

TU
Graz

Image Features

Readings:


Szeliski: Chapter 4.1

Some Slides adapted from Univ. of Washington


Horst Bischof Computer Vision II SS11

TU
Graz

Image matching/Correspondence problem



by [Diva Sian](#)



by [swashford](#)

Horst Bischof Computer Vision II SS11

TU
Graz

Harder case



by [Diva Sian](#)



by [scqbt](#)

Horst Bischof Computer Vision II SS11

Even harder case

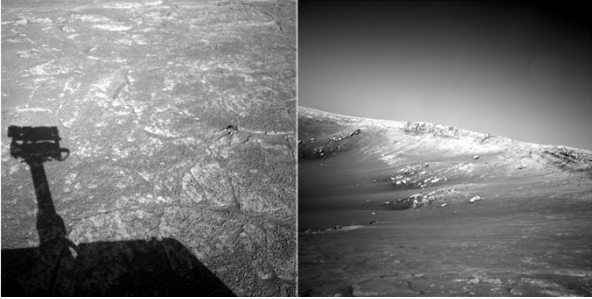


"How the Afghan Girl was Identified by Her Iris Patterns" Read the [story](#)



Horst Bischof | SS11

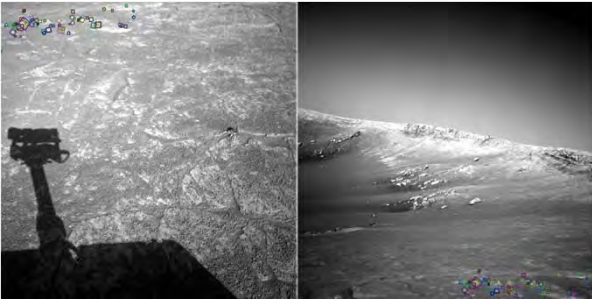
Harder still?



NASA Mars Rover images

Horst Bischof | Computer Vision II SS11

Answer below (look for tiny colored squares...)



NASA Mars Rover images with SIFT feature matches
Figure by Noah Snavely

Horst Bischof | Computer Vision II SS11

TU
Graz

Image Matching

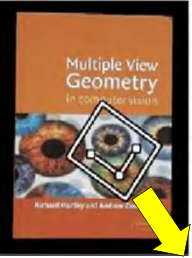



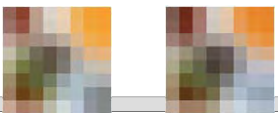

Horst BischofComputer Vision II SS11

7

TU
Graz

Image Matching



Horst BischofComputer Vision II SS11


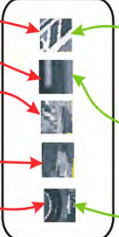
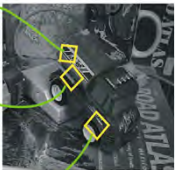
8

TU
Graz

Invariant local features

Find features that are invariant to transformations

- geometric invariance: translation, rotation, scale
- photometric invariance: brightness, exposure, ...

Feature Descriptors

Horst BischofComputer Vision II SS11

9

TU
Graz

Advantages of local features

Locality

- features are local, so robust to occlusion and clutter

Distinctiveness:

- can differentiate a large database of objects

Quantity

- hundreds or thousands in a single image

Efficiency

- real-time performance achievable

Generality

- exploit different types of features in different situations

Horst Bischof Computer Vision II SS11

TU
Graz

More motivation...


Feature points are used for:

- Image alignment (e.g., mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- ... other

Horst Bischof Computer Vision II SS11

TU
Graz

What makes a good feature?



Horst Bischof Computer Vision II SS11

TU
Graz

Want uniqueness

Look for image regions that are unusual

- Lead to unambiguous matches in other images

How to define "unusual"?

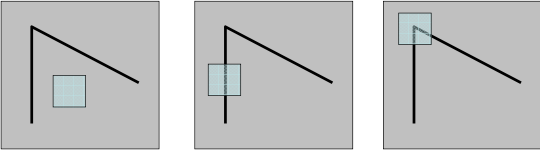
Horst Bischof Computer Vision II SS11 13

TU
Graz

Local measures of uniqueness

Suppose we only consider a small window of pixels

- What defines whether a feature is a good or bad candidate?



Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

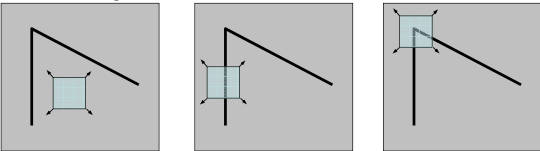
Horst Bischof Computer Vision II SS11 14

TU
Graz

Feature detection

Local measure of feature uniqueness

- How does the window change when you shift it?
- Shifting the window in *any direction* causes a *big change*



"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

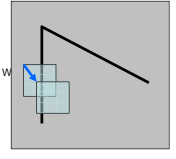
Horst Bischof Computer Vision II SS11 15

TU
Graz

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences (SSD)
- this defines an SSD "error" of $E(u,v)$:



$$E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2$$

Horst Bischof Computer Vision II SS11

TU
Graz

Small motion assumption

Taylor Series expansion of I :

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{higher order terms}$$

If the motion (u,v) is small, then first order approx is good

$$I(x + u, y + v) \approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

$$\approx I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix}$$

shorthand: $I_x = \frac{\partial I}{\partial x}$

Plugging this into the formula on the previous slide...

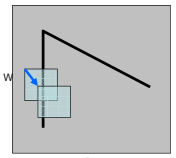
Horst Bischof Computer Vision II SS11

TU
Graz

Feature detection: the math

Consider shifting the window W by (u,v)

- how do the pixels in W change?
- compare each pixel before and after by summing up the squared differences
- this defines an "error" of $E(u,v)$:



$$E(u, v) = \sum_{(x,y) \in W} [I(x + u, y + v) - I(x, y)]^2$$

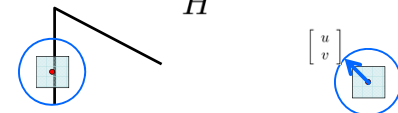
$$\approx \sum_{(x,y) \in W} [I(x, y) + [I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} - I(x, y)]^2$$

$$\approx \sum_{(x,y) \in W} \left[[I_x \ I_y] \begin{bmatrix} u \\ v \end{bmatrix} \right]^2$$

Hic S11

Feature detection: the math

This can be rewritten:

$$E(u, v) = \sum_{(x,y) \in W} [u \ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_H \begin{bmatrix} u \\ v \end{bmatrix}$$


For the example above

- You can move the center of the green window to anywhere on the blue unit circle
- Which directions will result in the largest and smallest E values?
- We can find these directions by looking at the eigenvectors of H

Horst Bischof Computer Vision II SS11 19

Quick eigenvalue/eigenvector review

The **eigenvectors** of a matrix **A** are the vectors **x** that satisfy:

$$Ax = \lambda x$$

The scalar λ is the **eigenvalue** corresponding to **x**

- The eigenvalues are found by solving:

$$\det(A - \lambda I) = 0$$
- In our case, $A = H$ is a 2×2 matrix, so we have

$$\det \begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} = 0$$
- The solution: $\lambda_{\pm} = \frac{1}{2} \left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2} \right]$

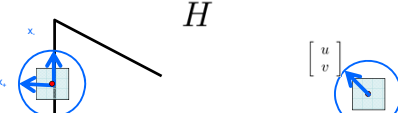
Once you know λ , you find **x** by solving

$$\begin{bmatrix} h_{11} - \lambda & h_{12} \\ h_{21} & h_{22} - \lambda \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 0$$

Horst Bischof Computer Vision II SS11 20

Feature detection: the math

This can be rewritten:

$$E(u, v) = \sum_{(x,y) \in W} [u \ v] \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_H \begin{bmatrix} u \\ v \end{bmatrix}$$



Eigenvalues and eigenvectors of H

- Define shifts with the smallest and largest change (E value)
- x_+ = direction of largest increase in E.
- λ_+ = amount of increase in direction x_+
- x_- = direction of smallest increase in E.
- λ_- = amount of increase in direction x_-

$$Hx_+ = \lambda_+ x_+$$

$$Hx_- = \lambda_- x_-$$

Horst Bischof Computer Vision II SS11 21




Feature detection: the math

How are λ_+ , x_+ , λ_- , and x_- relevant for feature detection?

