EE 417 Introduction to Computer Vision / EE 569 3D vision

Corner Detection

Instructor: Associate Prof. Mehmet Keskinöz

Communication Theory &Technologies (CTT) Group,
Electronics Engineering Program,
Computer Science and Engineering,
Cyber-Security Program
Faculty of Engineering and Natural Sciences

Email: keskinoz@sabanciuniv.edu

Sabancı Üniversitesi

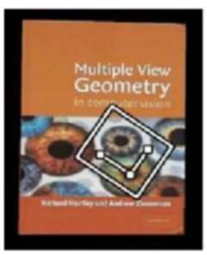
1

Features

Some slides from S. Seitz

2

Image Matching



3

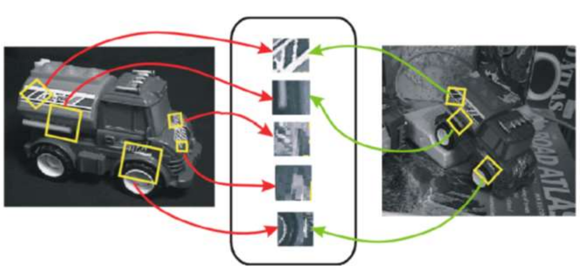
Image Matching

4

Invariant local features

Find features that are invariant to transformations

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



Feature Descriptors

5

Advantages of local features

Locality

- features are local, so robust to occlusion and clutter

Distinctiveness

- can differentiate a large database of objects

Quantity

- hundreds or thousands in a single image

Efficiency

- real-time performance achievable

Generality

- exploit different types of features in different situations

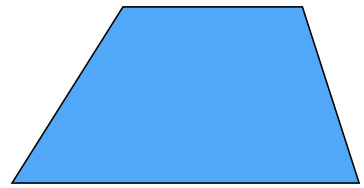
6

More motivation…

Feature points are used for:

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

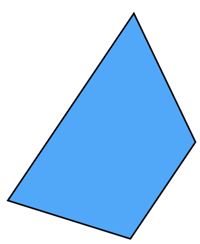
7



Pick a point in the image.
Find it again in the next image.

What type of feature would you select?

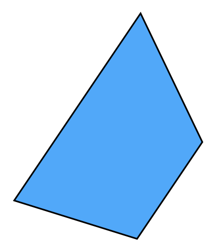
8



Pick a point in the image.
Find it again in the next image.

What type of feature would you select?

9



Pick a point in the image.
Find it again in the next image.

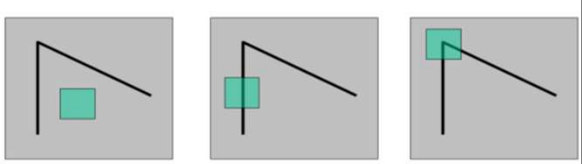
What type of feature would you select?
a corner

10

Local measures of uniqueness

Suppose we only consider a small window of pixels

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



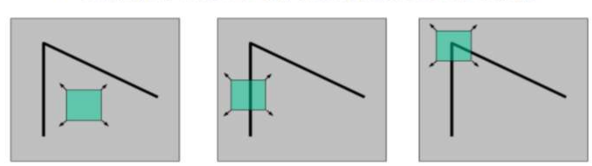
Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

11

Feature detection

Local measure of feature uniqueness

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



"flat" region:
no change in all directions

"edge":
no change along the edge direction

"corner":
significant change in all directions

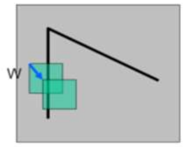
Slide adapted from Darya Frolova, Denis Simakov, Weizmann Institute.

12

Feature detection: the math

Consider shifting the window W by (u,v)

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



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

$\begin{bmatrix} u \\ v \end{bmatrix} \leftarrow$ shifting

13

Taylor Series for 2D Functions

$$f(x+u, y+v) = f(x,y) + uf_x(x,y) + vf_y(x,y) +$$

First partial derivatives

$$\frac{1}{2!}\left[u^2 f_{xx}(x,y) + uvf_{xy}x, y + v^2 f_{yy}(x,y)\right] +$$

Second partial derivatives

$$\frac{1}{3!}\left[u^3 f_{xxx}(x,y) + u^2 v f_{xxy}(x,y) + uv^2 f_{xyy}(x,y) + v^3 f_{yyy}(x,y)\right]$$

Third partial derivatives

$+ \ldots$ (Higher order terms)

First order approx

$$f(x+u, y+v) \approx f(x,y) + uf_x(x,y) + vf_y(x,y)$$

bilinear approximation

14

Small motion assumption

Taylor Series expansion of I:

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

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

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

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

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

Plugging this into the formula on the previous slide…

15

$$E = \sum_W [I(x+u, y+v) - I(x,y)]^2$$

$$\approx \sum [\cancel{I(x,y)} + uI_x + vI_y - \cancel{I(x,y)}]^2 \quad \text{First order approx}$$

$$= \sum u^2 I_x^2 + 2uvI_xI_y + v^2 I_y^2$$

$\vec{x}^T A \vec{x} \leftarrow$ quadr

$$= \sum_{x,y\in W} [u \ v]\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} \quad \text{Rewrite as matrix equation}$$

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

gradient covariance matrix

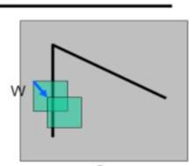
For small shifts $[u, v]$ we have a *bilinear* approximation

16

Feature detection: the math

Consider shifting the window W by (u,v)

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



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

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

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

$ABC = A(BC)$

$(a^T b)^2 = b^T (a)(a^T)(b)$

17

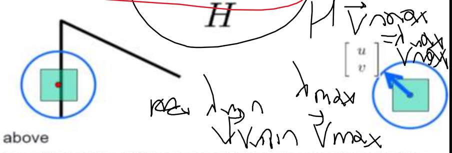
Feature detection: the math

This can be rewritten:

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

$\downarrow$ high

$H \vec{v}_{max} = \lambda_{max} \vec{v}_{max}$



For the example above

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

18

$$\sum_{x,y \in W} \underbrace{\begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}}_{H} = H$$

Note: these are just products of components of the gradient, Ix, Iy

19

Feature detection: the error function

➢ A new corner measurement by investigating the **shape** of the error function

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

quadratic

symmetric

$\mathbf{u}^T H \mathbf{u}$ represents a **quadratic function**;

Thus, we can analyze E's shape by looking at the property of **H**

$$\underline{\underline{H}} = \sum_{x,y \in W} \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$

20

Quadratic forms

Quadratic form (homogeneous polynomial of degree two) of n variables x_i

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \le j}}^{n} c_{ij} x_i x_j \qquad \vec{x}^T \underline{\underline{A}} \vec{x}$$

symmetric

Examples

$$4x_1^2 + 5x_2^2 + 3x_3^2 + 2x_1x_2 + 4x_1x_3 + 6x_2x_3$$

$$= \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix} \begin{pmatrix} 4 & 1 & 2 \\ 1 & 5 & 3 \\ 2 & 3 & 3 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

