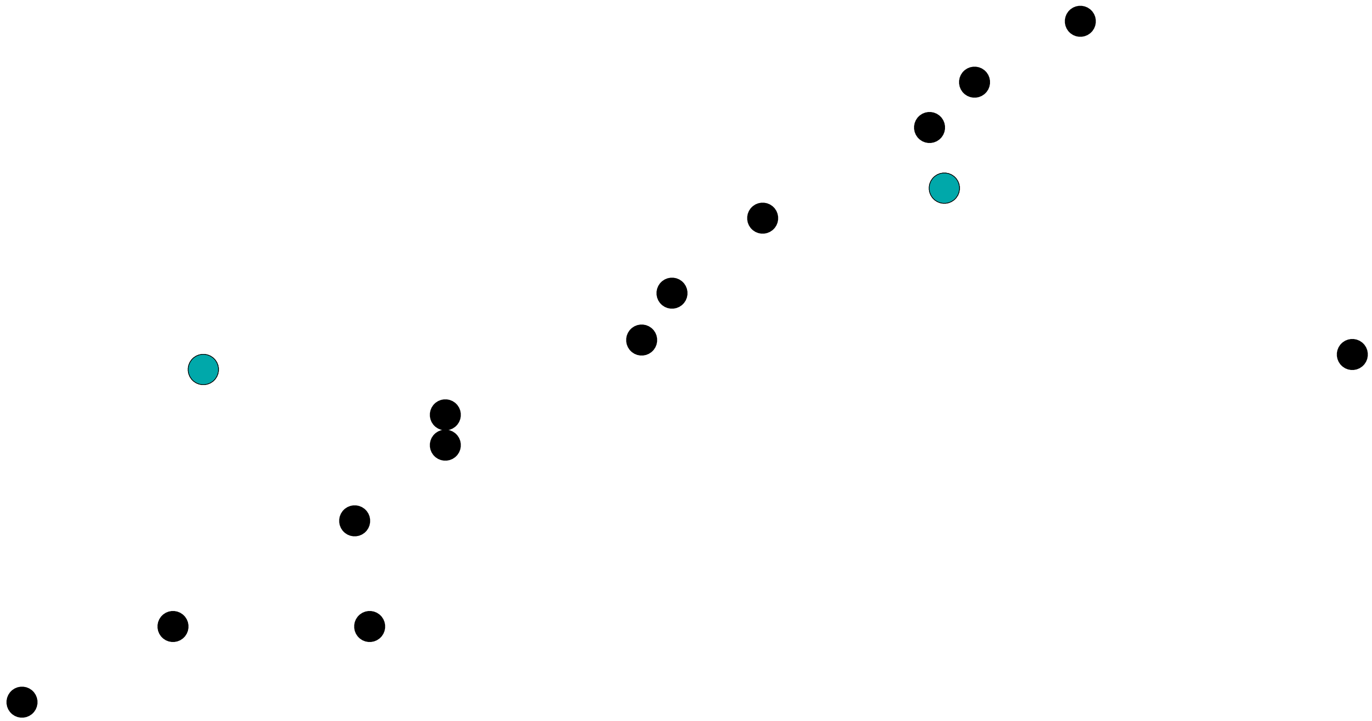


How do we sample many lines?

RANSAC Line Fitting Example

How do we sample many lines?

