BLG453E COMPUTER VISION

Fall 2021 Term

İTÜ

İstanbul Technical University
Computer Engineering Department

Instructor: Prof. Gözde ÜNAL

Teaching Assistant: Yusuf Hüseyin ŞAHİN

1

Learning Outcomes of the Course

Students will be able to:

1. Discuss the main problems of computer (artificial) vision, its uses and applications
2. Design and implement various image transforms: point-wise transforms, neighborhood operation-based spatial filters, and geometric transforms over images
3. Define and construct segmentation, feature extraction, and visual motion estimation algorithms to extract relevant information from images
4. Construct least squares solutions to problems in computer vision
5. Describe the idea behind dimensionality reduction and how it is used in data processing
6. Apply object and shape recognition approaches to problems in computer vision

2

Week 7: Template Matching and Feature Extraction: Hough Method, Detection of Lines, circles etc, Corner Detection

At the end of Week 7-8: Students will be able to:

3. Define and construct segmentation, feature extraction, and visual motion estimation algorithms to extract relevant information from images

3

Image Feature Extraction: Why we need features?

Example: Build a Panorama

This panorama was generated using **AUTOSTITCH** (freeware), available at http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

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

45/59 © R. Siegwart, I. Nourbakhsh

4

Image Feature Extraction: Why we need features?

How do we build panorama?

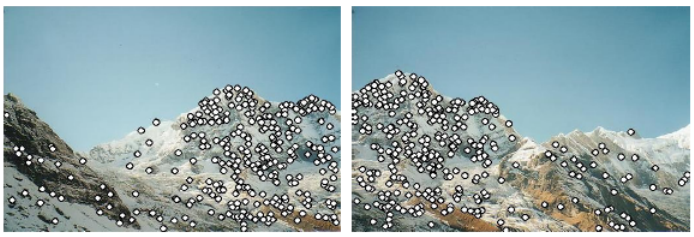
- We need to match (align) images

46/59

5

Matching with Features

•Detect feature points in both images

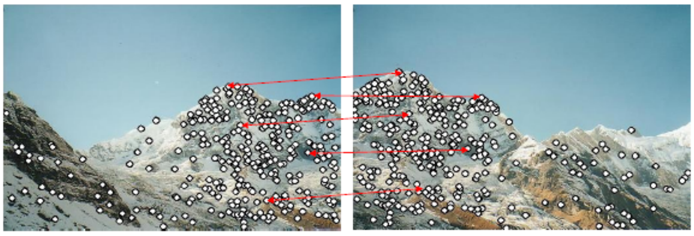


47/59

6

Matching with Features

- •Detect feature points in both images
- •Find corresponding pairs



48/59

7

Matching with Features

- •Detect feature points in both images
- •Find corresponding pairs
- •Use these pairs to align images

49/59

8

More Motivation:

Feature points are also used for:

- 3D Reconstruction
- Robot Navigation
- Object Recognition
- Indexing and database retrieval
- Image Alignment Panaromas
- Motion Tracking
- Other …

9

Motivation for Feature extraction and matching: 3D Vision– The Fundamental Problem

Problem: how to recover the 3-D geometry of the scene?
What makes this problem difficult?

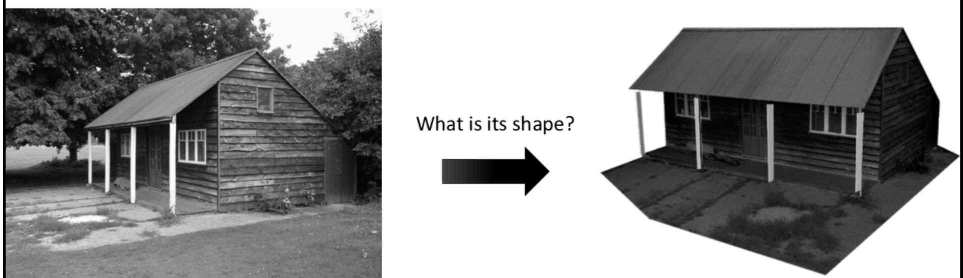


Image source: Internet

10

Motivation for Feature extraction and matching: 3D Vision– The Fundamental Problem

Problem: how to recover the 3-D geometry of the scene?
What makes this problem difficult?

A: we typically do NOT know the viewpoints from which the images were taken. Furthermore, some of the camera parameters such as the focal length is also unknown.

Input: Corresponding "features" in multiple perspective images.
Output: Camera pose, calibration, scene structure representation.

11

Traditional 3D Reconstruction Pipeline

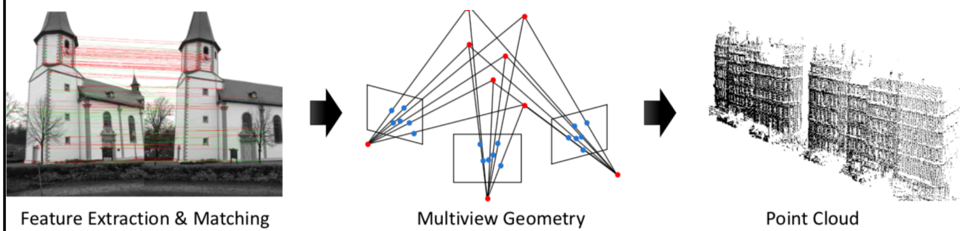
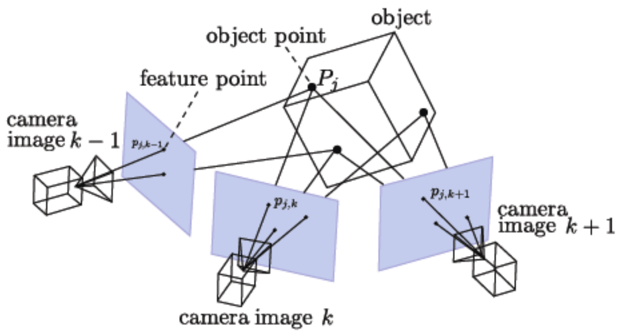


Image source: Internet

12

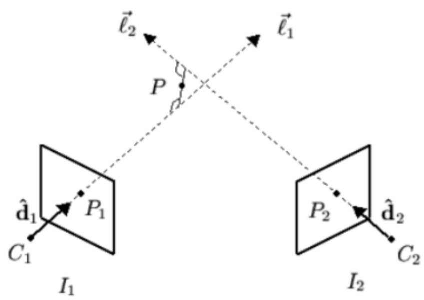
SLAM



The basic working principle of V-SLAM, from point observation and intrinsic camera parameters, the real time 3D structure of a scene is computed from the estimated motion of the camera.

13

3D Triangulation



14

Matching with Features

- Problem 1:
  - *Detect the* same *point* independently *in both images*

no chance to match!

We need a repeatable detector

50/59 © R. Siegwart, I. Nourbakhsh

15

Matching with Features

- Problem 2:
  - *For each point correctly recognize the corresponding one*

We need a reliable and distinctive descriptor

51/59 © R. Siegwart, I. Nourbakhsh

16

Most Popular Feature Detectors

* Hough method for line and other parametric structure detection (1959, 1972)

* Harris Corner Detector (1988)

* **SIFT Feature Detector (2004)** (one of the most popular feature extractors of early 2000's.)

Before we study the above feature detectors, first let us look at historically one of the most basic feature detectors: template matching next →

17

Template Matching

- Simple filtering method of detecting a particular feature in the image
- Provided that the appearance of this feature in the image is known accurately, we can detect it with a TEMPLATE.
- **Template:** a subimage that looks like the image of the object

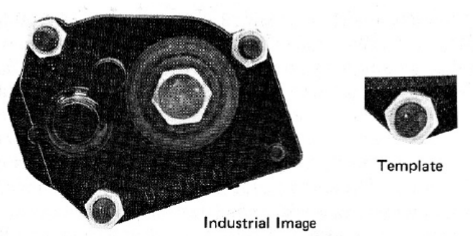
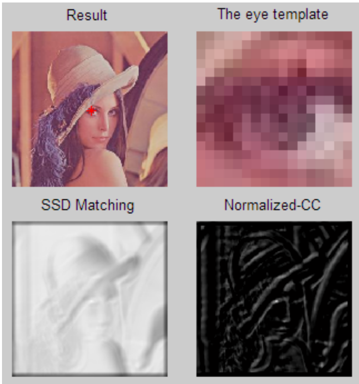


Fig. 3.3 An industrial image and template for a hexagonal nut.