21

Symmetric matrices

Quadratic forms can be represented by a real symmetric matrix A where

$$a_{ij} = \begin{cases} c_{ij} & \text{if } i = j, \\ \frac{1}{2}c_{ij} & \text{if } i < j, \\ \frac{1}{2}c_{ji} & \text{if } i > j. \end{cases}$$

$$\sum_{i=1}^{n} \sum_{\substack{j=1 \\ i \le j}}^{n} c_{ij} x_i x_j = \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} x_i x_j$$

$$= \begin{pmatrix} x_1 & \dots & x_n \end{pmatrix} \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

$$= \mathbf{x}^t A \mathbf{x}$$

22

Quick review: eigenvalue/eigenvector

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

$$Ax = \lambda x$$

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

- The eigenvalues are found by solving:

$$det(A - \lambda I) = 0$$

- In our case, **A = H** is a 2x2 matrix, so we have

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

- The solution:

$$\lambda_{\pm} = \frac{1}{2}\left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2}\right]$$

λ_- ← minimum

λ_+ ← maximum eigval

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

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

23

Eigenvalues of symmetric matrices

suppose $A \in \mathbf{R}^{n \times n}$ is symmetric, *i.e.*, $A = A^T$

fact: the eigenvalues of A are real

suppose $Av = \lambda v$, $v \neq 0$, $v \in \mathbf{C}^n$

$$\overline{v}^T A v = \overline{v}^T (Av) = \lambda \overline{v}^T v = \lambda \sum_{i=1}^{n} |v_i|^2$$

$$\overline{v}^T A v = \overline{(Av)}^T v = \overline{(\lambda v)}^T v = \overline{\lambda} \sum_{i=1}^{n} |v_i|^2$$

we have $\lambda = \overline{\lambda}$, *i.e.*, $\lambda \in \mathbf{R}$

(hence, can assume $v \in \mathbf{R}^n$)

24

$A = P\Lambda P^{-1} = P\Lambda P^T$

Eigenvectors of symmetric matrices

suppose $A \in \mathbf{R}^{n\times n}$ is symmetric, *i.e.*, $A = A^T$

fact: there is a set of orthonormal eigenvectors of A

$A = Q\Lambda Q^T$

where Q is an orthogonal matrix (the columns of which are eigenvectors of A), and Λ is real and diagonal (having the eigenvalues of A on the diagonal)

$P = [\vec{v}_1 \; \cdots \; \vec{v}_n]$ ← modal matrix

rotational

$\vec{v}_i^T \vec{v}_j = 0$ for $i \neq j$ $\quad P \quad P^T = \ldots$

$\|\vec{v}_i\|^2 = 1$ $\quad \boxed{P^{-1} = P^T}$

25

Eigenvectors of symmetric matrices

suppose $A \in \mathbf{R}^{n\times n}$ is symmetric, *i.e.*, $A = A^T$

fact: there is a set of orthonormal eigenvectors of A

$A = Q\Lambda Q^T$

$\mathbf{x}^T A \mathbf{x}$

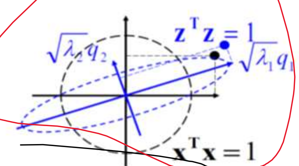
$= \mathbf{x}^T \mathbf{Q}\,\Lambda\,\mathbf{Q}^T \mathbf{x}$

$= (\mathbf{Q}^T\mathbf{x})^T \Lambda (\mathbf{Q}^T\mathbf{x})$

$= \mathbf{y}^T \Lambda \mathbf{y}$

$= (\Lambda^{\frac{1}{2}}\mathbf{y})^T(\Lambda^{\frac{1}{2}}\mathbf{y})$

$= \mathbf{z}^T\mathbf{z}$



$\vec{z} = \Lambda^{1/2}\vec{y}$

26

$f(\vec{x}) = \vec{x}^T A \vec{x}$ $\quad f(\vec{x}): \vec{x} \in \mathbb{R}^n \to \mathbb{R}$

assume A is symmetric

$= \vec{x}^T Q \Lambda Q^T \vec{x}$

$= (Q^T\vec{x})^T \Lambda Q^T \vec{x}$ $\quad A$

$= \vec{y}^T \Lambda \vec{y} =$

$\Lambda = \mathrm{diag}(\lambda_1 \ldots \lambda_n)$

$\vec{y} = Q^T\vec{x}$

$\Lambda = \Lambda^{1/2}\Lambda^{1/2}$

27

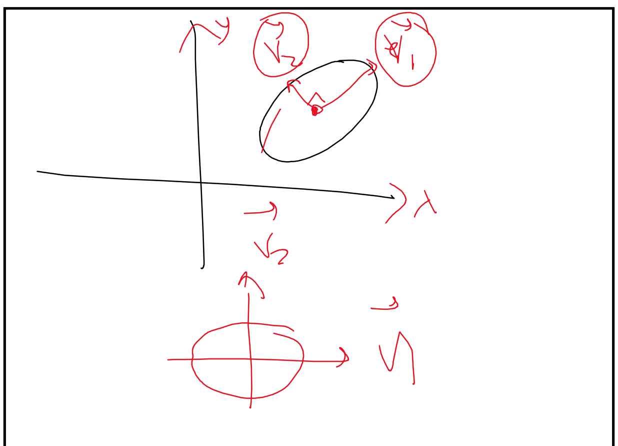
$\Lambda^{1/2} = ?$ $\quad \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_n} \end{bmatrix}$

Assume $\lambda_1 \ldots \lambda_n \geq 0$

$\Lambda = \underbrace{\Lambda^{1/2}\;\Lambda^{1/2}}_{?}$

$\begin{bmatrix} \lambda_1 & \\ & \lambda_n \end{bmatrix} = \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_n} \end{bmatrix} \begin{bmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_n} \end{bmatrix}$

28



29

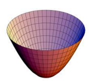
Visualizing quadratics

30

Equation of a circle

$x^2+y^2=r^2$

$1 = x^2 + y^2$



Equation of a 'bowl' (paraboloid)

$z = f(x,y) = x^2 + y^2$

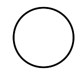
$z > 0$

If you slice the bowl at

$f(x,y) = 1$

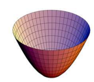
what do you get?

31



Equation of a circle

$1 = x^2 + y^2$



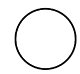
Equation of a 'bowl' (paraboloid)

$f(x,y) = x^2 + y^2$

If you slice the bowl at

$f(x,y) = 1$

what do you get?



