# Features

Digital Visual Effects, Spring 2008

*Yung-Yu Chuang*

2008/3/18

*with slides by Trevor Darrell Cordelia Schmid, David Lowe, Darya Frolova, Denis Simakov, Robert Collins and Jiwon Kim*

## Announcements

- Project #1 was due at midnight Friday. You have a total of 10 delay days without penalty, but you are advised to use them wisely.
- We reserve the rights for not including late homework for artifact voting.
- Project #2 handout will be available on the web later this week.

## Outline

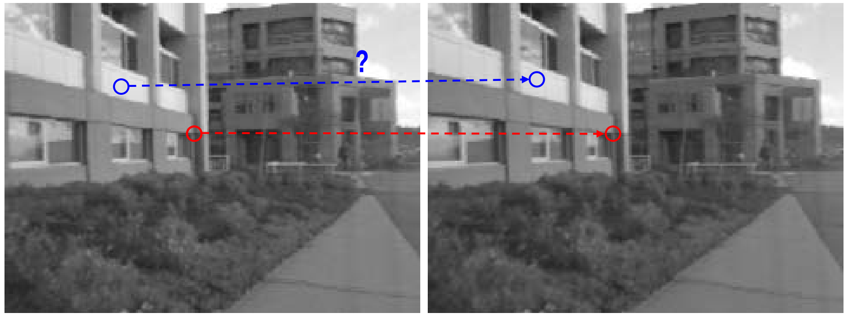
- Features
- Harris corner detector
- SIFT

# Features

## Features

DigiVFX

- Also known as interesting points, salient points or keypoints. Points that you can easily point out their correspondences in multiple images using only local information.




## Desired properties for features

DigiVFX

- Distinctive: a single feature can be correctly matched with high probability.
- Invariant: invariant to scale, rotation, affine, illumination and noise for robust matching across a substantial range of affine distortion, viewpoint change and so on. That is, it is repeatable.

## Applications

DigiVFX

- Object or scene recognition
- Structure from motion
- Stereo
- Motion tracking
- …

## Components

DigiVFX

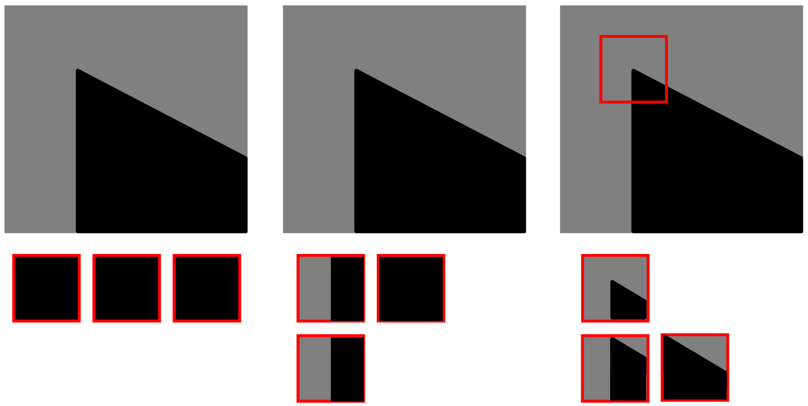
- Feature detection: locate where they are
- Feature description: describe what they are
- Feature matching: decide whether two are the same one

# Harris corner detector

## Moravec corner detector (1980)

- 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

## Moravec corner detector



flat

edge

corner
isolated point

## Moravec corner detector

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

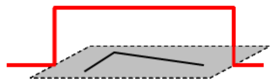
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2$$

window function

shifted intensity

intensity

Window function $w(x,y)$ =



1 in window, 0 outside

Four shifts: (u,v) = (1,0), (1,1), (0,1), (-1, 1)
Look for local maxima in *min{E}*

## Problems of Moravec detector

- Noisy response due to a binary window function
- Only a set of shifts at every 45 degree is considered
- Only minimum of E is taken into account

⇨ Harris corner detector (1988) solves these problems.

