Image Matching

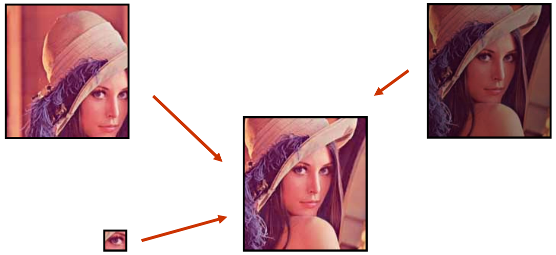
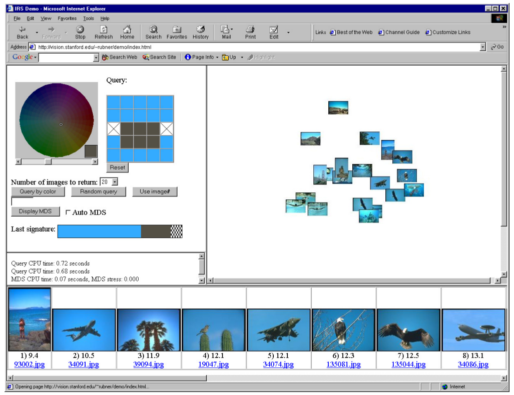


Image Retrieval



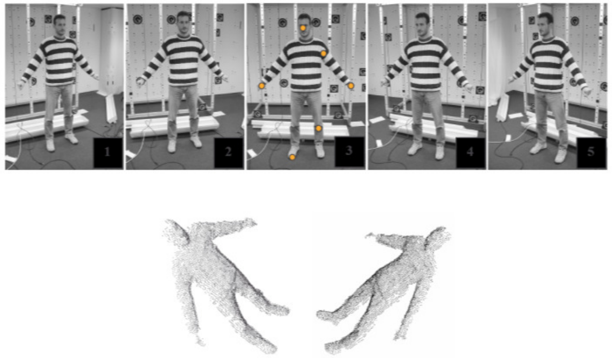
Object Recognition

Motion Estimation and Optical Flow Tracking

Example: Mosiacing (Panorama)

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Example – 3D Reconstruction



Source: http://www.photogrammetry.ethz.ch/general/persons/fabio/fabio_spie0102.pdf

Image Matching

Three approaches:

- **Shape Matching**
  - Assume shape has been extracted
- **Direct (appearance-based) registration**
  - Search for alignment where most pixels agree
- **Feature-based registration**
  - Find a few matching features in both images
  - compute alignment

Direct Method (brute force)

The simplest approach is a brute force search

- Need to define image distance function:
  SSD, Normalized Correlation, Mutual Information, etc.
- Search over all parameters within a reasonable range:

  e.g. for translation:

```
for Δx=x0:step:x1,
  for Δy=y0:step:y1,
    calculate Dist(image1(x,y),image2(x+Δx,y+Δy))
  end;
end;
```

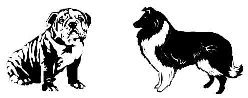
Shape Representation

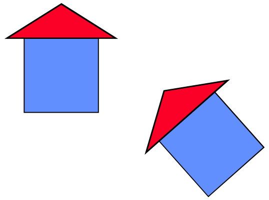
- Region Based Representation
- Area / Circumference / Width
- Euler Number

- Moments
- Quad Trees

- Edge Based Representation
- Chain Code
- Fourier Descriptor

- Interior Based Representation
- MAT / Skeleton
- Hierarchical Representations

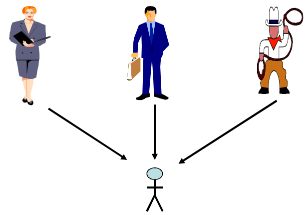




Shape Representation

Shape representation must be GOOD:

- Different shapes ⇔ Different Codes
- Location / Rotation /Scale Invariant
- Convenient
- Stable
- Generative

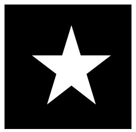


Moments

$$I(x,y) = \begin{cases} 1 & \text{If pixel (x,y) is IN object} \\ 0 & \text{otherwise} \end{cases}$$

ij –Moment: $$M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$$

Area: $$M_{00} = \sum_x \sum_y I(x,y)$$



Average x-coordinate: $$\bar{x} = \frac{M_{10}}{M_{00}}$$

Average y-coordinate: $$\bar{y} = \frac{M_{01}}{M_{00}}$$

Center of Mass: $$(\bar{x},\bar{y}) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right)$$

Moments

Central Moment: $$\mu_{ij} = \sum_x \sum_y (x-\bar{x})^i (y-\bar{y})^j I(x,y)$$

Moment expressions that are invariant to translation, rotation and/or scale:

1. For first-order moments, $\mu_{0,1} = \mu_{1,0} = 0$, (always invariant).
2. For second-order moments, $(p + q = 2)$, the invariants are

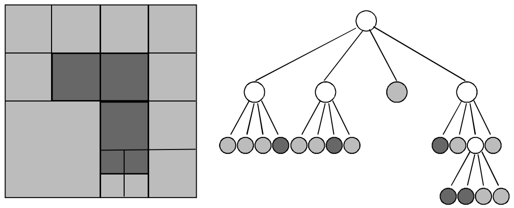
$$\phi_1 = \mu_{2,0} + \mu_{0,2}$$
$$\phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \quad (9.80)$$

3. For third-order moments $(p + q = 3)$, the invariants are

$$\phi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2$$
$$\phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2$$

wide domain, not unique, not unambiguous, not generative, not stable, invariant to translation, rotation. Very convenient.

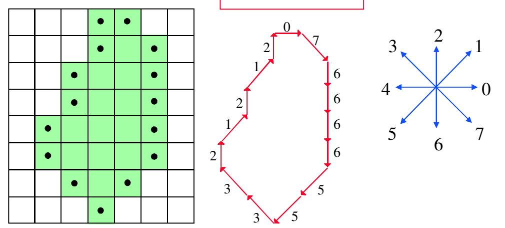
Quad Tree Representation



wide domain,
unique, unambiguous, generative – up to error tolerance
partially stable
Not invariant to translation, rotation scale.
Inefficient for comparison

Edge Based Representation

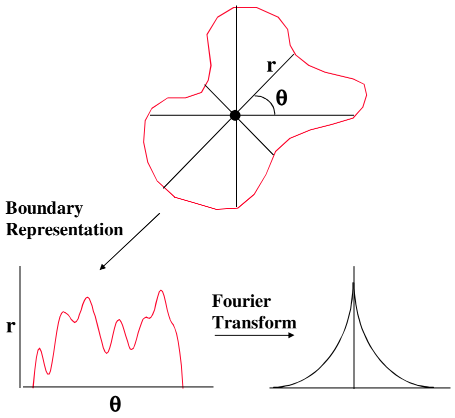
Chain Code

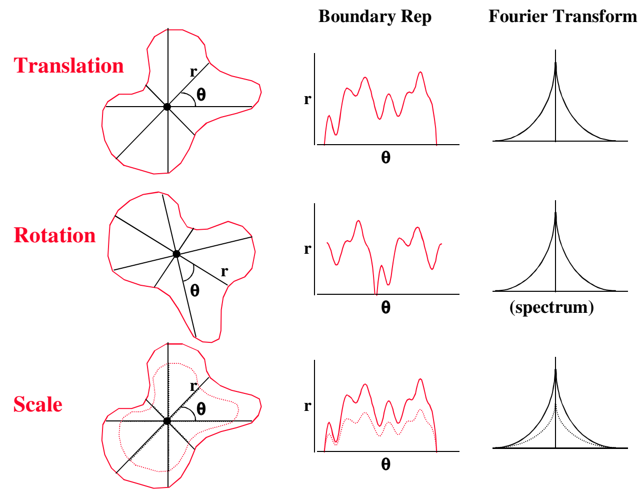


000102011717211

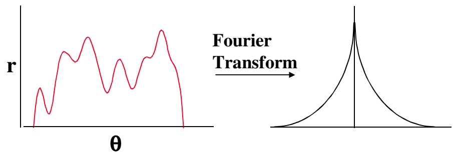
wide domain, Unique, unambiguous, generative - 2D only,
Not very stable Invariant to translation. Rotation (x90 deg)

Fourier Descriptors



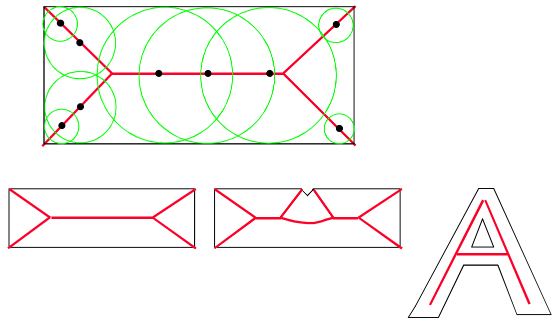


Fourier Descriptors



wide domain, Unique, unambiguous, generative, Stable (depends on tolerance), Invariant to translation. Rotation, Scale.

Interior Based representation – MAT, Skeleton



wide domain, unique, unambiguous, generative
not stable - small changes affect dramatically

Pattern Matching – Direct approach (Appearance based)

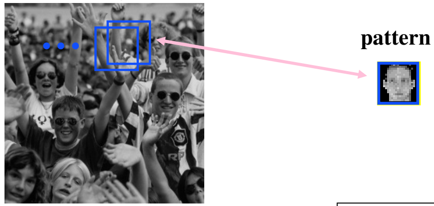
image

pattern

Finding a pattern in an Image

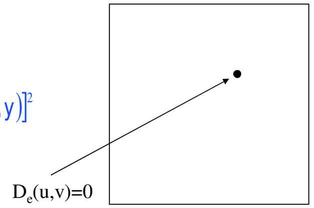
image

pattern



Look for minimum of:

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



Finding a pattern in an Image

$$d_e(u,v)=\sum_{x,y\in N}\left[I(u+x,v+y)-P(x,y)\right]^2$$

$$=\sum_{x,y\in N} I(u+x,v+y)^2+P(x,y)^2-2I(u+x,v+y)P(x,y)$$

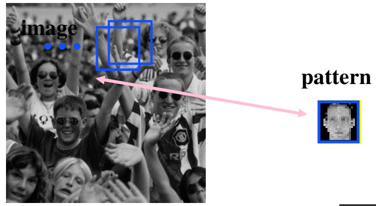
$$=\sum_{x,y\in N} I(u+x,v+y)^2+\sum_{x,y\in N} P(x,y)^2-2\sum_{x,y\in N} I(u+x,v+y)P(x,y)$$

Sum of squares of the window

Sum of squares of the pattern CONSTANT

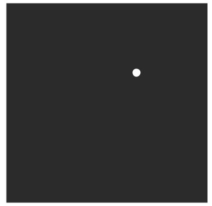
Correlation

Finding a pattern in an Image - Correlation

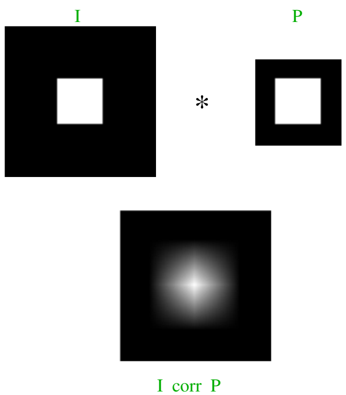


Look for maximum of:

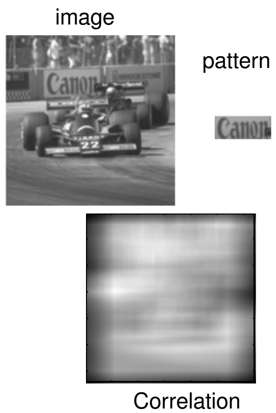
$$\sum_{x,y\in N}\left[I(u+x,v+y)P(x,y)\right]$$



Correlation



Real Image – Correlation Example



Correlation value is dependent on the local gray value of the pattern and the image window.

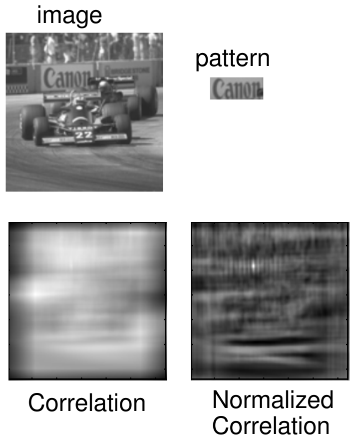
Normalized Correlation

$$\frac{\sum_{x,y\in N}\left[I(u+x,v+y)-\bar{I}_{uv}\right]\left[P(x,y)-\bar{P}\right]}{\left[\sum_{x,y\in N}\left[I(u+x,v+y)-\bar{I}_{uv}\right]^2\sum_{x,y\in N}\left[P(x,y)-\bar{P}\right]^2\right]^{1/2}}$$

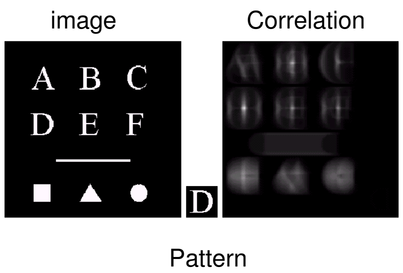
Correlation value is in (-1..1)

Correlation value is **independent** of the local gray value of the pattern and the image window.

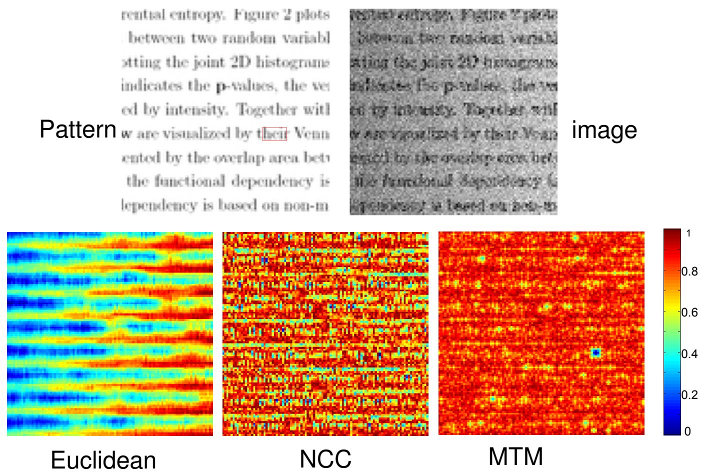
Normalized Correlation - Example



Normalized Correlation - Example

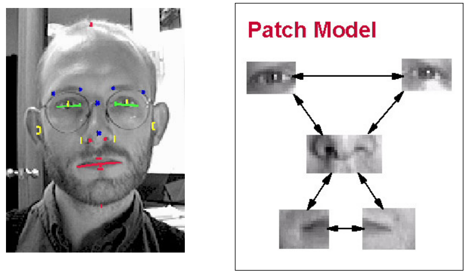


Pattern Matching - Example

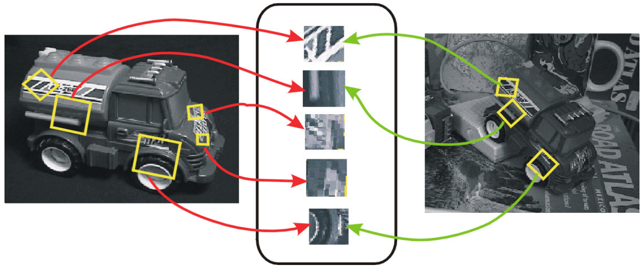


Pairs for Image Matching

Feature Based Object Detection



Feature Based Object Detection



Features Descriptors

Features: Issues to be addressed

- What are “good” features to extract?
  - –Distinctive
  - –Invariant to different acquisition conditions
    - –Different view-points, different illuminations, different cameras, etc.
- How can we find corresponding features in both images?

no chance to match!