32

$$f(x,y) = x^2 + y^2$$

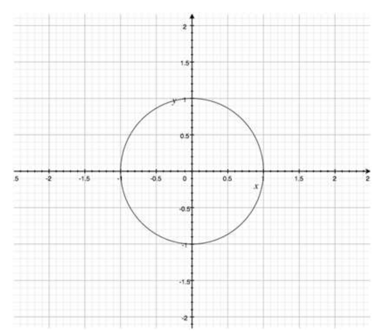
can be written in matrix form like this…

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

33

$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

'sliced at 1'



34

What happens if you ***increase*** *coefficient on* ***x****?*

$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

and slice at 1

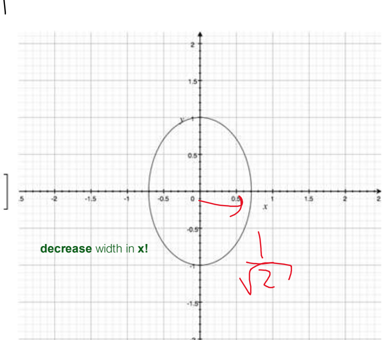
35

② $x^2 + y^2 = 1$

What happens if you ***increase*** *coefficient on* ***x****?*

$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$

and slice at 1



$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1$$

36

*What happens if you **increase** coefficient on **y**?*

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

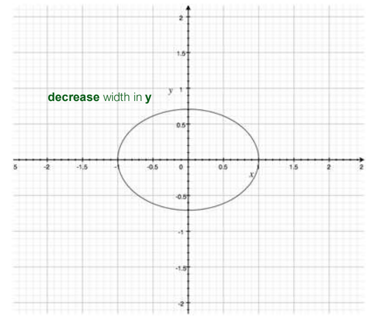
and slice at 1

37

*What happens if you **increase** coefficient on **y**?*

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

and slice at 1



38

$$f(x,y) = x^2 + y^2$$

can be written in matrix form like this…

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{A} \begin{bmatrix} x \\ y \end{bmatrix}$$

What's the shape?
What are the eigenvectors?
What are the eigenvalues?

39

$$f(x,y) = x^2 + y^2$$

can be written in matrix form like this…

$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Result of Singular Value Decomposition (SVD)

eigenvectors — eigenvalues along diagonal

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{\top}$$

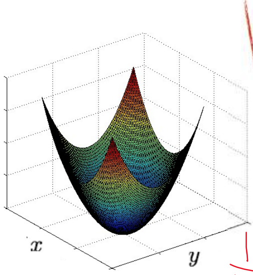
axis of the 'ellipse slice' — Inverse sqr of length of the quadratic along the axis

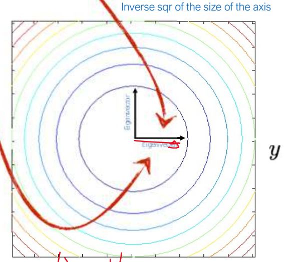
40

Eigenvectors — Eigenvalues

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{T}$$

Inverse sqr of the size of the axis

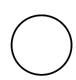




$\frac{1}{\lambda_i^{1/2}} = (\lambda_i)^{-1/2}$

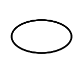
41

Recall:



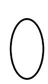
$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

you can smash this bowl in the **y** direction



$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

you can smash this bowl in the **x** direction

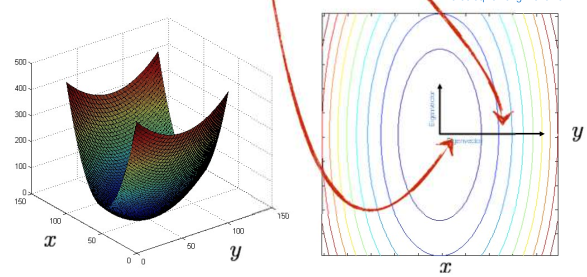


$$f(x,y) = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

42

$$\mathbf{A} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\text{Eigenvectors}} \overbrace{\begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}}^{\text{Eigenvalues}} \underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T}_{\text{Eigenvectors}}$$

Inverse sqr of length of axis

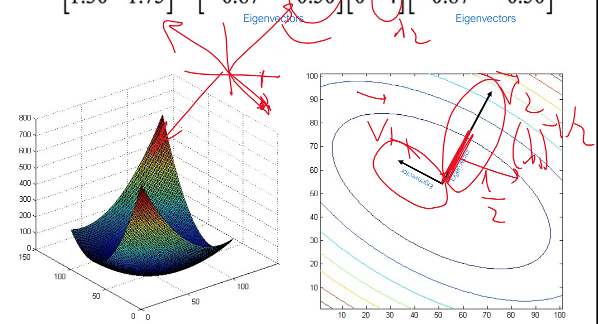


43

$\vec{v}_1$ $\vec{v}_2$ λ_1

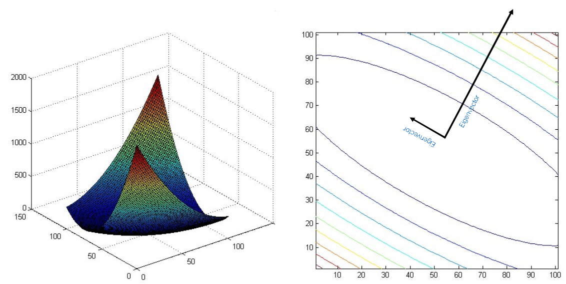
$$\mathbf{A} = \begin{bmatrix} 3.25 & 1.30 \\ 1.30 & 1.75 \end{bmatrix} = \underbrace{\begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}}_{\text{Eigenvectors}} \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}}^{\text{Eigenvalues}} \underbrace{\begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}^T}_{\text{Eigenvectors}}$$

λ_2



44

$$\mathbf{A} = \begin{bmatrix} 7.75 & 3.90 \\ 3.90 & 3.25 \end{bmatrix} = \underbrace{\begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}}_{\text{Eigenvectors}} \overbrace{\begin{bmatrix} 1 & 0 \\ 0 & 10 \end{bmatrix}}^{\text{Eigenvalues}} \underbrace{\begin{bmatrix} 0.50 & -0.87 \\ -0.87 & -0.50 \end{bmatrix}^T}_{\text{Eigenvectors}}$$



45

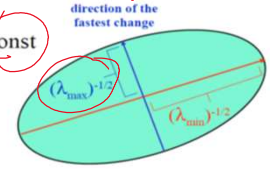
Ellipse $E(u,v) = \text{const}$

$\vec{v}_2$

direction of the fastest change

$\vec{v}_1$

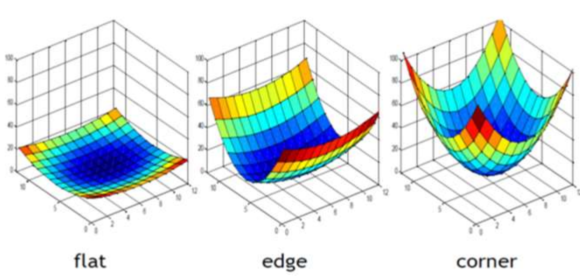
direction of the slowest change



46

Feature detection: the error function

High-level idea: what shape of the error function will we prefer for features?