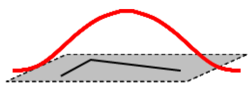
## Harris corner detector

Noisy response due to a binary window function

➢ Use a Gaussian function

$$w(x,y) = \exp\left(-\frac{(x^2+y^2)}{2\sigma^2}\right)$$

Window function $w(x,y)$ =



Gaussian

## Harris corner detector

Only a set of shifts at every 45 degree is considered

➢ Consider all small shifts by Taylor's expansion

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

$$E(u,v) = Au^2 + 2Cuv + Bv^2$$

$$A = \sum_{x,y} w(x,y) I_x^2(x,y)$$

$$B = \sum_{x,y} w(x,y) I_y^2(x,y)$$

$$C = \sum_{x,y} w(x,y) I_x(x,y) I_y(x,y)$$

## Harris corner detector

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

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

, where $\mathbf{M}$ is a 2×2 matrix computed from image derivatives:

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

## Harris corner detector (matrix form)

$$E(\mathbf{u}) = |I(\mathbf{x_0}+\mathbf{u}) - I(\mathbf{x_0})|^2$$

$$= \left| \left( I_0 + \frac{\partial I}{\partial \mathbf{u}}^T \mathbf{u} \right) - I_0 \right|^2$$

$$= \left| \frac{\partial I}{\partial \mathbf{u}}^T \mathbf{u} \right|^2$$

$$= \mathbf{u}^T \frac{\partial I}{\partial \mathbf{u}} \frac{\partial I}{\partial \mathbf{u}}^T \mathbf{u}$$

$$= \mathbf{u}^T \mathbf{M} \mathbf{u}$$

## Harris corner detector

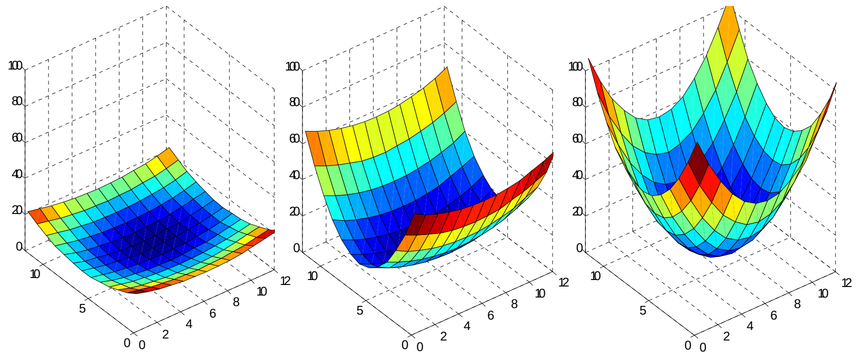
Only minimum of $E$ is taken into account

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

$\mathbf{u}^T\mathbf{M}\mathbf{u}$ represents a quadratic function; Thus, we can analyze $E$'s shape by looking at the property of $\mathbf{M}$

## Harris corner detector

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



flat edge corner

## Quadratic forms

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

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

- Examples

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

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

## Symmetric matrices

- Quadratic forms can be represented by a real symmetric matrix $\mathbf{A}$ where

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

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

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

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

## 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 $v \in \mathbf{R}^n$)

*Brad Osgood*

## 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 $A$

$$A = Q \Lambda Q^T$$

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

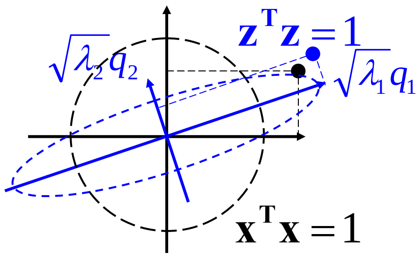
$$= \mathbf{x}^T \mathbf{Q} \, \mathbf{\Lambda} \, \mathbf{Q}^T \mathbf{x}$$

