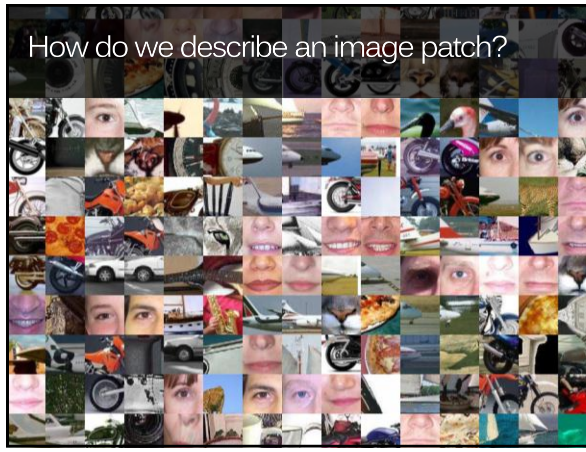
Patch Descriptors

CSE P 576

Larry Zitnick (larryz@microsoft.com)
Many slides courtesy of Steve Sietz

How do we describe an image patch?

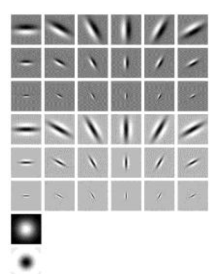


How do we describe an image patch?

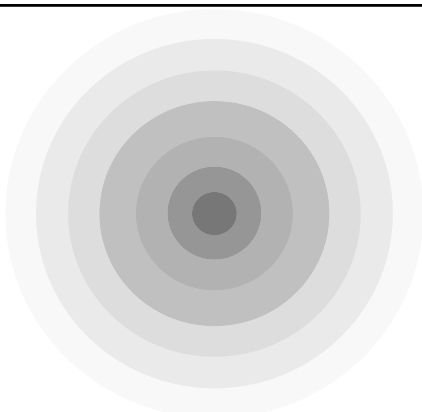
Patches with similar content should have similar descriptors.

What do human use?

Gabor filters…



… and many other things.



Encoding the gradients

Poster

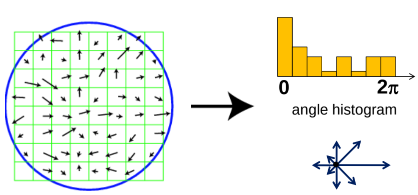
Object recognition from **local scale-invariant features**
DG Lowe - iccv, 1999 - computer.org
An **object recognition** system has been developed that uses a new class of **local** image **features**. The **features** are invariant to image scaling, translation, and rotation, and partially in- variant to illumination changes and affine or 3D projection. These **features** share ...
Cited by 3131 - Related articles - All 96 versions

Distinctive image **features** from **scale-invariant** keypoints
DG Lowe - International journal of computer vision, 2004 - Springer
... has been named the **Scale Invariant** Fea- ture Transform (SIFT), as it transforms image data into **scale-invariant** coordinates relative to **local features**. ... The quantity of **features** is particularly important for **object recognition**, where the ability to detect small objects in ...
Cited by 9470 - Related articles - BL Direct - All 207 versions

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute gradient orientation for each pixel
- Create histogram over edge orientations weighted by magnitude

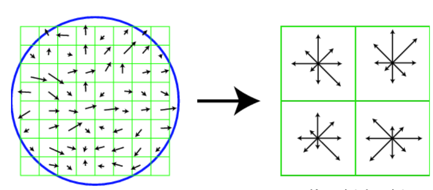


Adapted from slide by David Lowe

SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor



Adapted from slide by David Lowe

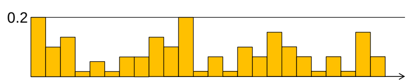
SIFT descriptor

Full version

- Divide the 16x16 window into a 4x4 grid of cells (2x2 case shown below)
- Compute an orientation histogram for each cell
- 16 cells * 8 orientations = 128 dimensional descriptor
- Threshold normalize the descriptor:

$$\sum_i d_i^2 = 1 \quad \text{such that:} \quad d_i < 0.2$$

Why?

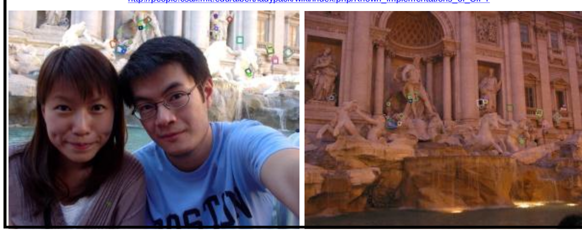


Adapted from slide by David Lowe

Properties of SIFT

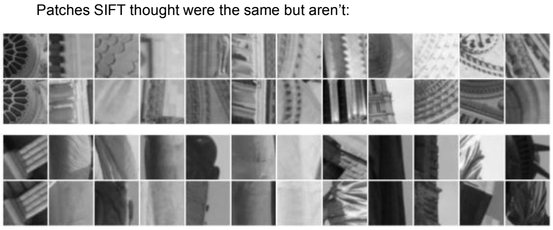
Extraordinarily robust matching technique

- Can handle changes in viewpoint
  - Up to about 30 degree 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



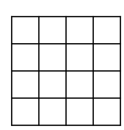
When does SIFT fail?