18

Template Matching

- A similarity measure is computed to measure how well the template matches the image data for each possible template location
- Select the point of maximal match as possible template location



Fast/Robust Template Matching
by Dirk-Jan Kroon
Matlab central exchange

https://www.mathworks.com/matlabcentral/fileexchange/24925-fast-robust-template-matching

19

Template Matching

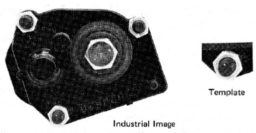


Fig. 3.3 An industrial image and template for a hexagonal nut.

- A widely used similarity measure between a function f and a template t:

Sum of Squared Differences (SSD):

$$d(\mathbf{y})^2 = \sum_{\mathbf{x}} \left[f(\mathbf{x}) - t(\mathbf{x}-\mathbf{y}) \right]^2$$

- Here $\sum_{\mathbf{x}}$ means $\sum_{x=-M}^{M} \sum_{y=-N}^{N}$, M and N: size of the template t
- If the image at point y is an exact match: $d(\mathbf{y}) \approx 0$

20

Template Matching

- Expand the SSD expression $d(\mathbf{y})^2$

$$d(\mathbf{y})^2 = \sum_{\mathbf{x}} \left[f^2(\mathbf{x}) - 2f(\mathbf{x})\ t(\mathbf{x}-\mathbf{y}) + t^2(\mathbf{x}-\mathbf{y}) \right]$$

- t^2 is constant, and assuming f^2 is constant, what is left is

Cross Correlation between f and t:

$$CC_{ft}(\mathbf{y}) = \sum_{\mathbf{x}} f(\mathbf{x})\ t(\mathbf{x}-\mathbf{y})$$

Maximized when portion of image f under t is identical to t!

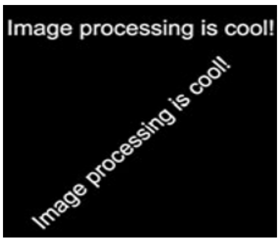
- Like convolution: Template-matching calculations are visualized as template being shifted across the image to different offsets, multiplied, and products are added

21

THQ: Template Matching

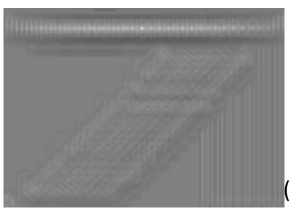
Template:

Image:



Template matching result:


(i)


(ii)


(iii)

22

Template Matching

Cross Correlation (CC): $$CC_{ft}(\mathbf{y}) = \sum_{\mathbf{x}} f(\mathbf{x})\ t(\mathbf{x}-\mathbf{y})$$

Note that this is the same as convolution of f(x) by t(-x)

→ template matching is a kind of Filtering for Object Detection

Template	*Image*	*Correlation*
1 1 1	1 1 0 0 0	7 4 2 x x
1 1 1	1 1 1 0 0	5 3 2 x x
1 1 1	1 0 1 0 0	2 1 9 x x
	0 0 0 0 0	x x x x x
	0 0 0 0 8	x x x x x
		x = undefined

Fig. 3.4 (a) A simple template. (b) An image with noise. (c) The aperiodic correlation array of the template and image. Ideally peaks in the correlation indicate positions of good match. Here the correlation is only calculated for offsets that leave the template entirely within the image. The correct peak is the upper left one at 0, 0 offset. The "false alarm" at offset 2, 2 is caused by the bright "noise point" in the lower right of the image.

23

THQ: 2Dimensional Correlation : Template detection

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & A & \bullet & \bullet & \bullet \\ \bullet & \bullet & \times & \bullet & \bullet \\ \bullet & \bullet & \bullet & B & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}$$

24

Template Matching

e.g. A bright spot in the image can badly influence the correlation matching!

CC may work if the average image intensity f varies slowly compared to the template size

$$CC_{ft}(u,v) = \sum_{x,y} f(x,y)\ t(x-u, y-v)$$

Normalized Cross Correlation:

$$NCC_{ft}(u,v) = \frac{\sum_{x,y}\left[f(x,y)-\bar{f}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\left\{\sum_{x,y}\left[f(x,y)-\bar{f}\right]^2 \sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2\right\}^{0.5}}$$

where the means are subtracted and divided by their standard deviations, NCC less dependent on the local properties of the template and the input image than CC

25

THQ: 2Dimensional Normalized Correlation : Template detection

$$\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 7 \end{bmatrix} = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & A & \bullet & \bullet & \bullet \\ \bullet & \bullet & \times & \bullet & \bullet \\ \bullet & \bullet & \bullet & B & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet \end{bmatrix}$$

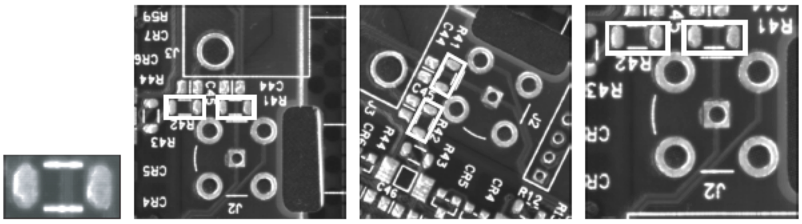
26

Normalized Cross Correlation

NCC is typically used in template matching problems of computer vision

$$NCC_{ft}(u,v) = \frac{\sum_{x,y}\left[f(x,y)-\bar{f}\right]\left[t(x-u,y-v)-\bar{t}\right]}{\left\{\sum_{x,y}\left[f(x,y)-\bar{f}\right]^2 \sum_{x,y}\left[t(x-u,y-v)-\bar{t}\right]^2\right\}^{0.5}}$$

Could the form of NCC above help with matching the parts below?



A B C D

When the pattern is rotated, scaled, or is imaged in a different lighting condition?

Different rotations and scales should be taken into account

Figure from National Instruments website: what to expect from a pattern matching tool?

27

Image Template

Advanced Methods (you are not responsible) but explore on your own

For illumination and appearance invariance: Use edge-based or sketch-like features of the object that is searched for

Sketch Template

http://www.wisdom.weizmann.ac.il/~vision/SelfSimilarities.html

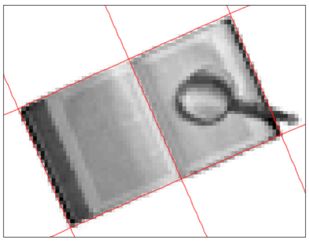
28

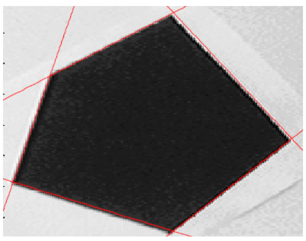
Next: Feature Extraction from images

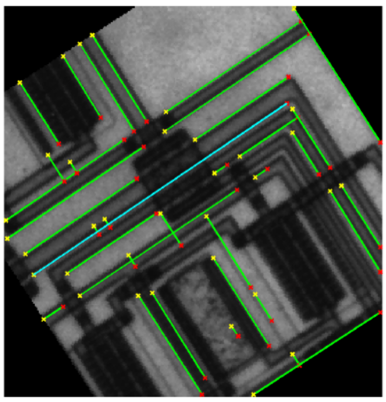
Lines
Quadratic forms
Corners
...

29

Line Detection and Parameterization: Hough Transform







30

Hough Method for Line Detection

Goal: Detecting Straight Lines in Images

Assume we have detected some points likely to be on lines in an image

Hough technique will organize these points into straight lines by considering all possible straight lines at once and rating each on how well it explains the data

31

Hough Method for Line Detection

32

Hough Method for Line Detection

Equation for a line: $y = mx + c$

Q:What are the lines that could pass through point (x',y') below?
A: All the lines that satisfy $y' = mx' + c$

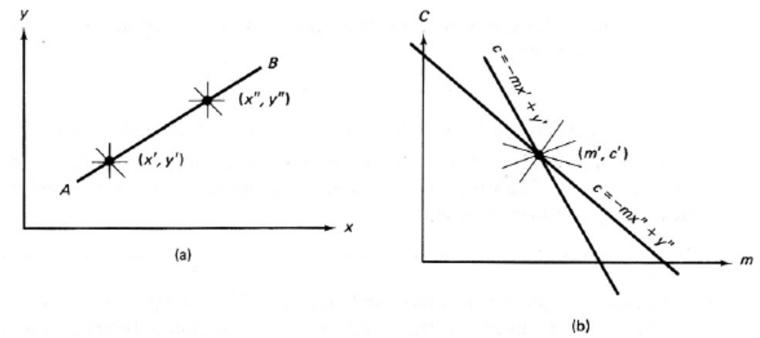
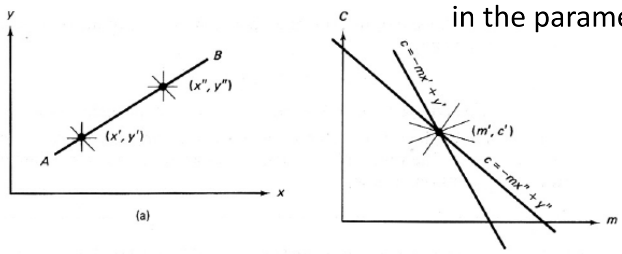


Fig. 4.5 A line (a) in image space; (b) in parameter space.