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

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

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

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



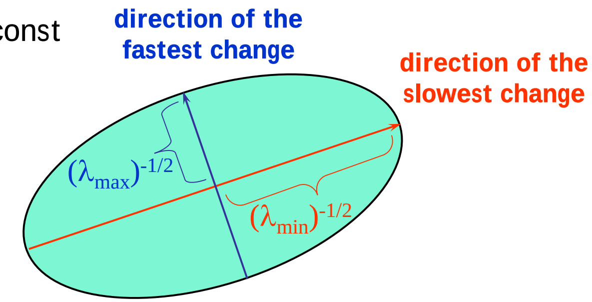

## Harris corner detector

Intensity change in shifting window: eigenvalue analysis

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

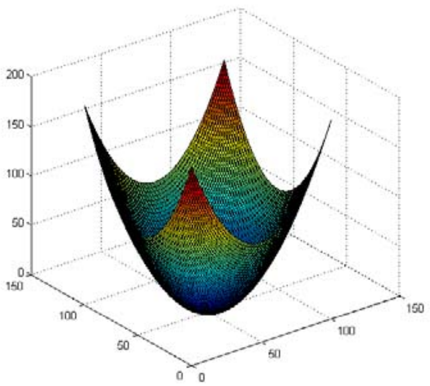
$\lambda_1, \lambda_2$ – eigenvalues of $\mathbf{M}$

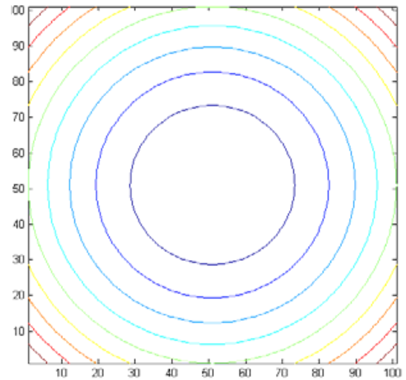
Ellipse *E(u,v)* = const



## Visualize quadratic functions

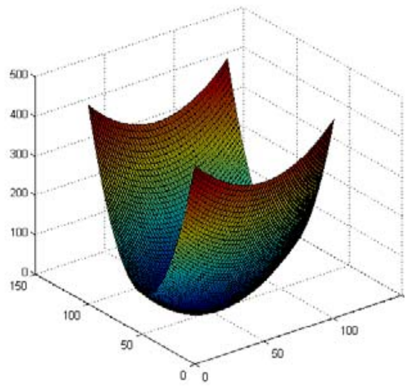
$$\mathbf{A}=\begin{bmatrix}1&0\\0&1\end{bmatrix}=\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}1&0\\0&1\end{bmatrix}^T$$

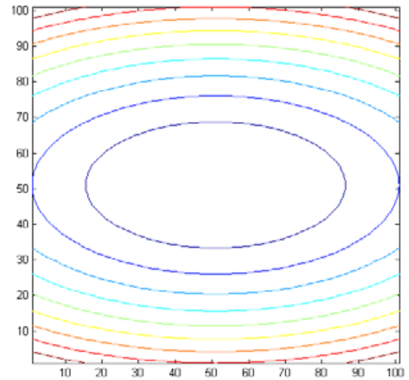




## Visualize quadratic functions

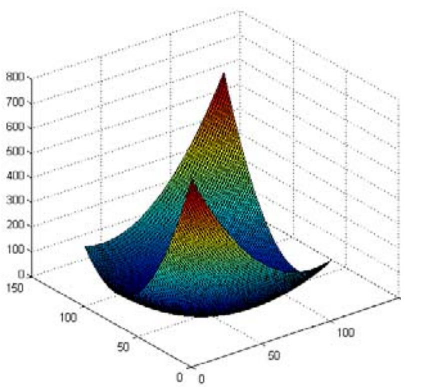
$$\mathbf{A}=\begin{bmatrix}4&0\\0&1\end{bmatrix}=\begin{bmatrix}1&0\\0&1\end{bmatrix}\begin{bmatrix}4&0\\0&1\end{bmatrix}\begin{bmatrix}1&0\\0&1\end{bmatrix}^T$$