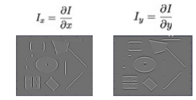


47

Finding corners

1. Compute image gradients over small region
2. Subtract mean from each image gradient
3. Compute the covariance matrix
4. Compute eigenvectors and eigenvalues
5. Use threshold on eigenvalues to detect corners

$I_x = \frac{\partial I}{\partial x}$ $\quad I_y = \frac{\partial I}{\partial y}$



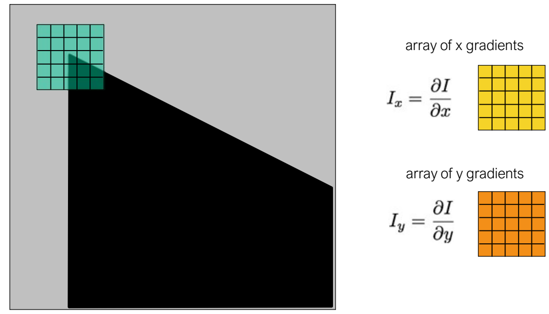
$$\begin{bmatrix} \sum_{p \in P} I_x I_x & \sum_{p \in P} I_x I_y \\ \sum_{p \in P} I_y I_x & \sum_{p \in P} I_y I_y \end{bmatrix}$$

48

1. Compute image gradients over a small region
(not just a single pixel)

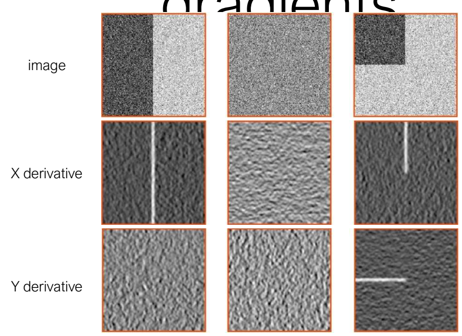
49

1. Compute image gradients over a small region
(not just a single pixel)

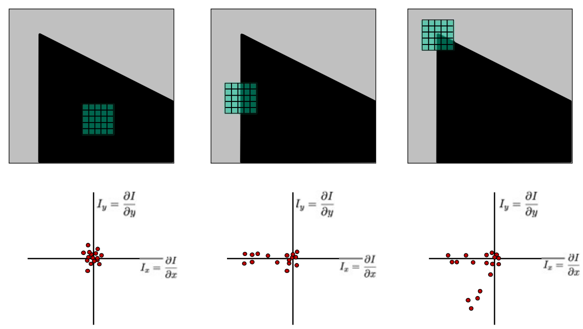


50

visualization of gradients

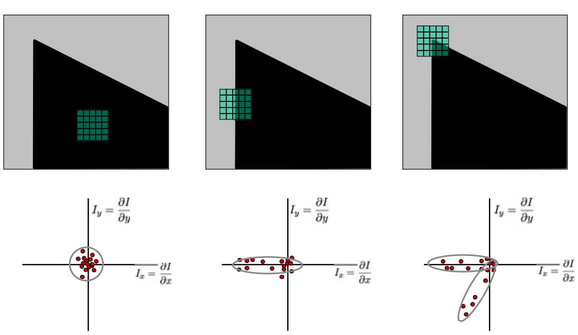


51



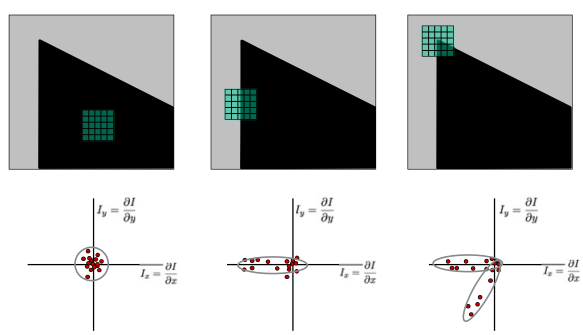
What does the distribution tell you about the region?

52



distribution reveals edge orientation and magnitude

53



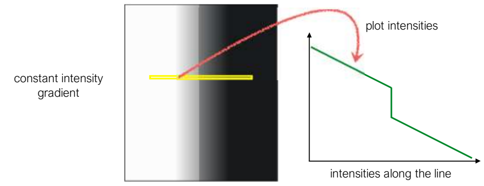
How do you quantify orientation and magnitude?

54

2. Subtract the mean from each image gradient

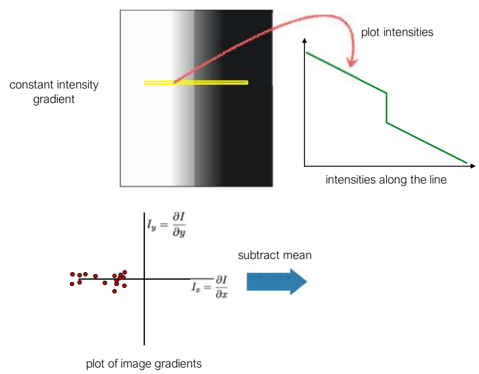
55

2. Subtract the mean from each image gradient



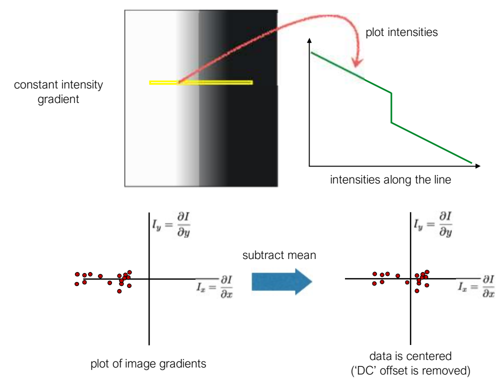
56

2. Subtract the mean from each image gradient



57

2. Subtract the mean from each image gradient



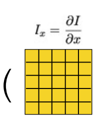
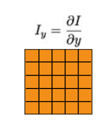
58

3. Compute the covariance matrix

59

3. Compute the covariance matrix

$$\begin{bmatrix} \sum\limits_{p \in P} I_x I_x & \sum\limits_{p \in P} I_x I_y \\ \sum\limits_{p \in P} I_y I_x & \sum\limits_{p \in P} I_y I_y \end{bmatrix}$$

$\sum\limits_{p \in P} I_x I_y$ =sum( .* )

Where does this covariance matrix come from?