33

Hough Method for Line Detection



- Fixing (x',y'), the line equation is now in the parameter space (m,c):

$$c = -mx + y$$

- A second point (x'',y'') will also have a line in the parameter space

→ All such lines will intersect at (m',c') → this point in the parameter space corresponds to the line AB in the image space with coord (x,y)

34

Hough Method for Line Detection

Algorithm 4.1: Line Detection with the Hough Algorithm

1. Quantize parameter space between appropriate maximum and minimum values for c and m.
2. Form an accumulator array $A(c, m)$ whose elements are initially zero.
3. For each point (x,y) in a *gradient* image such that the strength of the gradient exceeds some threshold, increment all points in the accumulator array along the appropriate line, i.e.,

$$A(c, m) := A(c, m) + 1$$

for m and c satisfying $c = -mx + y$ within the limits of the digitization.

4. Local maxima in the accumulator array now correspond to collinear points in the image array. The values of the accumulator array provide a measure of the number of points on the line.

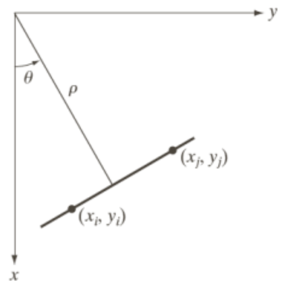
Q: Do you see where we might run into problem with this algorithm?

35

Hough Method for Line Detection

In the previous algorithm, when there is a vertical line in the image, m: the slope of the line is ∞

Recommended Rather than (c,m) parameters, use the line polar parameterization with : (ρ,θ)



$$x\cos\theta + y\sin\theta = \rho$$

Same idea as before, but with different parameterization of the line: now the lines become sinusoidal curves in the parameter space →

36

Hough Method for Line Detection

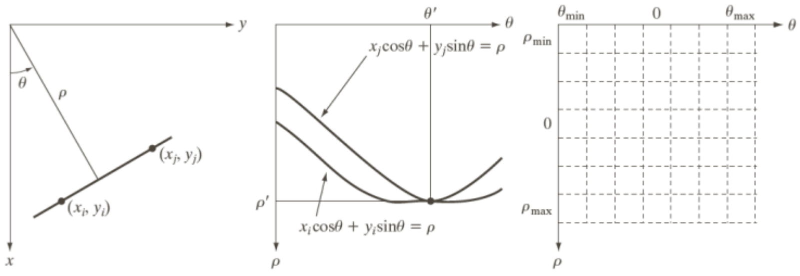


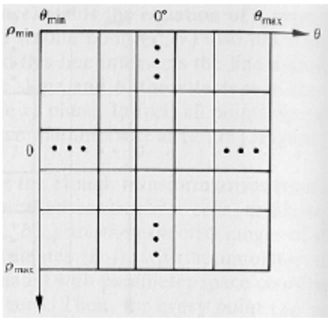
FIGURE 10.32 (a) (ρ, θ) parameterization of line in the xy-plane. (b) Sinusoidal curves in the $\rho\theta$-plane; the point of intersection (ρ', θ') corresponds to the line passing through points (x_i, y_i) and (x_j, y_j) in the xy-plane. (c) Division of the $\rho\theta$-plane into accumulator cells.

The lines with the parameterization: $x\cos\theta + y\sin\theta = \rho$

become sinusoidal curves in the parameter space

Hough Algorithm

1. Quantize the parameter space
$P[\rho_{min}, \ldots, \rho_{max}][\theta_{min}, \ldots, \theta_{max}]$ (accumulator array)

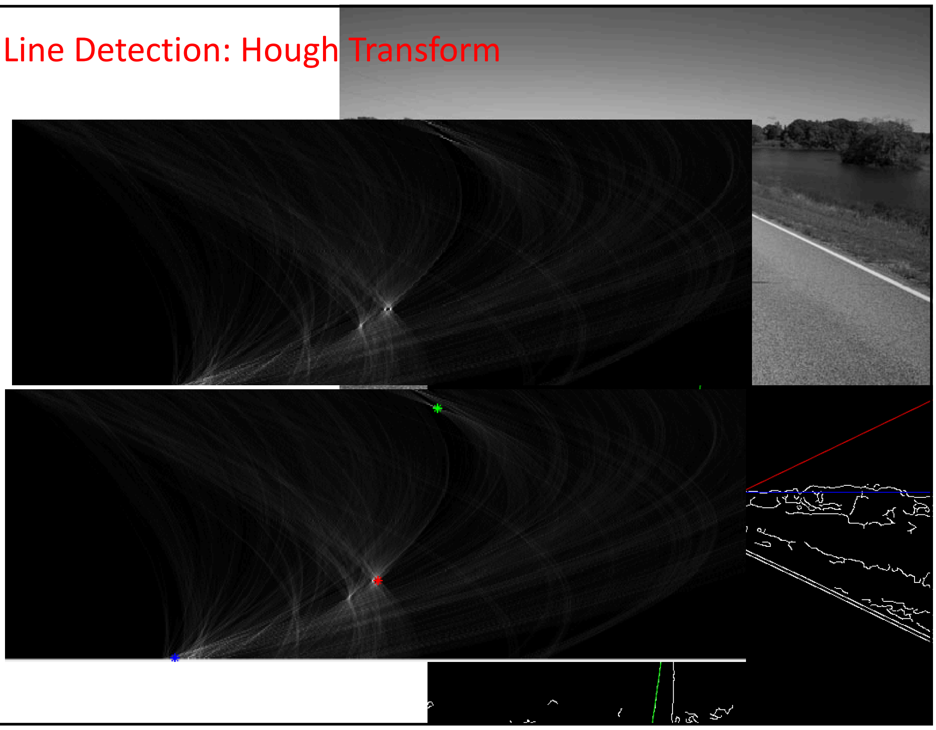


2. For each edge point (x, y)

```
For(θ = θmin; θ ≤ θmax; θ++) {
  ρ = xcosθ + ysinθ; /* round off if needed *
  (P[ρ][θ])++; /* voting */
}
```

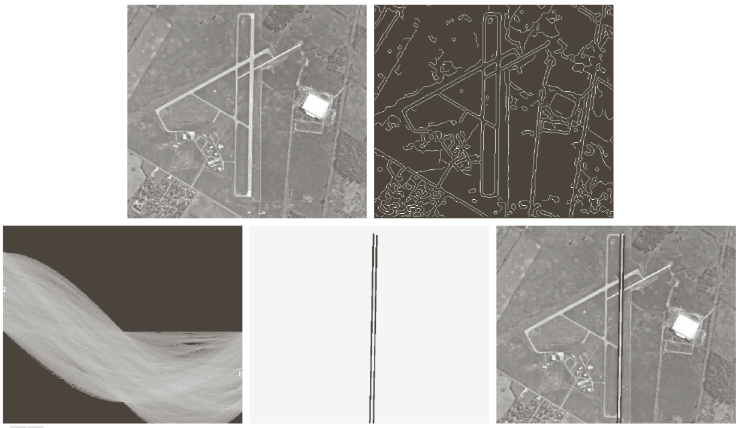
3. Find local maxima in $P[\rho][\theta]$

Line Detection: Hough Transform



39

Line Detection: Hough Transform



a b
c d e

FIGURE 10.34 (a) A 502 × 564 aerial image of an airport. (b) Edge image obtained using Canny's algorithm. (c) Hough parameter space (the boxes highlight the points associated with long vertical lines). (d) Lines in the image plane corresponding to the points highlighted by the boxes. (e) Lines superimposed on the original image.

