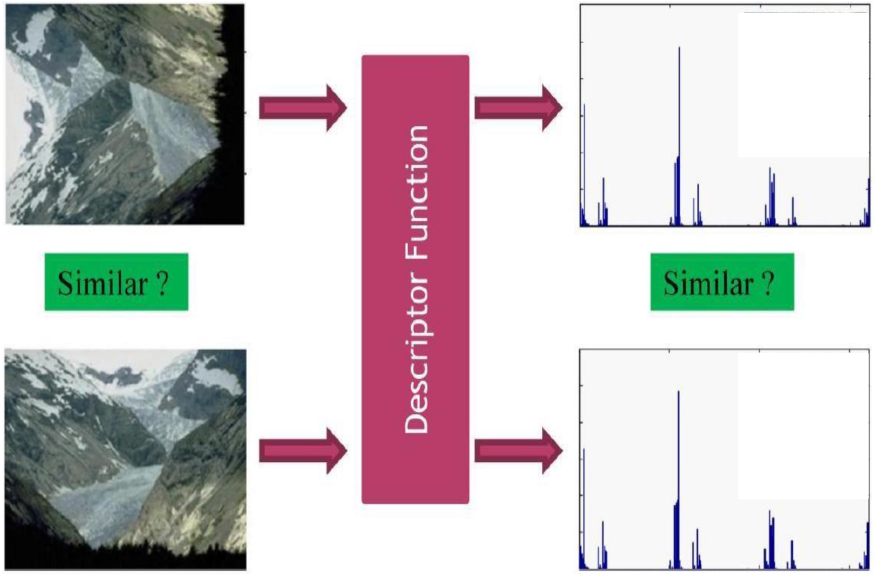
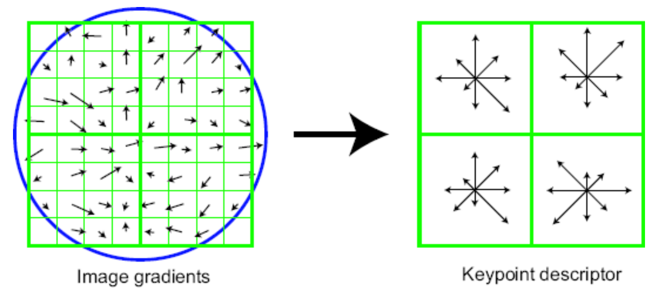
Feature Matching



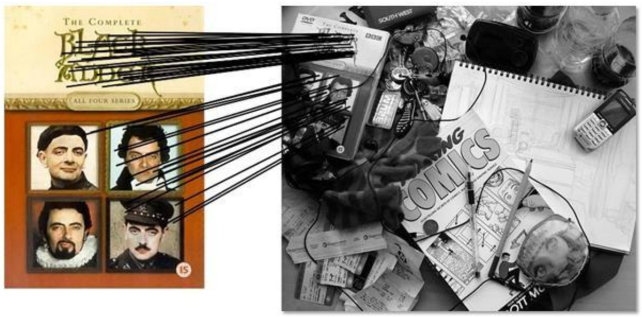
Computer Vision

Adduru U G Sankararao, IIIT Sri City

Previous Class



- Local Descriptors:
  - Discriminative
  - Robust
  - Compact
  - Efficient



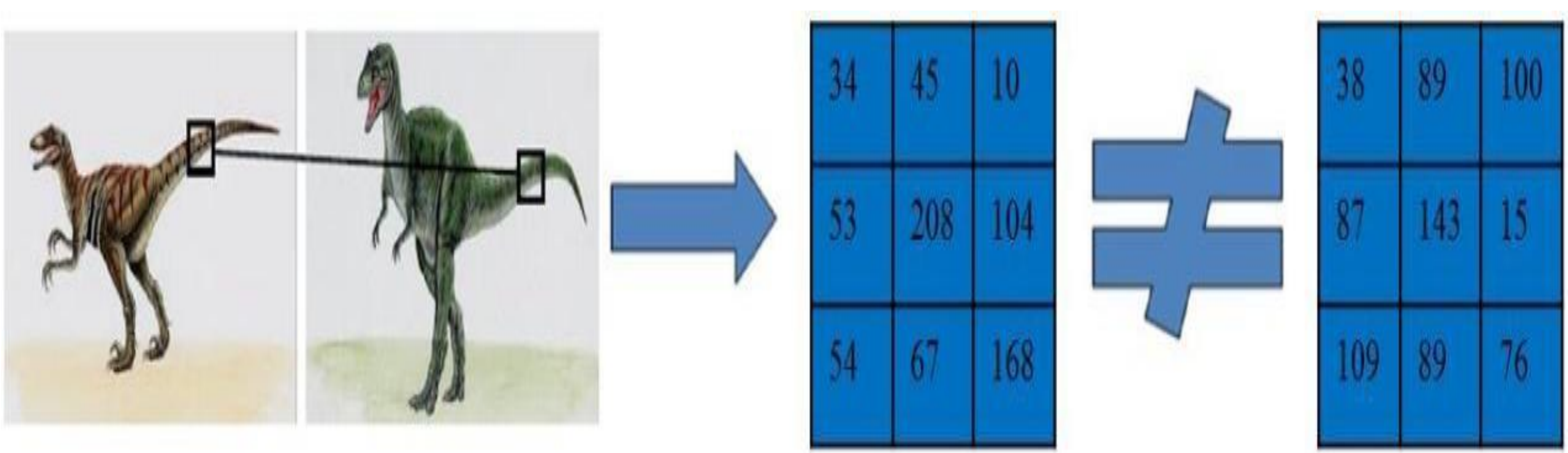
- Scale Invariant Feature Transform
  - Utilizes gradient information
  - Gradient direction binning

- Next:
  - Feature Matching
  - Evaluation Metrics

Image/Region Matching

- Automatically recognize whether two images/regions contain the similar content.

- Comparing the image pixels as they are, will not work.



Image/Region Matching

- Pixel-based distances on high-dimensional data (and images especially) can be very unintuitive.

