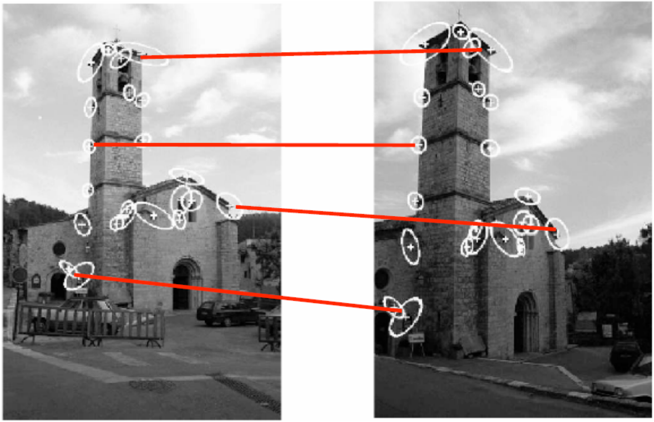
Harris Corner Detector

[redacted]

Motivation: Matching Problem

- Vision tasks such as stereo and motion estimation or tracking require finding corresponding features across two or more images.

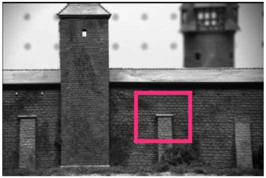


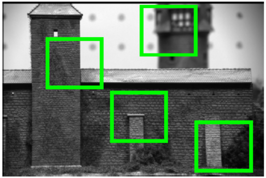
Motivation: Patch Matching

Elements to be matched are image patches of fixed size

Task: find the best (most similar) patch in a second image

Not all patches are created equal

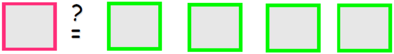




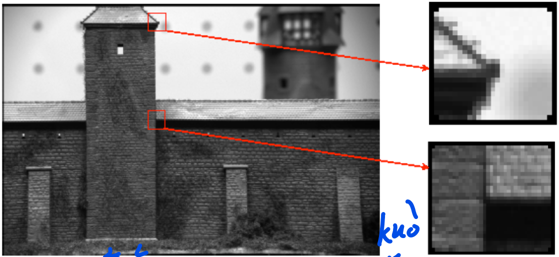
Inituition: this would be a good patch for matching, since it is very distinctive (there is only one patch in the second frame that looks similar).

Not all patches are created equal

Initiution: this would be a BAD patch for matching, since it is not very distinctive (there are many similar patches in the second frame)



What are corners?



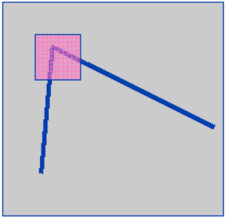
kuò

交点

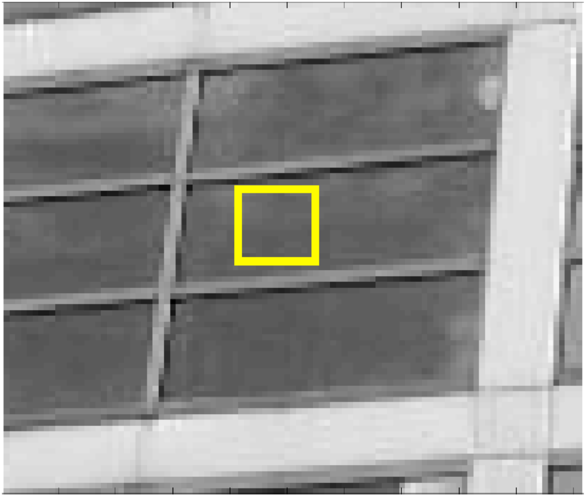
- Intuitively, junctions of contours. 轮廓
- Generally more stable features over changes of viewpoint
- Intuitively, large variations in the neighborhood of the point in all directions
- They are good features to match!

Corner points: Basic idea

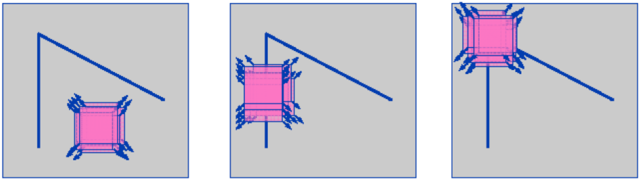
- We should easily recognize the point by looking at intensity values within a small window
- Shifting the window in *any direction* should yield a *large change* in appearance.