Digital Filtering of Images, Gonzalez and Woods Book

40

General Hough Method

- Can be extended to other functions

where a is a parameter vector:

$$f(\mathbf{x}, \mathbf{a}) = 0$$

For example, Circle : $(x-a)^2 + (y-b)^2 = r^2$

$$f(\mathbf{x}, \mathbf{a}) = (x-a)^2 + (y-b)^2 - r^2$$

$$\mathbf{x} = (x, y) \qquad \mathbf{a} = (a, b, r)$$

known feature coordinates, e.g. Edge point coordinates

Parameter vector to be estimated

41

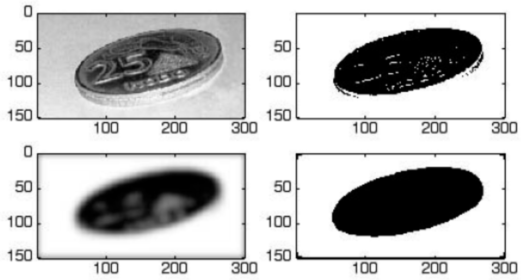
General Hough Method

Try to obtain these yourself

Check out for further information on conic sections:
https://www.andrews.edu/~calkins/math/webtexts/numb19.htm

42

Ellipse Detection



General Ellipse equation (including rotation) is a quadratic form:

$$ax^2 + bxy + cy^2 + dx + ey + f = 0$$

We can use the LEAST SQUARES technique, to solve for the parameters of the ellipse.

Recall: Ellipse not centered but with no rotation:

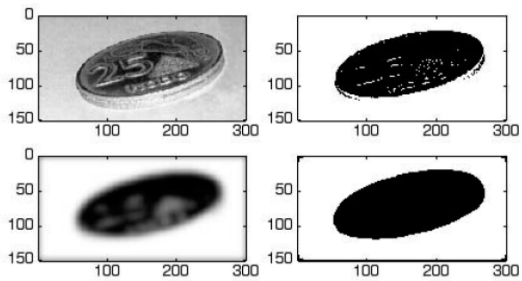
$$((x - x0)/a)^2 + ((y - y0)/b)^2 = 1$$

43

Ellipse Detection

ON BOARD: Construct the LEAST SQUARES problem using this equation, and solve for the parameters of the ellipse

E.g. Take your ellipse equation: $$ax^2 + bxy + cy^2 + dx + ey = 1$$



Overlay output ellipse curve (in blue) with estimated parameters: a, b, c, d, e, f :

44

THQ: Hough Method Summary

A preprocessing step extracts landmark points of interest from the image

The feature of interest should be represented by a parametric model with finite number of parameters

Parameter values are exhaustively traversed to accumulate the result of evaluation of the landmark points in the feature's parametric representation

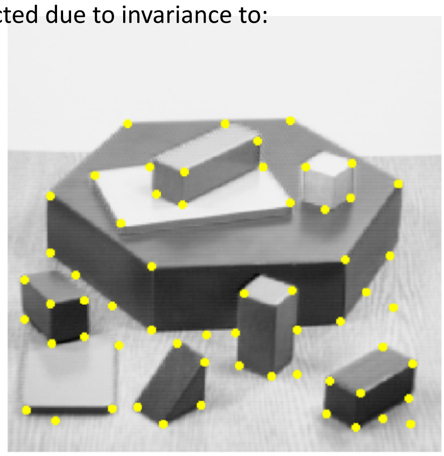
The detection is carried out in the parameter space by analyzing the maxima

45

Popular Feature Detectors

Popular interest points in images are selected due to invariance to:

- scale, rotation
- illumination variation
- image noise



A good feature should be distinctive

Next: corner detection

E.g. Corners

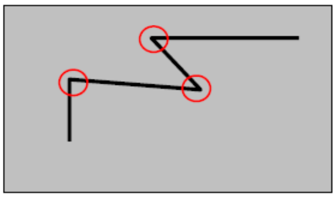
Harris Detector (1988)

46

Corner Detection: Harris Operator

Harris corner detector

An introductory example:

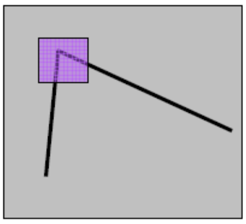


54/59 C.Harris, M.Stephens. "A Combined Corner and Edge Detector". 1988 © R. Siegwart, I. Nourbakhsh

47

The Basic Idea

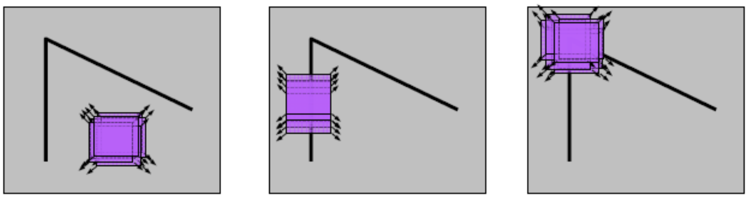
- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



55/59 © R. Siegwart, I. Nourbakhsh

48

Harris Detector: Basic Idea



"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

56/59

49

Corner (Harris) Detector (Mathematics)

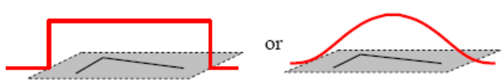
Let's analyze the local intensity changes :

Change of intensity for the shift [u,v]:

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

Window function | Shifted intensity | Intensity

Window function $w(x,y)$ =



1 in window, 0 outside — or — Gaussian

57/59

50

Corner Detector (Mathematics)

For small shifts (u,v), we have the following approximation (after Taylor series expansion on I(x+u,y+v) and expanding the quadratic term):

$$E(u,v) \cong \begin{bmatrix} u & v \end{bmatrix} G \begin{bmatrix} u \\ v \end{bmatrix}$$

where G is a 2x2 matrix computed from image derivatives in the given window:

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

G is called the Image Structure Tensor

51

Image Structure Tensor

$$G(x,y) = \begin{bmatrix} \sum_W (I_x(x_i,y_i))^2 & \sum_W (I_x(x_i,y_i) I_y(x_i,y_i)) \\ \sum_W (I_x(x_i,y_i) I_y(x_i,y_i)) & \sum_W (I_y(x_i,y_i))^2 \end{bmatrix}$$

- W is the window of a fixed size in your image, (xi,yi) pixel coordinates in that window.

Q: How do you calculate G or Ix and Iy?

I_x and I_y are the local approximations to the first order partial derivatives of the image J, which is the filtered image I with a Gaussian filter (as we've seen in previous week) :

$$I_x = \frac{\partial I}{\partial x} = \frac{I(x+1,y) - I(x-1,y)}{2} \qquad I_y = \frac{\partial I}{\partial y} = \frac{I(x,y+1) - I(x,y-1)}{2}$$

52

Corner Detector (Mathematics)

$$G(x,y)=\begin{bmatrix}\sum_W (I_x(x_i,y_i))^2 & \sum_W (I_x(x_i,y_i)I_y(x_i,y_i)) \\ \sum_W (I_x(x_i,y_i)I_y(x_i,y_i)) & \sum_W (I_y(x_i,y_i))^2\end{bmatrix}$$

Q: How to analyze G? (or ~ intensity change in shifting window?)