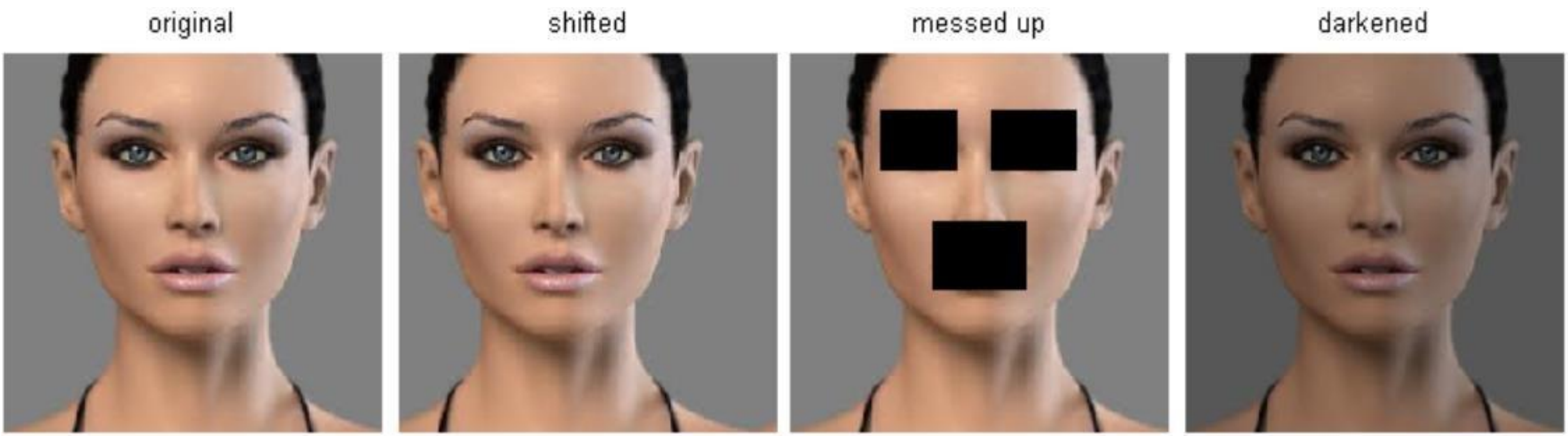


Image Source: cs231n, Oxford University

Challenges

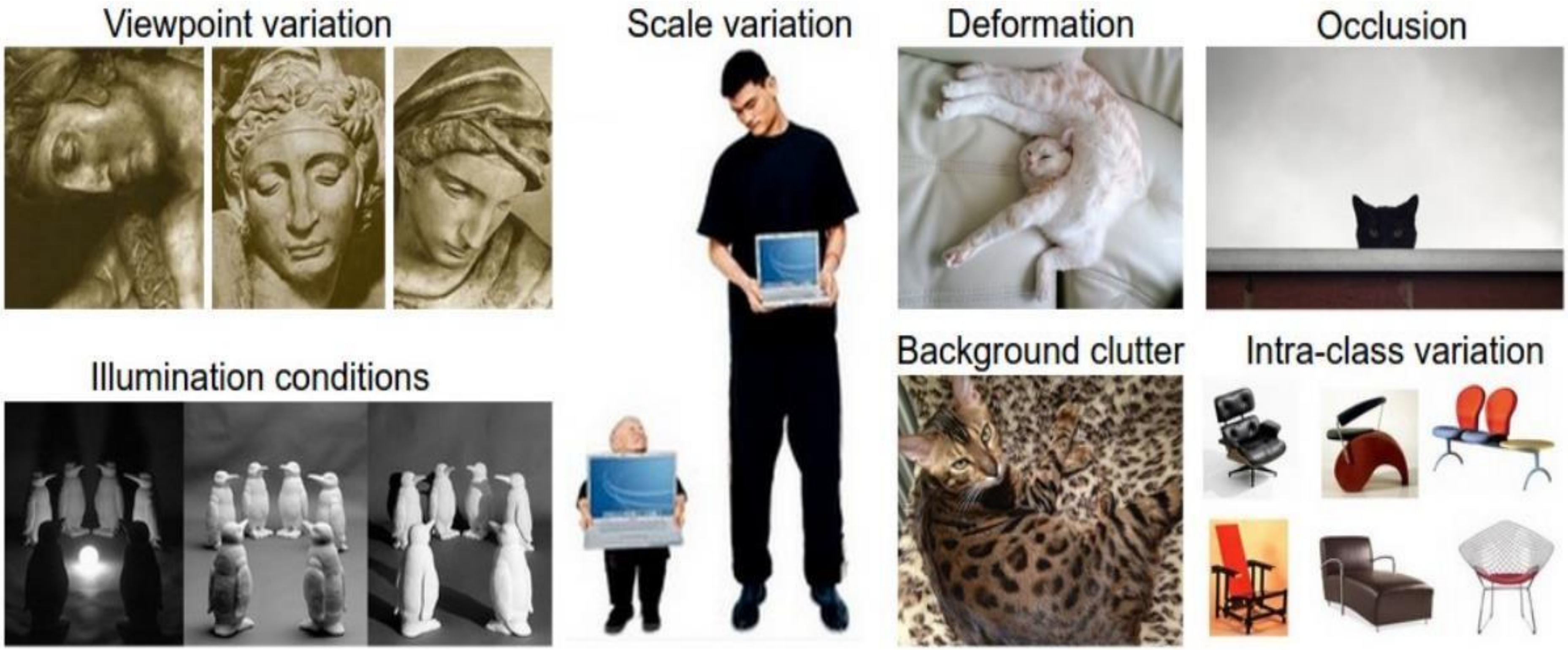
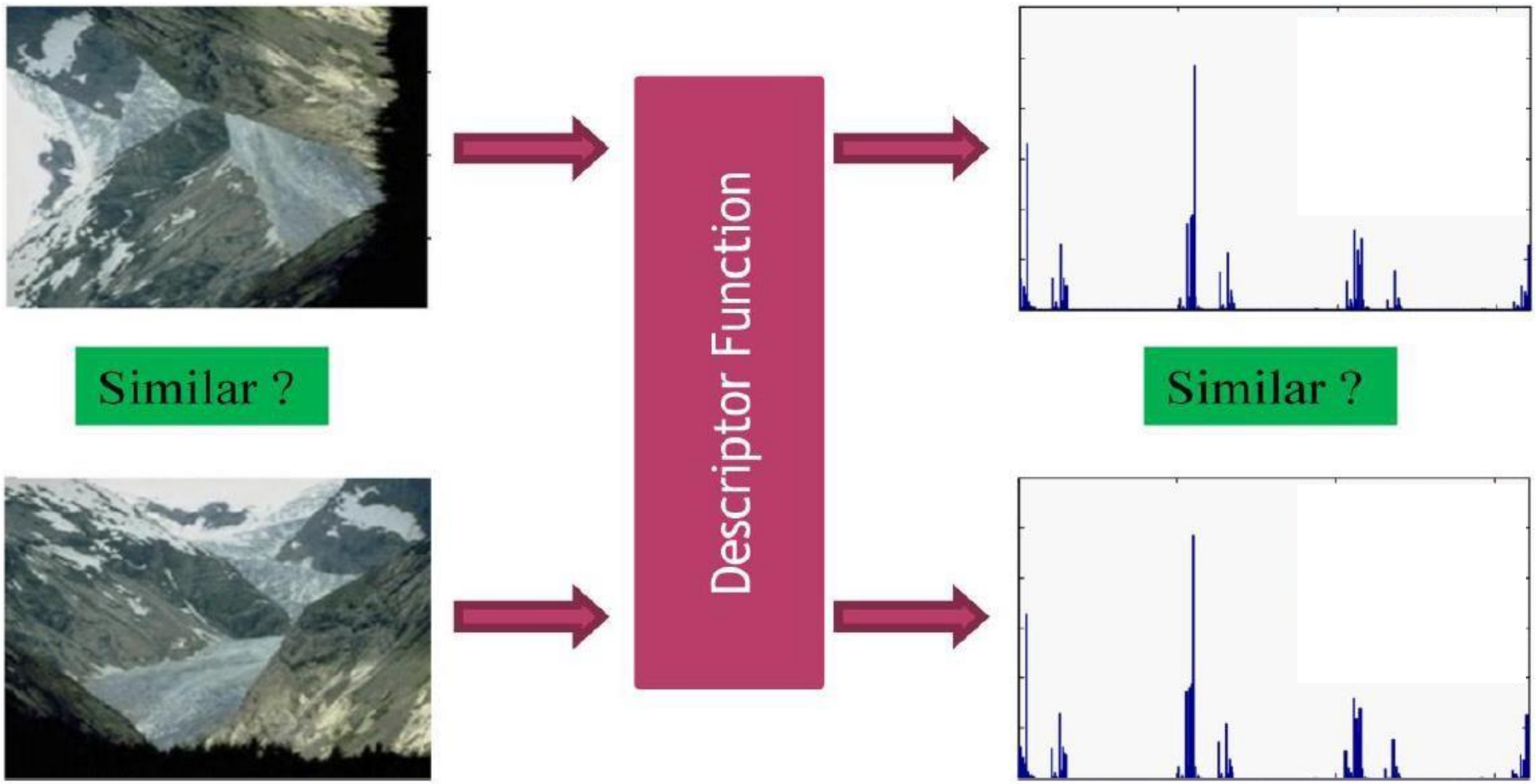


Image Source: cs231n, Oxford University

Solution

- Descriptors allow certain differences between the images.



Comparing using descriptor function
(images are taken from Corel-database and RSHD descriptor is used)

Dubey et al. Rotation and scale invariant hybrid image descriptor and retrieval. *Computers & Electrical Engineering*, 2015.

Feature matching

Given a feature in I_1, how to find the best match in I_2?

1. Define distance function that compares two descriptors
2. Test all the features in I_2, find the one with min distance

How to define the difference between two features f_1, f_2?

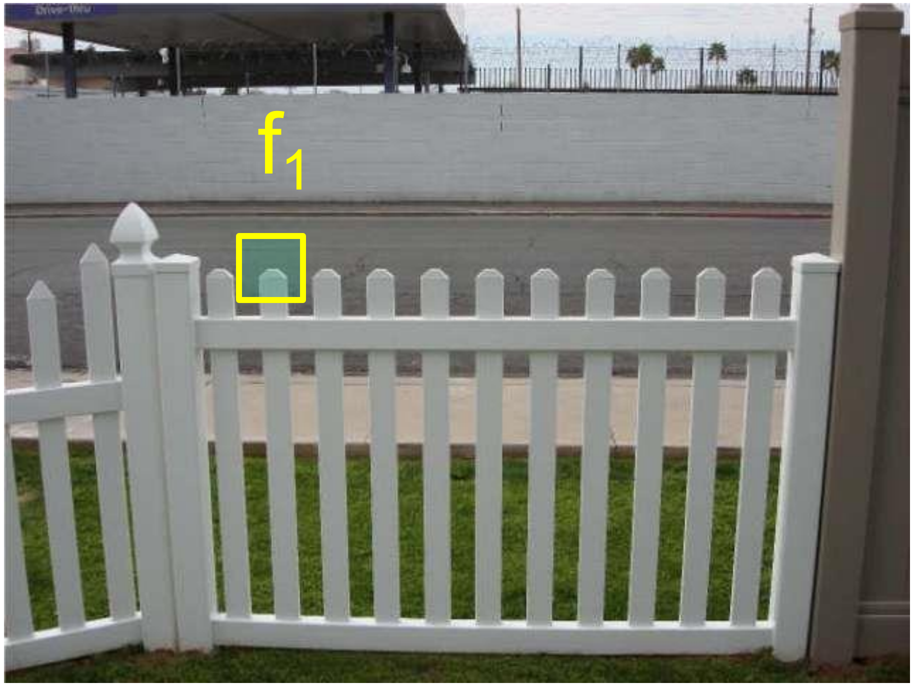
- Simple approach: L_2 distance, $||f_1 - f_2||$

Techniques:

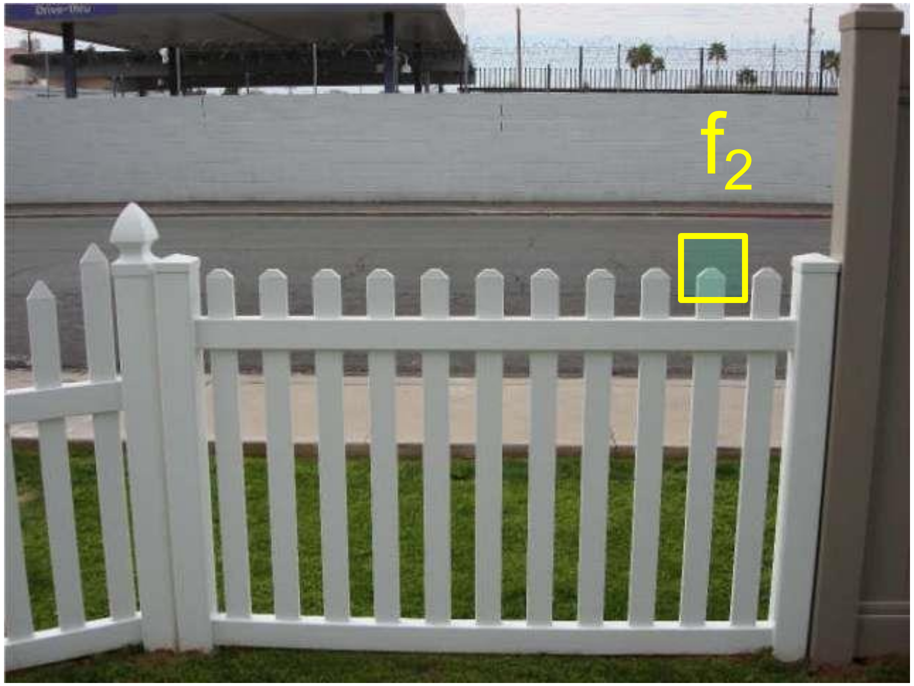
- Brute Force matching (L2, cv::BFMatcher in OpenCV)
- Nearest Neighbour matching
- FLANN-based matching

Brute Force Matcher (BFMatcher)

- Compares each descriptor in one image with all descriptors in the other image.
- Finds the closest match based on a distance metric (eg: L2 norm (Euclidean))
- Simple, easy to use, but Slow when there are lots of features.
- Can give good scores to ambiguous (incorrect) matches