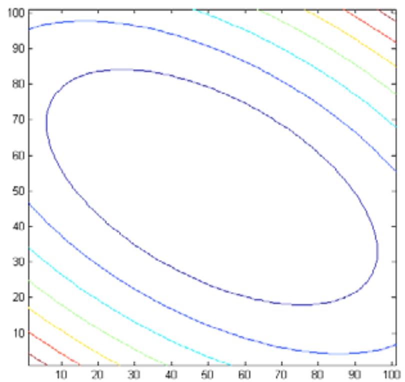




## Visualize quadratic functions

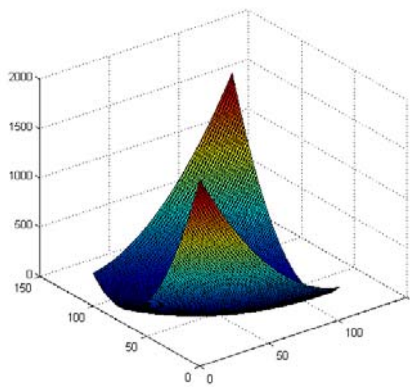
$$\mathbf{A}=\begin{bmatrix}3.25&1.30\\1.30&1.75\end{bmatrix}=\begin{bmatrix}0.50&-0.87\\-0.87&-0.50\end{bmatrix}\begin{bmatrix}1&0\\0&4\end{bmatrix}\begin{bmatrix}0.50&-0.87\\-0.87&-0.50\end{bmatrix}^T$$

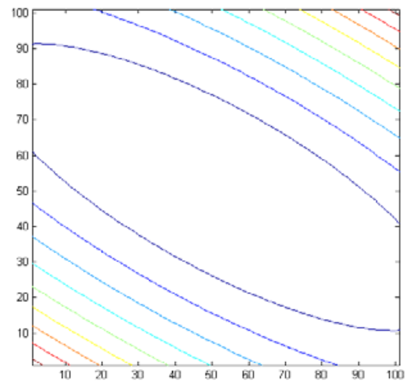




## Visualize quadratic functions

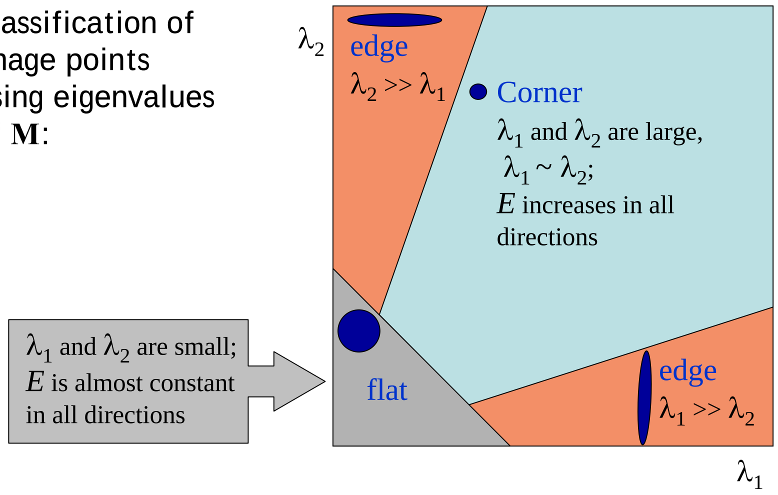
$$\mathbf{A}=\begin{bmatrix}7.75&3.90\\3.90&3.25\end{bmatrix}=\begin{bmatrix}0.50&-0.87\\-0.87&-0.50\end{bmatrix}\begin{bmatrix}1&0\\0&10\end{bmatrix}\begin{bmatrix}0.50&-0.87\\-0.87&-0.50\end{bmatrix}^T$$





## Harris corner detector

Classification of image points using eigenvalues of $\mathbf{M}$:




## Harris corner detector

$$\lambda = \frac{a_{00} + a_{11} \pm \sqrt{(a_{00} - a_{11})^2 + 4a_{10}a_{01}}}{2}$$