- What's our feature scoring function?

Horst Bischof Computer Vision II SS11
22



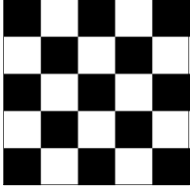
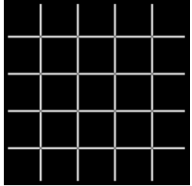
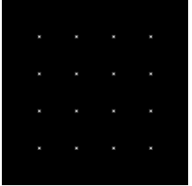
Feature detection: the math

How are λ_+ , x_+ , λ_- , and x_- relevant for feature detection?


- What's our feature scoring function?

Want $E(u,v)$ to be *large* for small shifts in *all* directions

- the *minimum* of $E(u,v)$ should be large, over all unit vectors $[u \ v]$
- this minimum is given by the smaller eigenvalue (λ_-) of H

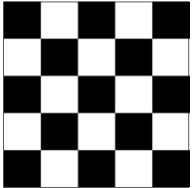
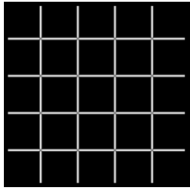
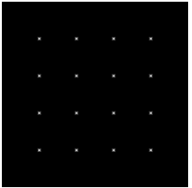
Horst Bischof I λ_+ Cof λ_- sion II SS11
23



Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the H matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response ($\lambda_- > \text{threshold}$)
- Choose those points where λ_- is a local maximum as features

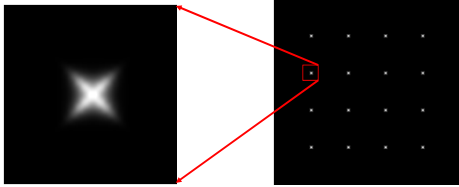




Horst Bischof I λ_+ Cof λ_- sion II SS11
24

Feature detection summary

Here's what you do

- Compute the gradient at each point in the image
- Create the H matrix from the entries in the gradient
- Compute the eigenvalues.
- Find points with large response ($\lambda_i > \text{threshold}$)
- Choose those points where λ_i is a local maximum as features



Horst Bischof Computer Vision II SS11 25

The Harris operator

λ_i is a variant of the "Harris operator" for feature detection

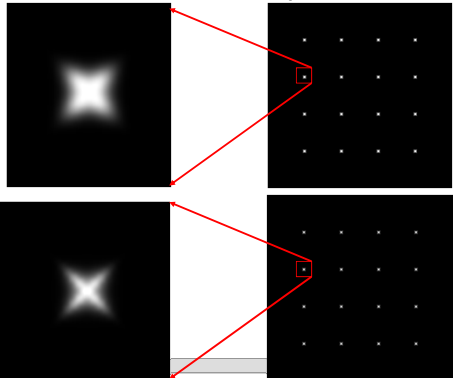
$$f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$$

$$= \frac{\text{determinant}(H)}{\text{trace}(H)}$$

- The *trace* is the sum of the diagonals, i.e., $\text{trace}(H) = h_{11} + h_{22}$
- Very similar to λ_i , but less expensive (no square root)
- Called the "Harris Corner Detector" or "Harris Operator"
- Lots of other detectors, this is one of the most popular