60

By computing the gradient covariance matrix…

$$\begin{bmatrix} \sum\limits_{p \in P} I_x I_x & \sum\limits_{p \in P} I_x I_y \\ \sum\limits_{p \in P} I_y I_x & \sum\limits_{p \in P} I_y I_y \end{bmatrix}$$

we are fitting a quadratic to the gradients over a small image region

61

4. Compute eigenvalues and eigenvectors

62

4. Compute eigenvalues and eigenvectors

eigenvalue

$$M\boldsymbol{e} = \lambda\boldsymbol{e}$$

eigenvector

$$(M - \lambda I)\boldsymbol{e} = 0$$

63

4. Compute eigenvalues and eigenvectors

eigenvalue

$$M\boldsymbol{e} = \lambda\boldsymbol{e}$$

eigenvector

$$(M - \lambda I)\boldsymbol{e} = 0$$

1. Compute the determinant of $M - \lambda I$
(returns a polynomial)

64

4. Compute eigenvalues and eigenvectors

eigenvalue

$$M\boldsymbol{e} = \lambda\boldsymbol{e}$$

eigenvector

$$(M - \lambda I)\boldsymbol{e} = 0$$

1. Compute the determinant of $M - \lambda I$
(returns a polynomial)

2. Find the roots of polynomial $\det(M - \lambda I) = 0$
(returns eigenvalues)

65

4. Compute eigenvalues and eigenvectors

eigenvalue

$$M\boldsymbol{e} = \lambda\boldsymbol{e}$$

eigenvector

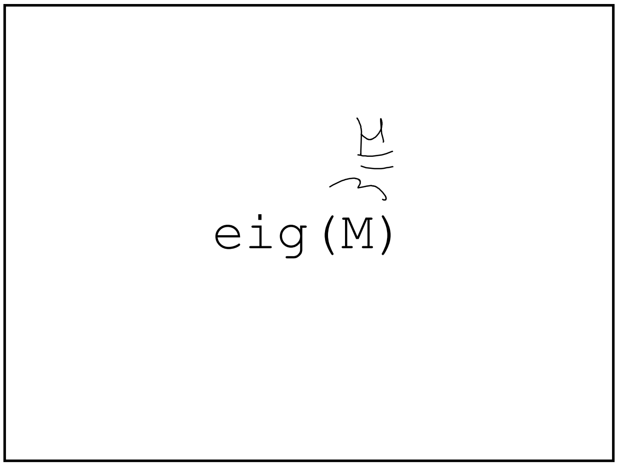
$$(M - \lambda I)\boldsymbol{e} = 0$$

1. Compute the determinant of $M - \lambda I$
(returns a polynomial)

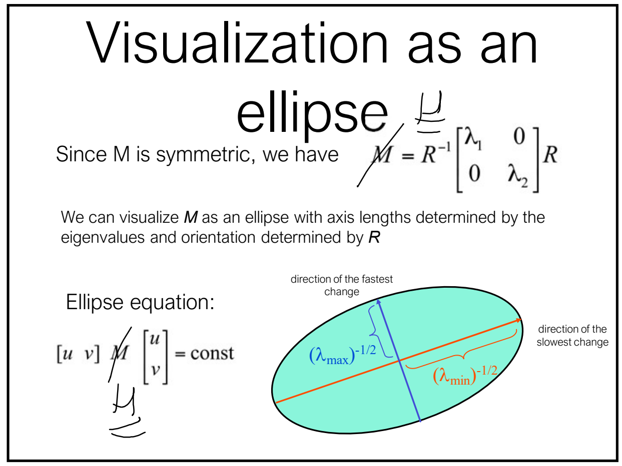
2. Find the roots of polynomial $\det(M - \lambda I) = 0$
(returns eigenvalues)