Appearance Change in a Local Neighbourhood

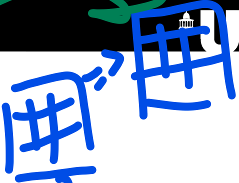


Harris Corner Detector: Basic Idea



“flat” region: no change in all directions	“edge”: no change along the edge direction	“corner”: significant change in all directions

Harris Corner Detector: Maths



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

1

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

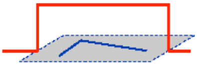
Window function

Shifted intensity

Intensity

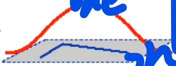
Intensity of the pixels in the patch

Window function $w(x,y)$ =



1 in window, 0 outside

or



Gaussian

Harris Corner Detector: Intuition

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

$$E(u,v) = \sum_{x,y} \underbrace{w(x,y)}_{\text{Window function}} \Big[\underbrace{I(x+u,y+v)}_{\text{Shifted intensity}} - \underbrace{I(x,y)}_{\text{Intensity}} \Big]^2$$

1

For nearly constant patches, this will be near 0.
For very distinctive patches, this will be larger.
Hence... we want patches where E(u,v) is LARGE.

我们当时另外的讲

下面就在讲（这是intensity的差别）

Taylor Series for 2D Functions

泰勒展开

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

First partial derivatives

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

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

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

Harris Corner Derivation

$\nabla I = \begin{bmatrix} I_x \\ I_y \end{bmatrix}$ ← sobel 算子

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

$$\approx \sum [I(x,y) + uI_x + vI_y - I(x,y)]^2$$ First order approx 一阶近似

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

harris Matrix / structure Tensor

$$= \sum \begin{bmatrix} u & v \end{bmatrix} \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}$$ Rewrite as matrix equation

$$= \begin{bmatrix} u & v \end{bmatrix} \left(\sum \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

Harris Detector: Maths I

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:

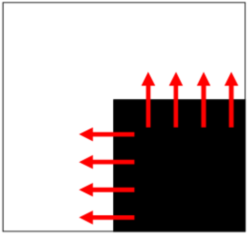
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Windowing function - computing a weighted sum (simplest case, w=1)

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

Interpreting the Second Moment Matrix

First, consider an axis-aligned corner:



Interpreting the Second Moment Matrix

对角化，表示为2个特征值，意味着与 axes x y 对齐

- First, consider an axis-aligned corner:

$$M = \underbrace{\begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}}_{B}$$

- This means dominant gradient directions align with x or y axis
- If either λ is close to 0, then this is **not** a corner, so look for locations where both are large.

找P: $P^{-1}AP = B$

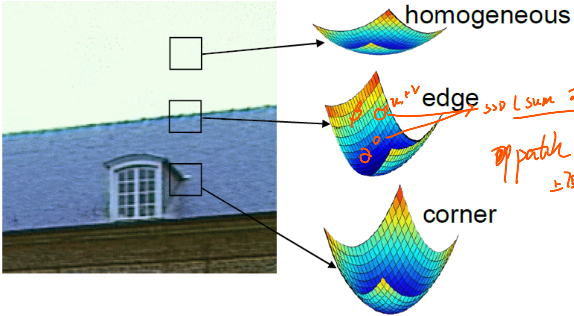
eigen vector

Harris Detector: Structure Tensor

- M captures the curvature of the local autocorrelation surface
- The eigenvalues of M are the principal curvatures
- They are the solutions to:

$$\lambda^2 - \lambda(\sum I_x^2 + \sum I_y^2) + (\sum I_x)^2(\sum I_y)^2 - (\sum I_x I_y)^2 = 0$$

Principal Axes (Eigenvectors)



SSD (sum of square difference)