Key idea: 2 points define a line
1. Pick 2 random points

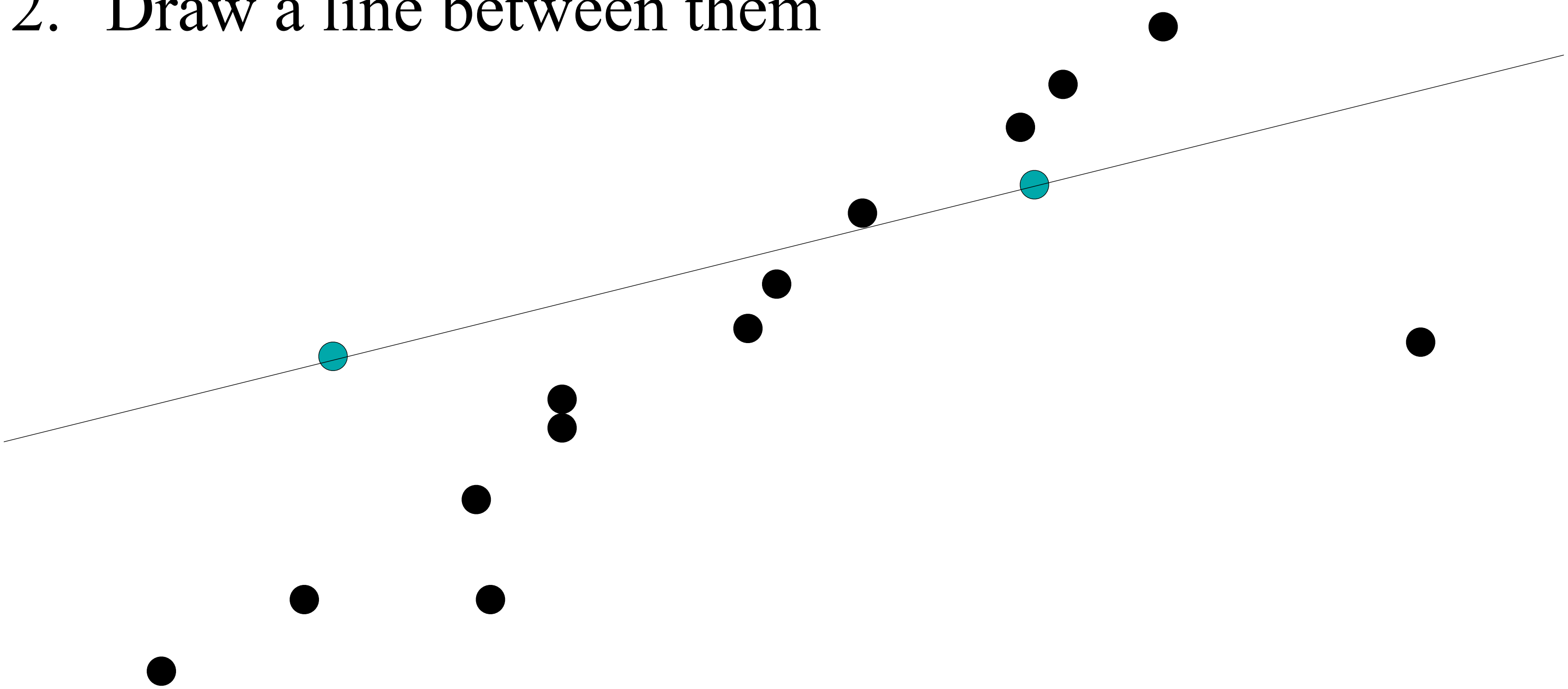


RANSAC Line Fitting Example

How do we sample many lines?