3. For each eigenvalue, solve $(M - \lambda I)\boldsymbol{e} = 0$
(returns eigenvectors)

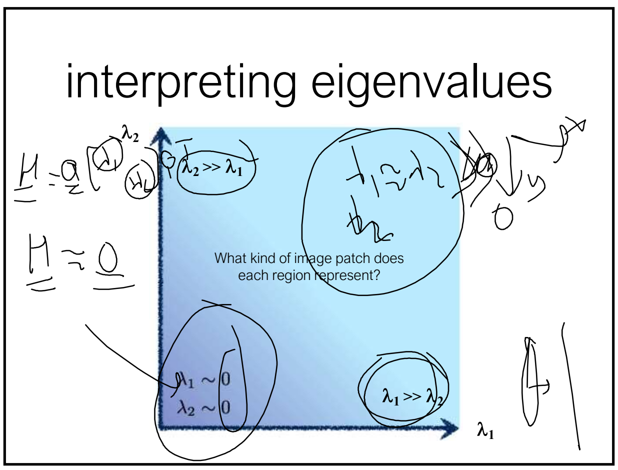
66



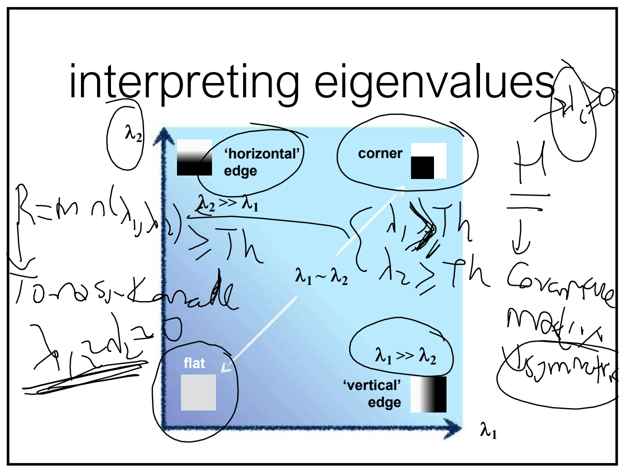
67



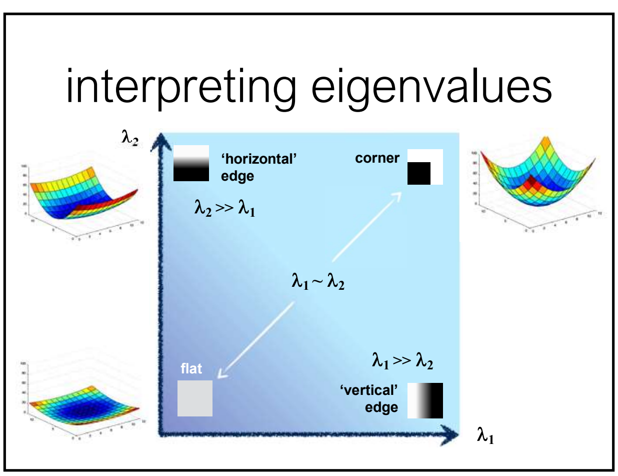
68



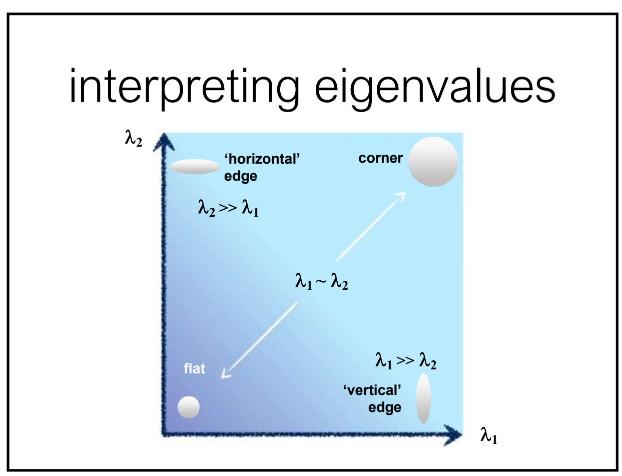
69



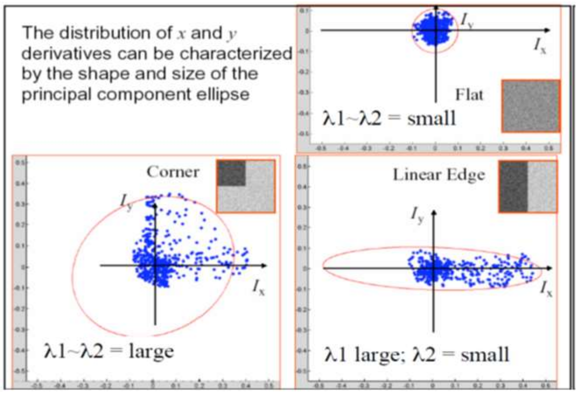
70



71

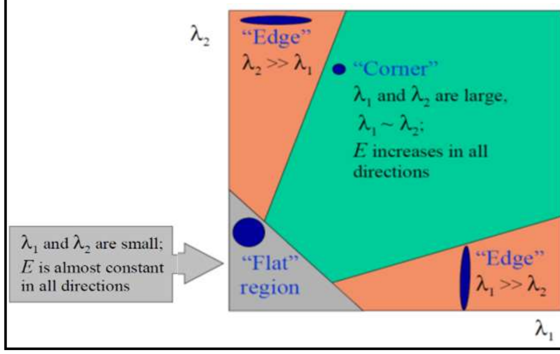


72



73

Classification via Eigenvalues

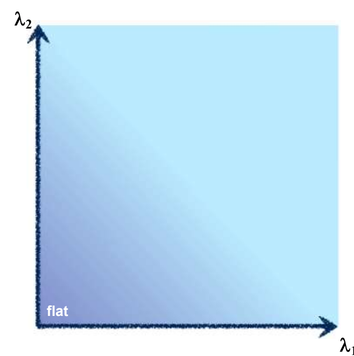


74

5. Use threshold on eigenvalues to detect corners

75

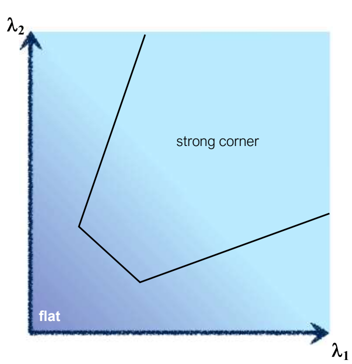
5. Use threshold on eigenvalues to detect corners



Think of a function to score 'cornerness'

76

5. Use threshold on eigenvalues to detect corners

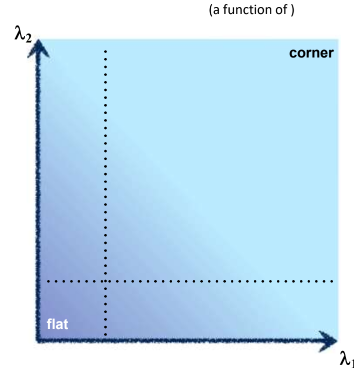


Think of a function to score 'cornerness'

77

5. Use threshold on eigenvalues to detect corners

(a function of)



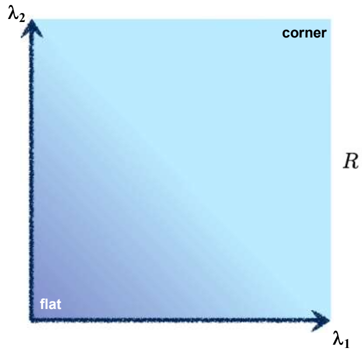
Use the smallest eigenvalue as the response function

$$R = \min(\lambda_1, \lambda_2)$$

78

5. Use threshold on eigenvalues to detect corners

(a function of)



Eigenvalues need to be bigger than one.

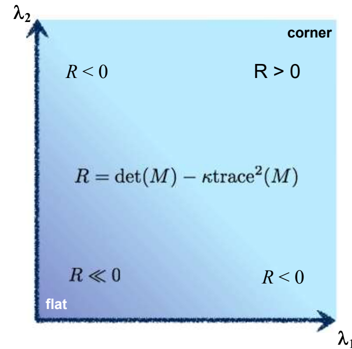
$$R = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2)^2$$

Can compute this more efficiently…

79

5. Use threshold on eigenvalues to detect corners

(a function of)



$\det M = \lambda_1\lambda_2$

$\mathrm{trace}\, M = \lambda_1 + \lambda_2$

$$det\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = ad - bc$$

$$trace\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}\right) = a + d$$

80

Harris & Stephens (1988)

$$R = \det(\cancel{M}) - \kappa \mathrm{trace}^2(\cancel{M})$$

Kanade & Tomasi (1994)

$$R = \min(\lambda_1, \lambda_2)$$

Nobel (1998)

$$R = \frac{\det(\cancel{M})}{\mathrm{trace}(\cancel{M}) + \epsilon}$$

$$R = \frac{\det(\underline{H})}{\mathrm{trace}(\underline{H})}$$

$\epsilon > 0^+$

81

Tomasi Kanade $R = \min(\lambda_1, \lambda_2)$

Nobel $R = \frac{\det(\underline{H})}{\mathrm{trace}(\underline{H})}$

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

$$\underline{H} = P \Lambda P^T$$

$\det(AB) = \det(A)\det(B)$

$$\det(H) = \det(P)\det(\Lambda)\det(P^T)$$

82

$$\underline{H} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$\det(\underline{H}) = ad - bc$$

$$\mathrm{trace}(\underline{H}) = a + d$$

83

Nobel $R = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{1}{\frac{1}{\lambda_2} + \frac{1}{\lambda_1}}$

Harmonic mean

Arithmetic mean $\frac{x+y}{2}$

Geometric mean $\sqrt{xy}$

84

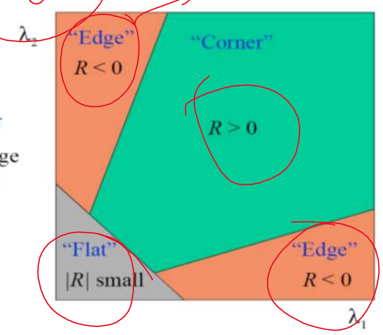
Harris $R = \det M - k \operatorname{trace}^2 M$