Invariant Feature Descriptors

- Schmid & Mohr 1997, Lowe 1999, Baumberg 2000, Tuytelaars & Van Gool 2000, Mikolajczyk & Schmid 2001, Brown & Lowe 2002, Matas et. al. 2002, Schaffalitzky & Zisserman 2002

Image Features

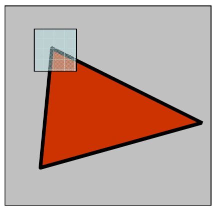
- Feature Detectors - where
- Feature Descriptors - what

- Methods:
  - Harris Corner Detector (multi-scale Harris)
  - SIFT (Scale Invariant Features Transform)

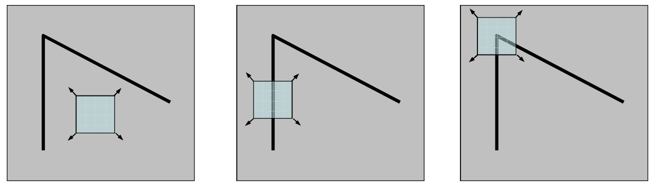
Harris Corner Detector

C.Harris, M.Stephens. “A Combined Corner and Edge Detector”. 1988

- We should easily recognize a corner by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



Harris Detector: Basic Idea



“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

Harris Detector: Mathematics

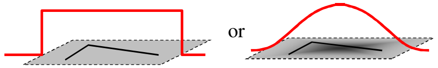
Corner at position (x,y) ?

Evaluate change of intensity for shift in [*u*,*v*] direction:

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

Window function — Shifted intensity — Intensity

Window function $w(x,y)$ =



1 in window, 0 outside — Gaussian

Harris Detector: Mathematics

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

For small [u,v]: $I(x+u, y+v) = I(x,y) + uI_x + vI_y$

We have:

$$E(u,v) = \sum_{x,y} w(x,y)\left\| \begin{bmatrix} I_x(x,y) & I_y(x,y) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \right\|^2 =$$

$$\begin{bmatrix} u & v \end{bmatrix} \sum w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Detector: Mathematics

For small shifts [*u*,*v*] we have a *bilinear* approximation:

$$E(u,v) \cong [u,v] \quad M \begin{bmatrix} u \\ v \end{bmatrix}$$

where *M* is a 2×2 matrix computed from image derivatives:

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Harris Detector: Mathematics

What is the direction [u,v] of greatest intensity change?

$$\arg\max_{\|(u,v)\|=1} E(u,v) = \mathbf{e}_{\max}$$

Denote by $\mathbf{e}_i$ the i[th] eigen-vector of M whose eigen-value is λ_i:

$$\mathbf{e}_i^T M \mathbf{e}_i = \lambda_i > 0$$

Conclusions:

$$E(\mathbf{e}_{\max}) = \lambda_{\max}$$

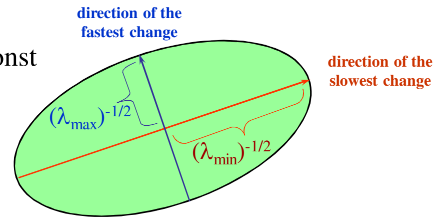
Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

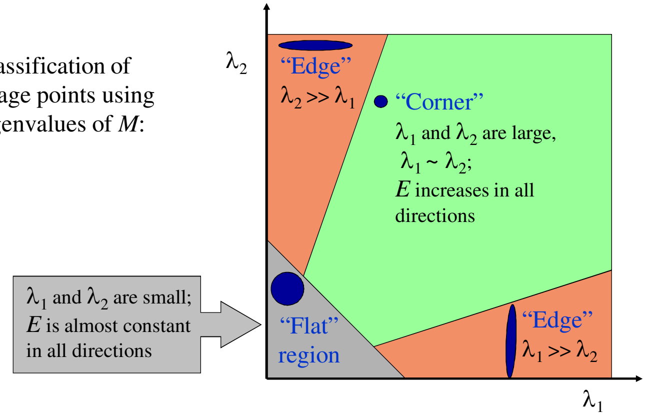
λ_1, λ_2 – eigenvalues of M

Ellipse $E(u,v)$ = const



Harris Detector: Mathematics

Classification of image points using eigenvalues of M:



Harris Detector: Mathematics

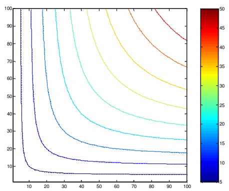
Measure of corner response (without calculating the e.v.):

$$R = \frac{\det M}{\operatorname{Trace} M}$$

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

R is associated with the smallest eigen-vector (why?)

R v.s. λ_1, λ_2



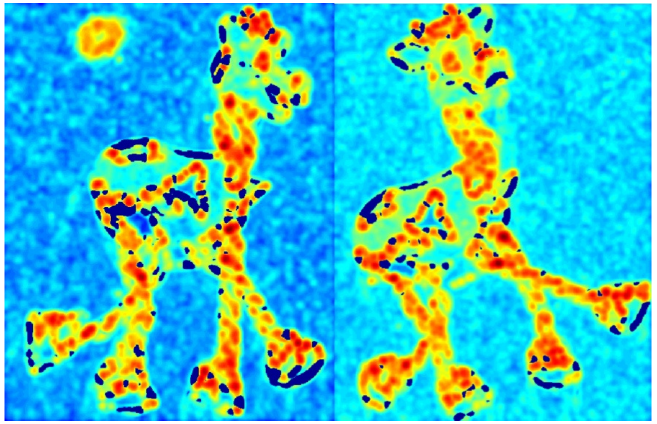
Harris Corner Detector

- The Algorithm:
  - Find points with large corner response function R (R > threshold)
  - Take the points of local maxima of R

Harris Detector: Workflow

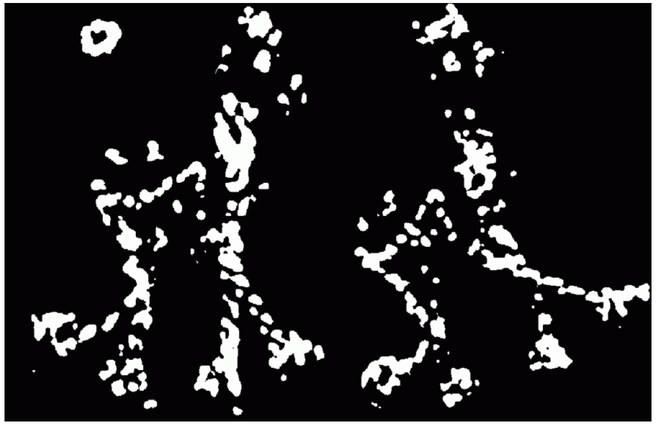
Harris Detector: Workflow

Compute corner response R



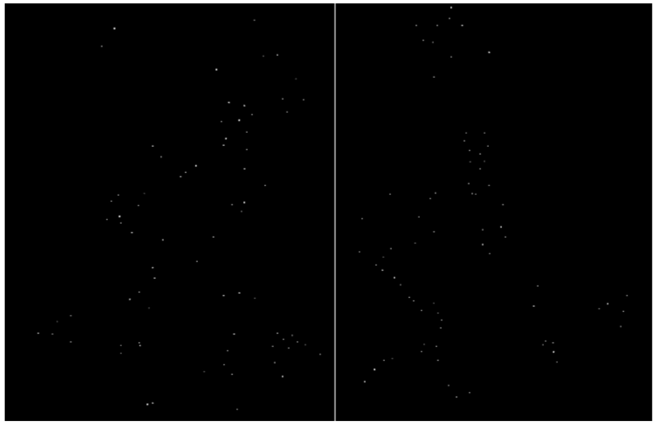
Harris Detector: Workflow

Find points with large corner response: R>threshold



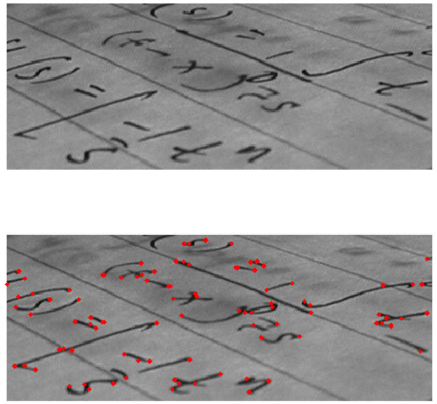
Harris Detector: Workflow

Take only the points of local maxima of R



Harris Detector: Workflow

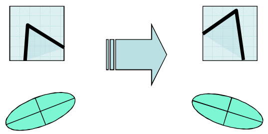
Harris Detector: Example



Harris Detector: Example

Harris Detector: Some Properties

- Rotation invariance



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

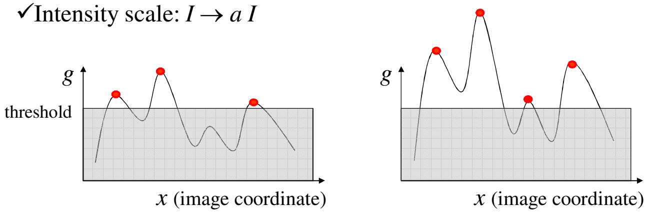
Corner response R is invariant to image rotation

Harris Detector: Some Properties

Partial invariance to affine intensity change

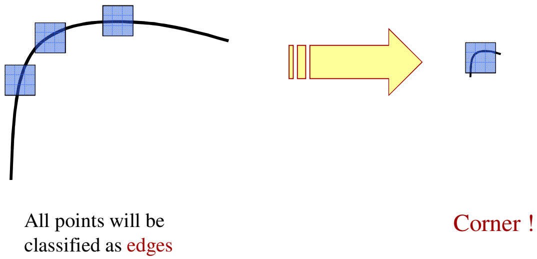
✓Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

✓Intensity scale: $I \rightarrow a\, I$



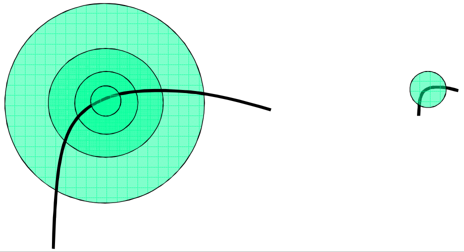
Harris Detector: Some Properties

- But: non-invariant to spatial scale!



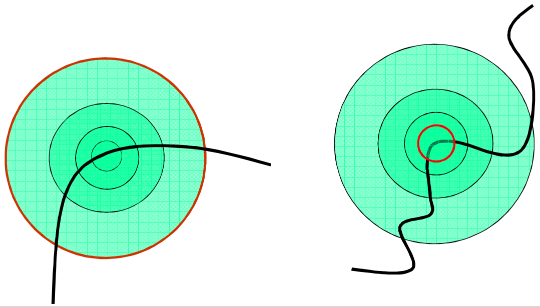
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



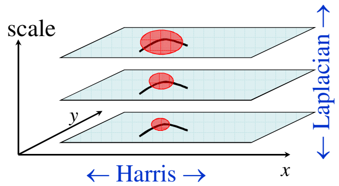
Scale Invariant Detection

- **The problem**: how do we choose corresponding circles *independently* in each image?
- **Solution**: choose the scale of the "best" corner.



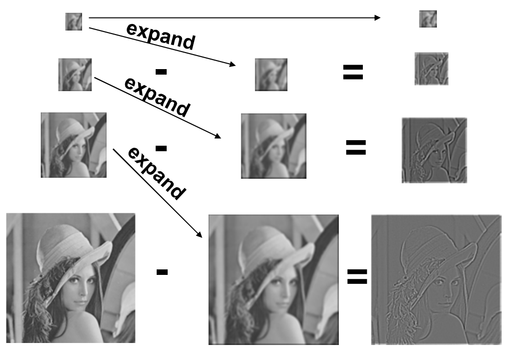
Harris-Laplacian Point Detector

- Harris-Laplacian
  Find local maximum of: Harris corner detector for a set of Laplacian images.



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

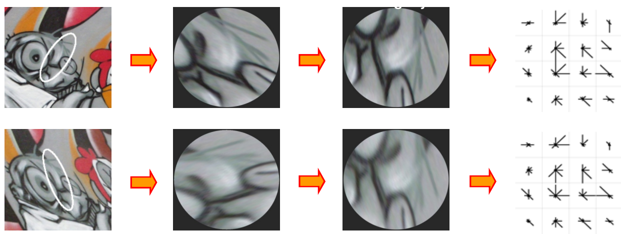
Memo: Gaussian / Laplacian Pyramids



Harris - Laplacian Detector

SIFT – Scale Invariant Feature Transform

David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110



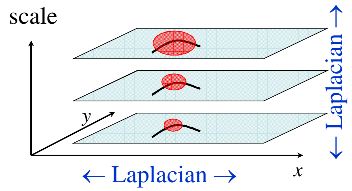
SIFT – Scale Invariant Feature Transform

David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

- *Give about 2000 stable "keypoints" for a typical 500* x *500 image*
- *Each keypoint is described by a vector of*
  4 x 4 x 8 = 128 elements
  (over 4x4 array of 8-bin gradient histograms keypoint neighborhood)

SIFT – Scale Invariant Feature Transform

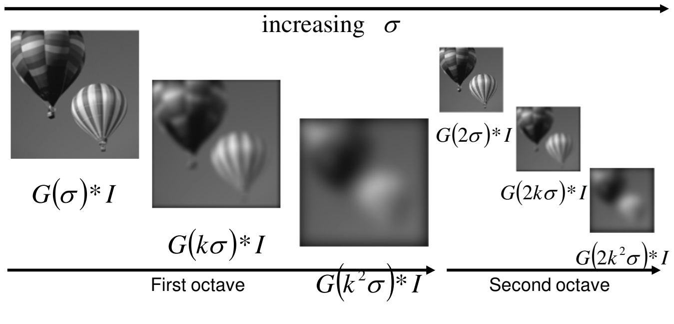
\- *Find local maximum of* Laplacian in space and scale



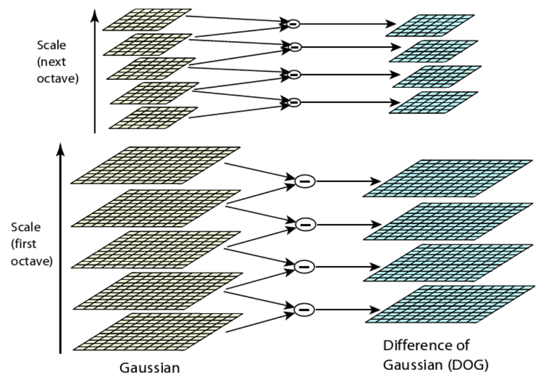
David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

SIFT - Point Detection

- Construct scale-space:



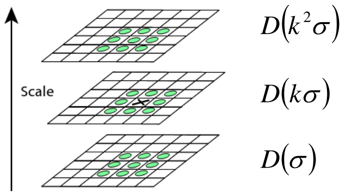
SIFT – Scale Space



SIFT – point detection

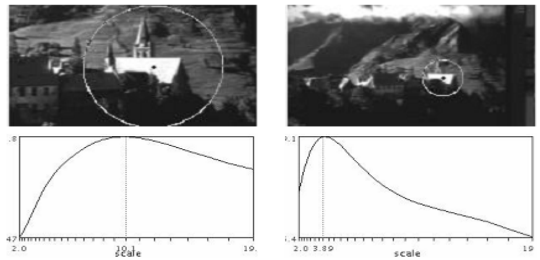
STEP 1:
Determine local Maxima in DoG pyramid (Laplacian Pyramid).