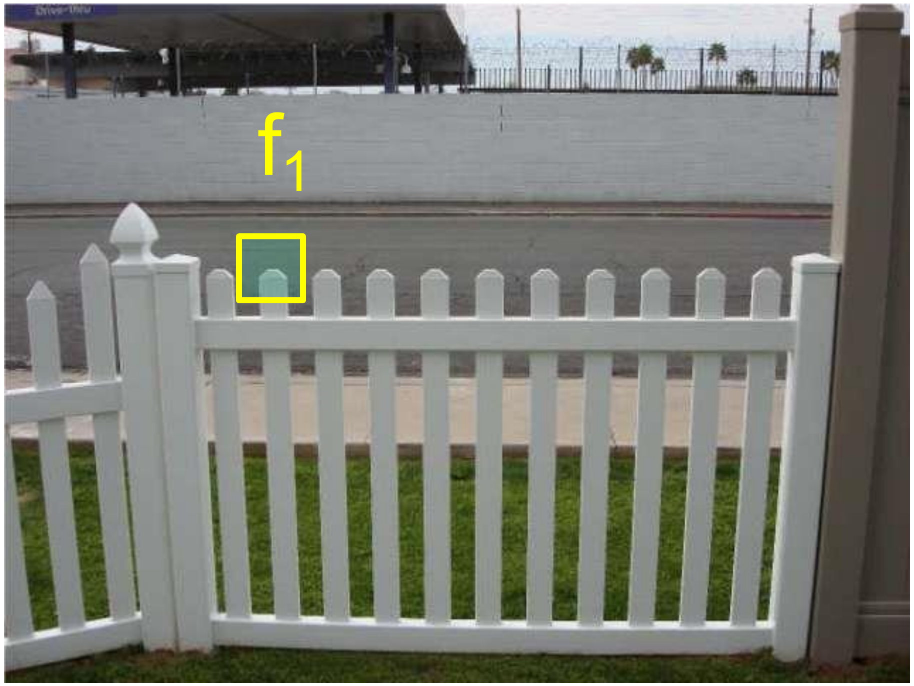
I_1



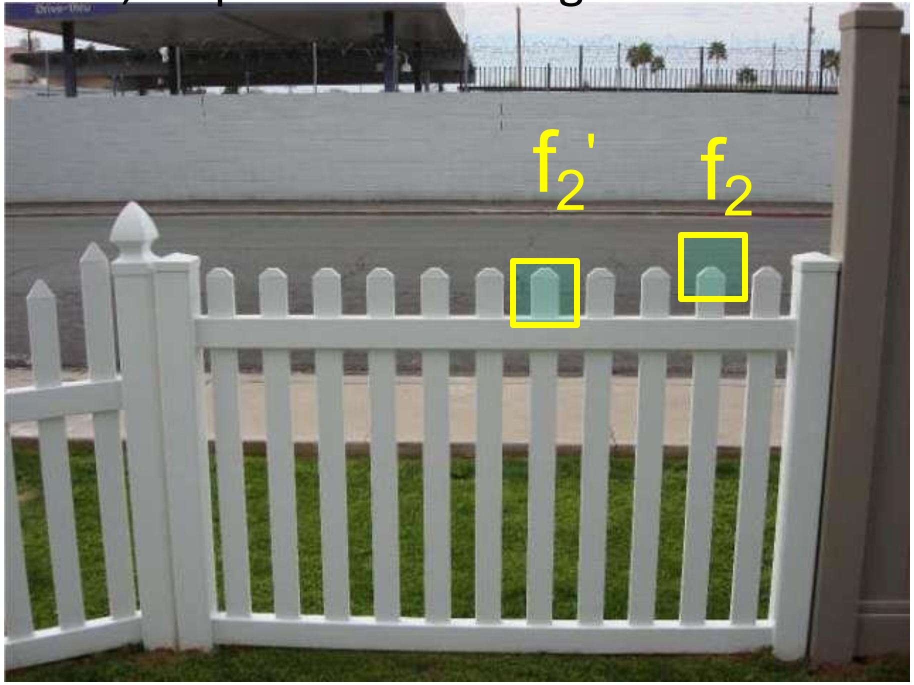
I_2

Nearest Neighbours based matching

- Instead of just taking the single closest match, it finds k best matches for each descriptor, (Usually k = 2).
- Then apply Lowe's ratio test to keep only reliable matches:
- Better approach: ratio distance = $||f_1 - f_2|| / || f_1 - f_2' ||$
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2nd best SSD match to f_1 in I_2
  - Keep a match if the ratio is less than a threshold
  - gives large values for ambiguous matches, helps remove ambiguous matches