Only for reference, you do not need them to compute R

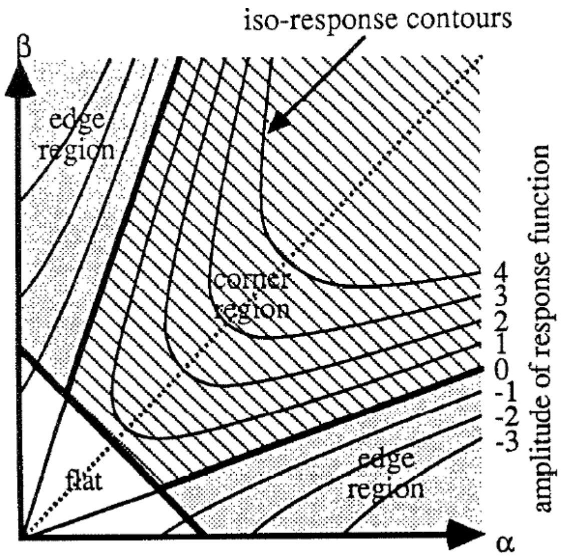
Measure of corner response:

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

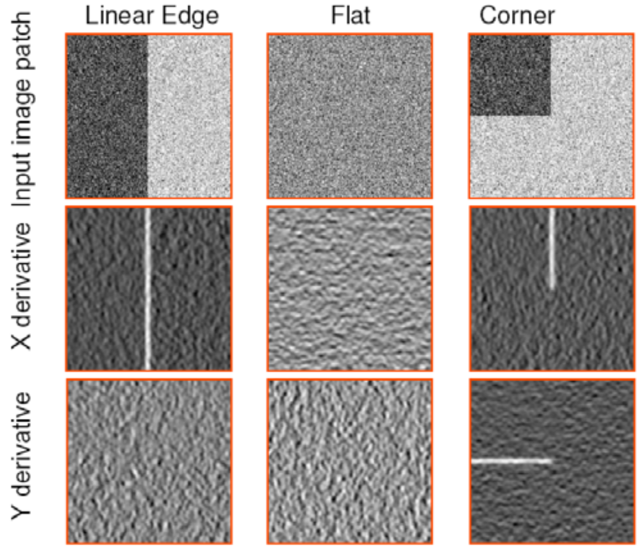
$$\det \mathbf{M} = \lambda_1 \lambda_2$$
$$\mathrm{trace}\ \mathbf{M} = \lambda_1 + \lambda_2$$