Key idea: 2 points define a line

1. Pick 2 random points
2. Draw a line between them

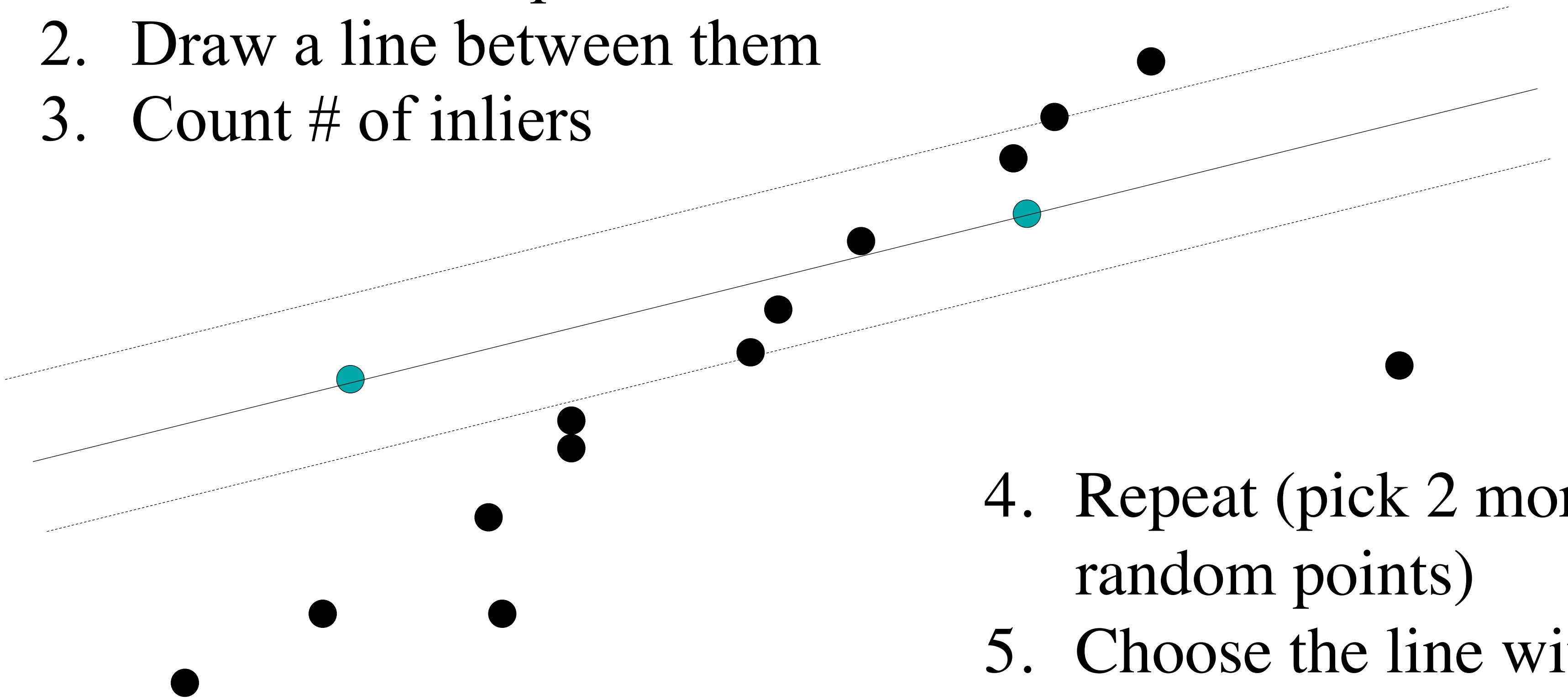


RANSAC Line Fitting Example

How do we sample many lines?

Key idea: 2 points define a line

1. Pick 2 random points
2. Draw a line between them
3. Count # of inliers



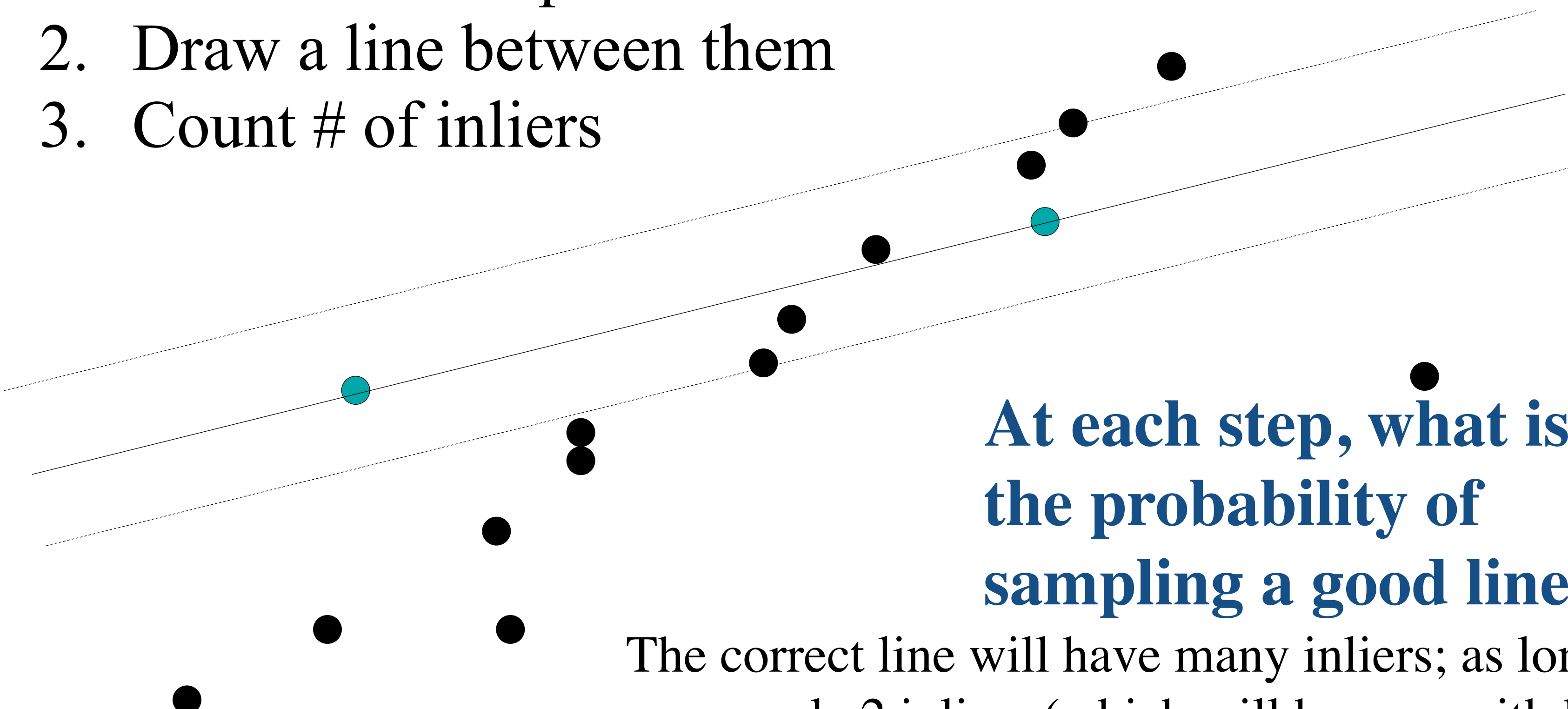
4. Repeat (pick 2 more random points)
5. Choose the line with the highest score