A: Perform eigenvalue analysis on G

Recall:

$$G = V\Sigma V^{-1}$$

$\lambda_2 \geq \lambda_1$: eigenvalues of G

Σ : covariance matrix of the data

V : Eigenvector matrix of G

$$G = V\begin{bmatrix}\lambda_2 & 0 \\ 0 & \lambda_1\end{bmatrix}V^{-1}$$

53

Corner Detector using Structure Tensor

Perform Eigenvalue analysis on G

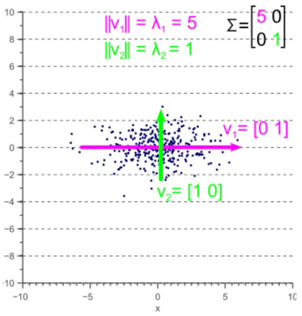
V : Eigenvector matrix of G

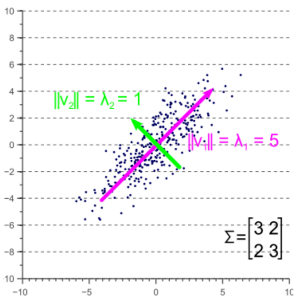
Σ : covariance matrix of the data

$$G = V\begin{bmatrix}\lambda_2 & 0 \\ 0 & \lambda_1\end{bmatrix}V^{T}$$

$\lambda_2 \geq \lambda_1$

: eigenvalues of G



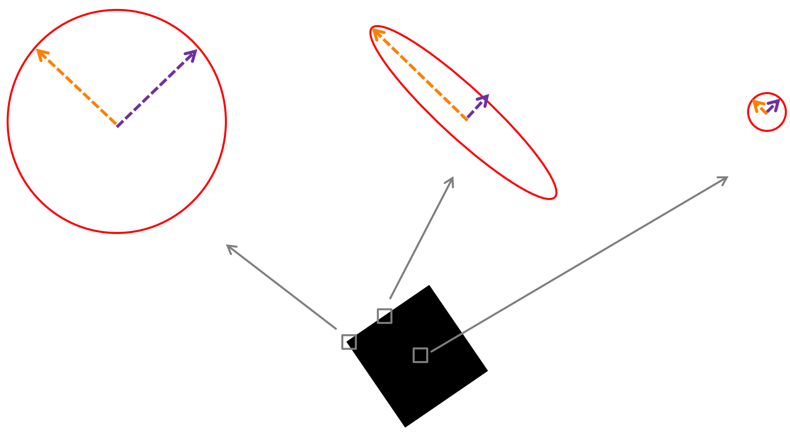


http://www.visiondummy.com/2014/04/geometric-interpretation-covariance-matrix/

54

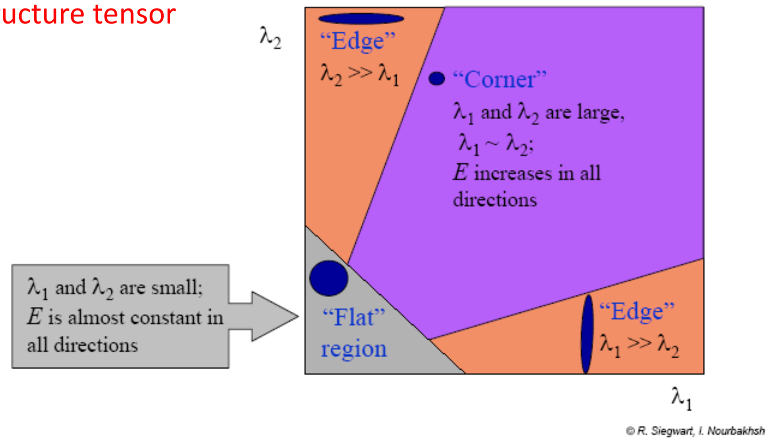
Eigenvalue analysis

- There are three common scenarios:
  - Both eigenvalues are large: **corner**
  - One eigenvalue is large, the other small: edge
  - Both eigenvalues small: homogeneous region



55

Classification of image points using eigenvalues of G
G: image structure tensor



1) If both eigenvalues of G are small, local autocorrelation function (E(u,v)) is FLAT, little change in any direction

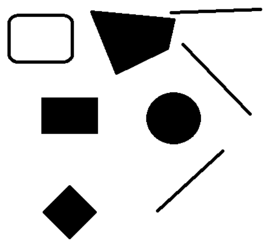
2) If one eigen value is high, the other low, then only there is high shift in one direction → EDGE

3) If both eigen values are high, the local autocorr fn. is sharply peaked → CORNER

56

Minimum eigenvalue corner detection

- Since the eigenvalues are positive, one can look at the *smaller* of the two eigenvalues. If this is *large enough*, then a corner has been identified.
- Example: looking at the value of λ_2 [Shi and Tomasi, '94]



Image

λ_2

```
% Compute min eigenvalue: see Equation 7 of
% http://www.soest.hawaii.edu/martel/Courses/GG303/Eigenvectors.pdf
lambda2 = 0.5*( (s11+s22)-((s11-s22).^2+4*s12.^2).^0.5 );
figure; imshow(lambda2, []);
```

57

Harris Corner Detector

Harris proposed looking for corners based on a cornerness function

$$C = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2$$

$$\det(G) = \lambda_1\lambda_2$$

$$trace(G) = \lambda_1 + \lambda_2$$

k – empirical constant, a small value in the range: k=0.04 – 0.06

59

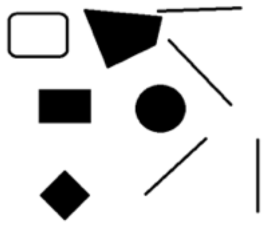
Harris Corner Detector defines

A Measure of Corner Response based on evalues of G:

$$R = C(G) = \det(G) - k\ trace^2(G)$$

$$\det(G) = \lambda_1 \lambda_2$$
$$trace(G) = \lambda_1 + \lambda_2$$

e.g. set k=0.05



Image

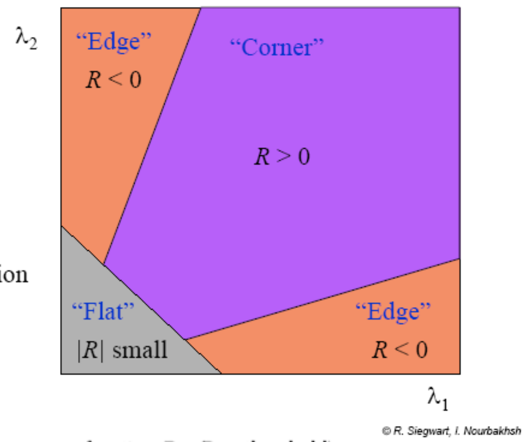
C

→ Using the 2x2 local structure tensor matrix G, the Harris corner detector THRESHOLDS the quantity R (or =C).

60

Harris Detector: Mathematics

- R depends only on eigenvalues of G
- R is large for a corner
- R is negative with large magnitude for an edge
- $|R|$ is small for a flat region

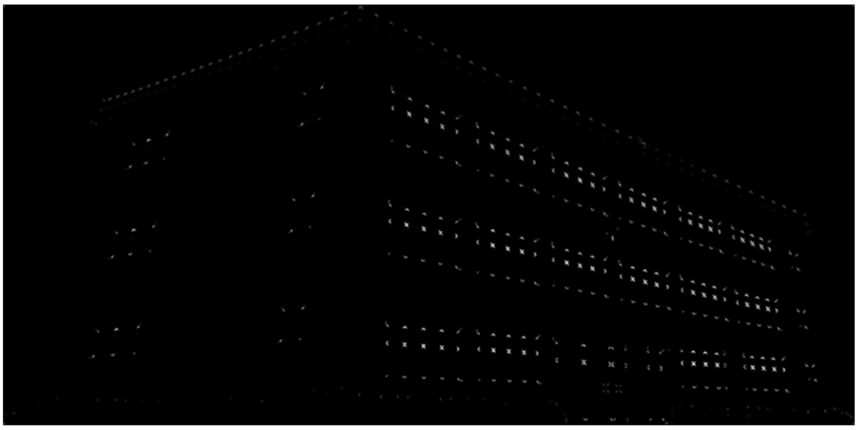


- The Algorithm:
  - *Find points with large corner response function* R (R > *threshold)*
  - *Take the points of local maxima of* R

62

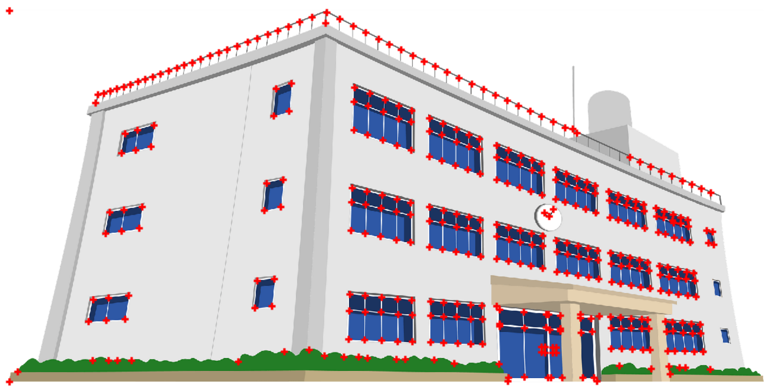
Harris Corner Detector on Clipart example