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

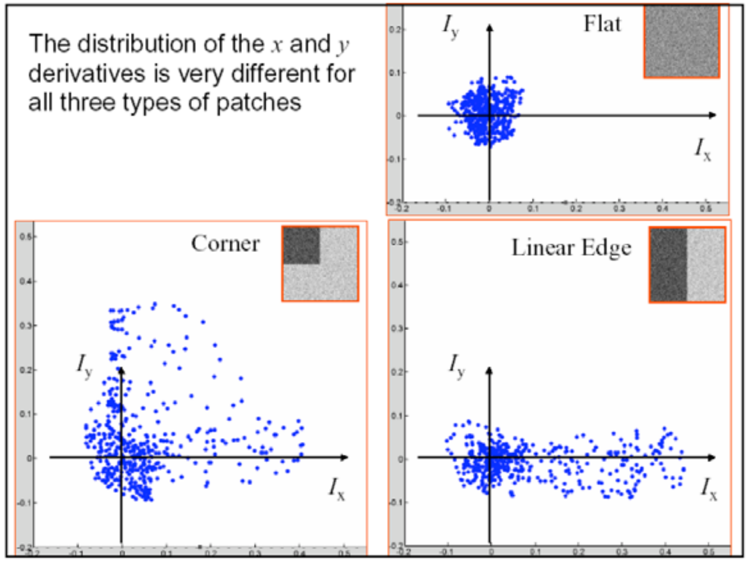
## Harris corner detector



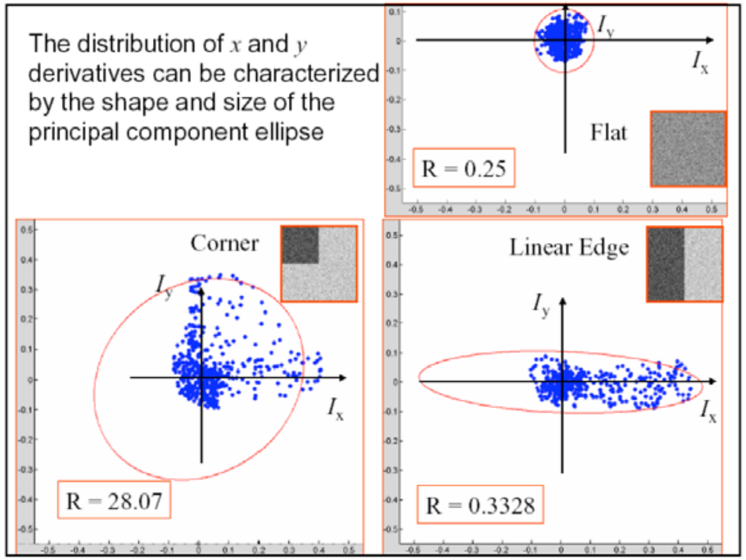

## Another view



## Another view




## Another view




## Summary of Harris detector

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

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

## Summary of Harris detector

4. Define the matrix at each pixel

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

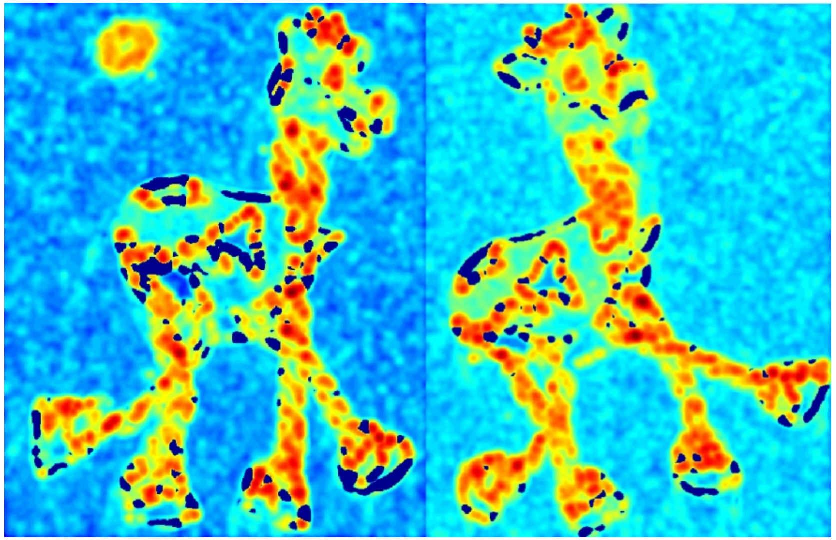
5. Compute the response of the detector at each pixel

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

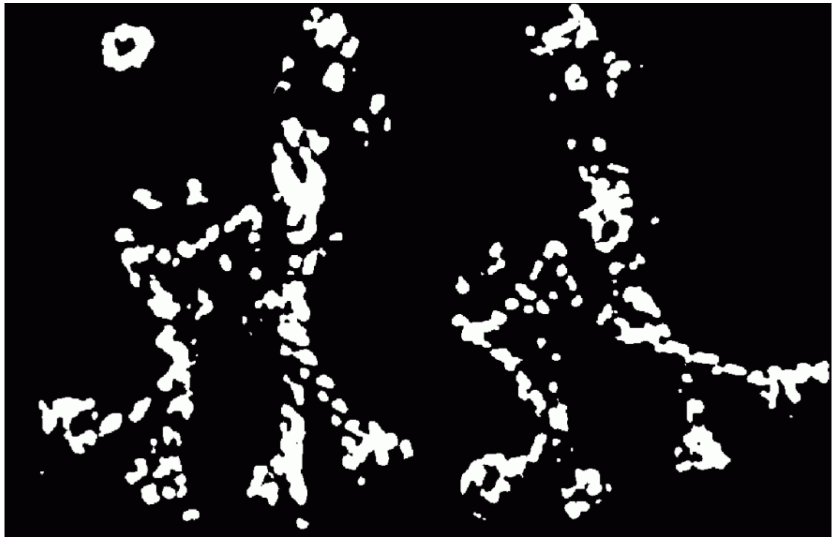
6. Threshold on value of R; compute nonmax suppression.