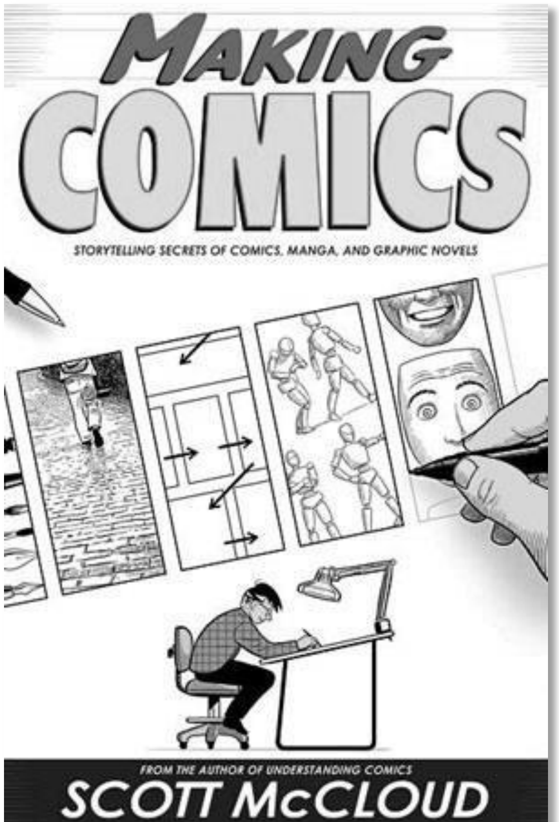


I_1

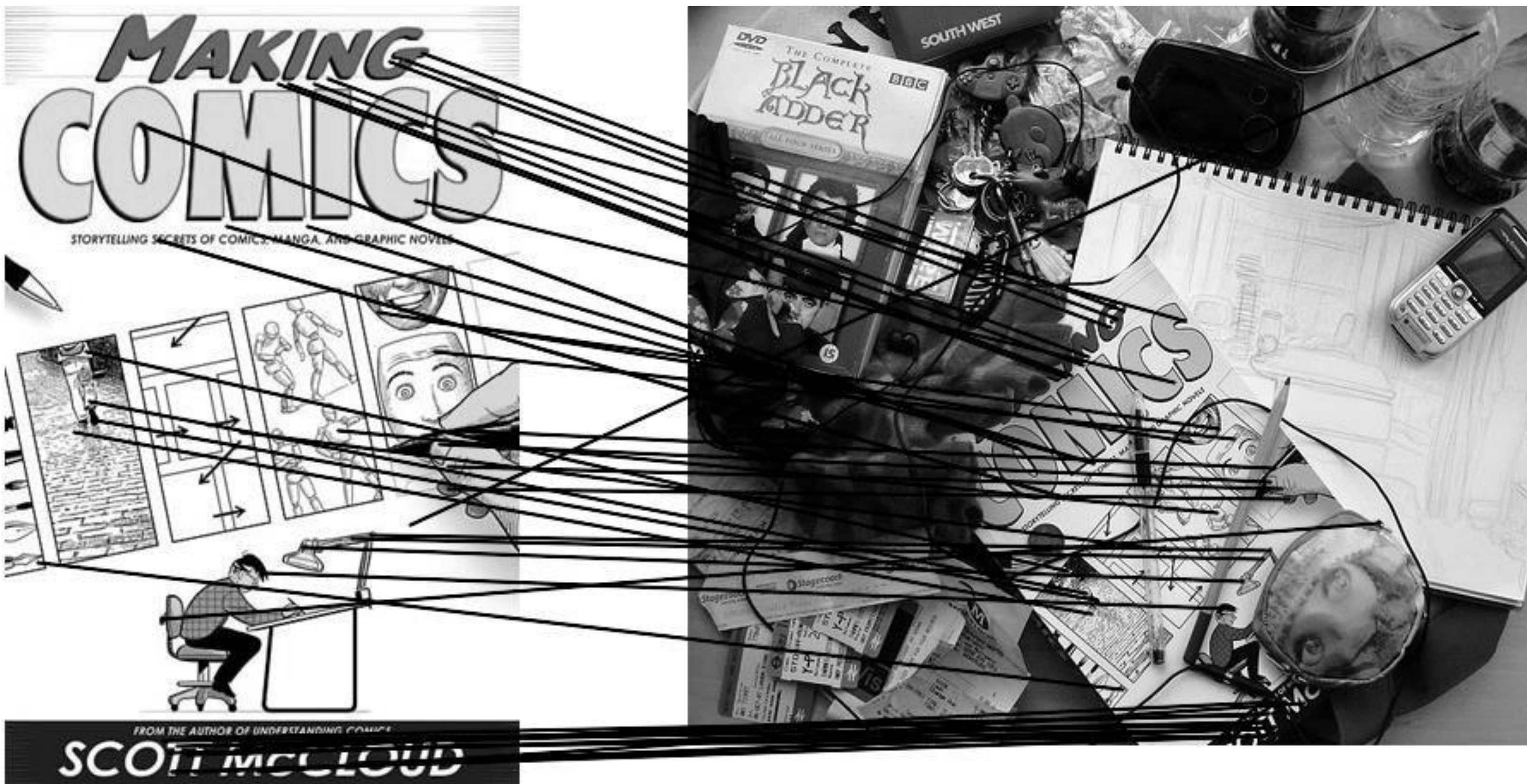


I_2

Feature matching example

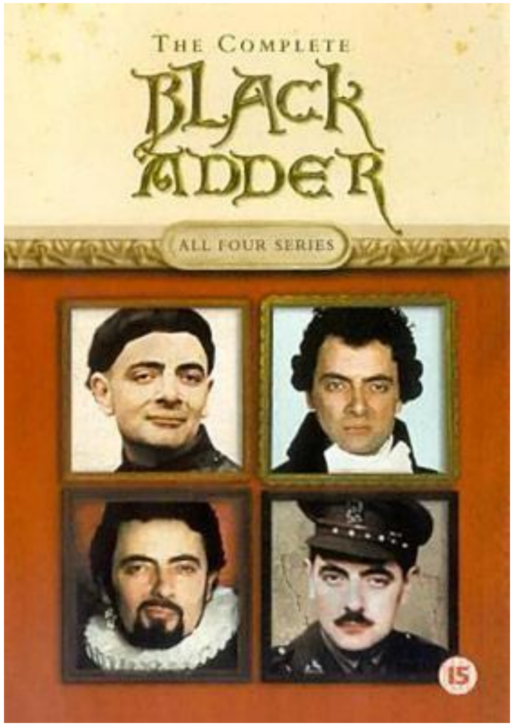


Feature matching example

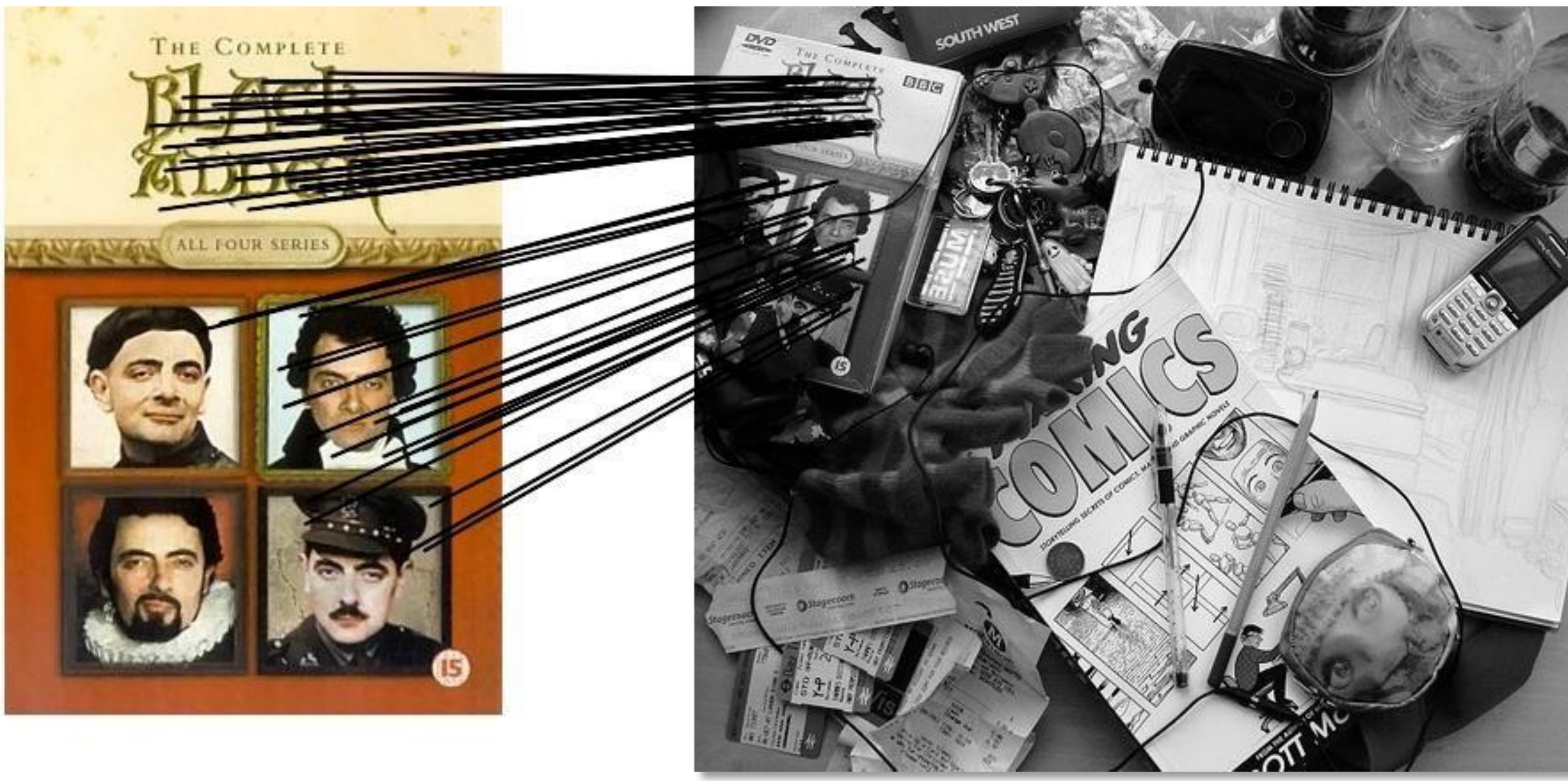


51 matches

Feature matching example

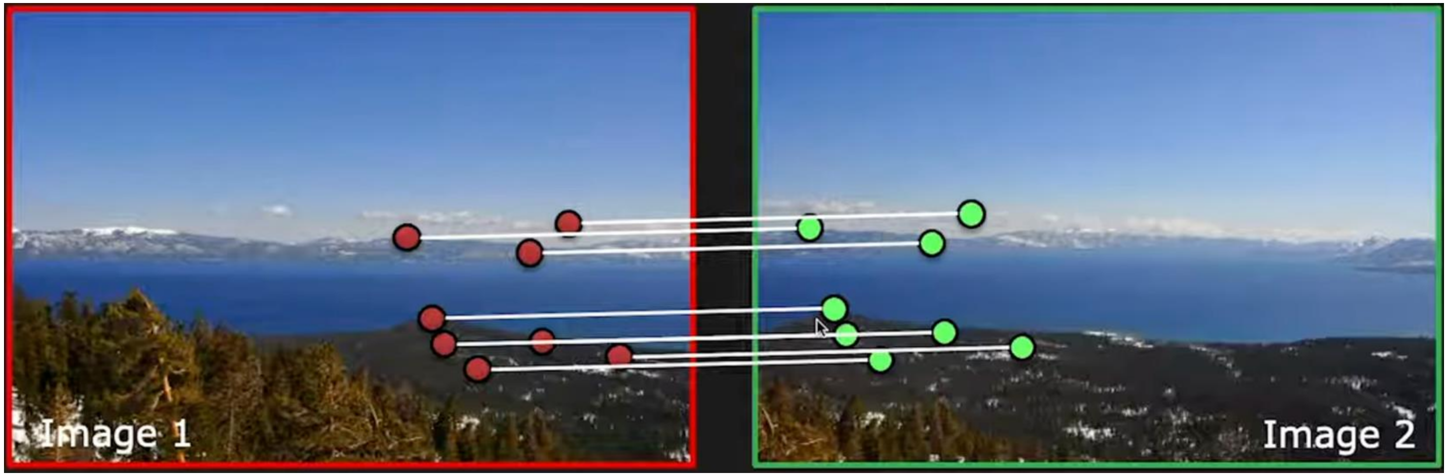


Feature matching example



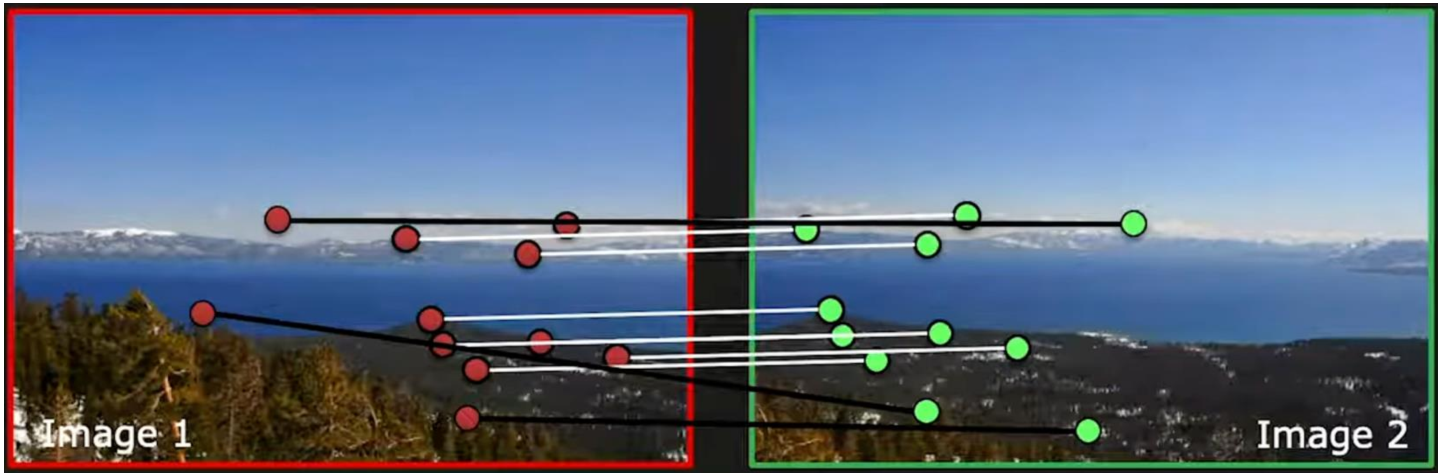
58 matches

Feature matching:



Correct matches: Inliers

Need to remove outliers



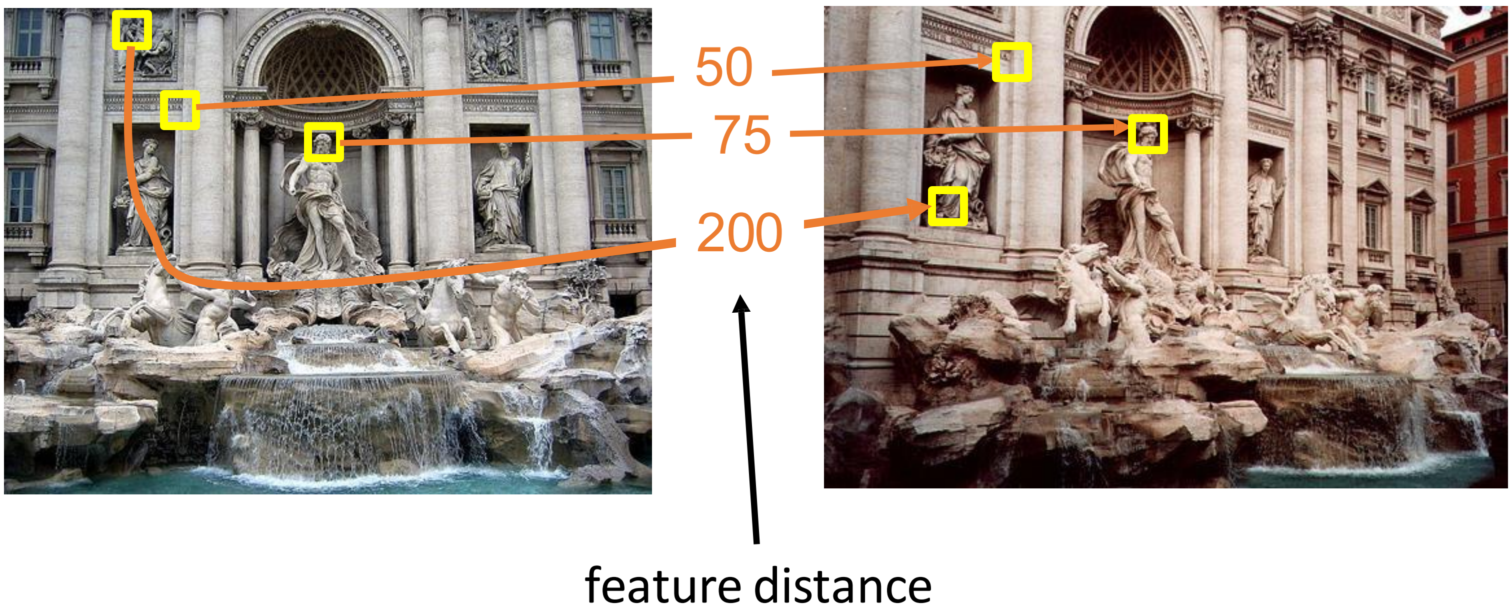
Incorrect matches: Outliers

Evaluating the results

How can we measure the performance of a feature matcher?

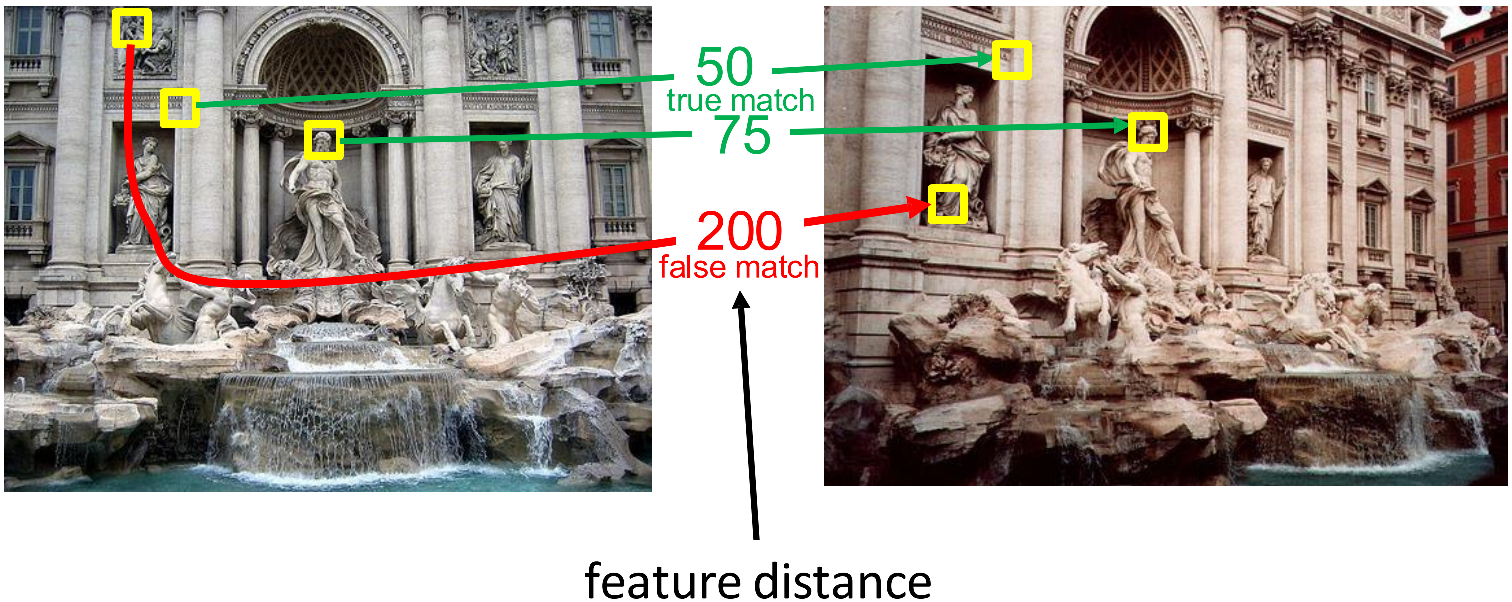
Evaluating the results

How can we measure the performance of a feature matcher?