RANSAC Line Fitting Example

How do we sample many lines?

Key idea: 2 points define a line

1. Pick 2 random points
2. Draw a line between them
3. Count # of inliers



At each step, what is the probability of sampling a good line?

The correct line will have many inliers; as long as we sample 2 inliers (which will happen with high probability), we have found the correct line!

RANSAC for line fitting

Repeat ***k*** times:

- Draw 2 points uniformly at random
- Fit line to these points
- Count # inliers to this line (i.e., points whose distance from the line is less than a threshold ***t***)

Return line with largest inlier count

Optional post processing:

Take the inliers of the returned line and perform a least-squares fit (using just the inliers)

Source: M. Pollefeys

General RANSAC

Repeat ***k*** times:

- Sample **n** points uniformly at random
- Fit model to these points (line, plane, sphere, etc)
- Count inliers to this model (i.e., points whose distance from the model is less than a threshold ***t***)

Return model with largest inlier count

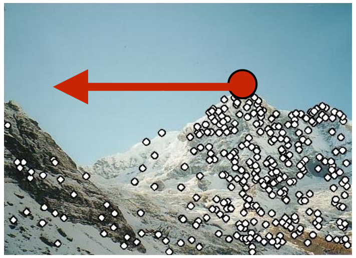
Optional post processing:

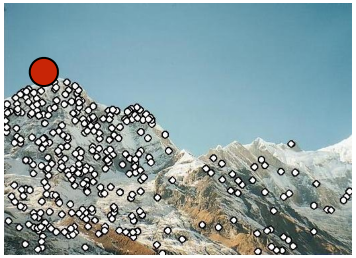
Take the inliers of the returned model and perform a least-squares fit (using just the inliers)

Source: M. Pollefeys

RANSAC for aligning images

Simplest version: Assume that the 2 images are defined by a translation





What is the minimum number of parameters that I need to define a translation of an image? 2

What is the minimum number of pairs of corresponding points that I need to define a translation? 1

Find lots of pairs of correspondences (NN descriptor distance)

Repeat ***k*** times:

- Sample **1** pair of corresponding points uniformly at random
- Fit translation to this pair points (subtract their coordinates = v)
- Count inliers to this model (i.e., every pair of points defines a vector v'; count the pairs where v' is within a threshold of v)

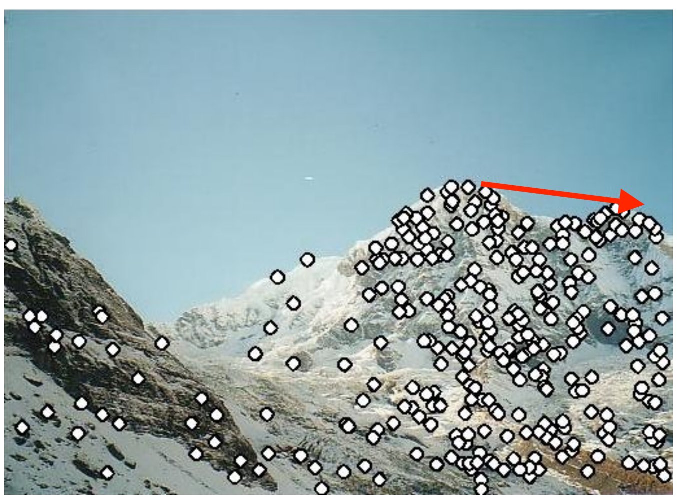
Return translation with largest inlier count