- Scale Space extrema detection.
- Choose all extrema within 3x3x3 neighborhood.



SIFT – point detection

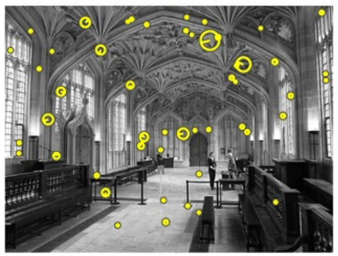
STEP 1:
Determine local Maxima in DoG pyramid (Laplacian Pyramid).



Experimentally, Maximum of Laplacian gives best notion of s

SIFT - Step 1: Interest Point Detection

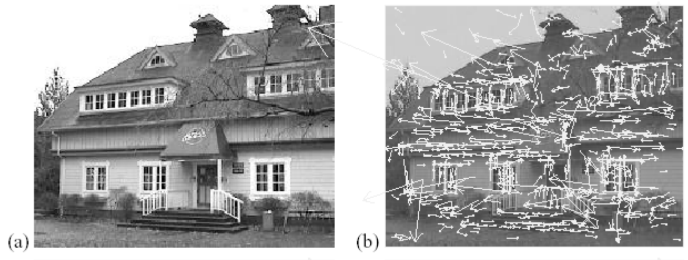
Detections at multiple scales



Some of the detected SIFT frames.

http://www.vlfeat.org/overview/sift.html

SIFT – point detection



233x189 image

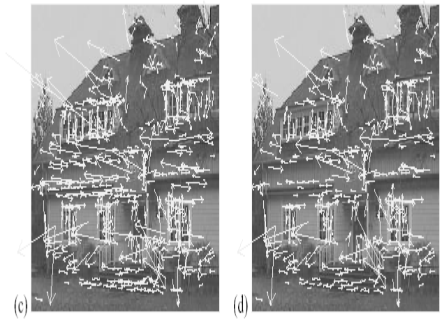
832 SIFT extrema

SIFT - Step 2: Interest Localization & Filtering

2) Remove bad Interest points:
   a) Remove points with low contrast
   b) Remove Edge points (Eigenvalues of Hessian Matrix must BOTH be large).

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Interest Points

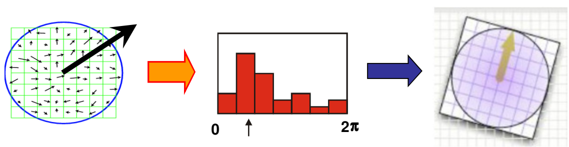


(c) 729 left after peak value threshold (from 832)
(d) 536 left after testing ratio of principle curvatures

SIFT – Descriptor Vector

STEP 3: Select canonical orientation

- Each SIFT interest point is associated with location (x,y) and scale (σ)
- Compute gradient magnitude and orientation for each SIFT point:

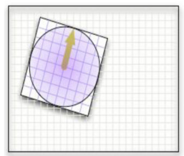


Assign canonical orientation at peak of smoothed histogram (fit parabola to better localize peak).

SIFT – Descriptor Vector

STEP 3: Select canonical orientation

- Each SIFT interest point is associated with location (x,y), scale (σ), gradient magnitude and orientation (m, θ).



- Compute SIFT feature - a vector of 128 entries.

SIFT – Descriptor Vector

STEP 4: Compute SIFT feature vector of 128 entries

- Gradients determined in 16x16 window at SIFT point in scale space.
- Histogram is computed for gradients of each 4x4 sub window in 8 **relative** directions.
- A 4x4x8 = 128 dimensional feature vector is produced.

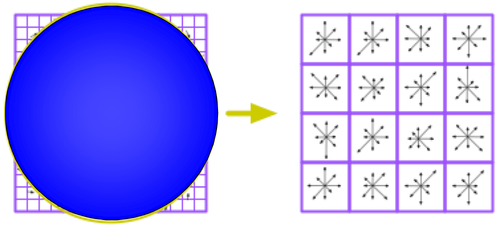
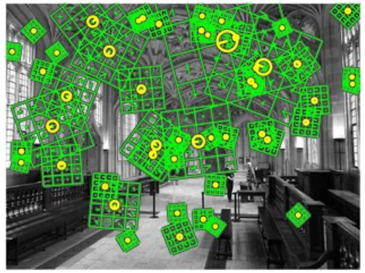


Image gradients   Keypoint descriptor

Image from: Jonas Hurrelmann

SIFT – Descriptor Vector

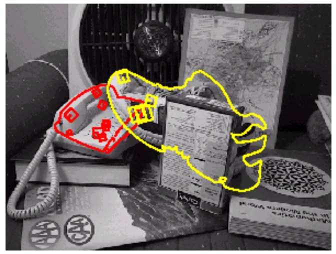
STEP 4: Compute feature vector



Object Recognition

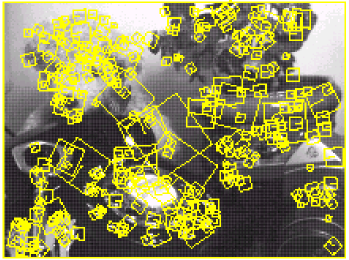
- Only 3 keys are needed for recognition, so extra keys provide robustness

Recognition under occlusion



Test of illumination Robustness

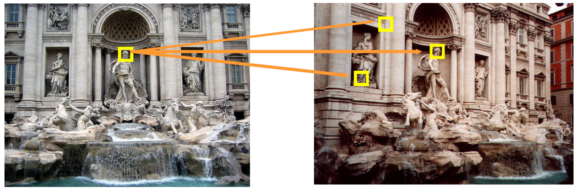
- Same **image** under differing illumination



273 keys verified in final match

Matching SIFT Features

- 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. Accept if below threshold.



I_1 I_2

Matching SIFT Features

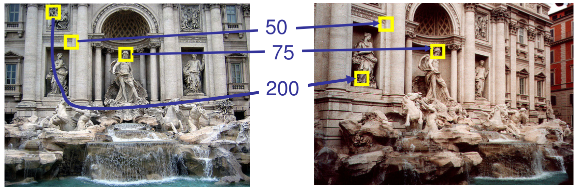
22 correct matches

Matching SIFT Features

33 correct matches

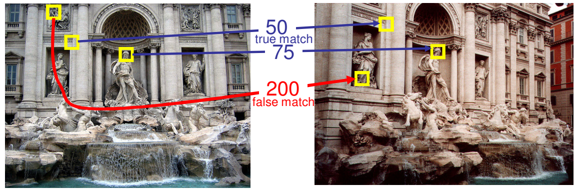
Matching SIFT Features

How to evaluate the performance of a feature matcher?



Matching SIFT Features

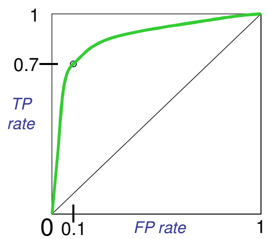
- Threshold t affects # of correct/false matches



- True positives (TP) = # of detected matches that are correct
- False positives (FP) = # of detected matches that are incorrect

Matching SIFT Features

- ROC Curve
  - Generated by computing (FP, TP) for different thresholds.
  - Maximize area under the curve (AUC).

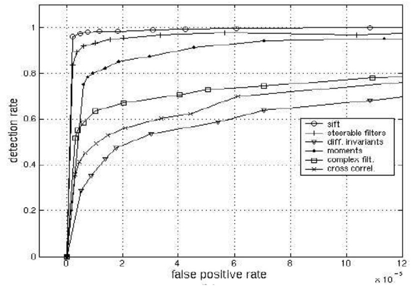


http://en.wikipedia.org/wiki/Receiver_operating_characteristic

Evaluating SIFT Features

- Empirically found[2] to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45^0



[1] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004
[2] K.Mikolajczyk, C.Schmid. "A Performance Evaluation of Local Descriptors". CVPR 2003

Example - Mosaicing

Source: Alexei Efros

Example: Mosiacing (Panorama)

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Image Matching

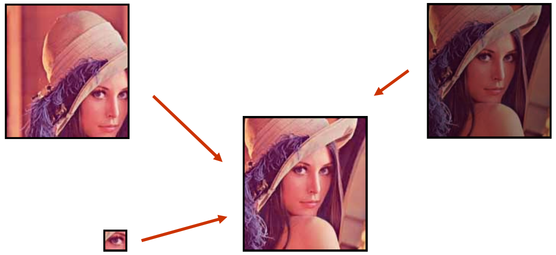
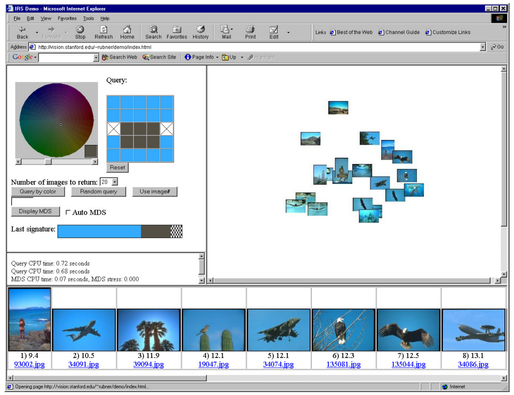


Image Retrieval



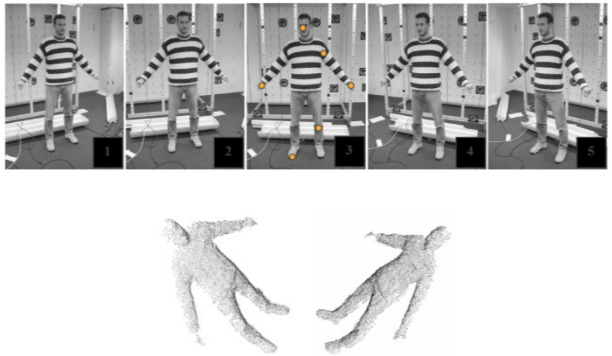
Object Recognition

Motion Estimation and Optical Flow Tracking

Example: Mosiacing (Panorama)

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Example – 3D Reconstruction



Source: http://www.photogrammetry.ethz.ch/general/persons/fabio/fabio_spie0102.pdf

Image Matching

Three approaches:

- **Shape Matching**
  - Assume shape has been extracted
- **Direct (appearance-based) registration**
  - Search for alignment where most pixels agree
- **Feature-based registration**
  - Find a few matching features in both images
  - compute alignment

Direct Method (brute force)

The simplest approach is a brute force search

- Need to define image distance function:
  SSD, Normalized Correlation, Mutual Information, etc.
- Search over all parameters within a reasonable range:

e.g. for translation:

```
for Δx=x0:step:x1,
 for Δy=y0:step:y1,
   calculate Dist(image1(x,y),image2(x+Δx,y+Δy))
 end;
end;
```

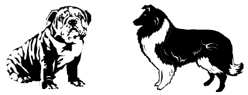
Shape Representation

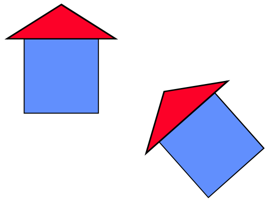
- Region Based Representation
- Area / Circumference / Width
- Euler Number

- Moments
- Quad Trees

- Edge Based Representation
- Chain Code
- Fourier Descriptor

- Interior Based Representation
- MAT / Skeleton
- Hierarchical Representations

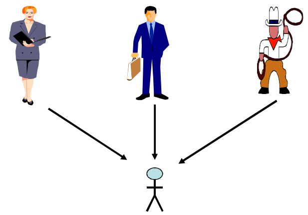




Shape Representation

Shape representation must be GOOD:

- Different shapes ⇔ Different Codes
- Location / Rotation /Scale Invariant
- Convenient
- Stable
- Generative

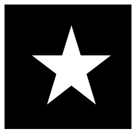


Moments

$$I(x,y) = \begin{cases} 1 & \text{If pixel (x,y) is IN object} \\ 0 & \text{otherwise} \end{cases}$$

ij –Moment: $$M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$$

Area: $$M_{00} = \sum_x \sum_y I(x,y)$$



Average x-coordinate:
Average y-coordinate:
$$\bar{x} = \frac{M_{10}}{M_{00}} \qquad \bar{y} = \frac{M_{01}}{M_{00}}$$

Center of Mass: $$(\bar{x}, \bar{y}) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}} \right)$$

Moments

Central Moment: $$\mu_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^j I(x,y)$$

Moment expressions that are invariant to translation, rotation and/or scale:

1. For first-order moments, $\mu_{0,1} = \mu_{1,0} = 0$, (always invariant).
2. For second-order moments, $(p + q = 2)$, the invariants are

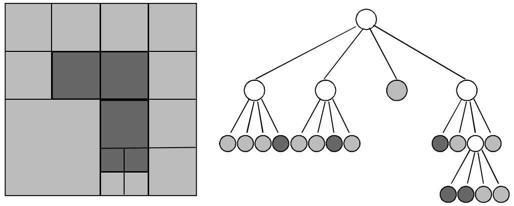
$$\phi_1 = \mu_{2,0} + \mu_{0,2}$$
$$\phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \qquad (9.80)$$

3. For third-order moments $(p + q = 3)$, the invariants are

$$\phi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2$$
$$\phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2$$

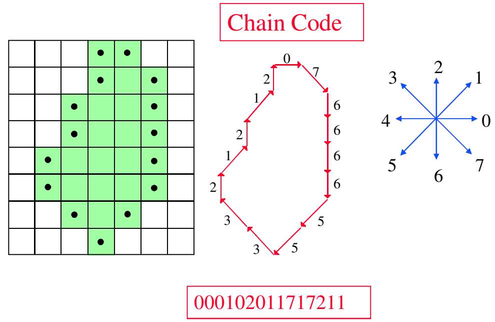
wide domain, not unique, not unambiguous, not generative, not stable, invariant to translation, rotation. Very convenient.