True/false positives

How can we measure the performance of a feature matcher?



The distance threshold affects performance

- True positives = # of detected matches that are correct
  - Suppose we want to maximize these—how to choose threshold?
- False positives = # of detected matches that are incorrect
  - Suppose we want to minimize these—how to choose threshold?

Evaluation Metrics

	True matches	True non-matches		
Predicted matches	TP = 18	FP = 4	P' = 22	PPV = 0.82
Predicted non-matches	FN = 2	TN = 76	N' = 78	
	P = 20	N = 80	Total = 100	

TPR = 0.90	FPR = 0.05	ACC = 0.94

- true positive rate (TPR),

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} = \frac{\mathrm{TP}}{\mathrm{P}};$$

- false positive rate (FPR),

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP+TN}} = \frac{\mathrm{FP}}{\mathrm{N}};$$

- positive predictive value (PPV),

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} = \frac{\mathrm{TP}}{\mathrm{P'}};$$

- accuracy (ACC),

$$\mathrm{ACC} = \frac{\mathrm{TP+TN}}{\mathrm{P+N}}.$$

PPV = Precision

TPR. = Recall

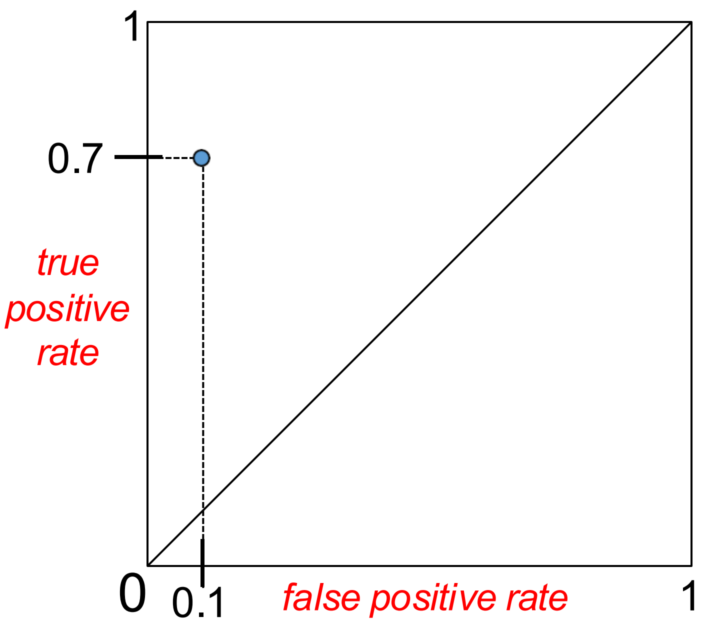
F1-Score = HM (Pre, Rec)

Evaluating the results

How can we measure the performance of a feature matcher?

$$\frac{\text{\# true positives}}{\text{\# correctly matched features (positives)}}$$

"recall"



$$\frac{\text{\# false positives}}{\text{\# incorrectly matched features (negatives)}}$$

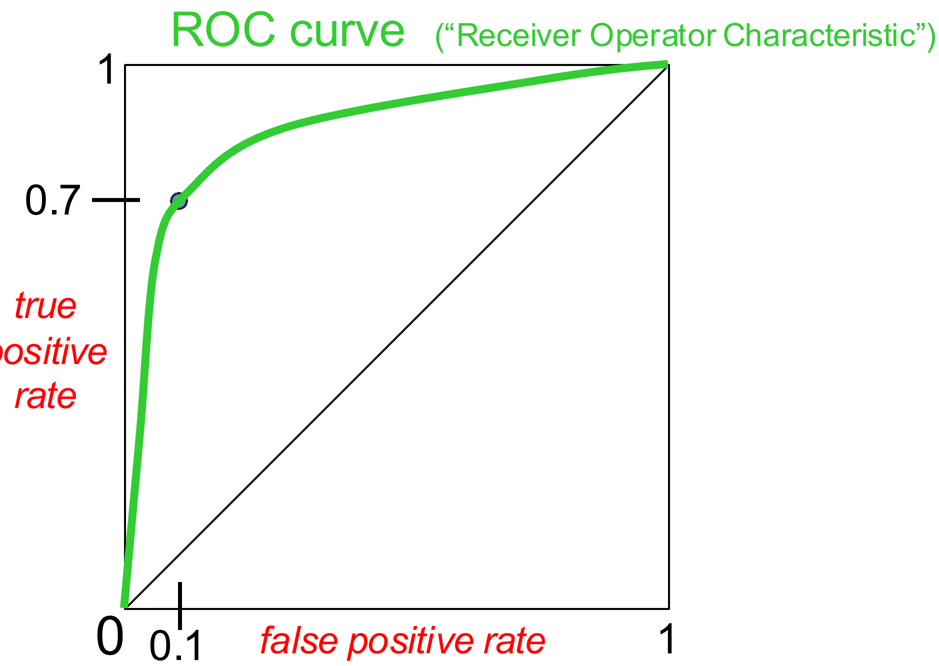
1 - "precision"

Evaluating the results

How can we measure the performance of a feature matcher?

Aera under ROC

AUC more → better



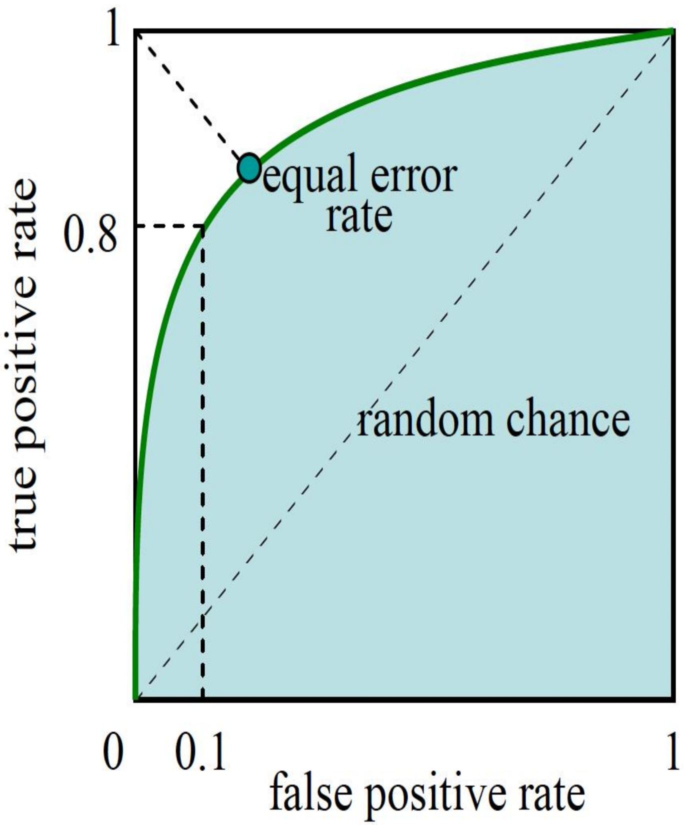
$$\frac{\text{\# true positives}}{\text{\# correctly matched features (positives)}}$$

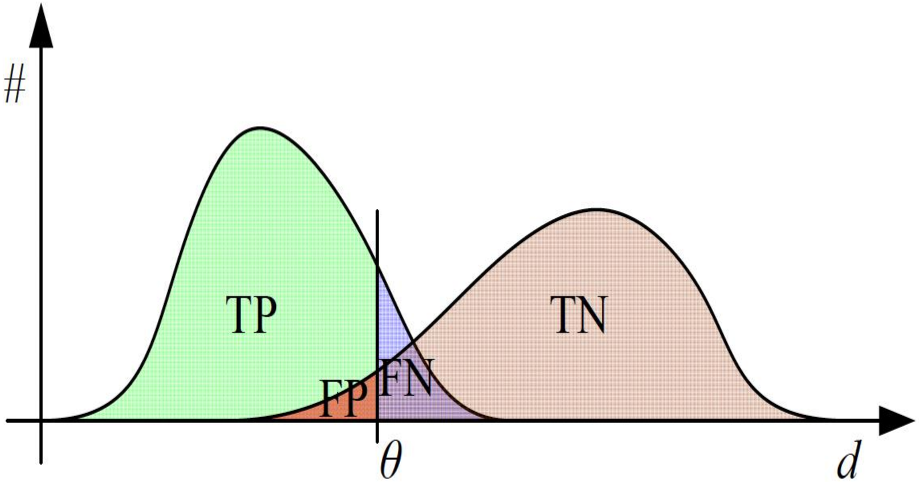
"recall"

$$\frac{\text{\# false positives}}{\text{\# incorrectly matched features (negatives)}}$$

1 - "precision"

Evaluating Results





- *As the threshold   is increased, the number of true positives (TP) and false positives (FP) increases.*

Image matching

dense registration*

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g*. stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

dense registration*

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g.* stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

dense registration*

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g.* stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