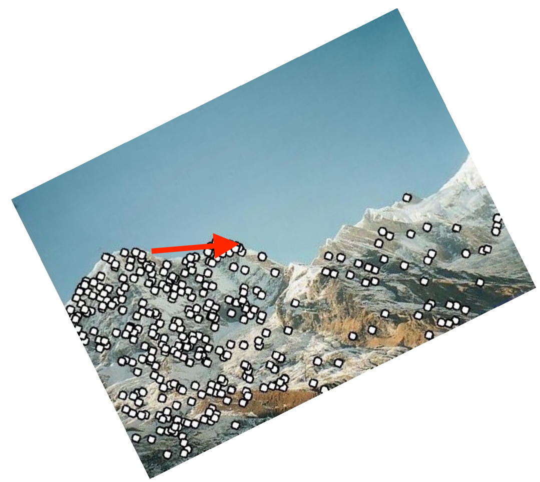
RANSAC for similarity alignment

What is the minimum number of pairs of corresponding points needed to fit a similarity (translation+rotation+scale) transform?

4 parameters (2 translation + 1 rotation + 1 scale)

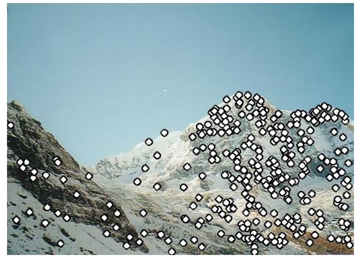
2 pairs of corresponding points

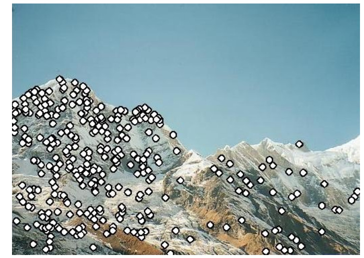




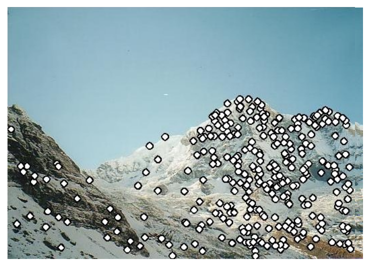
Models for aligning images

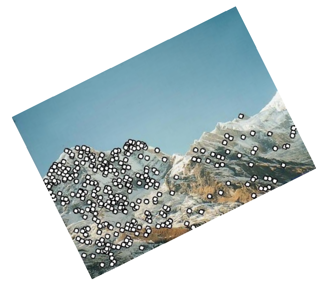
Translation

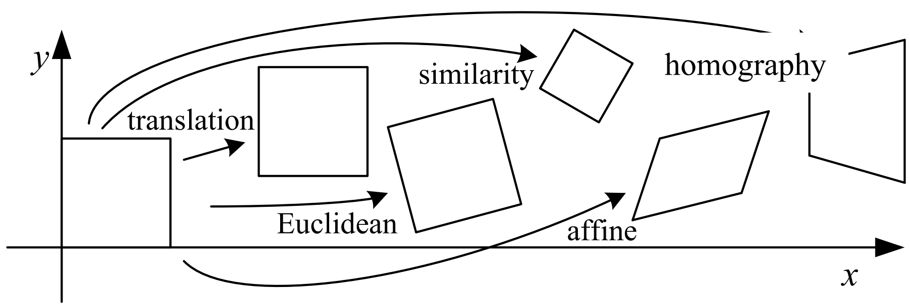




Similarity
(translation+rotation+scale)