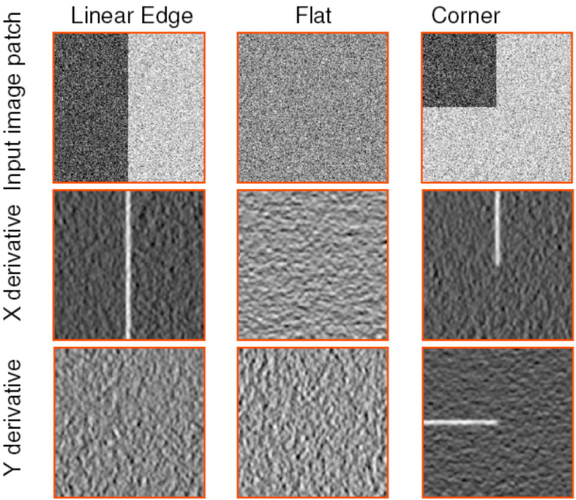
即 path ② → path ①

上面个公式 page 11

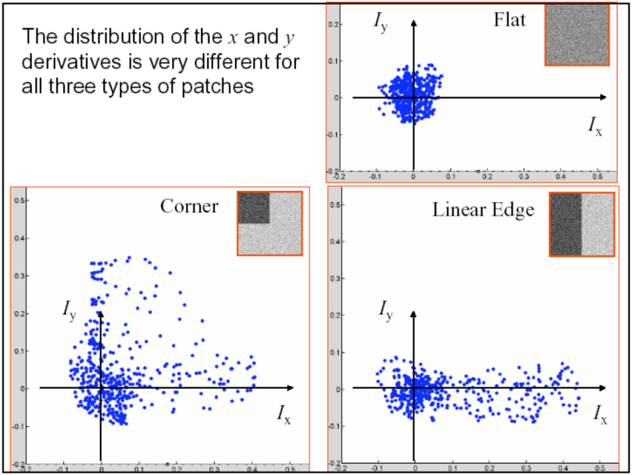
E(u, v)

Corner = both eigenvalues are large

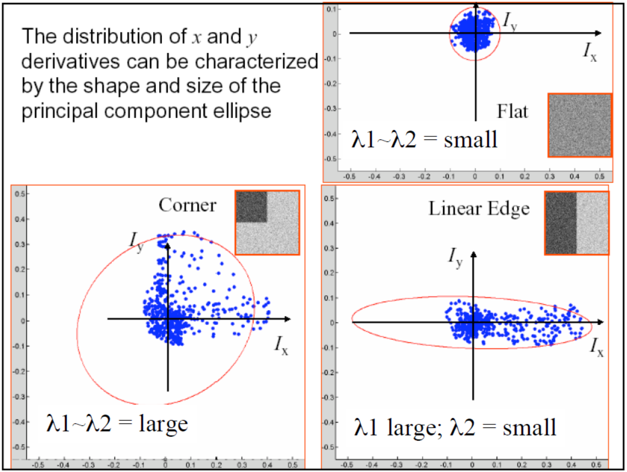
Example: Cases and 2D Derivatives



Plotting Derivatives as 2D Points

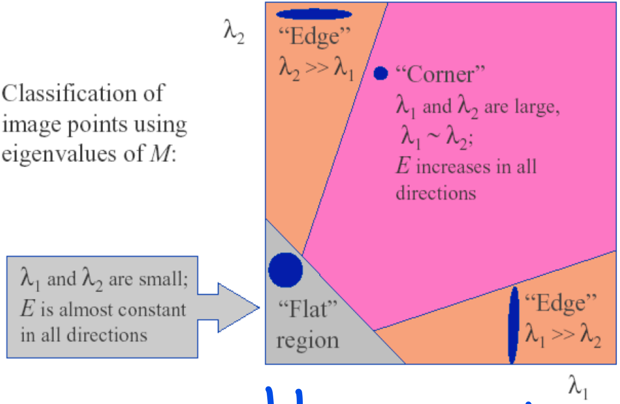


Fitting an Ellipse to each Set of Points



Classification via Eigenvalues

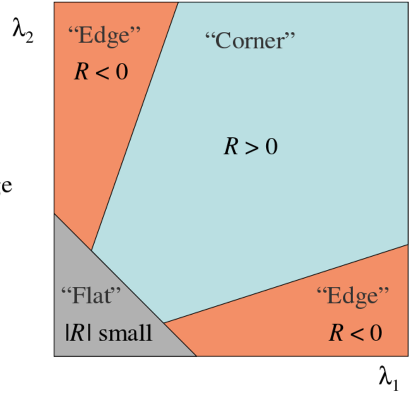
Classification of image points using eigenvalues of M:



另一种表达形式

Classification via Eigenvalues

- 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



Corner Response Measure

Measure of corner response:

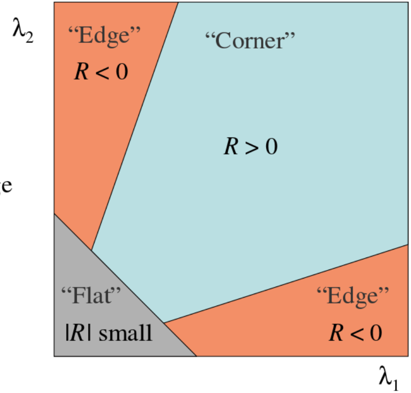
$$R = \det M - k\left(\operatorname{trace} M\right)^2$$