Quad Tree Representation



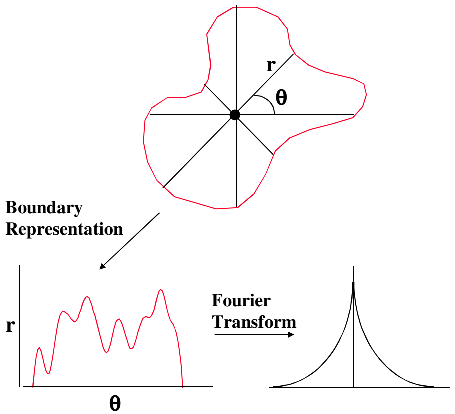
wide domain,
unique, unambiguous, generative – up to error tolerance
partially stable
Not invariant to translation, rotation scale.
Inefficient for comparison

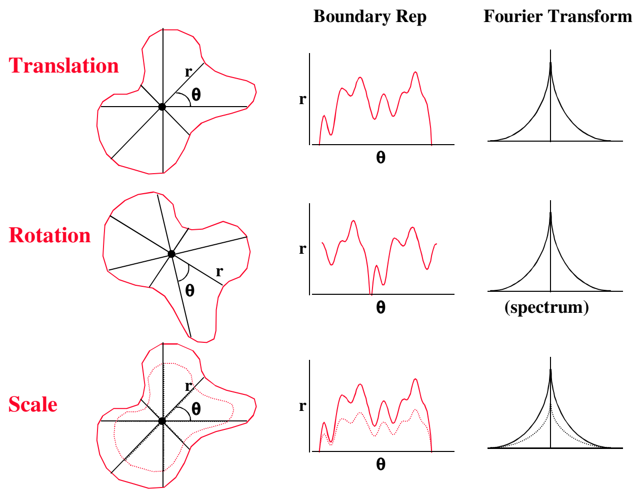
Edge Based Representation



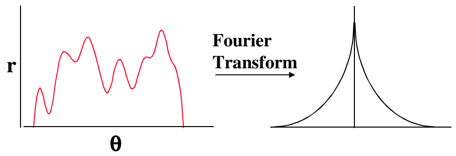
wide domain, Unique, unambiguous, generative - 2D only,
Not very stable Invariant to translation. Rotation (x90 deg)

Fourier Descriptors



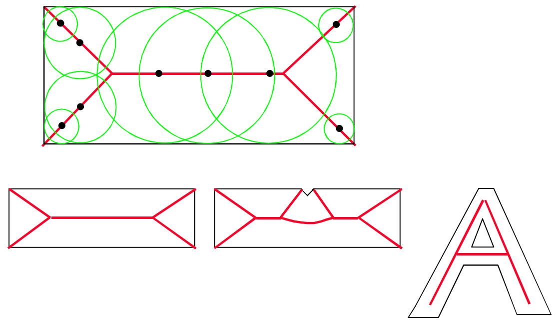


Fourier Descriptors



wide domain, Unique, unambiguous, generative, Stable (depends on tolerance), Invariant to translation. Rotation, Scale.

Interior Based representation – MAT, Skeleton



wide domain, unique, unambiguous, generative
not stable - small changes affect dramatically

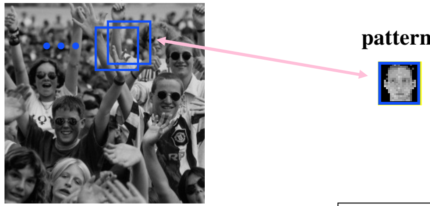
Pattern Matching – Direct approach (Appearance based)

image

pattern

Finding a pattern in an Image

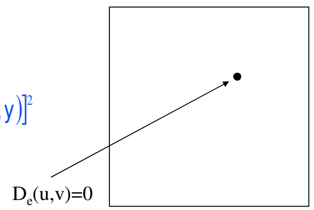
image



pattern

Look for minimum of:

$$d_e(u,v)=\sum_{x,y\in N}\left[I(u+x,v+y)-P(x,y)\right]^2$$



Finding a pattern in an Image

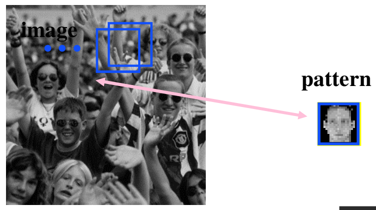
$$d_e(u,v)=\sum_{x,y\in N}\left[I(u+x,v+y)-P(x,y)\right]^2$$

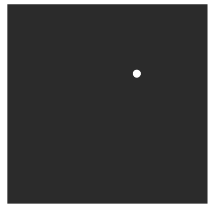
$$=\sum_{x,y\in N} I(u+x,v+y)^2+P(x,y)^2-2I(u+x,v+y)P(x,y)$$

$$=\sum_{x,y\in N} I(u+x,v+y)^2+\sum_{x,y\in N} P(x,y)^2-2\sum_{x,y\in N} I(u+x,v+y)P(x,y)$$

- Sum of squares of the window
- Sum of squares of the pattern CONSTANT
- Correlation

Finding a pattern in an Image - Correlation

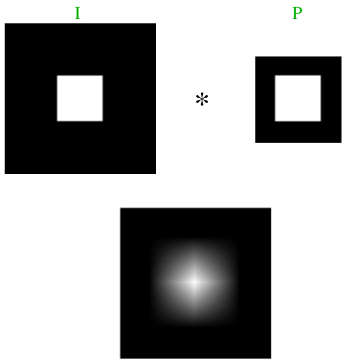




Look for maximum of:

$$\sum_{x,y\in N}\left[I(u+x,v+y)P(x,y)\right]$$

Correlation

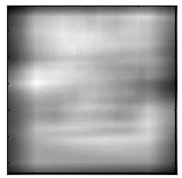


I corr P

Real Image – Correlation Example

image

pattern



Correlation

Correlation value is dependent on the local gray value of the pattern and the image window.

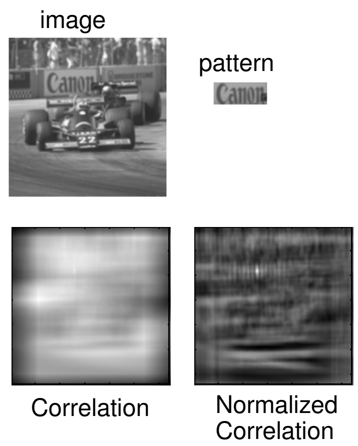
Normalized Correlation

$$\frac{\sum_{x,y\in N}\left[I(u+x,v+y)-\bar{I}_{uv}\right]\left[P(x,y)-\bar{P}\right]}{\left[\sum_{x,y\in N}\left[I(u+x,v+y)-\bar{I}_{uv}\right]^2\sum_{x,y\in N}\left[P(x,y)-\bar{P}\right]^2\right]^{1/2}}$$

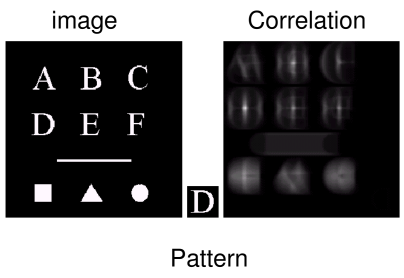
Correlation value is in (-1..1)

Correlation value is **independent** of the local gray value of the pattern and the image window.

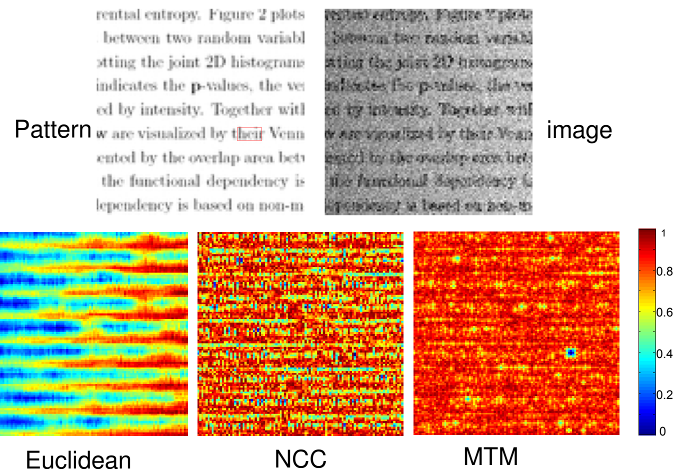
Normalized Correlation - Example



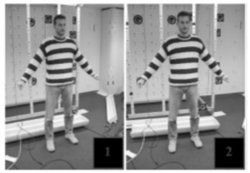
Normalized Correlation - Example



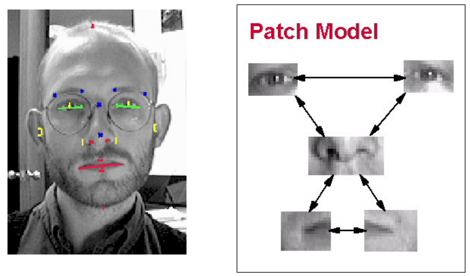
Pattern Matching - Example



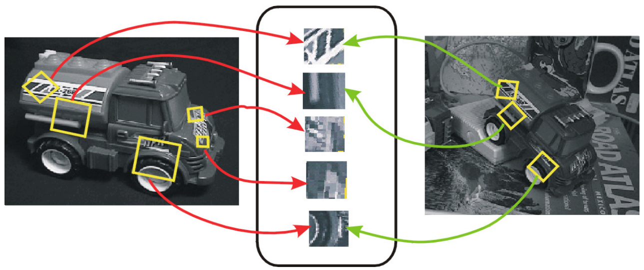
Pairs for Image Matching



Feature Based Object Detection



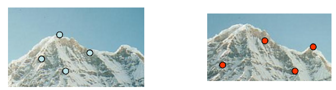
Feature Based Object Detection



Features Descriptors

Features: Issues to be addressed

- What are “good” features to extract?
  - –Distinctive
  - –Invariant to different acquisition conditions
    - –Different view-points, different illuminations, different cameras, etc.
- How can we find corresponding features in both images?



no chance to match!

Invariant Feature Descriptors

- Schmid & Mohr 1997, Lowe 1999, Baumberg 2000, Tuytelaars & Van Gool 2000, Mikolajczyk & Schmid 2001, Brown & Lowe 2002, Matas et. al. 2002, Schaffalitzky & Zisserman 2002

Image Features

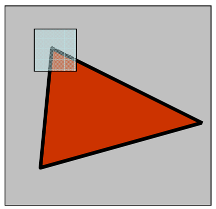
- Feature Detectors - where
- Feature Descriptors - what

- Methods:
  - Harris Corner Detector (multi-scale Harris)
  - SIFT (Scale Invariant Features Transform)

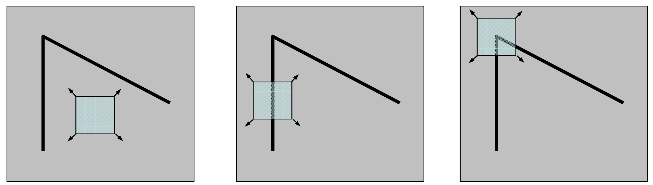
Harris Corner Detector

C.Harris, M.Stephens. “A Combined Corner and Edge Detector”. 1988

- We should easily recognize a corner by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



Harris Detector: Basic Idea



“flat” region: no change in all directions

“edge”: no change along the edge direction

“corner”: significant change in all directions

Harris Detector: Mathematics

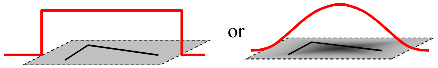
Corner at position (x,y) ?

Evaluate change of intensity for shift in [*u,v*] direction:

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

Window function — Shifted intensity — Intensity

Window function $w(x,y)$ =



1 in window, 0 outside — Gaussian

Harris Detector: Mathematics

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

For small [u,v]: $I(x+u,y+v)=I(x,y)+uI_x+vI_y$

We have:

$$E(u,v)=\sum_{x,y} w(x,y)\left\|\begin{bmatrix}I_x(x,y) & I_y(x,y)\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix}\right\|^2 =$$

$$\begin{bmatrix}u & v\end{bmatrix}\sum w(x,y)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}\begin{bmatrix}u\\ v\end{bmatrix}=\begin{bmatrix}u & v\end{bmatrix}M\begin{bmatrix}u\\ v\end{bmatrix}$$

Harris Detector: Mathematics

For small shifts [*u,v*] we have a *bilinear* approximation:

$$E(u,v)\cong\left[u,v\right]\ M\begin{bmatrix}u\\ v\end{bmatrix}$$

where *M* is a 2×2 matrix computed from image derivatives:

$$M=\sum_{x,y} w(x,y)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}$$

Harris Detector: Mathematics

What is the direction [u,v] of greatest intensity change?

$$\arg\max_{\|(u,v)\|=1} E(u,v)=\mathbf{e}_{\max}$$

Denote by $\mathbf{e}_i$ the i[th] eigen-vector of M whose eigen-value is λ_i:

$$\mathbf{e}_i^T M\mathbf{e}_i=\lambda_i>0$$

Conclusions:

$$E(\mathbf{e}_{\max})=\lambda_{\max}$$

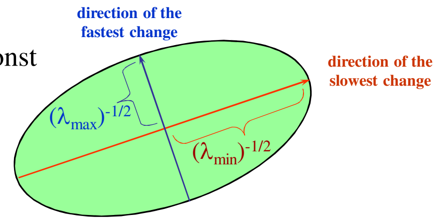
Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

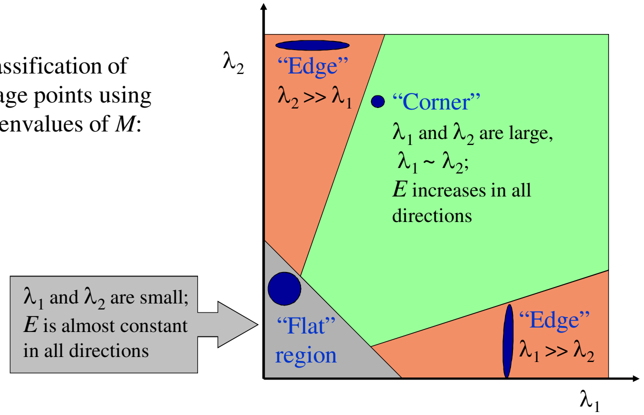
λ_1, λ_2 – eigenvalues of M

Ellipse $E(u,v)$ = const



Harris Detector: Mathematics

Classification of image points using eigenvalues of M:



Harris Detector: Mathematics

Measure of corner response (without calculating the e.v.):

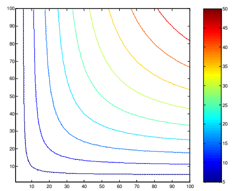
$$R = \frac{\det M}{\operatorname{Trace} M}$$

$\det M = \lambda_1 \lambda_2$

$\operatorname{trace} M = \lambda_1 + \lambda_2$

R is associated with the smallest eigen-vector (why?)