Horst Bischof Computer Vision II SS11 26

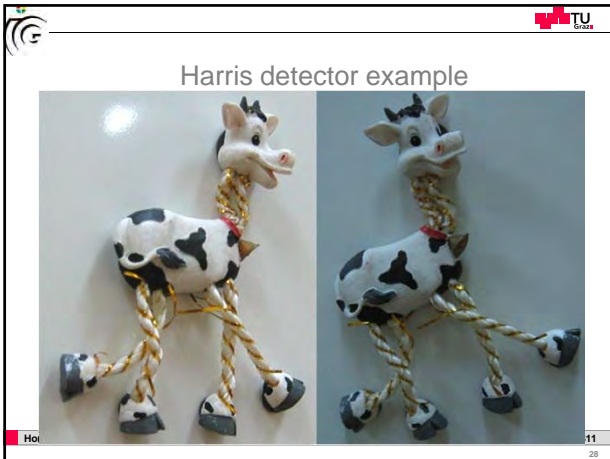
The Harris operator

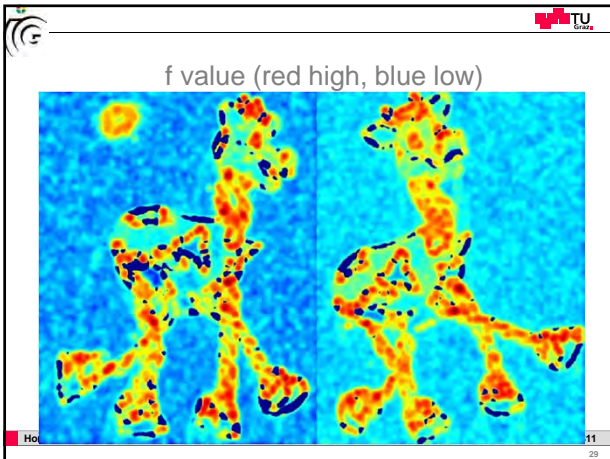


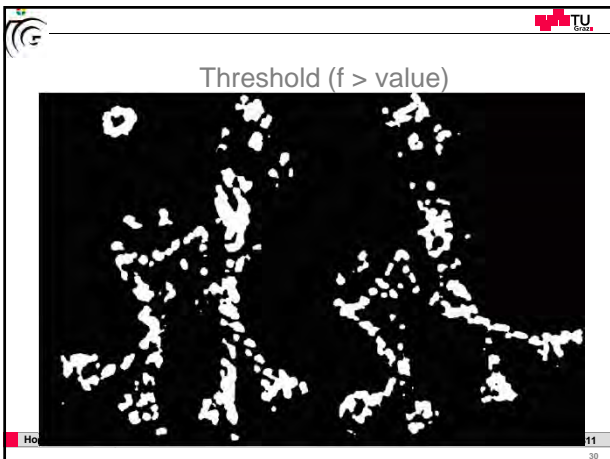
Harris operator

λ_i

Horst Bischof Computer Vision II SS11 27







Find local maxima of f

Horst TU 11 31

Harris features (in red)

Horst TU 11 32

Invariance

Suppose you **rotate** the image by some angle
 – Will you still pick up the same features?

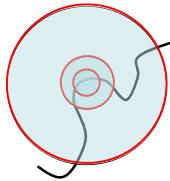
What if you change the brightness?

Scale?

Horst Bischof Computer Vision II SS11 33

Scale invariant detection

Suppose you're looking for corners




Key idea: find scale that gives local maximum of f

- f is a local maximum in both position and scale
- Common definition of f : Laplacian
(or difference between two Gaussian filtered images with different sigmas)