63



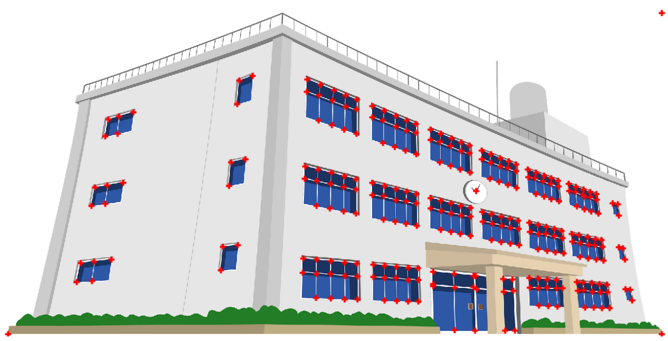
64

R is thresholded by 10



65

R is thresholded by 50



66

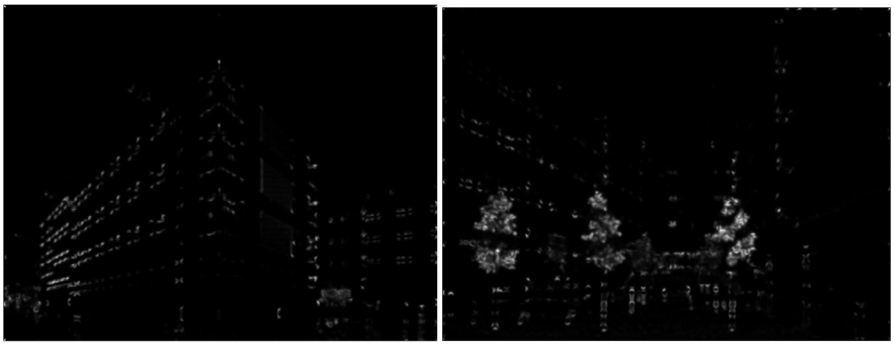
R is thresholded by 200

67

Harris Corner Detector Example

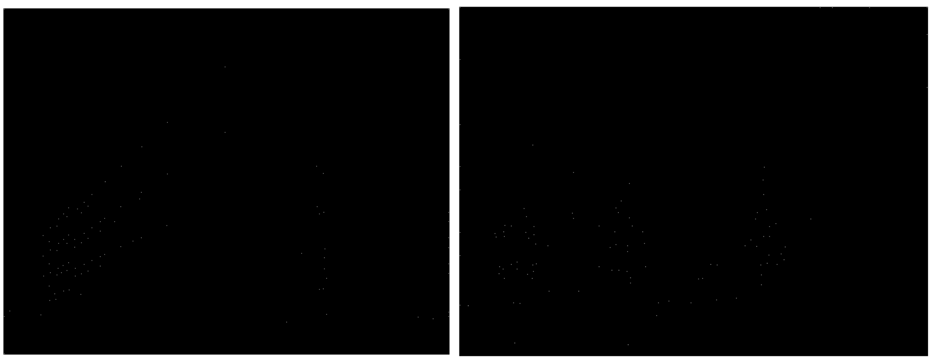
68

Corner Response R



69

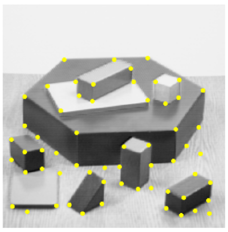
Those points with both Large Corner Response R >Threshold and Local Maxima of R



70

71

THQ: Corner detection Summary (Harris method/Image structure tensor)



Which of the following is NOT true for a corner detector?

A corner feature in an image has a significant change along only one direction

Significant intensity changes in more than one direction are required

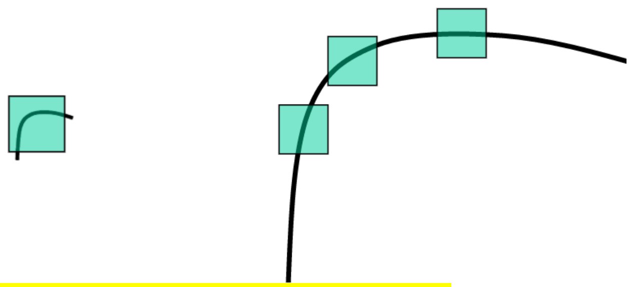
Popular corner detector Harris detector is based on the image structure tensor, which is derived from a measure of local intensity differences

Image structure tensor at a point is calculated using image spatial derivatives in two orthogonal directions over a window of fixed size around the given point

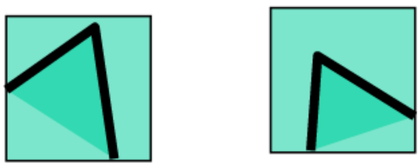
Two large eigenvalues in the image structure tensor at a pixel point indicates a corner location

72

THQ: Feature exraction Scale: T or F: Image structure tensor can be calculated using a fixed window size to detect both of the corners shown in the picture.



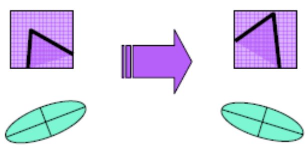
THQ: Feature exraction Rotation: T or F:



73

Harris Detector: Some Properties

- Rotation invariance



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

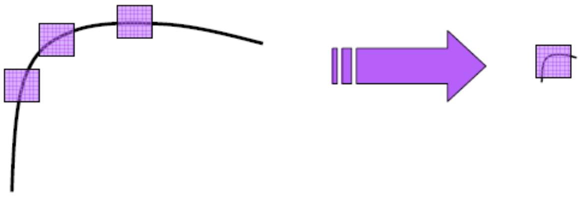
Corner response R is invariant to image rotation

69/59

74

Harris Detector: Some Properties

- But: non-invariant to *image scale*!



All points will be classified as edges

Corner !

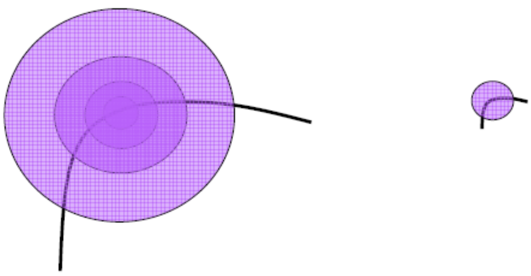
70/59

75

Feature Detection

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

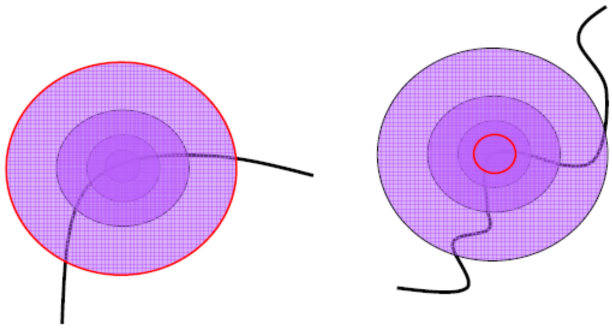


72/59

76

Scale Invariant Detection

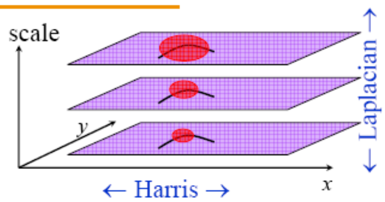
- The problem: how do we choose corresponding circles ***independently*** in each image?