R v.s. λ_1, λ_2



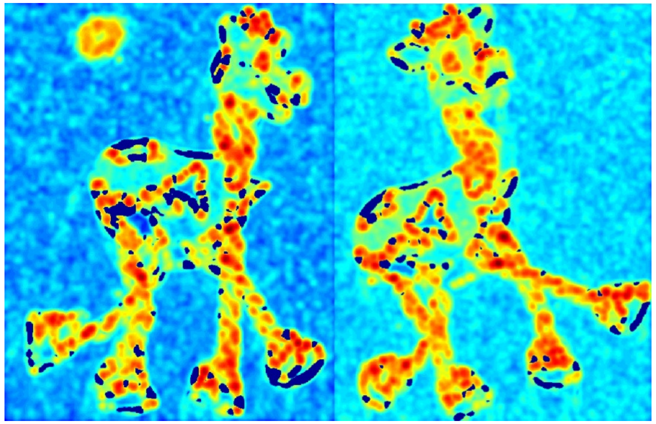
Harris Corner Detector

- The Algorithm:
  - Find points with large corner response function R (R > threshold)
  - Take the points of local maxima of R

Harris Detector: Workflow

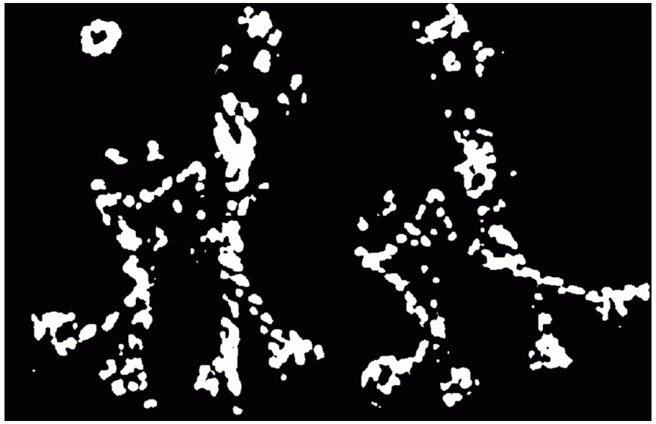
Harris Detector: Workflow

Compute corner response R



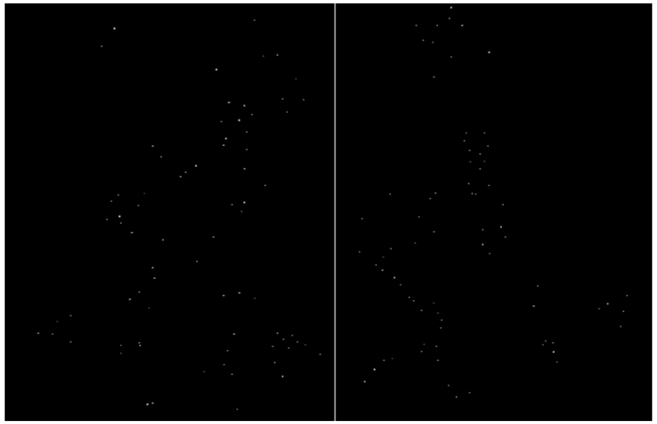
Harris Detector: Workflow

Find points with large corner response: R>threshold



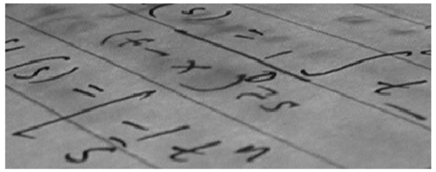
Harris Detector: Workflow

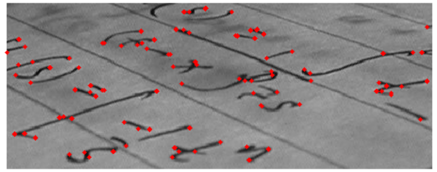
Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector: Example

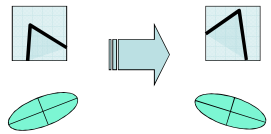




Harris Detector: Example

Harris Detector: Some Properties

- Rotation invariance



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

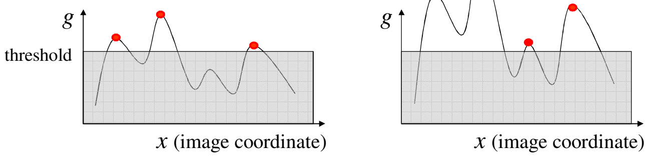
Corner response R is invariant to image rotation

Harris Detector: Some Properties

Partial invariance to affine intensity change

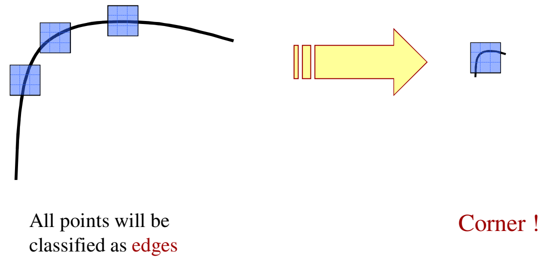
✓Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

✓Intensity scale: $I \rightarrow a\, I$



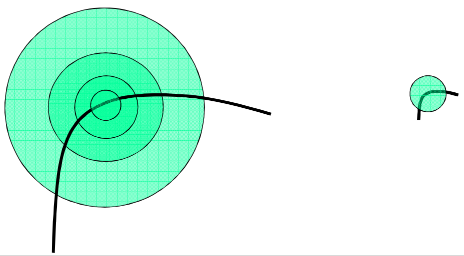
Harris Detector: Some Properties

- But: non-invariant to spatial scale!



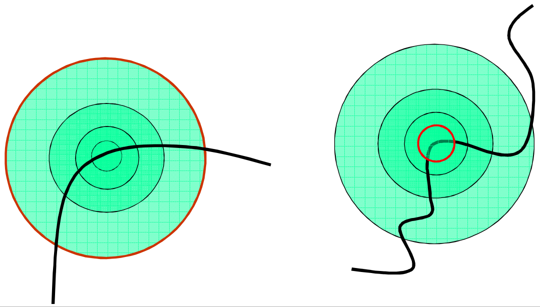
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



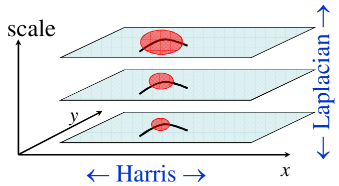
Scale Invariant Detection

- **The problem**: how do we choose corresponding circles *independently* in each image?
- **Solution**: choose the scale of the "best" corner.



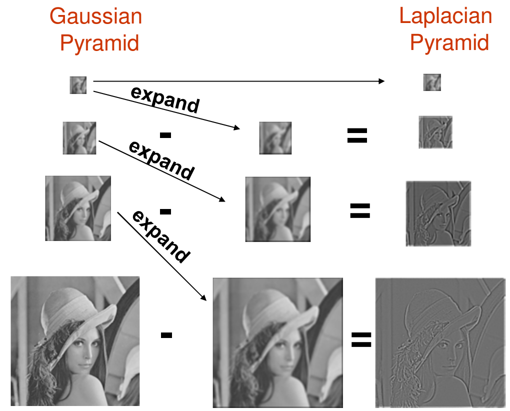
Harris-Laplacian Point Detector

- Harris-Laplacian
  Find local maximum of: Harris corner detector for a set of Laplacian images.



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

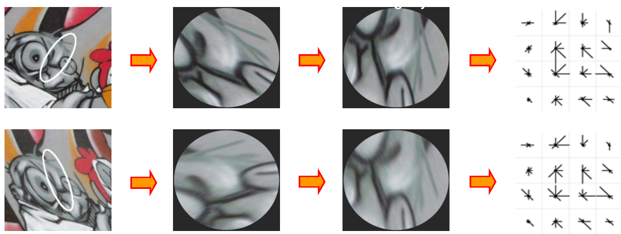
Memo: Gaussian / Laplacian Pyramids



Harris - Laplacian Detector

SIFT – Scale Invariant Feature Transform

David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110



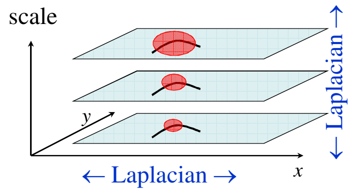
SIFT – Scale Invariant Feature Transform

David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

- *Give about 2000 stable "keypoints" for a typical 500* x *500 image*
- *Each keypoint is described by a vector of*
  4 x 4 x 8 = 128 elements
  (over 4x4 array of 8-bin gradient histograms keypoint neighborhood)

SIFT – Scale Invariant Feature Transform

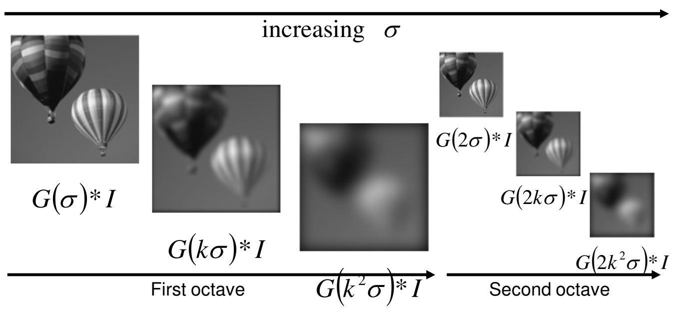
- *Find local maximum of* Laplacian in space and scale



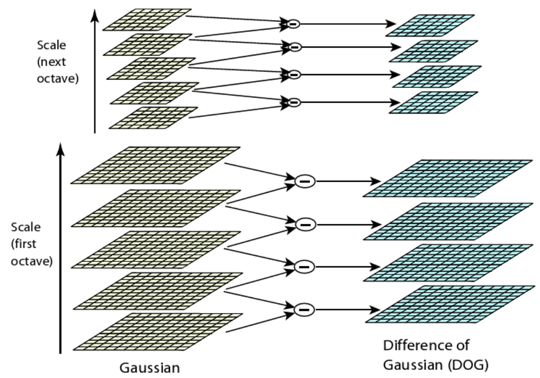
David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

SIFT - Point Detection

- Construct scale-space:



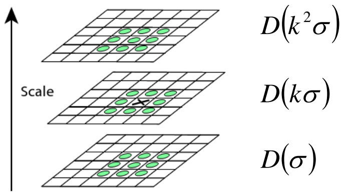
SIFT – Scale Space



SIFT – point detection

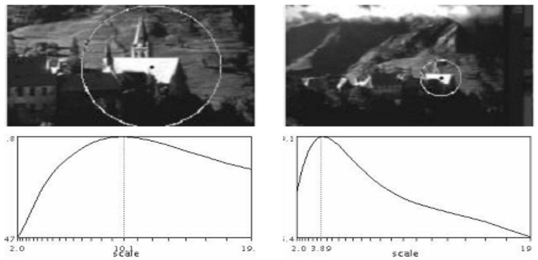
STEP 1:
Determine local Maxima in DoG pyramid (Laplacian Pyramid).

- Scale Space extrema detection.
- Choose all extrema within 3x3x3 neighborhood.



SIFT – point detection

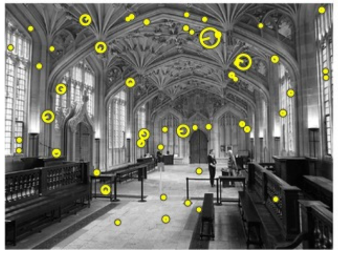
STEP 1:
Determine local Maxima in DoG pyramid (Laplacian Pyramid).



Experimentally, Maximum of Laplacian gives best notion of s

SIFT - Step 1: Interest Point Detection

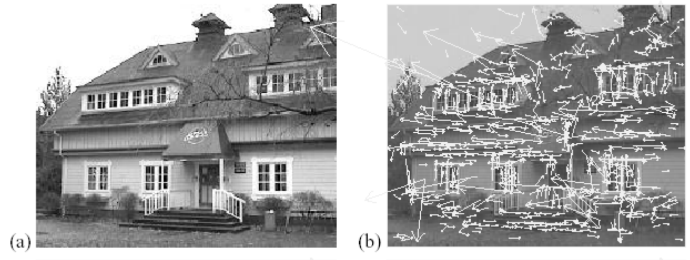
Detections at multiple scales



Some of the detected SIFT frames.

http://www.vlfeat.org/overview/sift.html

SIFT – point detection



(a) 233x189 image (b) 832 SIFT extrema

SIFT - Step 2: Interest Localization & Filtering

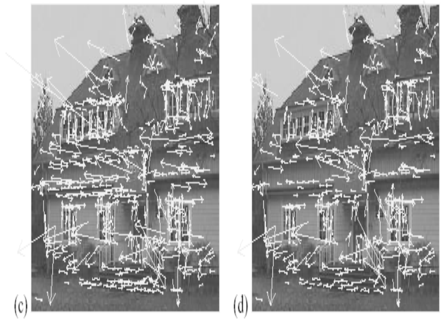
2) Remove bad Interest points:

a) Remove points with low contrast

b) Remove Edge points (Eigenvalues of Hessian Matrix must BOTH be large).

$$A^T A = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Interest Points

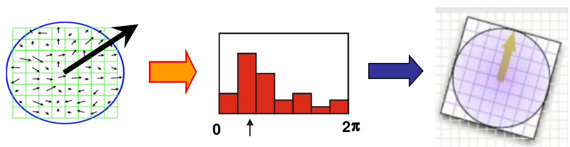


(c) 729 left after peak value threshold (from 832)
(d) 536 left after testing ratio of principle curvatures

SIFT – Descriptor Vector

STEP 3: Select canonical orientation

- Each SIFT interest point is associated with location (x,y) and scale (σ)
- Compute gradient magnitude and orientation for each SIFT point:

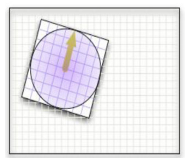


Assign canonical orientation at peak of smoothed histogram (fit parabola to better localize peak).

SIFT – Descriptor Vector

STEP 3: Select canonical orientation

- Each SIFT interest point is associated with location (x,y), scale (σ), gradient magnitude and orientation (m, θ).



- Compute SIFT feature - a vector of 128 entries.