Horst Bischof Computer Vision II SS11


Automatic scale selection

Lindeberg et al., 1996

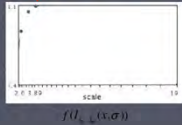


Slide from Tinne Tuytelaars

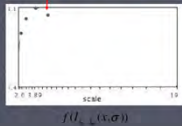
Automatic scale selection



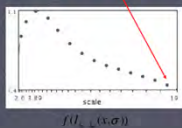
Automatic scale selection



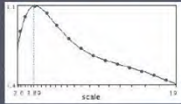
Automatic scale selection



Automatic scale selection

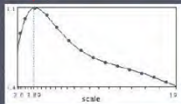


Automatic scale selection

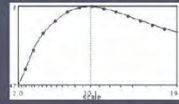


$f(D_{x, \sigma}(x, \sigma))$

Automatic scale selection



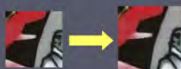
$f(D_{x, \sigma}(x, \sigma))$



$f(D_{x, \sigma'}(x, \sigma'))$

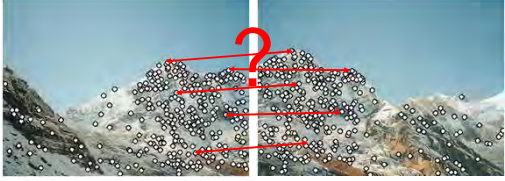
Automatic scale selection

Normalize: rescale to fixed size



Feature descriptors

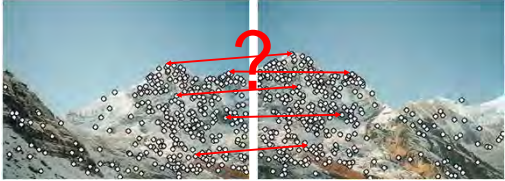
We know how to detect good points
 Next question: **How to match them?**



Horst Bischof Computer Vision II SS11 43

Feature descriptors

We know how to detect good points
 Next question: **How to match them?**



Lots of possibilities (this is a popular research area)

- Simple option: match square windows around the point
- State of the art approach: SIFT
 - David Lowe, UBC <http://www.cs.ubc.ca/~lowe/keypoints/>

Horst Bischof Computer Vision II SS11 44

Invariance

Suppose we are comparing two images I_1 and I_2

- I_2 may be a transformed version of I_1
- What kinds of transformations are we likely to encounter in practice?

Horst Bischof Computer Vision II SS11 45

TU
Graz

Invariance

Suppose we are comparing two images I_1 and I_2

- I_2 may be a transformed version of I_1
- What kinds of transformations are we likely to encounter in practice?

We'd like to find the same features regardless of the transformation

- This is called transformational ***invariance***
- Most feature methods are designed to be invariant to
 - Translation, 2D rotation, scale
- They can usually also handle
 - Limited 3D rotations (SIFT works up to about 60 degrees)
 - Limited affine transformations (some are fully affine invariant)
 - Limited illumination/contrast changes

Horst Bischof Computer Vision II SS11

TU
Graz

How to achieve invariance

Need both of the following:

1. Make sure your detector is invariant
 - Harris is invariant to translation and rotation
 - Scale is trickier
 - common approach is to detect features at many scales using a Gaussian pyramid (e.g., MOPS)
 - More sophisticated methods find "the best scale" to represent each feature (e.g., SIFT)
2. Design an invariant feature *descriptor*
 - A descriptor captures the information in a region around the detected feature point
 - The simplest descriptor: a square window of pixels
 - What's this invariant to?
 - Let's look at some better approaches...


Horst Bischof Computer Vision II SS11

TU
Graz

Rotation invariance for feature descriptors

Find dominant orientation of the image patch

- This is given by \mathbf{x}_+ , the eigenvector of \mathbf{H} corresponding to λ_+
 - λ_+ is the *larger* eigenvalue
- Rotate the patch according to this angle

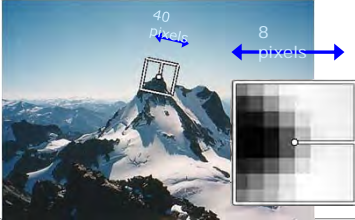


Horst Bischof Computer Vision II SS11

Multiscale Oriented PatcheS descriptor

Take 40x40 square window around detected feature

- Scale to 1/5 size (using prefiltering)
- Rotate to horizontal
- Sample 8x8 square window centered at feature
- Intensity normalize the window by subtracting the mean, dividing by the standard deviation in the window



Horst Bischof Computer Vision II SS11

Detections at multiple scales

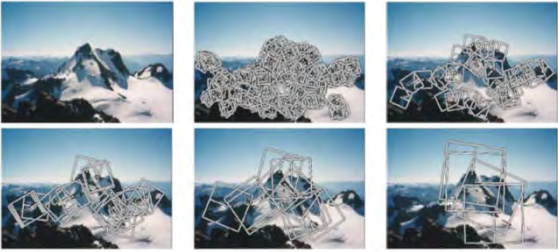


Figure 1. Multi-scale Oriented Patches (MOPS) extracted at five pyramid levels from one of the Matter images. The boxes show the feature orientation and the region from which the descriptor vector is sampled.