## Harris corner detector (input)

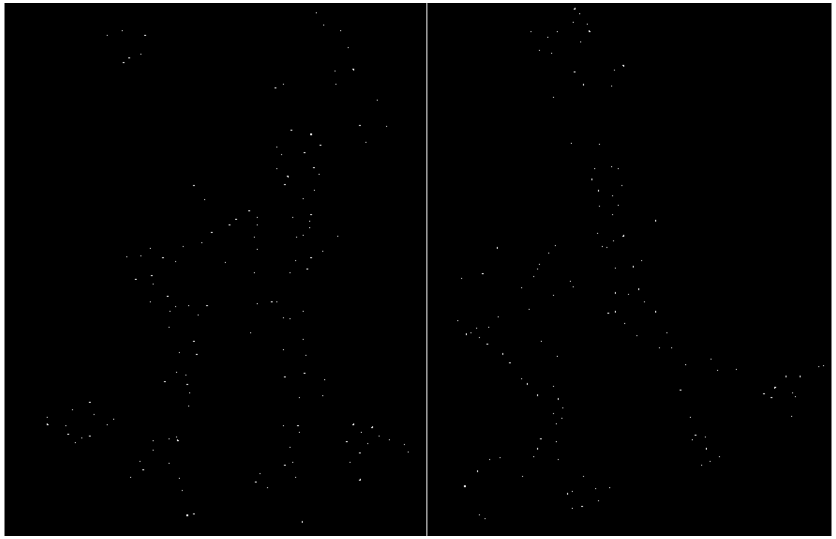
## Corner response R



## Threshold on R



## Local maximum of R



## Harris corner detector

## Harris detector: summary

- Average intensity change in direction $[u,v]$ can be expressed as a bilinear form:

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

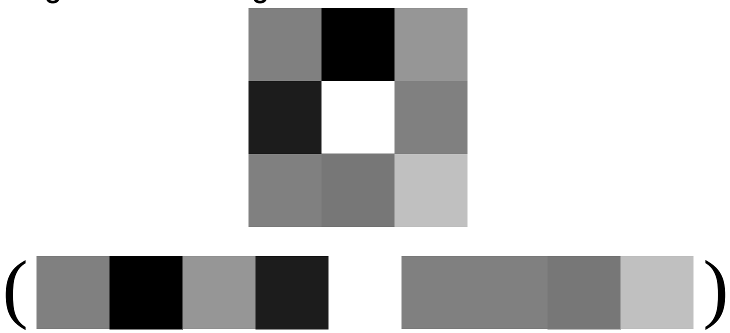
- Describe a point in terms of eigenvalues of *M*: *measure of corner response*

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

- A good (corner) point should have a *large intensity change* in *all directions*, i.e. *R* should be large positive

## Now we know where features are

- But, how to match them?
- What is the 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.

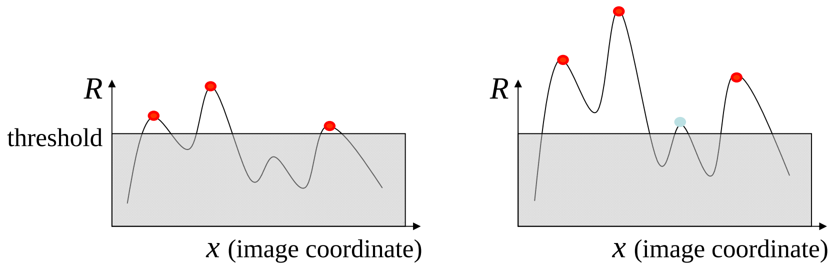


## Harris detector: some properties

- Partial invariance to *affine intensity* change

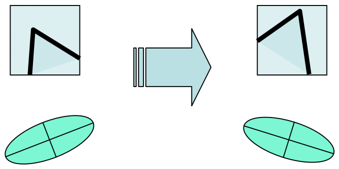
  ✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