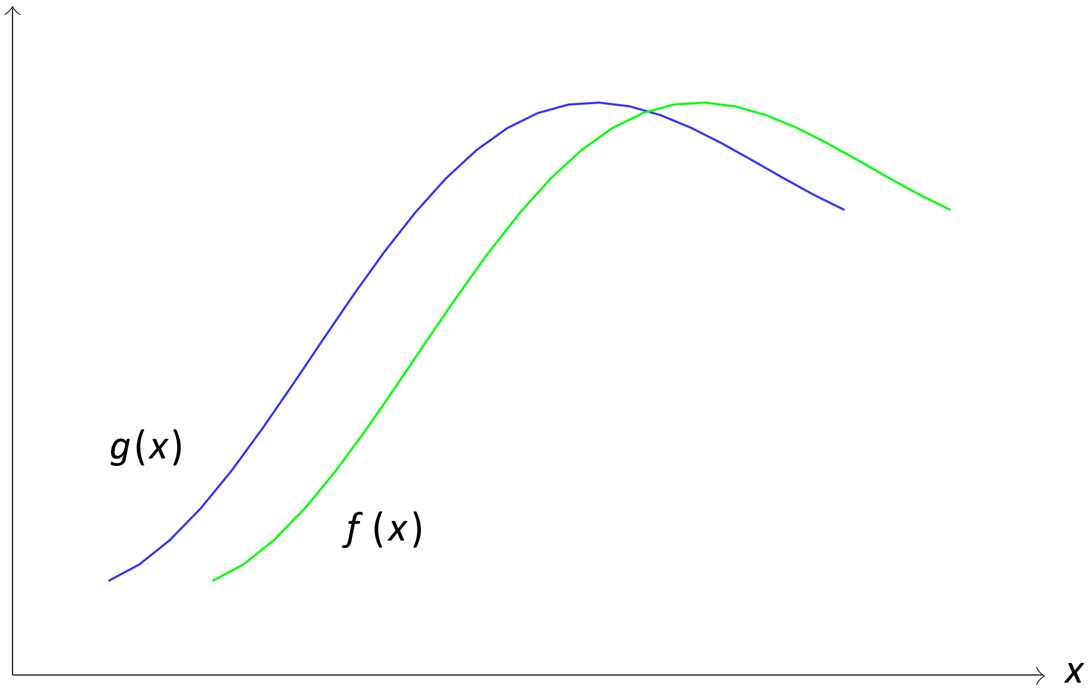
dense registration*

[Lucas and Kanade 1981]

- for each location in an image, find a displacement with respect to another reference image
- appropriate for small displacements, *e.g.* stereopsis or optical flow

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

one dimension*

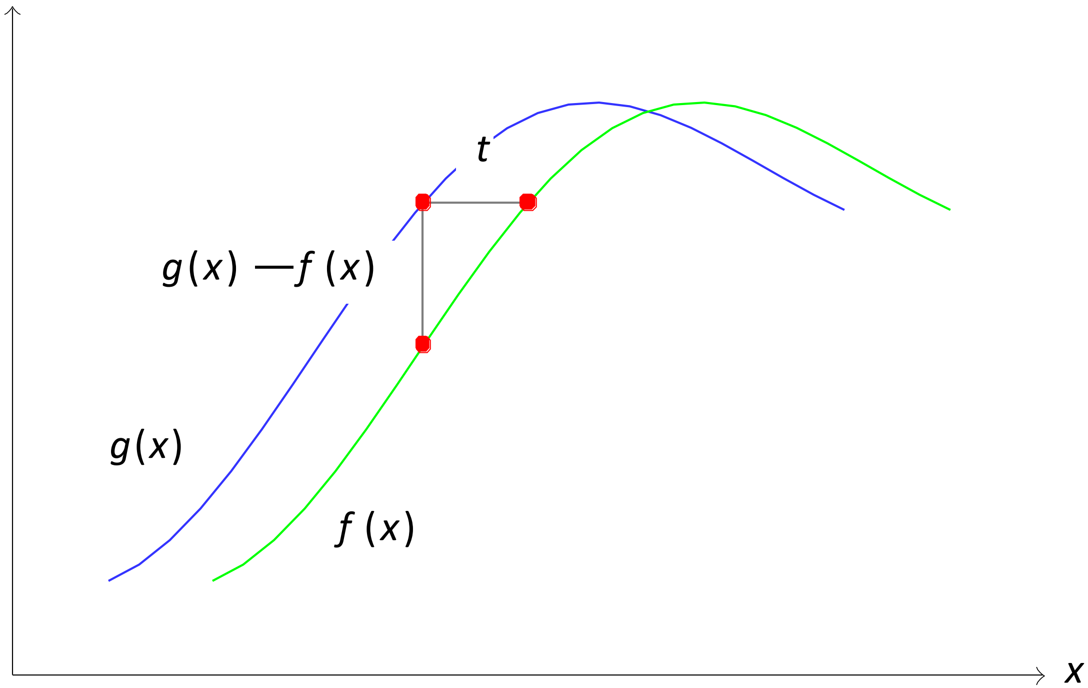


- assuming $g(x) = f(x + t)$ and t is small,

$$\frac{df}{dx}(x) \approx \frac{f(x+t) - f(x)}{t} = \frac{g(x) - f(x)}{t}$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

one dimension*

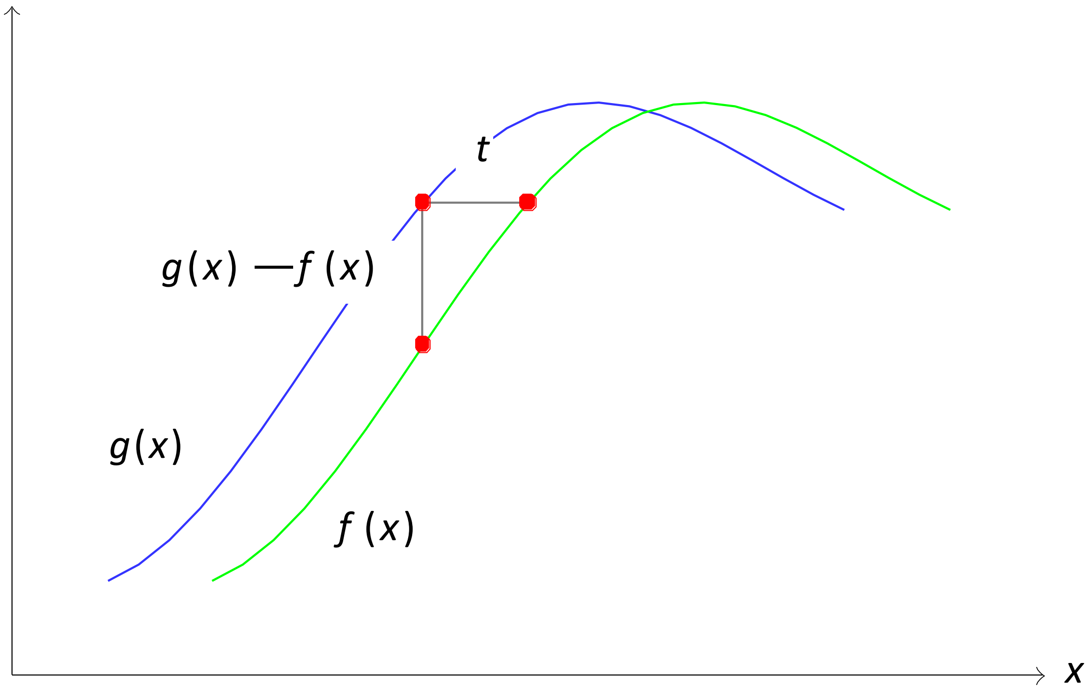


- assuming $g(x) = f(x + t)$ and t is small,

$$\frac{df}{dx}(x) \approx \frac{f(x+t) - f(x)}{t} = \frac{g(x) - f(x)}{t}$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

one dimension*



- assuming $g(x) = f(x + t)$ and t is small,

$$\frac{df}{dx}(x) \approx \frac{f(x+t) - f(x)}{t} = \frac{g(x) - f(x)}{t}$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

two dimensions: least squares*

- again, assume an image patch defined by window w; what is the error between the patch shifted by $\mathbf{t}$ in reference image f and a patch at the origin in shifted image g?

$$E(\mathbf{t}) = \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}+\mathbf{t}) - g(\mathbf{x}))^2$$
$$\approx \sum_{\mathbf{x}} w(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))^2$$

- error minimized when gradient vanishes

$$\mathbf{0} = \frac{\partial E}{\partial \mathbf{t}} = \sum_{\mathbf{x}} w(\mathbf{x}) 2 \nabla f(\mathbf{x})(f(\mathbf{x}) + \mathbf{t}^\top \nabla f(\mathbf{x}) - g(\mathbf{x}))$$

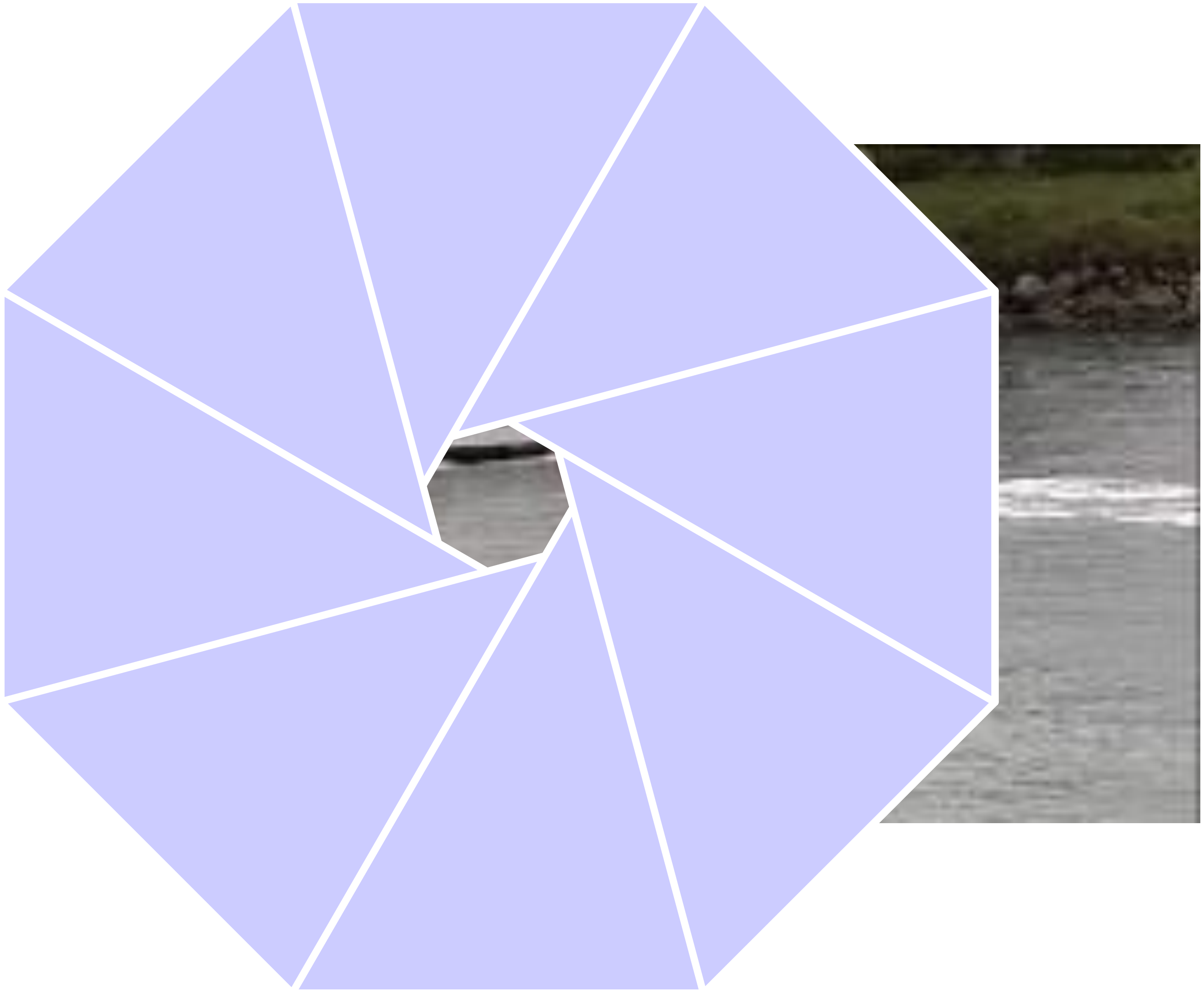
- least-squares solution

$$\left(w * (\nabla f)(\nabla f)^\top\right) \mathbf{t} = w * ((\nabla f)(g - f))$$