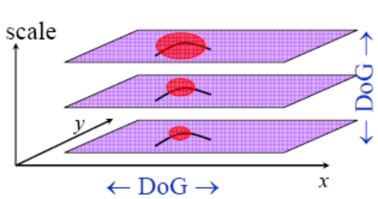
Recall: to create scale-space, typically, image gradient based operators are used

Scale Invariant Detectors

- Harris-Laplacian[1]
  Find local maximum of:
  - *Harris corner detector in space (image coordinates)*
  - *Laplacian in scale*



- SIFT (Lowe)[2]
  Find local maximum of:
  - Difference of Gaussians in space and scale



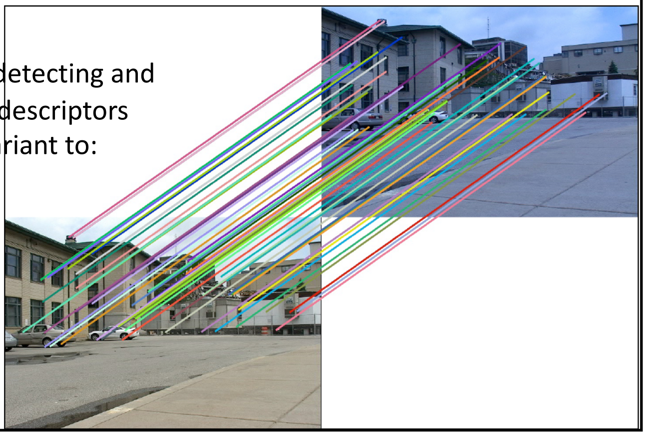
[1] K.Mikolajczyk, C.Schmid. "Indexing Based on Scale Invariant Interest Points". ICCV 2001
[2] D.Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". IJCV 2004

Feature Matching Problem

- After you detect features in 2 images, how to match them?
- You have to define a descriptor for a feature: The simplest solution is the intensities of its spatial neighbors. This might not be robust to brightness change or small shift/rotation.
- Therefore, typically, features are based on gradients of images

e.g. SIFT detector:
SIFT is an approach for detecting and extracting local feature descriptors that are reasonably invariant to:

- Rotation
- Scaling
- Small changes in viewpoint
- illumination



79

SIFT
(Scale Invariant Feature Transform)

80

80

SIFT stages:

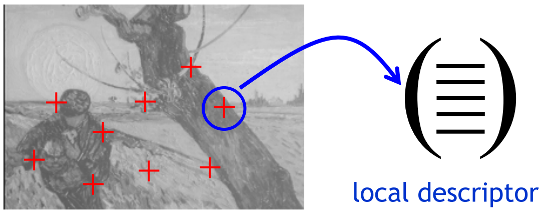
- Scale-space extrema detection
- Keypoint localization

detector

- Orientation assignment
- Keypoint descriptor generation

descriptor

matching



A 500x500 image gives about 2000 features

81

81

We want invariance!!!

- Good features should be robust to all sorts of nastiness that can occur between images.

82

82

Types of invariance

- Illumination

83

83

Types of invariance

- Illumination
- Scale

84

84

Types of invariance

- Illumination
- Scale
- Rotation

85

85

Types of invariance

- Illumination
- Scale
- Rotation
- Affine

86

86

Types of invariance

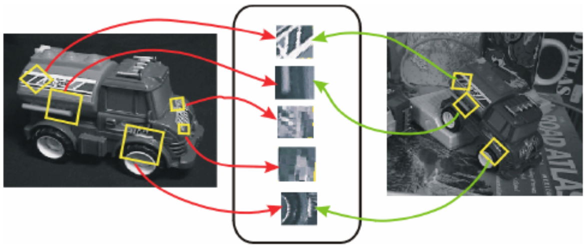
- Illumination
- Scale
- Rotation
- Affine
- Full Perspective

87

87

Invariant Local Features

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



81/59

© R. Siegwart, I. Nourbakhsh

88

Detection stages for SIFT features

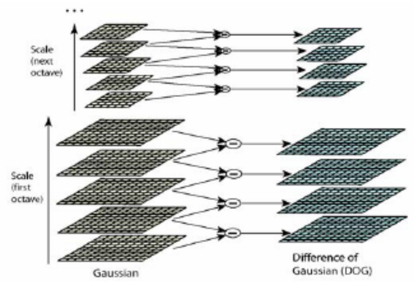
- Scale-space extrema detection
- Keypoint localization
- Orientation assignment
- Generation of keypoint descriptors.

82/59

89

Scale-space extrema detection

- The first step toward the detection of interest points is the convolution of the image with Gaussian filters at different scales, and the generation of Difference-of-Gaussian images (DoG) from the difference of adjacent blurred images

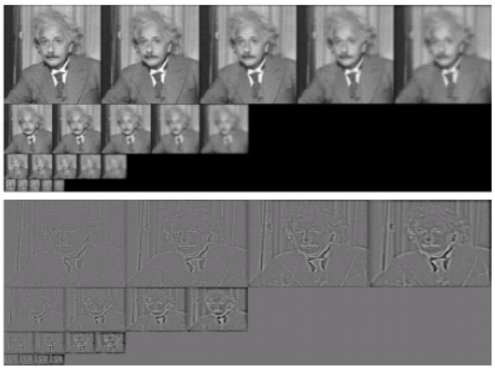


83/59

90

Scale-space extrema detection

- Gaussian blurred images grouped by octave
- Difference of Gaussian images (DoG) grouped by octave

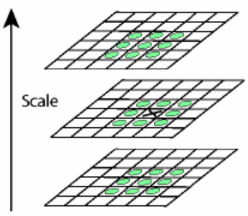


84/59

91

Scale-space extrema detection

- SIFT keypoints are identified as local maxima or minima of the DoG images across scales.
- Each pixel in the DoG images is compared to its 8 neighbors at the same scale, plus the 9 corresponding neighbors at neighboring scales.
- If the pixel is a local maximum or minimum, candidate keypoint.

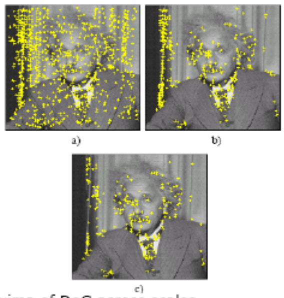


85/59

92

Key point localization

- For each candidate keypoint:
  - *Interpolation of nearby data is used to accurately determine its position.*
  - *Keypoints with low contrast are removed (b).*
  - *Responses along edges are eliminated (c).*

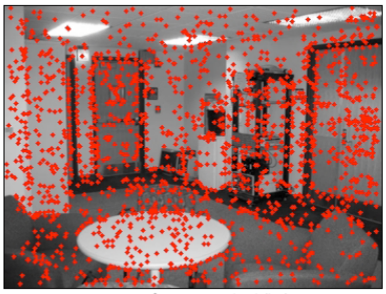


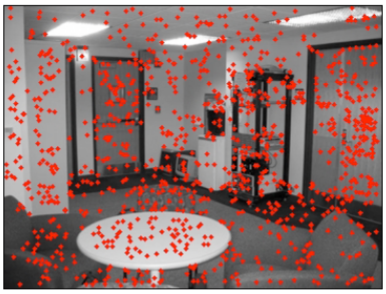
a) Maxima of DoG across scales
b) Remaining keypoints after removal of low contrast points
c) Remaining keypoints after removal of edge responses

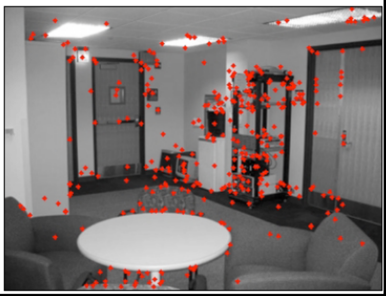
86/59

93

Remove low contrast and edges





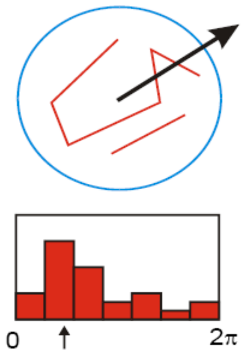


94

94

Orientation assignment

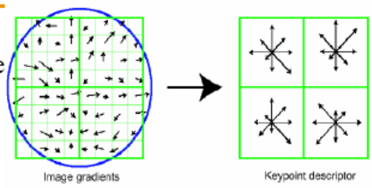
- To determine the keypoint orientation, a gradient orientation histogram is computed in the neighborhood of the keypoint.
- Peaks in the histogram correspond to dominant orientations. If more than one peak is found, a separated feature is assigned to the same point location.
- All the properties of the keypoint are measured relative to the keypoint orientation, this provides invariance to rotation.



87/59

95

Generation of keypoint descriptor



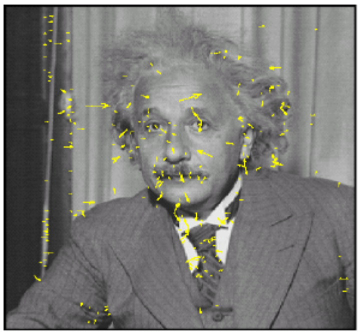
- A keypoint descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left.
- These samples are then accumulated into orientation histograms summarizing the contents over 4x4 sub regions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region.
- The descriptor is formed from a vector containing the values of all the orientations histogram entries, corresponding to the arrows of the right side.
- This vector is then normalized to enhance invariance to changes in illumination.