Horst Bischof Computer Vision II SS11

Scale Invariant Feature Transform

Basic idea:

- Take 16x16 square window around detected feature
- Compute edge orientation (angle of the gradient - 90°) for each pixel
- Throw out weak edges (threshold gradient magnitude)
- Create histogram of surviving edge orientations

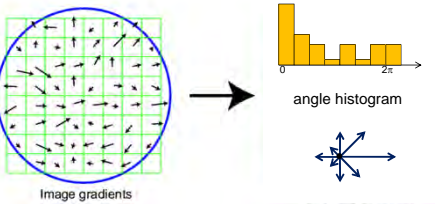


Image gradients → angle histogram

Adapted from slide by David Lowe

Horst Bischof Computer Vision II SS11

TU
Graz

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


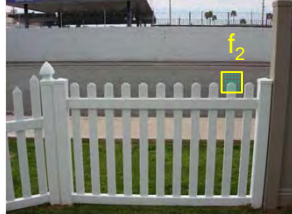
Horst Bischof Computer Vision II SS11

TU
Graz

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

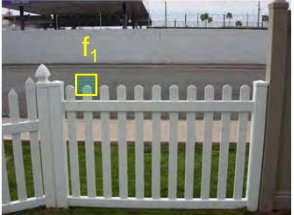
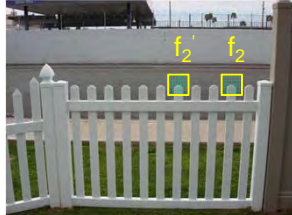
Horst Bischof Computer Vision II SS11

TU
Graz

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 2nd best SSD match to f_1 in I_2
 - gives small values for ambiguous matches

Horst Bischof Computer Vision II SS11

TU
Graz

Evaluating the results

How can we measure the performance of a feature matcher?

feature distance

Horst Bischof Computer Vision II SS11 58

TU
Graz

True/false positives

feature distance

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?

Horst Bischof Computer Vision II SS11 59

TU
Graz

Evaluating the results

How can we measure the performance of a feature matcher?

true positives
matching features (positives) true positive rate

false positives
unmatched features (negatives) false positive rate

Horst Bischof Computer Vision II SS11 60

TU
Graz

Evaluating the results

How can we measure the performance of a feature matcher?

ROC curve ("Receiver Operator Characteristic")

true positives / # matching features (positives) = true positive rate

false positives / # unmatched features (negatives) = false positive rate

ROC Curves

- Generated by counting # current/incorrect matches, for different thresholds
- 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

Horst Bischof Computer Vision II SS11

TU
Graz

More on feature detection/description

Affine Covariant Regions

Publications

Region detectors

- Horst-Affine & Hessian-Affine: K. Mikolajczyk and C. Schmid. Scale and Affine invariant interest point detectors. In *ICV* 1450-1458, 2004. [PDF](#)
- MSER: J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC* p. 384-393, 2002. [PDF](#)
- ISF & SRR: T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. In *ICV* 1 (59) 61-85, 2004. [PDF](#)
- Salient regions: T. Radle, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *ECCV* p. 404-416, 2004. [PDF](#)

Region descriptors

- SIFT: D. Lowe. Distinctive image features from scale invariant keypoints. In *ICCV* 2(69) 91-110, 2004. [PDF](#)

Performance evaluation

- K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, T. Matas, F. Schaffalitzky, T. Felsi and L. Van Gool. A comparison of affine region detectors. Technical Report, accepted to *ICV*. [PDF](#)
- K. Mikolajczyk, C. Schmid. A performance evaluation of local descriptors. Technical Report, accepted to *PAMI*. [PDF](#)

Horst Bischof Computer Vision II SS11

TU
Graz

Lots of applications

Features are used for:

- Image alignment (e.g., mosaics)
- 3D reconstruction
- Motion tracking
- Object recognition
- Indexing and database retrieval
- Robot navigation
- ... other

Horst Bischof Computer Vision II SS11