SIFT – Descriptor Vector

STEP 4: Compute SIFT feature vector of 128 entries

- Gradients determined in 16x16 window at SIFT point in scale space.
- Histogram is computed for gradients of each 4x4 sub window in 8 **relative** directions.
- A 4x4x8 = 128 dimensional feature vector is produced.

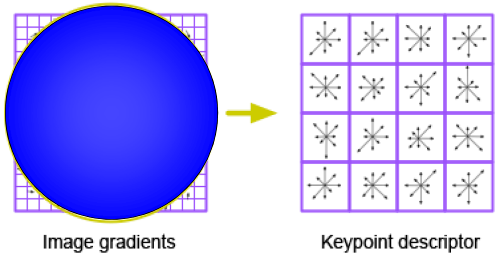
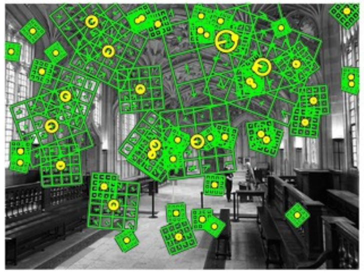


Image from: Jonas Hurrelmann

SIFT – Descriptor Vector

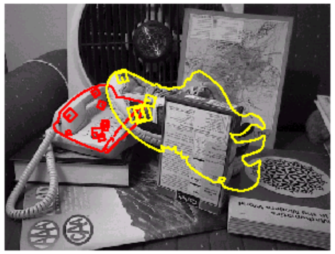
STEP 4: Compute feature vector



Object Recognition

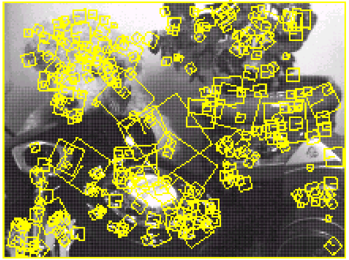
- Only 3 keys are needed for recognition, so extra keys provide robustness

Recognition under occlusion



Test of illumination Robustness

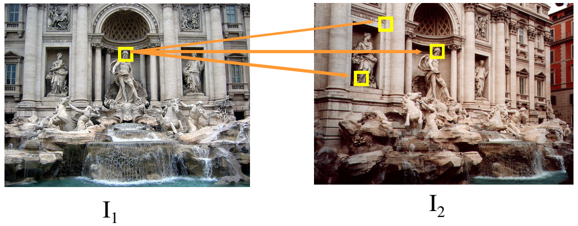
- Same **image** under differing illumination



273 keys verified in final match

Matching SIFT Features

- 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. Accept if below threshold.



Matching SIFT Features

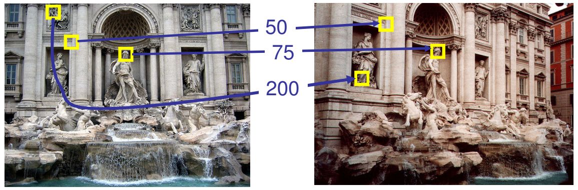
22 correct matches

Matching SIFT Features

33 correct matches

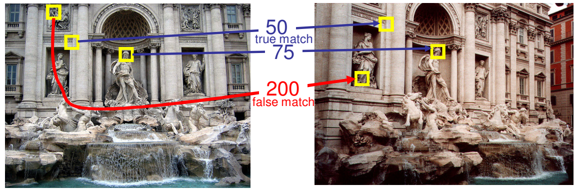
Matching SIFT Features

How to evaluate the performance of a feature matcher?



Matching SIFT Features

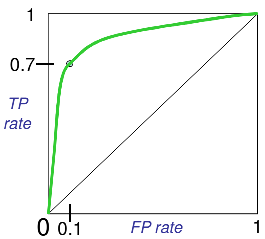
- Threshold t affects # of correct/false matches



- True positives (TP) = # of detected matches that are correct
- False positives (FP) = # of detected matches that are incorrect

Matching SIFT Features

- ROC Curve
  - Generated by computing (FP, TP) for different thresholds.
  - Maximize area under the curve (AUC).

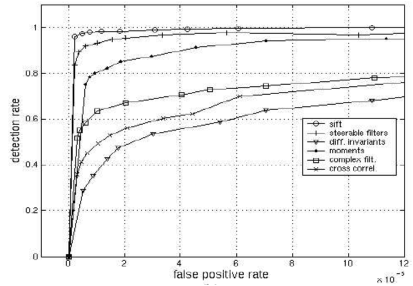


http://en.wikipedia.org/wiki/Receiver_operating_characteristic

Evaluating SIFT Features

- Empirically found[2] to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45^0



[1] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004
[2] K.Mikolajczyk, C.Schmid. "A Performance Evaluation of Local Descriptors". CVPR 2003

Example - Mosaicing

Source: Alexei Efros

Example: Mosiacing (Panorama)

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Image Matching

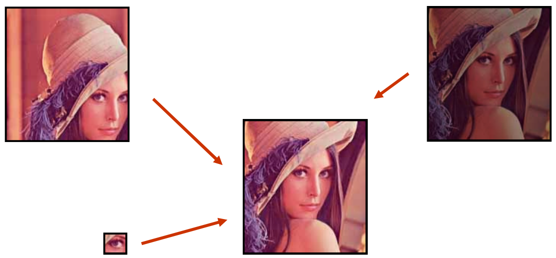
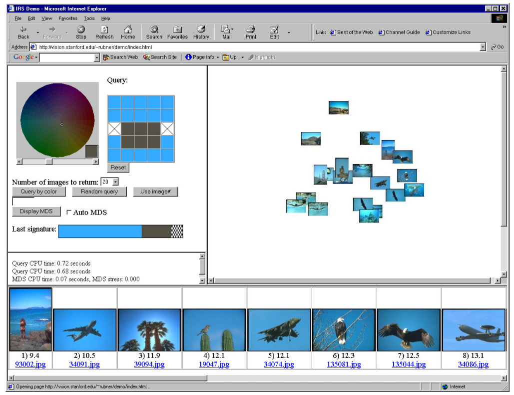


Image Retrieval



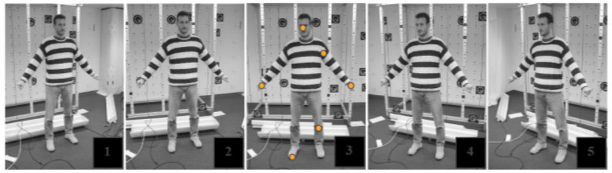
Object Recognition

Motion Estimation and Optical Flow Tracking

Example: Mosiacing (Panorama)

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003

Example – 3D Reconstruction



Source: http://www.photogrammetry.ethz.ch/general/persons/fabio/fabio_spie0102.pdf

Image Matching

Three approaches:

- **Shape Matching**
  - Assume shape has been extracted
- **Direct (appearance-based) registration**
  - Search for alignment where most pixels agree
- **Feature-based registration**
  - Find a few matching features in both images
  - compute alignment

Direct Method (brute force)

The simplest approach is a brute force search

- Need to define image distance function:
  SSD, Normalized Correlation, Mutual Information, etc.
- Search over all parameters within a reasonable range:

e.g. for translation:

```
for Δx=x0:step:x1,
 for Δy=y0:step:y1,
   calculate Dist(image1(x,y),image2(x+Δx,y+Δy))
 end;
end;
```

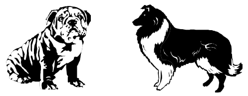
Shape Representation

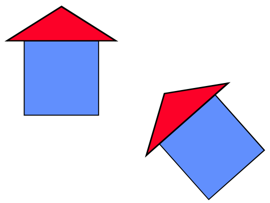
- Region Based Representation
- Area / Circumference / Width
- Euler Number

- Moments
- Quad Trees

- Edge Based Representation
- Chain Code
- Fourier Descriptor

- Interior Based Representation
- MAT / Skeleton
- Hierarchical Representations

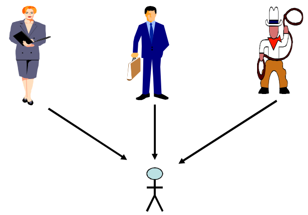




Shape Representation

Shape representation must be GOOD:

- Different shapes ⇔ Different Codes
- Location / Rotation /Scale Invariant
- Convenient
- Stable
- Generative

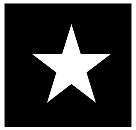


Moments

$$I(x,y) = \begin{cases} 1 & \text{If pixel (x,y) is IN object} \\ 0 & \text{otherwise} \end{cases}$$

ij –Moment: $M_{ij} = \sum_x \sum_y x^i y^j I(x,y)$

Area: $M_{00} = \sum_x \sum_y I(x,y)$



Average x-coordinate: $\bar{x} = \frac{M_{10}}{M_{00}}$

Average y-coordinate: $\bar{y} = \frac{M_{01}}{M_{00}}$

Center of Mass: $(\bar{x},\bar{y}) = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right)$

Moments

Central Moment: $\mu_{ij} = \sum_x \sum_y (x-\bar{x})^i (y-\bar{y})^j I(x,y)$

Moment expressions that are invariant to translation, rotation and/or scale:

1. For first-order moments, $\mu_{0,1} = \mu_{1,0} = 0$, (always invariant).
2. For second-order moments, $(p + q = 2)$, the invariants are

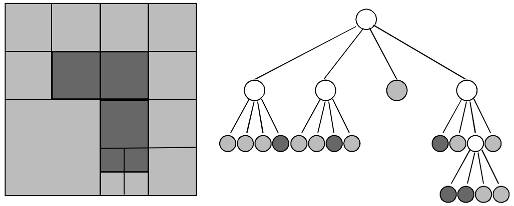
$$\phi_1 = \mu_{2,0} + \mu_{0,2}$$
$$\phi_2 = (\mu_{2,0} - \mu_{0,2})^2 + 4\mu_{1,1}^2 \quad (9.80)$$

3. For third-order moments $(p + q = 3)$, the invariants are

$$\phi_3 = (\mu_{3,0} - 3\mu_{1,2})^2 + (\mu_{0,3} - 3\mu_{2,1})^2$$
$$\phi_4 = (\mu_{3,0} + \mu_{1,2})^2 + (\mu_{0,3} + \mu_{2,1})^2$$

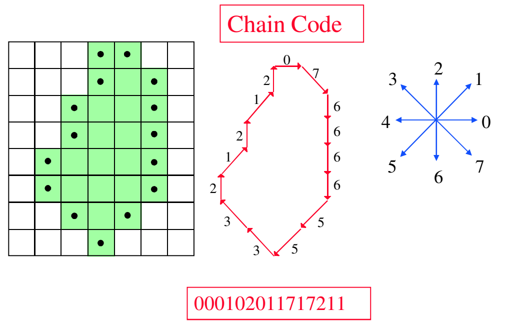
wide domain, not unique, not unambiguous, not generative, not stable, invariant to translation, rotation. Very convenient.

Quad Tree Representation



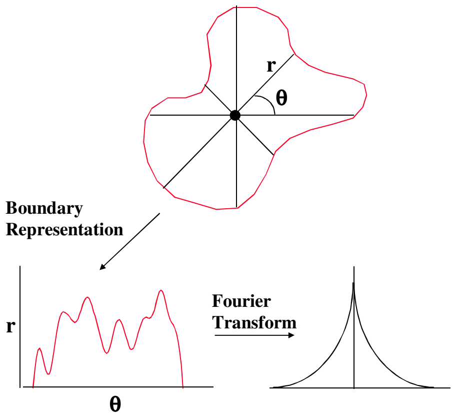
wide domain,
unique, unambiguous, generative – up to error tolerance
partially stable
Not invariant to translation, rotation scale.
Inefficient for comparison

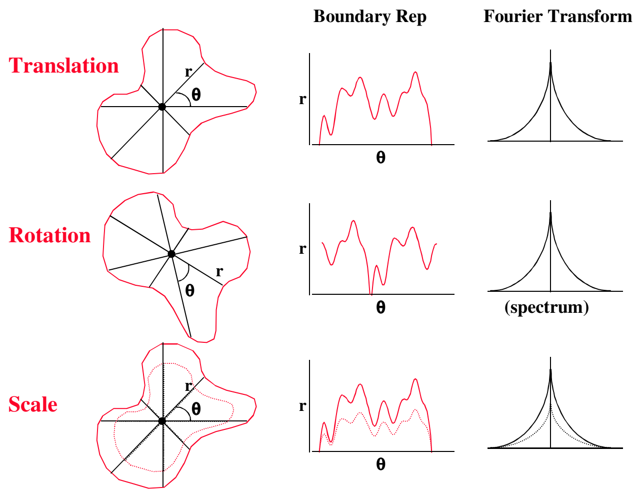
Edge Based Representation



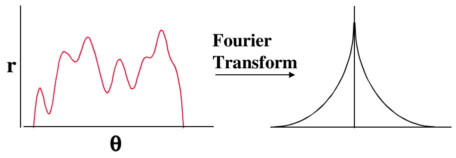
wide domain, Unique, unambiguous, generative - 2D only,
Not very stable Invariant to translation. Rotation (x90 deg)

Fourier Descriptors



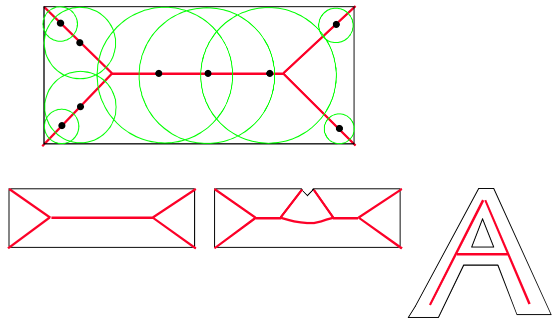


Fourier Descriptors



wide domain, Unique, unambiguous, generative, Stable (depends on tolerance), Invariant to translation. Rotation, Scale.