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

(k is an empirically determined constant; $k = 0.04$ - 0.06)

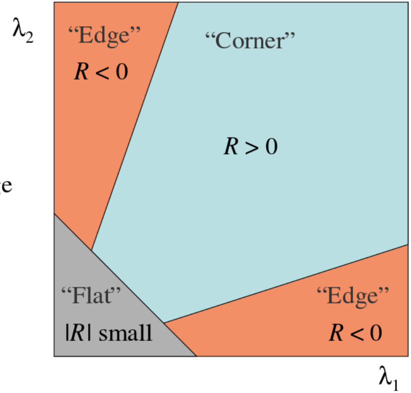
Corner Response Measure

- 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



Harris Corner Detector: Algorithm

- 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

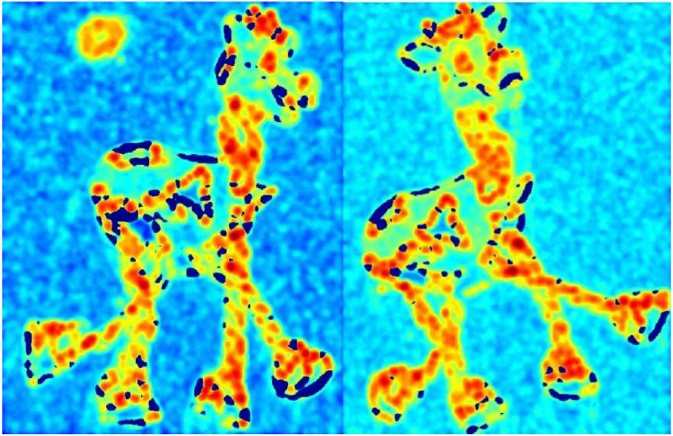


Harris Detector: Workflow

Harris Detector: Workflow

Compute corner response R

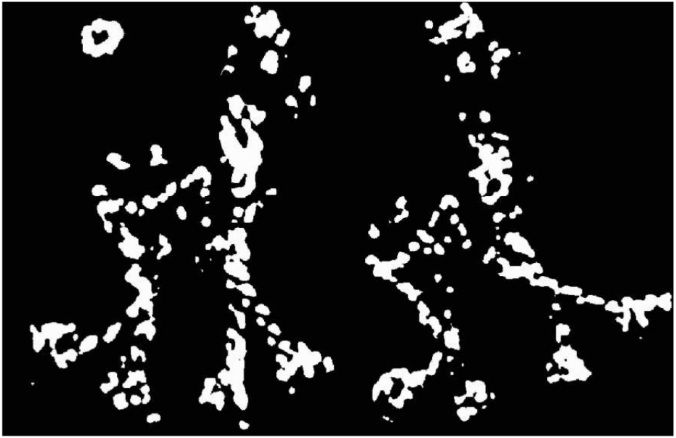
Compute corner response R



Harris Detector: Workflow

Find points with large corner response R¿ threshold

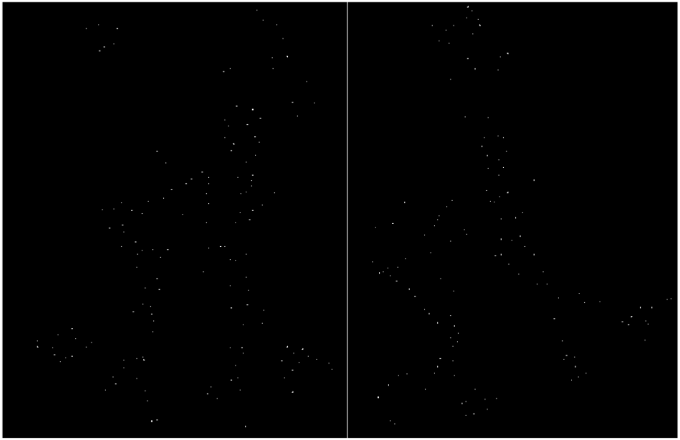
Find points with large corner response: R>threshold



Harris Detector: Workflow

Non-maxima suppression

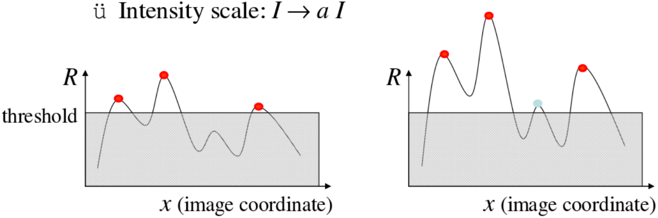
Take only the points of local maxima of R