Lucas and Kanade IJCAI 1981. An Iterative Image Registration Technique With an Application to Stereo Vision.

the aperture problem*

the aperture problem*



wide-baseline matching

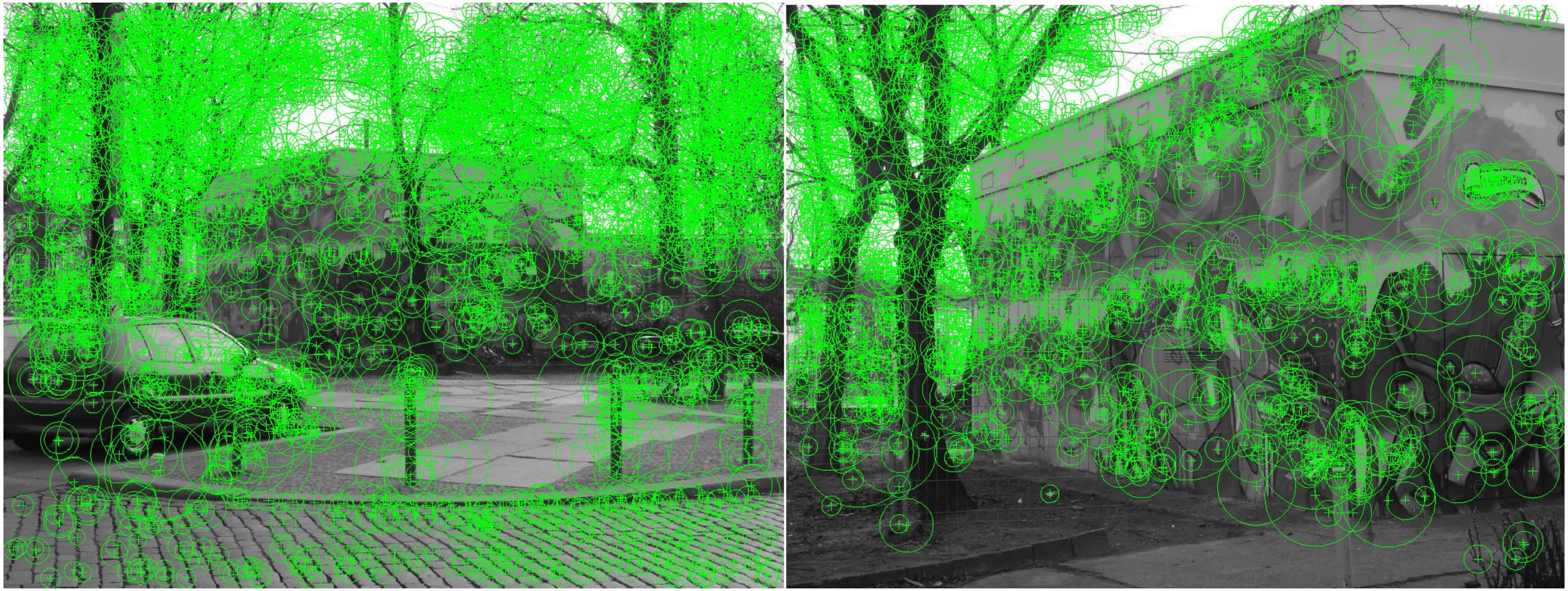
- in dense registration, we started from a local "template matching" process and found an efficient solution based on a Taylor approximation
- both make sense for small displacements

- in wide-baseline matching, every part of one image may appear anywhere in the other
- we start by pairwise matching of local descriptors without any order and then attempt to enforce some geometric consistency according to a rigid motion model

wide-baseline matching

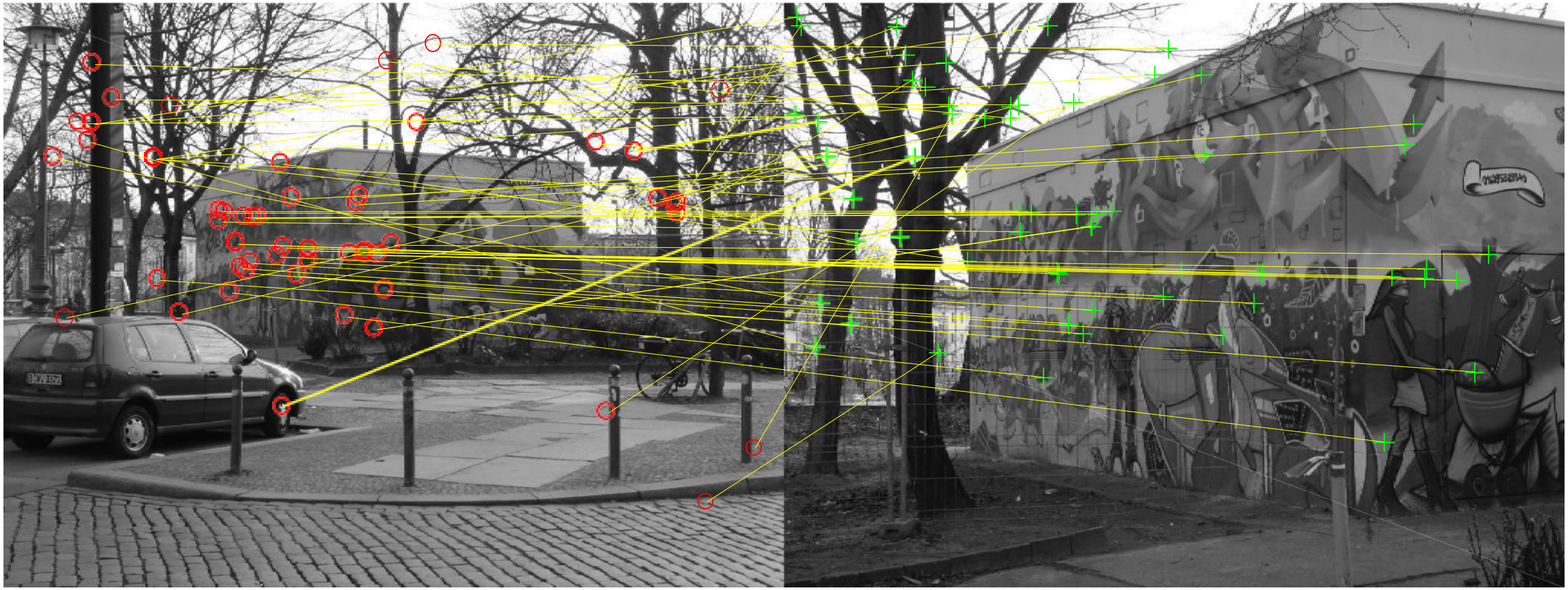
- a region in one image may appear anywhere in the other

wide-baseline matching



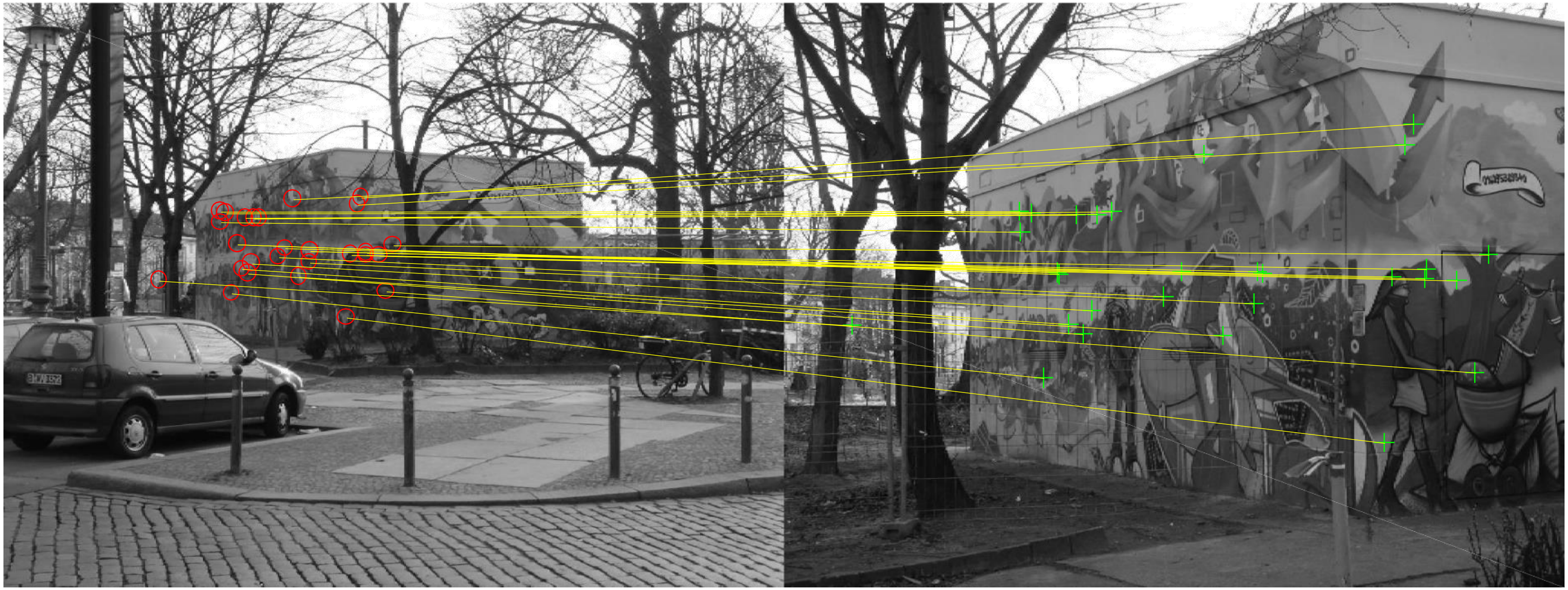
- features detected independently in each image

wide-baseline matching



- tentative correspondences by pairwise descriptor matching

wide-baseline matching



- subset of correspondences that are ‘inlier’ to a rigid transformation

descriptor extraction

for each detected feature in each image

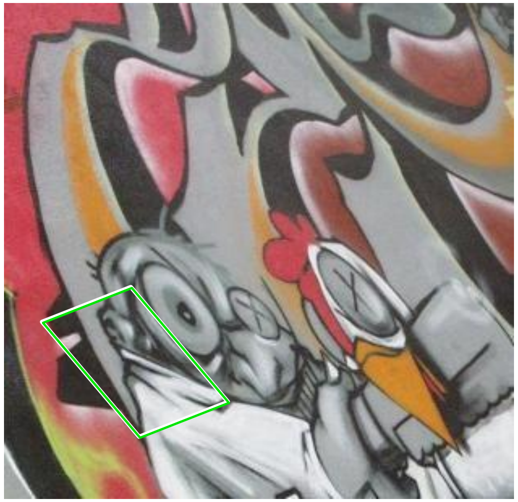
- construct a local histogram of gradient orientations
- find one or more dominant orientations corresponding to peaks in the histogram
- resample local patch at given location, scale, affine shape and orientation
- extract one descriptor for each dominant orientation