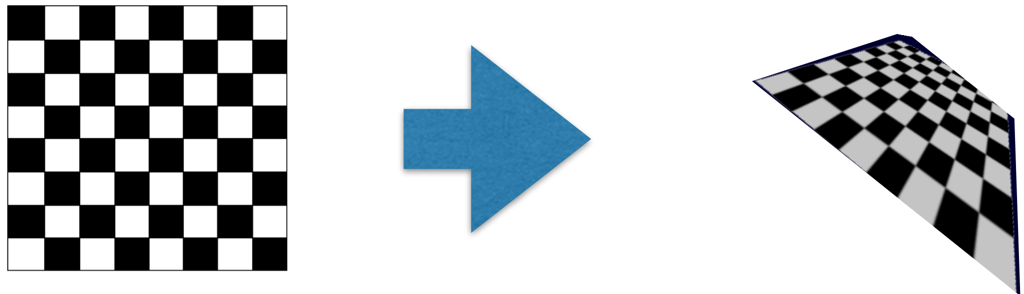




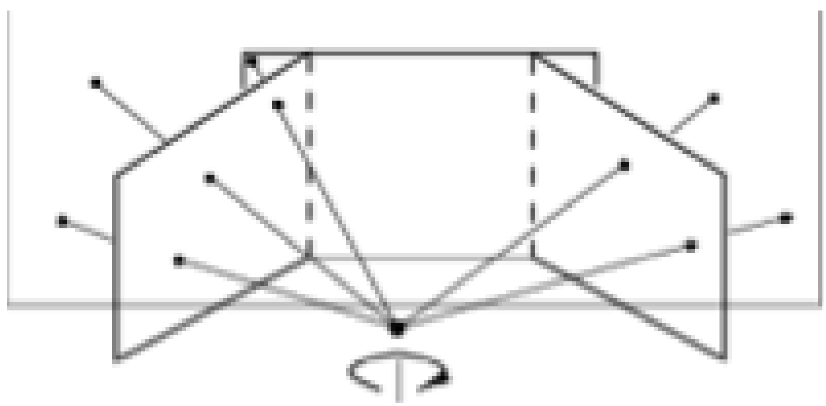
(In HW1, you'll explore homography)

Homography transformations

1. Models perspective effects for a planar scene



2. Models perspective effects from camera rotations



3. Homography mapping is a 3x3 matrix with 8 degrees of freedom (scaling the matrix by a constant doesn't matter)

RANSAC parameters

How many samples will I need before I find a good match?

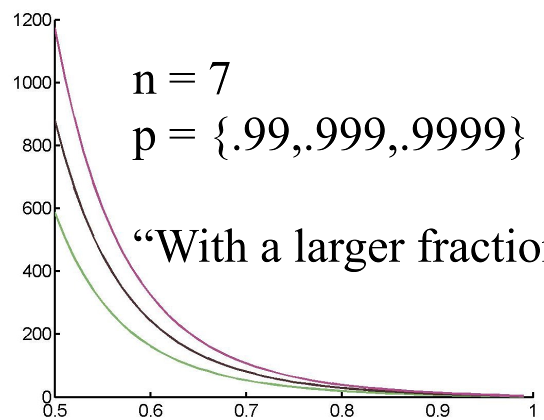
- w: fraction of inliers in data
- n: number of points I need to sample to estimate a model
  (translation: 1 pair; similarity: 2 pairs)
- k: number of random trials

Let's compute probabilities:

- w^n :prob that all n sampled points will be inliers (good)
- $1- w^n$:prob that at least 1 out of n sampled points is an outlier (bad)
- $(1- w^n)^k$:prob that RANSAC fails across all k trials (all samples are bad)
- $p = 1- (1- w^n)^k$:prob that RANSAC returns a good model (at least 1 of k trials is good)

Goal: How many trials do we need to succeed with a given probability p?