Interior Based representation – MAT, Skeleton



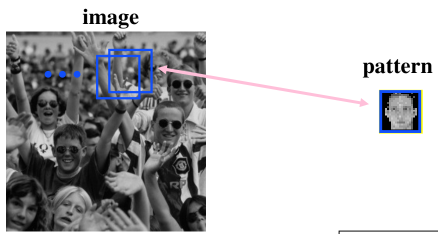
wide domain, unique, unambiguous, generative
not stable - small changes affect dramatically

Pattern Matching – Direct approach (Appearance based)

image

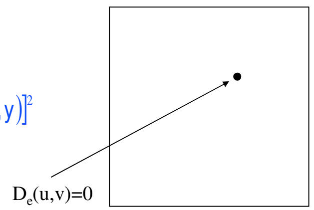
pattern

Finding a pattern in an Image



Look for minimum of:

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



Finding a pattern in an Image

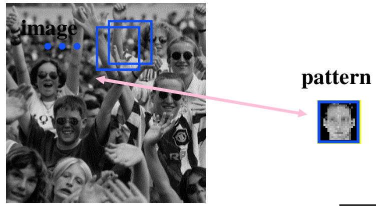
$$d_e(u,v) = \sum_{x,y \in N} [I(u+x, v+y) - P(x,y)]^2$$

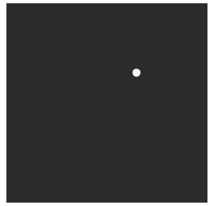
$$= \sum_{x,y \in N} I(u+x, v+y)^2 + P(x,y)^2 - 2I(u+x, v+y)P(x,y)$$

$$= \sum_{x,y \in N} I(u+x, v+y)^2 + \sum_{x,y \in N} P(x,y)^2 - 2\sum_{x,y \in N} I(u+x, v+y)P(x,y)$$

Sum of squares of the window — Sum of squares of the pattern CONSTANT — Correlation

Finding a pattern in an Image - Correlation





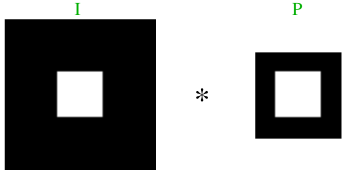
Look for maximum of:

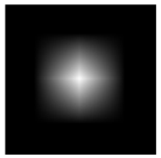
$$\sum_{x,y \in N} [I(u+x, v+y)P(x,y)]$$

Correlation

I

P



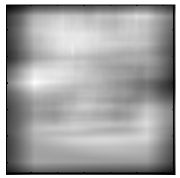


I corr P

Real Image – Correlation Example

image

pattern



Correlation

Correlation value is dependent on the local gray value of the pattern and the image window.

Normalized Correlation

$$\frac{\sum_{x,y\in N}\left[I(u+x,v+y)-\bar{I}_{uv}\right]\left[P(x,y)-\bar{P}\right]}{\left[\sum_{x,y\in N}\left[I(u+x,v+y)-\bar{I}_{uv}\right]^2\sum_{x,y\in N}\left[P(x,y)-\bar{P}\right]^2\right]^{1/2}}$$

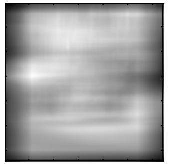
Correlation value is in (-1..1)

Correlation value is **independent** of the local gray value of the pattern and the image window.

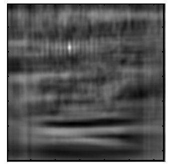
Normalized Correlation - Example

image

pattern



Correlation

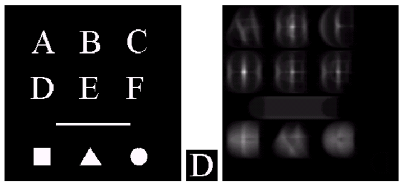


Normalized Correlation

Normalized Correlation - Example

image

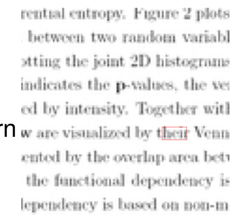
Correlation

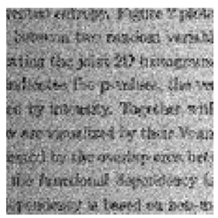


Pattern

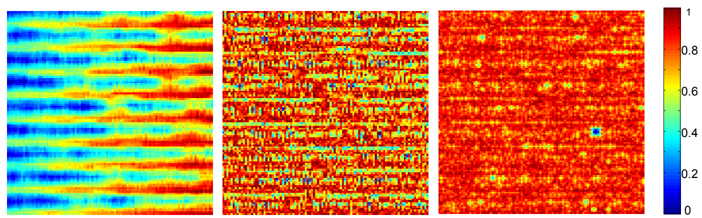
Pattern Matching - Example

Pattern



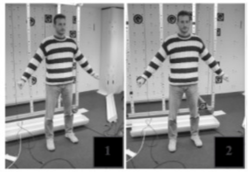


image

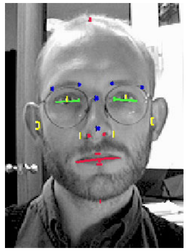


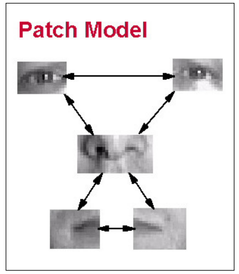
Euclidean NCC MTM

Pairs for Image Matching

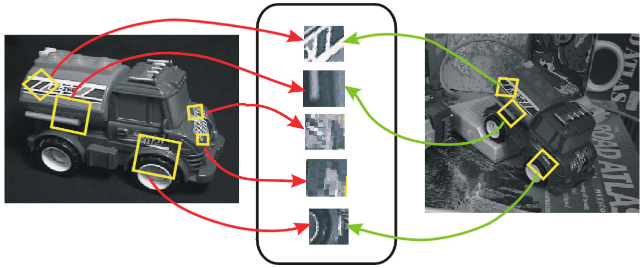


Feature Based Object Detection





Feature Based Object Detection



Features Descriptors

Features: Issues to be addressed

- What are “good” features to extract?
  - –Distinctive
  - –Invariant to different acquisition conditions
    - –Different view-points, different illuminations, different cameras, etc.
- How can we find corresponding features in both images?

no chance to match!

Invariant Feature Descriptors

- Schmid & Mohr 1997, Lowe 1999, Baumberg 2000, Tuytelaars & Van Gool 2000, Mikolajczyk & Schmid 2001, Brown & Lowe 2002, Matas et. al. 2002, Schaffalitzky & Zisserman 2002

Image Features

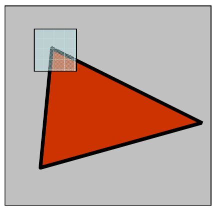
- Feature Detectors - where
- Feature Descriptors - what

- Methods:
  - Harris Corner Detector (multi-scale Harris)
  - SIFT (Scale Invariant Features Transform)

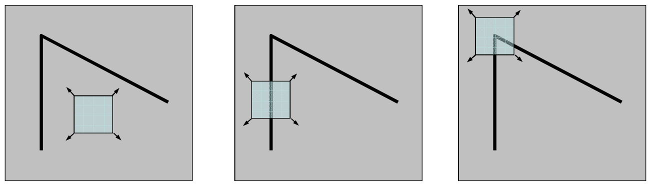
Harris Corner Detector

C.Harris, M.Stephens. "A Combined Corner and Edge Detector". 1988

- We should easily recognize a corner by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



Harris Detector: Basic Idea



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Harris Detector: Mathematics

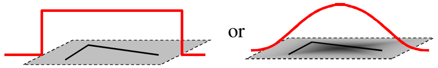
Corner at position (x,y) ?

Evaluate change of intensity for shift in [*u,v*] direction:

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

Window function — Shifted intensity — Intensity

Window function $w(x,y)$ =



or

1 in window, 0 outside — Gaussian

Harris Detector: Mathematics

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

For small [u,v]: $I(x+u,y+v)=I(x,y)+uI_x+vI_y$

We have:

$$E(u,v)=\sum_{x,y} w(x,y)\left\|\begin{bmatrix}I_x(x,y) & I_y(x,y)\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix}\right\|^2 =$$

$$\begin{bmatrix}u & v\end{bmatrix}\sum w(x,y)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}\begin{bmatrix}u\\v\end{bmatrix}=\begin{bmatrix}u & v\end{bmatrix}M\begin{bmatrix}u\\v\end{bmatrix}$$

Harris Detector: Mathematics

For small shifts [*u,v*] we have a *bilinear* approximation:

$$E(u,v)\cong\begin{bmatrix}u,v\end{bmatrix}\; M \begin{bmatrix}u\\v\end{bmatrix}$$

where *M* is a 2×2 matrix computed from image derivatives:

$$M=\sum_{x,y} w(x,y)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}$$

Harris Detector: Mathematics

What is the direction [u,v] of greatest intensity change?

$$\arg\max_{\|(u,v)\|=1} E(u,v)=\mathbf{e}_{\max}$$

Denote by $\mathbf{e}_i$ the i[th] eigen-vector of M whose eigen-value is λ_i:

$$\mathbf{e}_i^T M\mathbf{e}_i=\lambda_i>0$$

Conclusions:

$$E(\mathbf{e}_{\max})=\lambda_{\max}$$

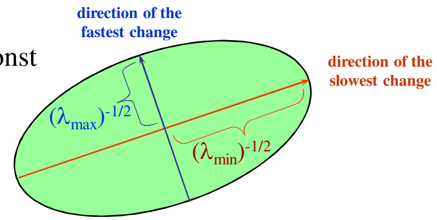
Harris Detector: Mathematics

Intensity change in shifting window: eigenvalue analysis

$$E(u,v) \cong [u,v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

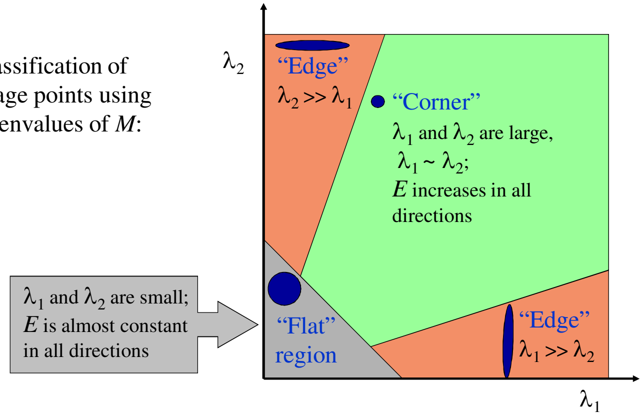
λ_1, λ_2 – eigenvalues of M

Ellipse $E(u,v)$ = const



Harris Detector: Mathematics

Classification of image points using eigenvalues of M:



Harris Detector: Mathematics

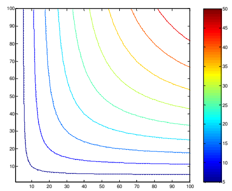
Measure of corner response (without calculating the e.v.):

$$R = \frac{\det M}{\operatorname{Trace} M}$$

$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

R is associated with the smallest eigen-vector (why?)

R v.s. λ_1, λ_2



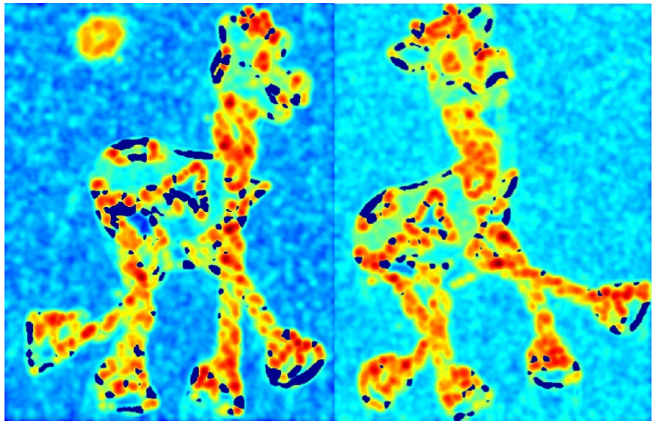
Harris Corner Detector

- The Algorithm:
  - Find points with large corner response function R (R > threshold)
  - Take the points of local maxima of R

Harris Detector: Workflow

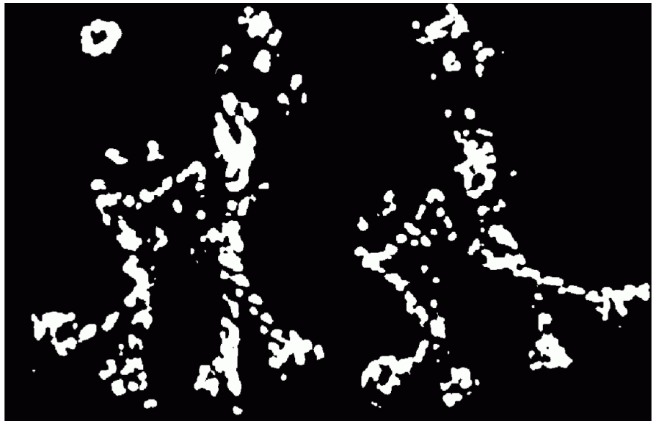
Harris Detector: Workflow

Compute corner response R



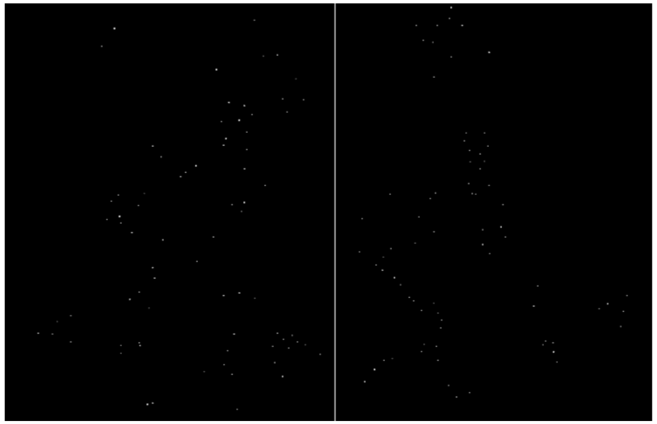
Harris Detector: Workflow

Find points with large corner response: R>threshold



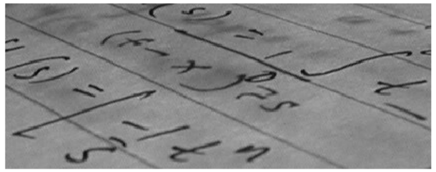
Harris Detector: Workflow

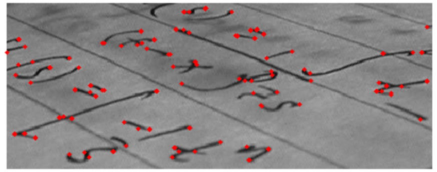
Take only the points of local maxima of R



Harris Detector: Workflow

Harris Detector: Example

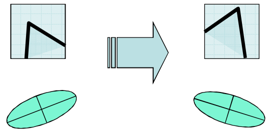




Harris Detector: Example

Harris Detector: Some Properties

- Rotation invariance



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

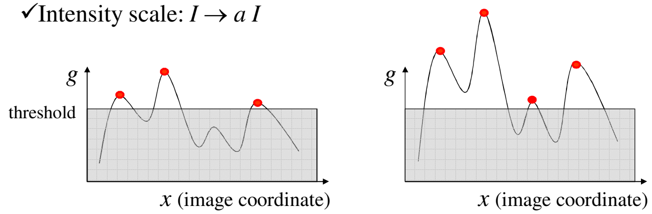
Corner response R is invariant to image rotation

Harris Detector: Some Properties

Partial invariance to affine intensity change

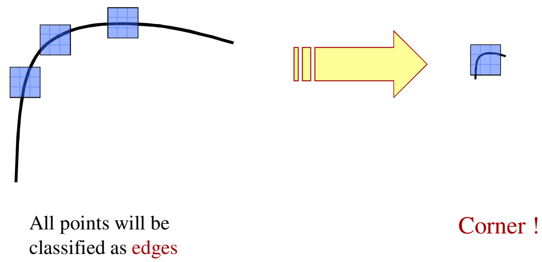
✓Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

✓Intensity scale: $I \rightarrow a\, I$



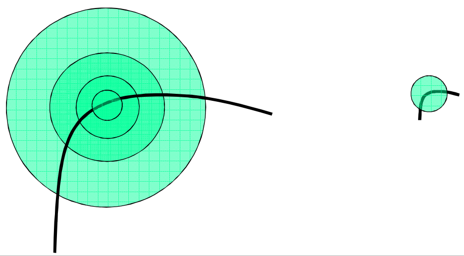
Harris Detector: Some Properties

- But: non-invariant to spatial scale!