Patches SIFT thought were the same but aren't:

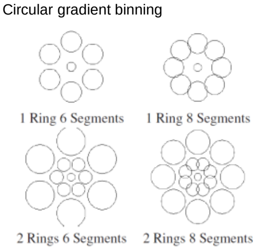


Other methods: Daisy

Circular gradient binning



SIFT



Daisy

Picking the best DAISY, S. Winder, G. Hua, M. Brown, CVPR 09

Other methods: SURF

For computational efficiency only compute gradient histogram with 4 bins:

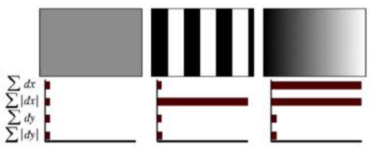


Fig. 3. The descriptor entries of a sub-region represent the nature of the underlying intensity pattern. Left: In case of a homogeneous region, all values are relatively low. Middle: In presence of frequencies in x direction, the value of $\sum |d_x|$ is high, but all others remain low. If the intensity is gradually increasing in x direction, both values $\sum d_x$ and $\sum |d_x|$ are high.

SURF: Speeded Up Robust Features
Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, ECCV 2006

Other methods: BRIEF

Randomly sample pair of pixels a and b.
1 if a > b, else 0. Store binary vector.

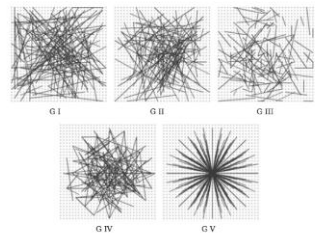


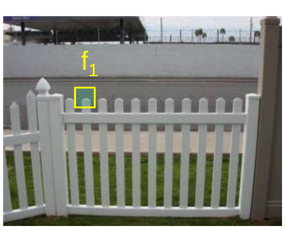
Fig. 2. Different approaches to choosing the test locations. All except the righmost one are selected by random sampling. Showing 128 tests in every image.

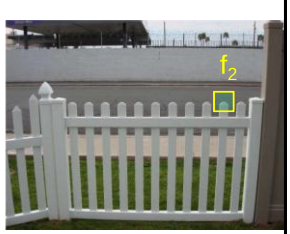
BRIEF: binary robust independent elementary features, Calonder, V Lepetit, C Strecha, ECCV 2010

Feature distance

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

- Simple approach is SSD(f_1, f_2)
  - sum of square differences between entries of the two descriptors
  - can give good scores to very ambiguous (bad) matches



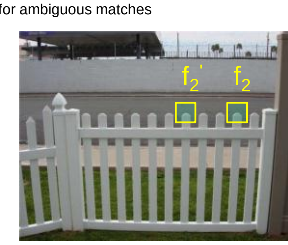


I_1 I_2

Feature distance

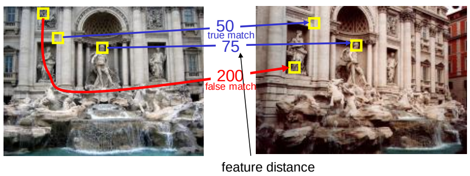
How to define the difference between two features f_1, f_2?

- Better approach: ratio distance = SSD(f_1, f_2) / SSD(f_1, f_2')
  - f_2 is best SSD match to f_1 in I_2
  - f_2' is 2[nd] best SSD match to f_1 in I_2
  - gives large values (~1) for ambiguous matches



I_1 I_2

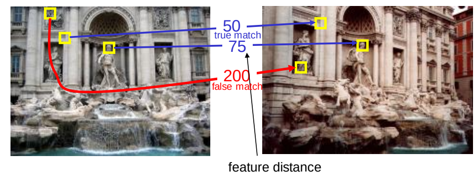
Eliminating bad matches



Throw out features with distance > threshold

- How to choose the threshold?

True/false positives

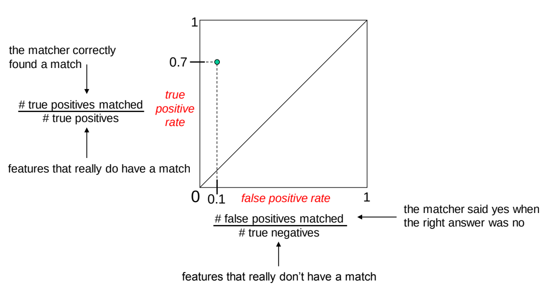


The distance threshold affects performance

- 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?

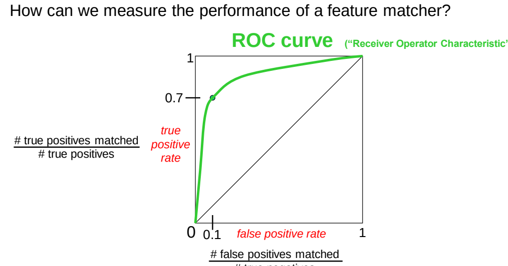
Evaluating the results

How can we measure the performance of a feature matcher?



Evaluating the results

How can we measure the performance of a feature matcher?



ROC Curves

- Generated by counting # current/incorrect matches, for different threholds
- Want to maximize area under the curve (AUC)
- Useful for comparing different feature matching methods
- For more info: http://en.wikipedia.org/wiki/Receiver_operating_characteristic

Some actual ROC curves