$$k = \frac{\log(1-p)}{\log(1-w^n)}$$



n = 7

p = {.99,.999,.9999}

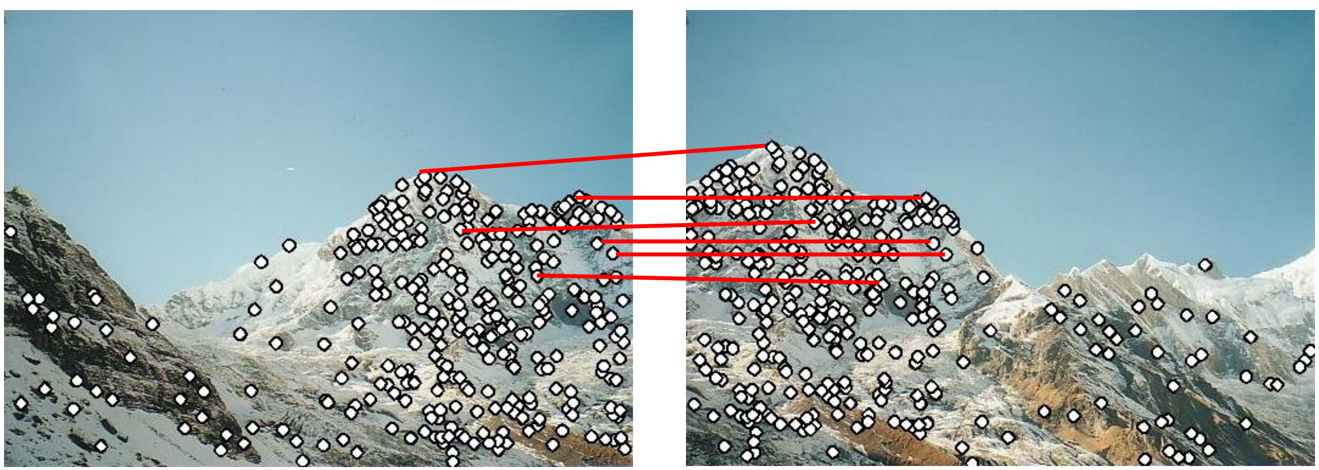
"With a larger fraction of inliers, fewer trials are needed"

w (fraction of inliers)