89/59

96

Key point localization

- Final keypoints with selected orientation and scale



88/59

97

Advantages of SIFT features

- Locality: features are local, so robust to occlusion and clutter (no prior segmentation)
- Distinctiveness: individual features can be matched to a large database of objects
- Quantity: many features can be generated for even small objects
- Efficiency: close to real-time performance

90/59

98

SIFT output: Feature Matching example

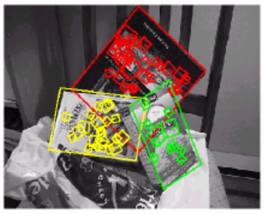
SIFT Detector Application Examples

Place recognition

SIFT Detector Application Examples

Planar recognition

- Planar surfaces can be reliably recognized at a rotation of 60° away from the camera
- Affine fit approximates perspective projection
- Only 3 points are needed for recognition



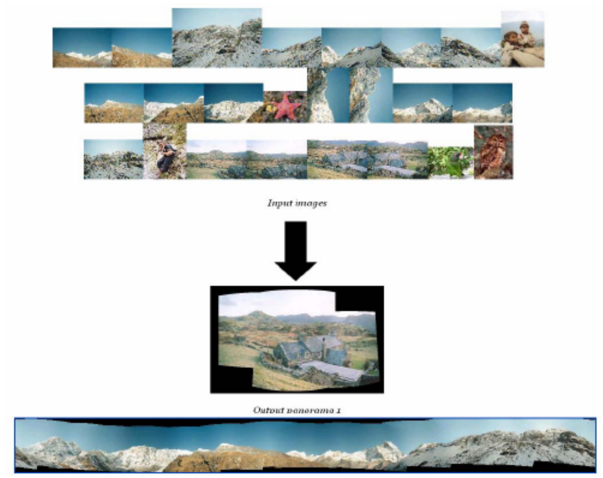
94/59

© R. Siegwart, I. Nourbakhsh

101

SIFT Detector Application Examples

Multiple panoramas from an unordered image set



99/59

© R. Siegwart, I. Nourbakhsh

102

END OF LECTURE on Feature Extraction

Recall Learning objectives of Week 8: Students are able to:

3. Define and construct segmentation, feature extraction, and visual motion estimation algorithms to extract relevant information from images

Next assignment: Work on feature extraction and template matching

Reading Assignments:

[Klette Book] 1.1, 1.3, 2.1.1, 2.1.2

[Gonzalez and Woods]: Chap 3.1 – 3.3

Those interested: Advanced Feature Extractors like SIFT: David Lowe, Distinctive Image Features from Scale-Invariant Key Points, International Journal of Computer Vision 2004.

103

Self Reading (Not included)

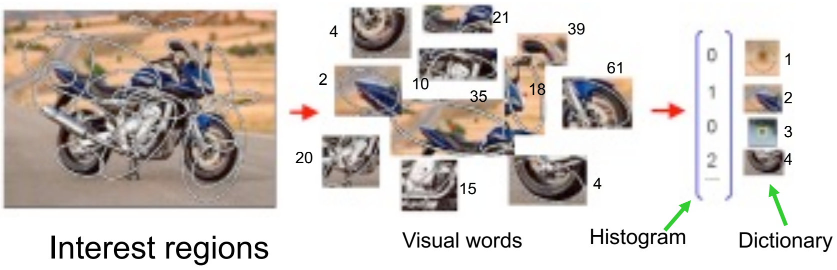
RANSAC : Random sample consensus used to eliminate outlier points in model fitting

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

Local Binary Patterns

104

Bag of features (words)



Stack visual word histograms
as columns in matrix

Throw away spatial information!

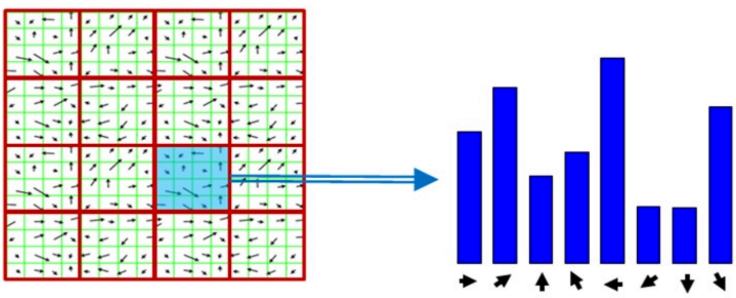
Note that for the bag of features approach to be effective, majority of each image's area must be occupied by the subject of the category, for example, an object or a type of scene.

For further information, e.g.: https://www.youtube.com/watch?v=iGZpJZhqEME&feature=plcp

111

Histogram of Oriented Gradients (HOG)

- The histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for the purpose of object detection.
- The technique counts occurrences of gradient orientation in localized portions of an image. This method is similar to that of scale-invariant feature transform descriptors (SIFT) and Speeded Up Robust Features (SURF) but differs in that it is computed on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization for improved accuracy.

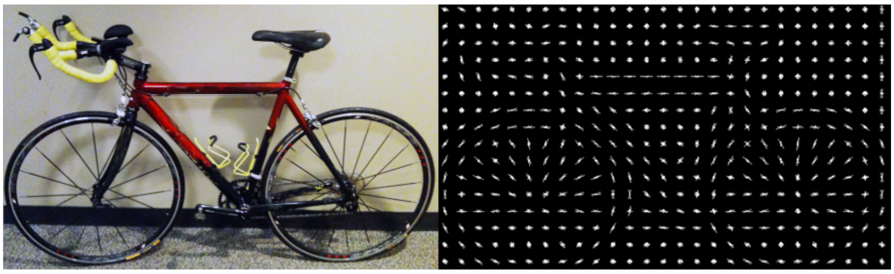


https://www.youtube.com/watch?v=0Zib1YEE4LU&feature=plcp

112

HOG Features

The HOG features are often used to detect objects such as people and cars. They are useful for capturing the overall shape of an object. For example, in the following visualization of the HOG features, you can see the outline of the bicycle.

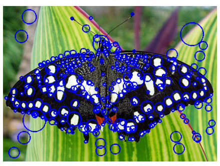


Matlab's trainCascadeObjectDetector

113

Blob Detection

- Blob detection methods are aimed at detecting regions in a digital image that differ in properties, such as brightness or colour, compared to surrounding regions. Informally, a blob is a region of an image in which some properties are constant or approximately constant; all the points in a blob can be considered in some sense to be similar to each other.
- Given some property of interest expressed as a function of position on the image, there are two main classes of blob detectors:
  - (i) differential methods, which are based on derivatives of the function with respect to position (such as The Laplacian of Gaussian), and
  - (ii) methods based on local extrema, which are based on finding the local maxima and minima of the function. With the more recent terminology used in the field, these detectors can also be referred to as interest point operators, or alternatively interest region operators

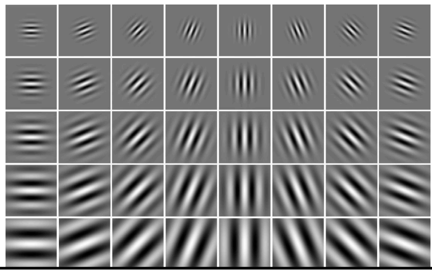


114

Gabor Filters

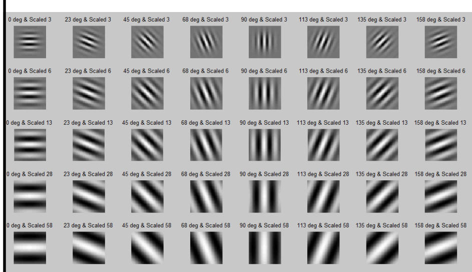
- Gabor Filter is a filter used for edge detection. Frequency and orientation representations of Gabor filters are similar to those of the human visual system, and they have been found to be particularly appropriate for texture representation and discrimination.
- In the spatial domain, a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave.

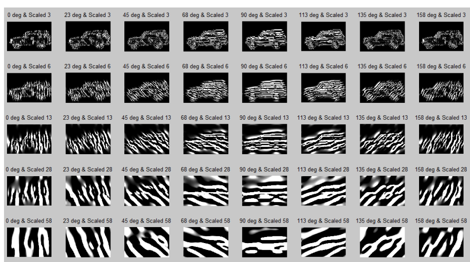
$$G(x,y) = \exp\left(-\frac{(X^2+\gamma^2 Y^2)}{2\sigma^2}\right)\cos\left(\frac{2\pi}{\lambda}X\right)$$



115

Gabor Filters





http://www.mathworks.com/matlabcentral/fileexchange/44630-gabor-feature-extraction

116