85

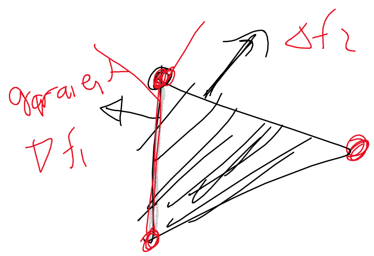
Harris Cornerness Score

$\det(M) - k \operatorname{trace}^2 M$

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

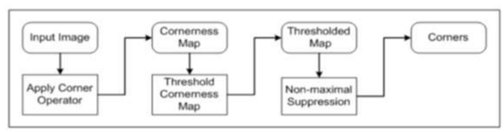


86



87

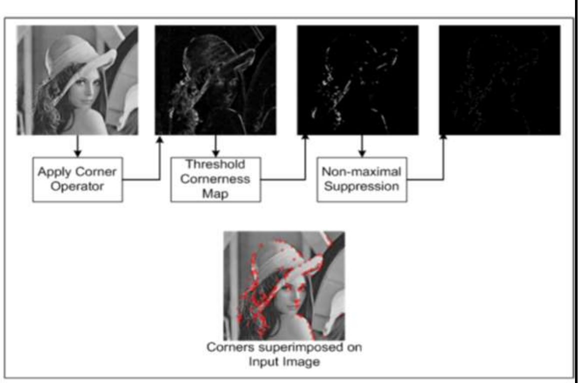
Steps in Corner Detection



1. For each pixel, the corner operator is applied to obtain a cornerness measure for this pixel.
2. Threshold cornerness map to eliminate weak corners.
3. Apply non-maximal suppression to eliminate points whose cornerness measure is not larger than the cornerness values of all points within a certain distance.

88

Steps in Corner Detection (cont'd)



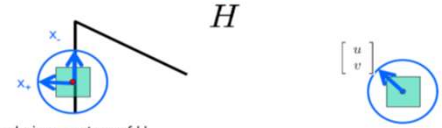
Corners superimposed on Input Image

89

Feature detection: the math

This can be rewritten:

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



Eigenvalues and eigenvectors of H

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

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

90

Tomasi-Kanade $R = (\min(\lambda_1, \lambda_2))$

$R > Th$

Cornerness measure

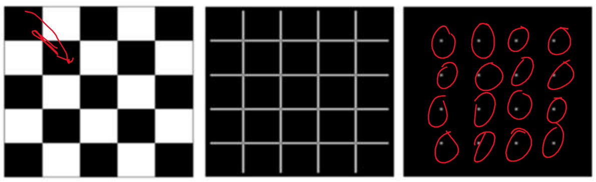
Feature detection: the math

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

- What's our feature scoring function?

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

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



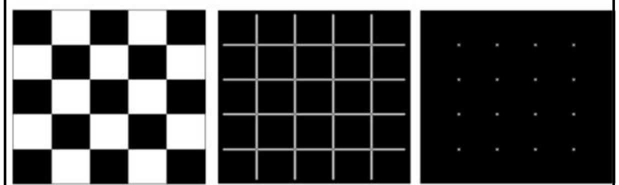
I λ_+ λ_-

91

Feature detection summary (Kanade-Tomasi)

Here's what you do

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



I λ_+ λ_-

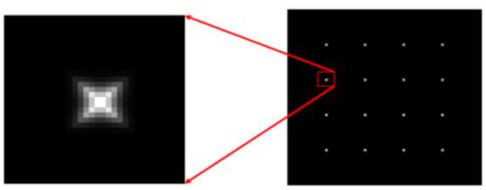
J. Shi and C. Tomasi (June 1994). "Good Features to Track". *9th IEEE Conference on Computer Vision and Pattern Recognition*. Springer.

92

Feature detection summary

Here's what you do

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



λ_-

93

The Harris operator

λ_- is a variant of the "Harris operator" for feature detection ($\lambda_- = \lambda_1$; $\lambda_+ = \lambda_2$)

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

Brown, Szeliski, and Winder (2005)

: harmonic mean,

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

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

$$^* \quad \lambda_{\pm} = \frac{1}{2}\left[(h_{11} + h_{22}) \pm \sqrt{4h_{12}h_{21} + (h_{11} - h_{22})^2}\right]$$

94

The Harris operator

Measure of corner response (Harris):

$$R = \det H - k(\operatorname{trace} H)^2$$

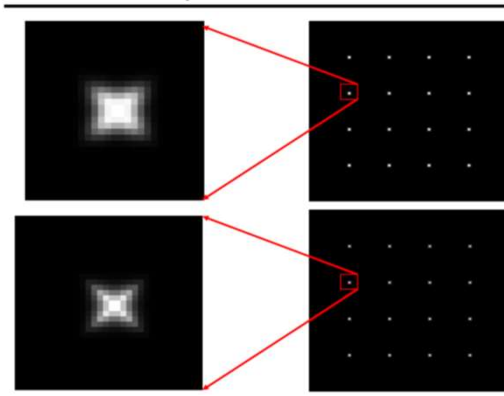
With:

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

(k – empirical constant, k = 0.04-0.06)

95

The Harris operator



Harris operator

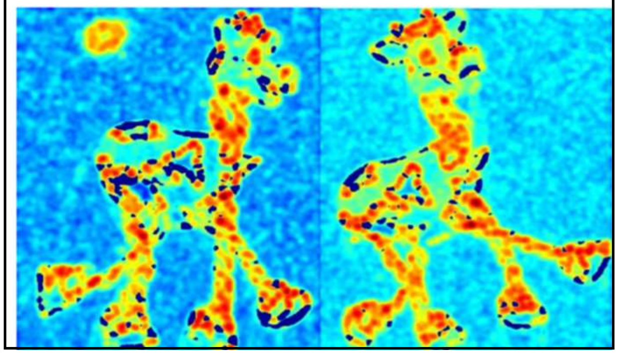
λ_-

96

Harris detector example

97

f value (red high, blue low)

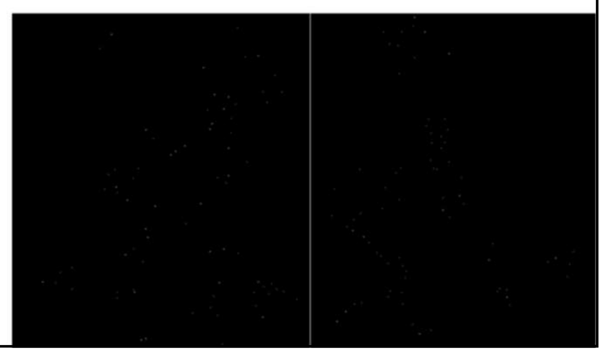


98

Threshold (f > value)

99

Find local maxima of f



100

Harris features (in red)

101

Harris detector: Steps

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix *H* in a Gaussian window around each pixel
3. Compute corner response function *R*
4. Threshold *R*
5. Find local maxima of response function (non-maximum suppression)

$$R = \det(H) - \alpha\,\mathrm{trace}(H)^2 = \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

α: constant (0.04 to 0.06)

C.Harris and M.Stephens. "A Combined Corner and Edge Detector," *Proceedings of the 4th Alvey Vision Conference*: pages 147—151, 1988.