Panorama Stitching using SIFT

Image 1

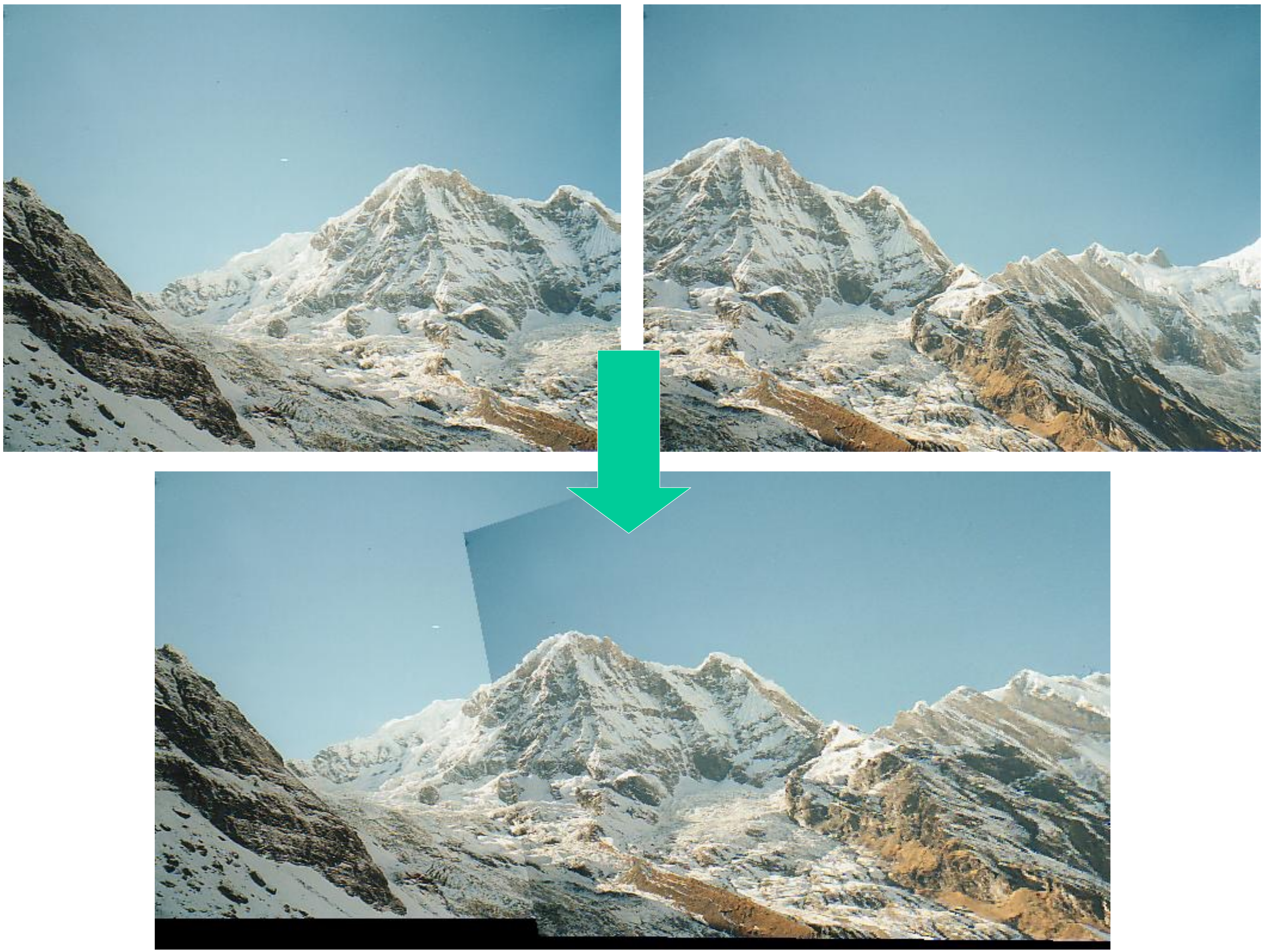
Image 2



Match SIFT Interest Points

[Autopano]

RANSAC for alignment



Outline

- Interest point detection (Harris Corners)
- Descriptors (SIFT, BRIEF, Filter Banks)
- RANSAC

References

[Autopano] Software to make panaromas using SIFT. http://user.cs.tu-berlin.de/~nowozin/autopano-sift/

[Brown and Lowe 2002] M. Brown and D. Lowe. “Invariant Features from Interest Point Groups”. *BMVC*, 2002.

[Harris and Stephens 1988] C. Harris and M. Stephens. “A Combined Corner and Edge Detector”. *4th Alvey Vision Conference*, 1988.

[Lowe 2004] D. Lowe.“Distinctive Image Features from Scale-Invariant Keypoints”. *IJCV*, 2004.

[Lindeberg 1994] T. Lindeberg. “Scale-Space Theory: A Basic Tool for Analysing Structures at Different Scales.” *J. of Applied Statistics,* 1994.

[Matas 2002] J. Matas, O. Chum, M. Urban, and T. Pajdla. “Robust Wide Baseline Stereo from Maximally Stable Extremal Regions. *BMVC*, 2002.

[Mikolajczyk 2002] K. Mikolajczyk. “Detection of Local Features Invariant to Affine Transformations.” *Ph.D. Thesis*, 2002.

References

[Mikolajczyk 2004] K. Mikolajczyk and C. Schmid. “Scale and Affine Invariant Interest Point Detectors.” *IJCV,* 2004.

[Mikolajczyk 2005] K. Mikolajczyk and C. Schmid. “A Performance Evaluation of Local Descriptors.” *PAMI,* 2005.

[SIFT] SIFT Binaries. http://www.cs.ubc.ca/~lowe/keypoints/

[Witkin 1983] A. Witkin. “Scale-Space Filtering”. *IJCAI*, 1983.

- **BRIEF, (D-BRIEF, binboost, LBGM, …):**
  - Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua . BRIEF: Binary Robust Independent Elementary Features, Proc. ECCV 2010.
  - http://cvlabwww.epfl.ch/~lepetit/
- **BRISK:**
  - Stefan Leutenegger, Margarita Chli and Roland Y. Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. Proc. ICCV 2011.
- **GIST:**
  - Aude Oliva, Antonio Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. International Journal of Computer Vision, Vol. 42(3): 145-175, 2001.
  - http://people.csail.mit.edu/torralba/code/spatialenvelope/

- K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, T. Kadir and L. Van Gool, A Comparison of Affine Region Detectors; International Journal of Computer Vision (IJCV), Volume 65, Number 1, 2005.
- Major resource for all the features (Vision Lab Features Library (VLFeat)):

http://www.vlfeat.org/api/index.html

- **SIFT:**
  - David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. IJCV (International Journal of Computer Vision), 2004.
  - http://www.cs.ubc.ca/~lowe/keypoints/
- **MSER:**
  - P-E. Forssen, P-E. and D. Lowe, Shape Descriptors for Maximally Stable Extremal Regions International Conference on Computer Vision (ICCV), 2007.
- **SURF:**
  - Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool, SURF: Speeded Up Robust Features, Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, 2008.
  - http://www.vision.ee.ethz.ch/~surf/
  - **DAISY:** Engin Tola, Vincent Lepetit, Pascal Fua, DAISY: An Efficient Dense Descriptor Applied to Wide-Baseline Stereo, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 2010.
- **LBP:**
  - Timo Ojala, Matti Pietikainen, and Topi Maenpa. Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 24, No. 7, July 2002
  - http://www.cse.oulu.fi/CMV/Research/LBP