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching



- detect features

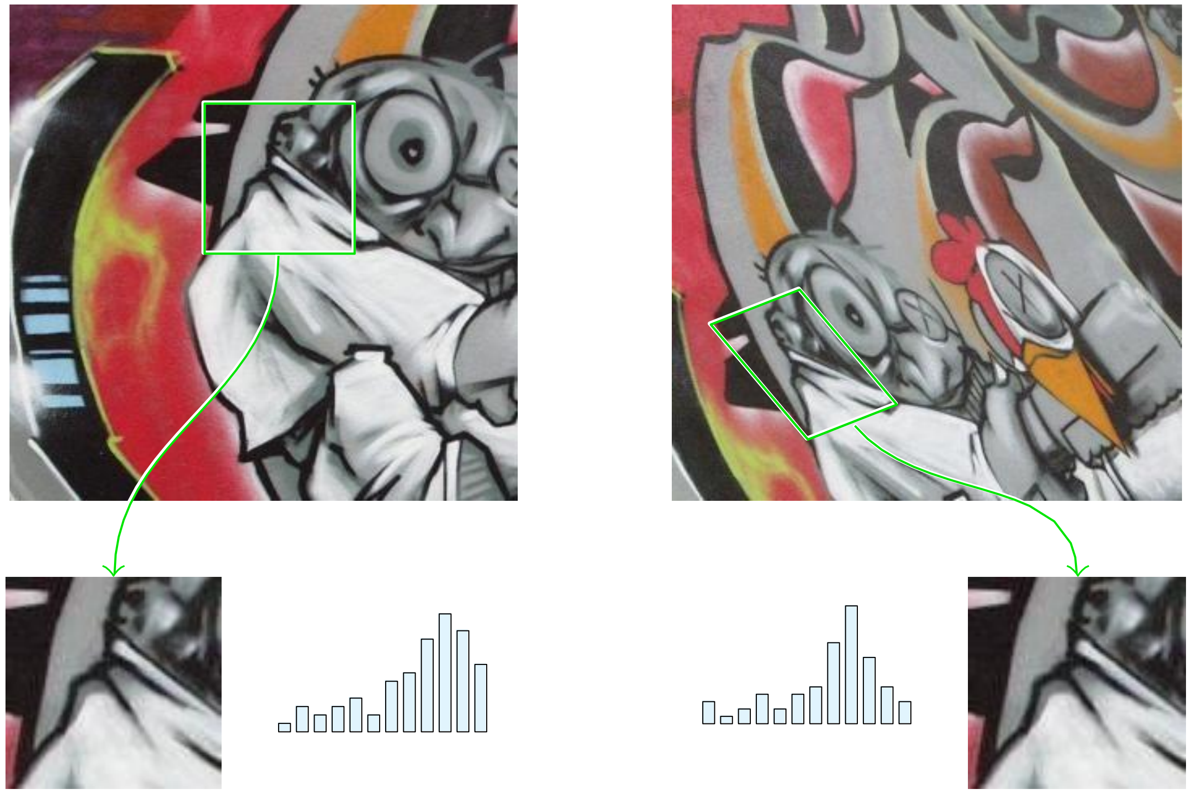
Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching

- detect features - find dominant orientation, resample patches

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

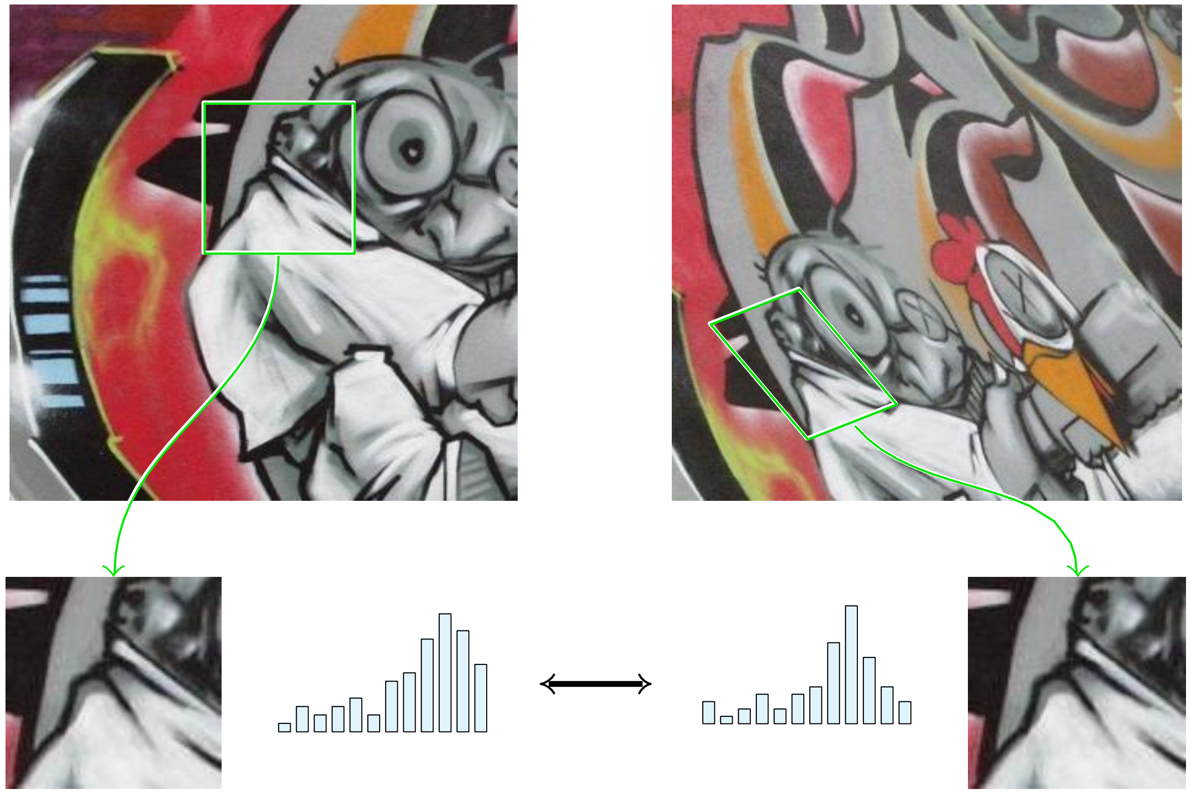
descriptor matching



- detect features - find dominant orientation, resample patches - extract descriptors

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching



- detect features - find dominant orientation, resample patches - extract descriptors - match pairwise

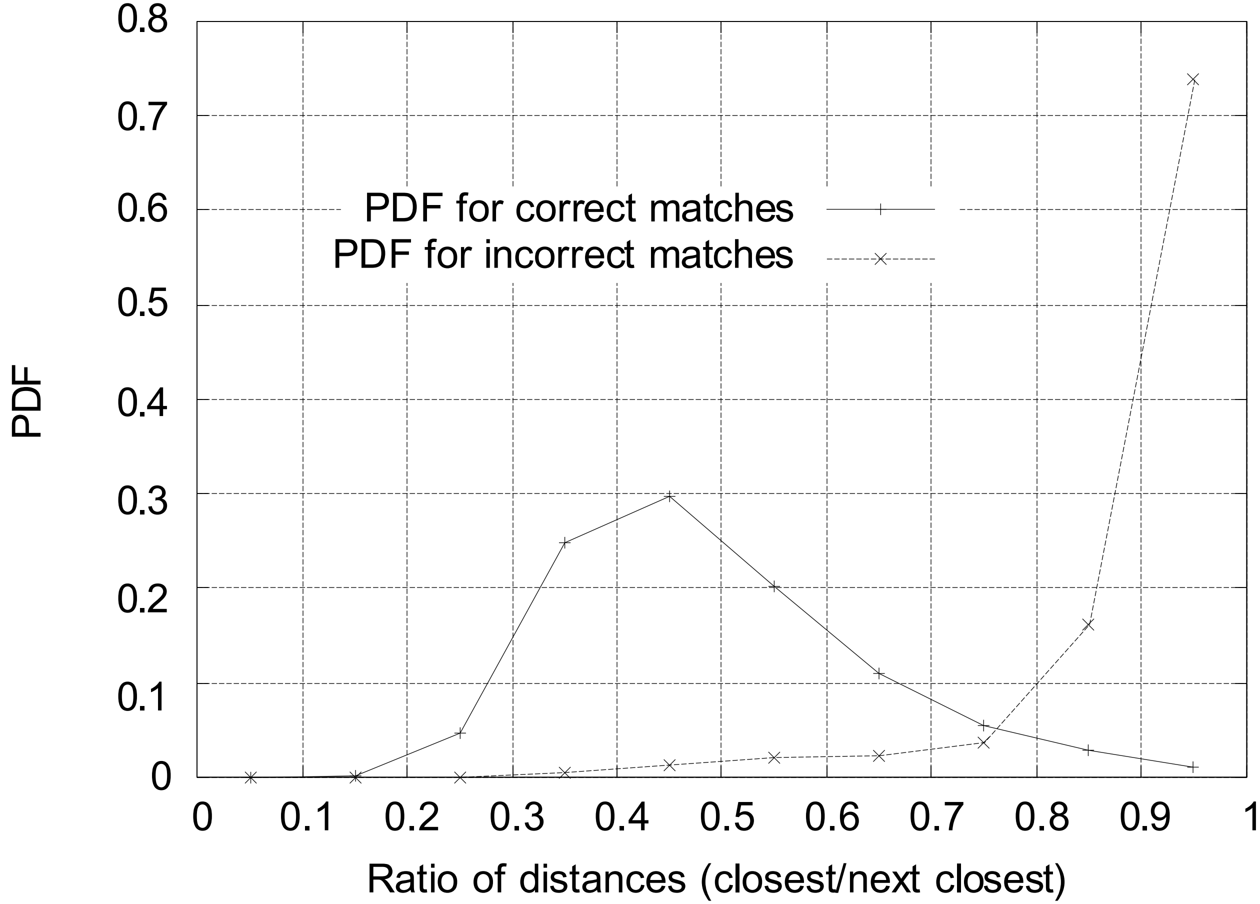
Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

descriptor matching

- for each descriptor in one image, find its two nearest neighbors in the other
- if ratio of distance of first to distance of second is small, make a correspondence
- this yields a list of tentative correspondences

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.

ratio test



- ratio of first to second nearest neighbor distance can determine the probability of a true correspondence

Lowe. IJCV 2004. Distinctive Image Features From Scale-Invariant Keypoints.