Harris Detector: Properties

不变性

- Partial invariance to *affine intensity* change
  - ü Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$
  - ü Intensity scale: $I \rightarrow a\,I$

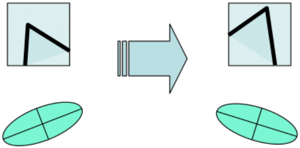


Harris Detector: Properties

Invariant to rotations

- Rotation invariance

好：对角化了
eigen 如二阶矩阵
最大的映射
∴ affine_trans
不干扰



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

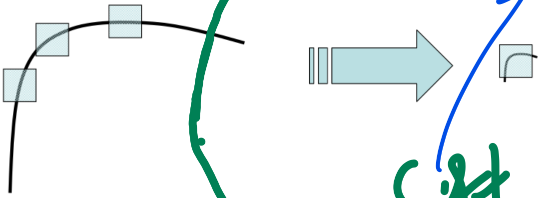
Corner response R is invariant to image rotation

Harris Detector: Properties

Not invariant to image scale!

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

Scale Invariant Feature Transform

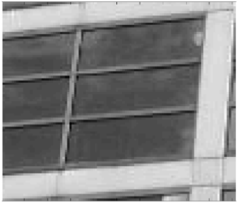


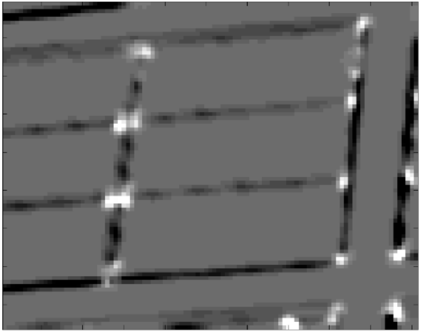
All points will be classified as edges

Corner

The Sift Detector and Descriptor

Harris Detector: Results

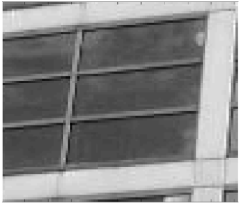


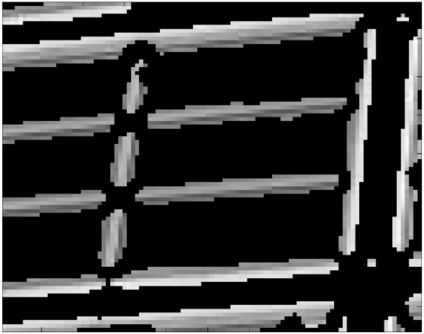


Harris R score.

Ix, Iy computed using Sobel operator
Windowing function w = Gaussian, sigma=1

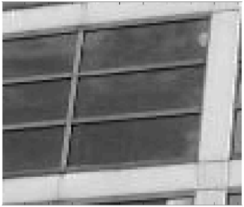
Harris Detector: Results

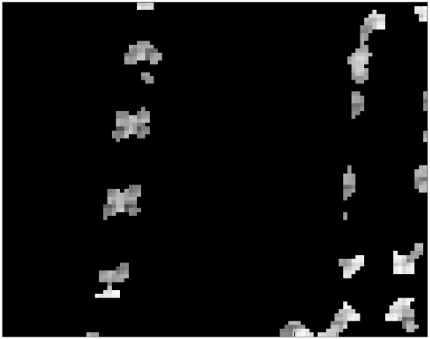




Threshold: R < -10000
(edges)

Harris Detector: Results





Threshold: > 10000
(corners)

Harris Detector: Algorithm Summary

1. Compute x and y derivatives of image

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

2. Compute products of derivatives at every pixel

$$I_{x2} = I_x.I_x \quad I_{y2} = I_y.I_y \quad I_{xy} = I_x.I_y$$

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

$$S_{x2} = G_{\sigma'} * I_{x2} \quad S_{y2} = G_{\sigma'} * I_{y2} \quad S_{xy} = G_{\sigma'} * I_{xy}$$

4. Define at each pixel (x, y) the matrix

$$H(x,y) = \begin{bmatrix} S_{x2}(x,y) & S_{xy}(x,y) \\ S_{xy}(x,y) & S_{y2}(x,y) \end{bmatrix}$$

5. Compute the response of the detector at each pixel

$$R = Det(H) - k(Trace(H))^2$$

6. Threshold on value of R. Compute nonmax suppression.