102

Harris Detector

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

1. Compute x and y derivatives of image

$$I_x = G_\sigma^x * I \qquad I_y = G_\sigma^y * I$$

2. Compute products of derivatives at every pixel

$$I_{x^2} = I_x \cdot I_x \qquad I_{y^2} = I_y \cdot I_y \qquad I_{xy} = I_x \cdot I_y$$

3. Compute the sums of the products of derivatives at each pixel

$$S_{x^2} = G_{\sigma'} * I_{x^2} \qquad S_{y^2} = G_{\sigma'} * I_{y^2} \qquad S_{xy} = G_{\sigma'} * I_{xy}$$

103

Harris Detector

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

4. Define the matrix at each pixel

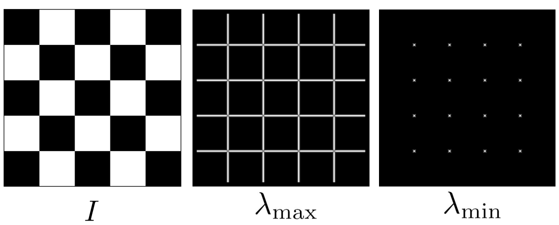
$$M(x,y) = \begin{bmatrix} S_{x^2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y^2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = \det M - k(\mathrm{trace} M)^2$$

6. Threshold on value of R; compute non-max suppression.

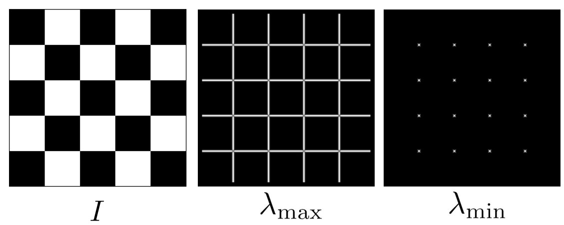
104



Yet another option: $f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2}$

How do you write this equivalently
using determinant and trace?

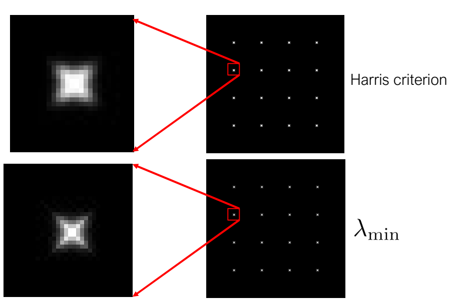
105



Yet another option: $f = \frac{\lambda_1 \lambda_2}{\lambda_1 + \lambda_2} = \frac{determinant(H)}{trace(H)}$

106

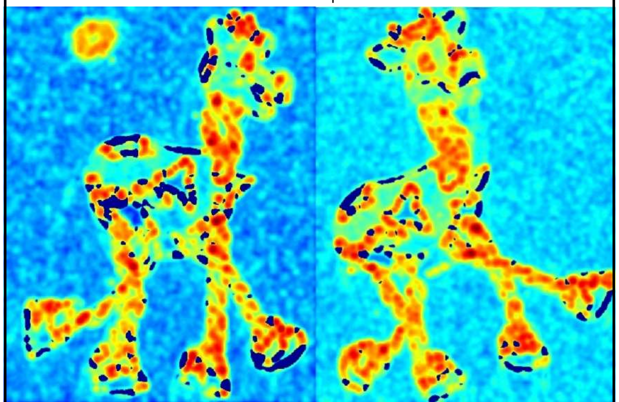
Different criteria



107

108

Corner response



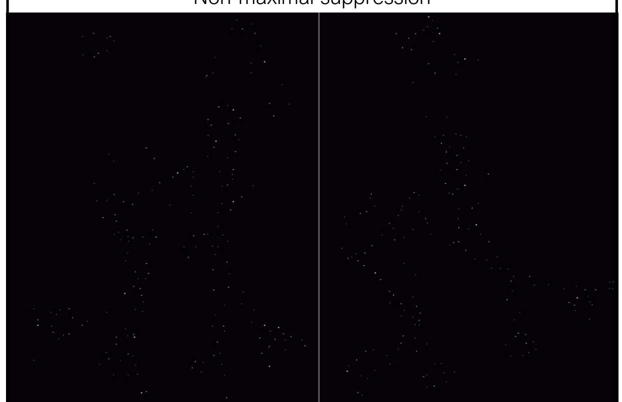
109

110

Thresholded corner response

111

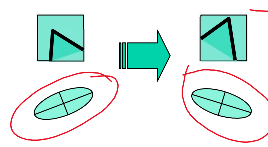
Non-maximal suppression



112

113

Harris corner response is invariant to rotation



Ellipse rotates but its shape (**eigenvalues**) remains the same

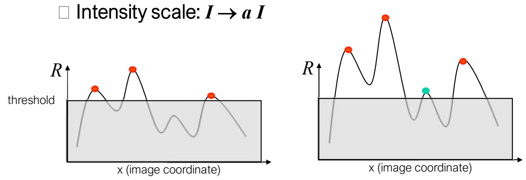
Corner response R is invariant to image rotation

114

Harris corner response is invariant to intensity changes

Partial invariance to *affine intensity* change

- Only derivatives are used => invariance to intensity shift $I \to I + b$
- Intensity scale: $I \to a\,I$



115

Linear Transform

$$T[a_1 x_1 + a_2 x_2] = a_1 T[x_1] + a_2 T[x_2]$$

116

Affine Transform

$$y = ax + b$$

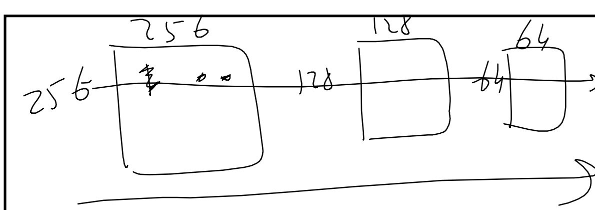
117

The Harris detector is not invariant to changes in …

118

The Harris corner detector is not invariant to scale

119



Multi-scale detection

120

References

Basic reading:
- Szeliski textbook, Sections 4.1.

121