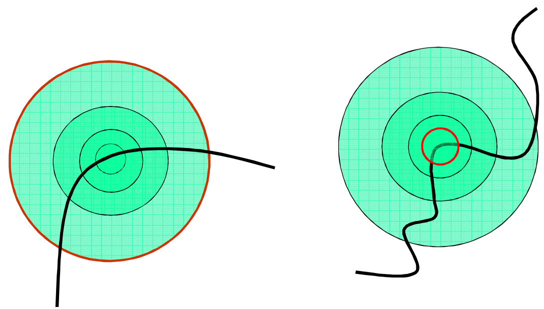
Scale Invariant Detection

- Consider regions (e.g. circles) of different sizes around a point
- Regions of corresponding sizes will look the same in both images



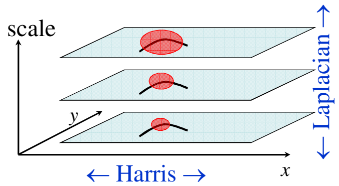
Scale Invariant Detection

- **The problem**: how do we choose corresponding circles *independently* in each image?
- **Solution**: choose the scale of the "best" corner.



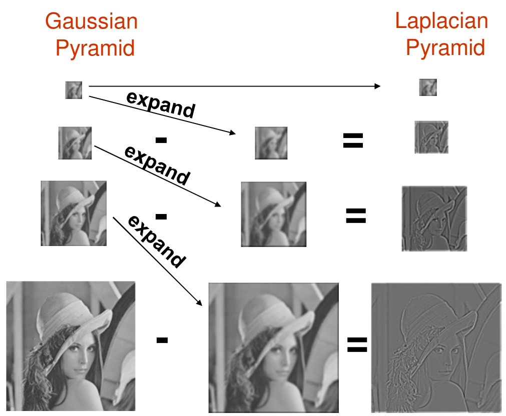
Harris-Laplacian Point Detector

- Harris-Laplacian
  Find local maximum of: Harris corner detector for a set of Laplacian images.



[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001

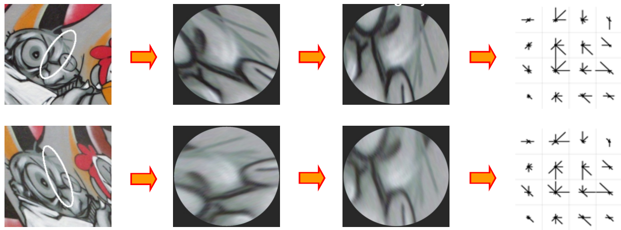
Memo: Gaussian / Laplacian Pyramids



Harris - Laplacian Detector

SIFT – Scale Invariant Feature Transform

David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110



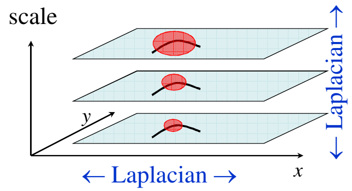
SIFT – Scale Invariant Feature Transform

David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

- *Give about 2000 stable "keypoints" for a typical 500* x *500 image*
- *Each keypoint is described by a vector of 4 x 4 x 8 = 128 elements (over 4x4 array of 8-bin gradient histograms keypoint neighborhood)*

SIFT – Scale Invariant Feature Transform

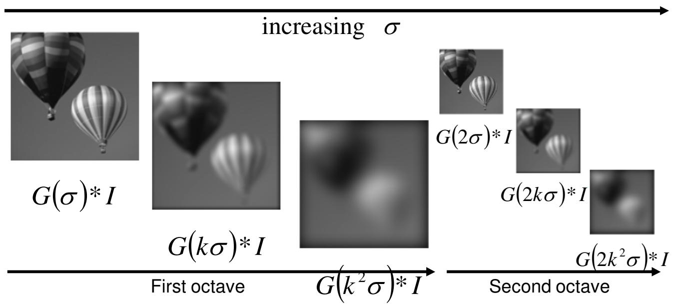
- *Find local maximum of* Laplacian in space and scale



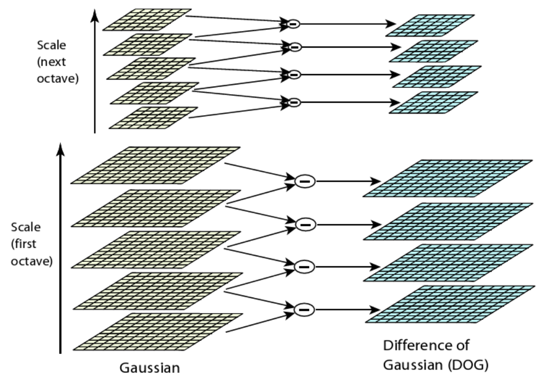
David G. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, 60, 2 (2004), pp. 91-110

SIFT - Point Detection

- Construct scale-space:



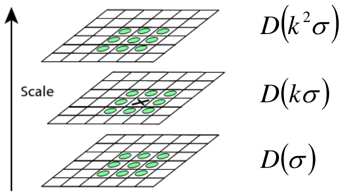
SIFT – Scale Space



SIFT – point detection

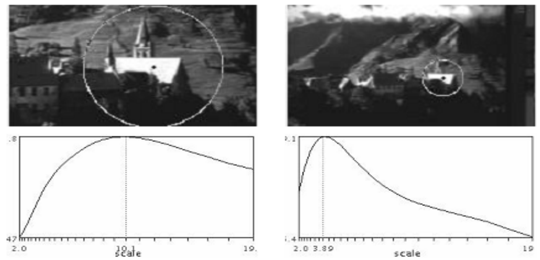
STEP 1:
Determine local Maxima in DoG pyramid (Laplacian Pyramid).

- Scale Space extrema detection.
- Choose all extrema within 3x3x3 neighborhood.



SIFT – point detection

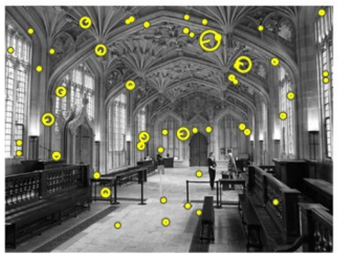
STEP 1:
Determine local Maxima in DoG pyramid (Laplacian Pyramid).



Experimentally, Maximum of Laplacian gives best notion of s

SIFT - Step 1: Interest Point Detection

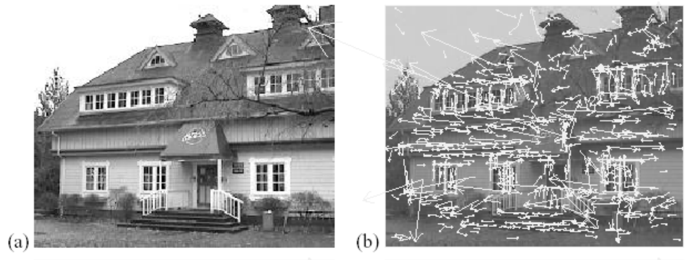
Detections at multiple scales



Some of the detected SIFT frames.

http://www.vlfeat.org/overview/sift.html

SIFT – point detection



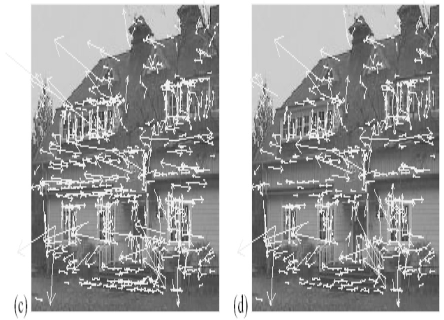
233x189 image   832 SIFT extrema

SIFT - Step 2: Interest Localization & Filtering

2) Remove bad Interest points:
   a) Remove points with low contrast
   b) Remove Edge points (Eigenvalues of Hessian Matrix must BOTH be large).

$$A^TA = \begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix} = \sum \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x \; I_y] = \sum \nabla I (\nabla I)^T$$

Interest Points

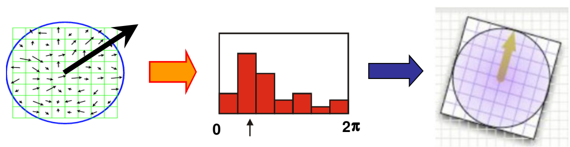


(c) 729 left after peak value threshold (from 832)
(d) 536 left after testing ratio of principle curvatures

SIFT – Descriptor Vector

STEP 3: Select canonical orientation

- Each SIFT interest point is associated with location (x,y) and scale (σ)
- Compute gradient magnitude and orientation for each SIFT point:

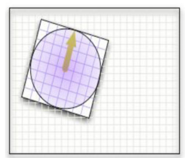


Assign canonical orientation at peak of smoothed histogram (fit parabola to better localize peak).

SIFT – Descriptor Vector

STEP 3: Select canonical orientation

- Each SIFT interest point is associated with location (x,y), scale (σ), gradient magnitude and orientation (m, θ).



- Compute SIFT feature - a vector of 128 entries.

SIFT – Descriptor Vector

STEP 4: Compute SIFT feature vector of 128 entries

- Gradients determined in 16x16 window at SIFT point in scale space.
- Histogram is computed for gradients of each 4x4 sub window in 8 **relative** directions.
- A 4x4x8 = 128 dimensional feature vector is produced.

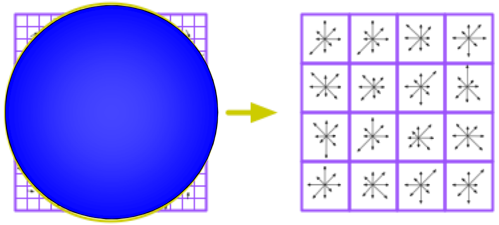
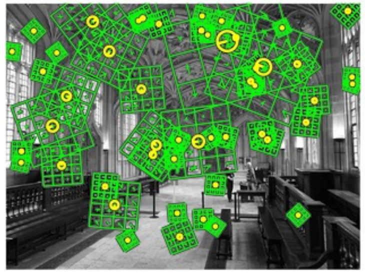


Image gradients | Keypoint descriptor

Image from: Jonas Hurrelmann

SIFT – Descriptor Vector

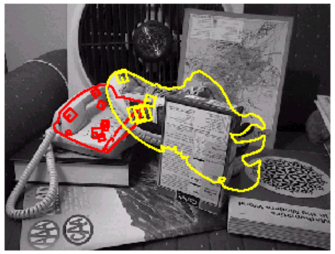
STEP 4: Compute feature vector



Object Recognition

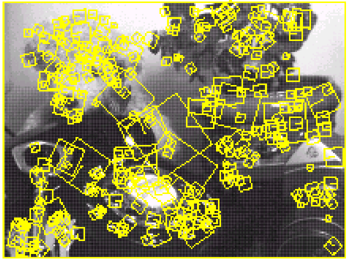
- Only 3 keys are needed for recognition, so extra keys provide robustness

Recognition under occlusion



Test of illumination Robustness

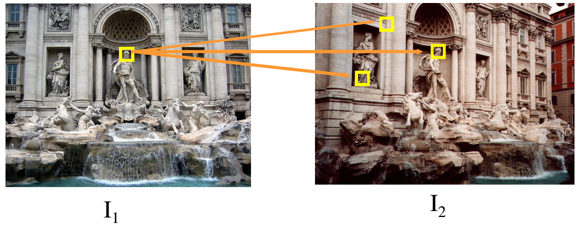
- Same **image** under differing illumination



273 keys verified in final match

Matching SIFT Features

- 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. Accept if below threshold.



Matching SIFT Features

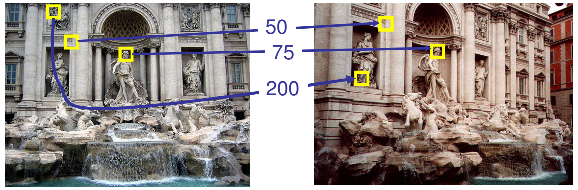
22 correct matches

Matching SIFT Features

33 correct matches

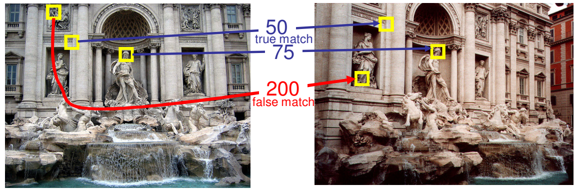
Matching SIFT Features

How to evaluate the performance of a feature matcher?



Matching SIFT Features

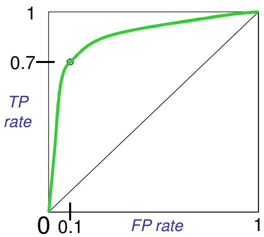
- Threshold t affects # of correct/false matches



- True positives (TP) = # of detected matches that are correct
- False positives (FP) = # of detected matches that are incorrect

Matching SIFT Features

- ROC Curve
  - Generated by computing (FP, TP) for different thresholds.
  - Maximize area under the curve (AUC).

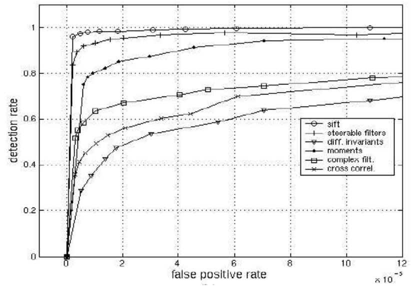


http://en.wikipedia.org/wiki/Receiver_operating_characteristic

Evaluating SIFT Features

- Empirically found[2] to show very good performance, invariant to *image rotation*, *scale*, *intensity change*, and to moderate *affine* transformations

Scale = 2.5
Rotation = 45^0



[1] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". Accepted to IJCV 2004
[2] K.Mikolajczyk, C.Schmid. "A Performance Evaluation of Local Descriptors". CVPR 2003

Example - Mosaicing

Source: Alexei Efros

Example: Mosiacing (Panorama)

M. Brown and D. G. Lowe. Recognising Panoramas. ICCV 2003