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



## Harris Detector: Some Properties

- Rotation invariance



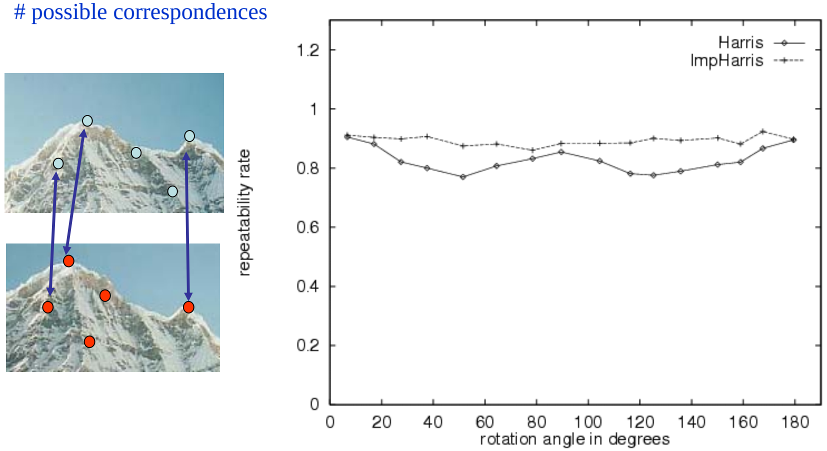
Ellipse rotates but its shape (i.e. eigenvalues) remains the same

*Corner response R* is invariant to image rotation

## Harris Detector is rotation invariant

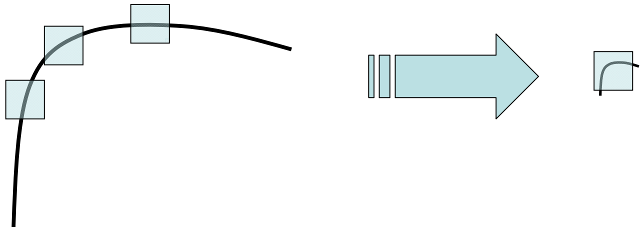
Repeatability rate:

$$\frac{\text{\# correspondences}}{\text{\# possible correspondences}}$$




## Harris Detector: Some Properties

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



All points will be classified as edges

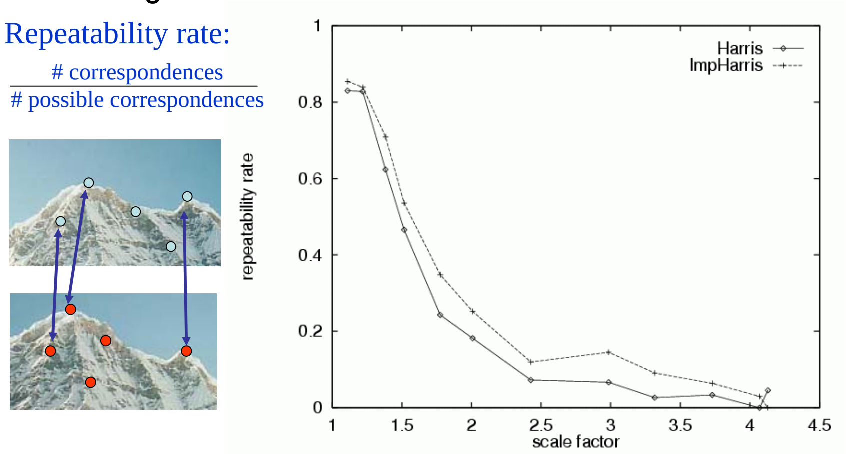
**Corner !**

## Harris detector: some properties

- Quality of Harris detector for different scale changes

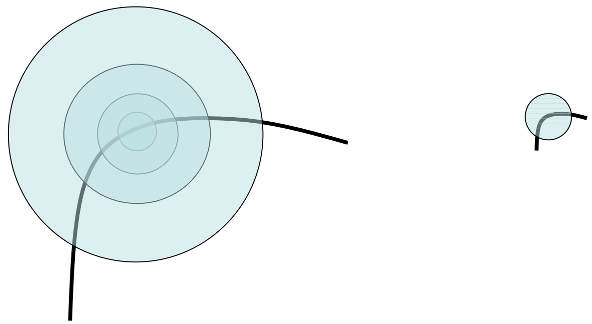
Repeatability rate:

$$\frac{\text{\# correspondences}}{\text{\# possible correspondences}}$$



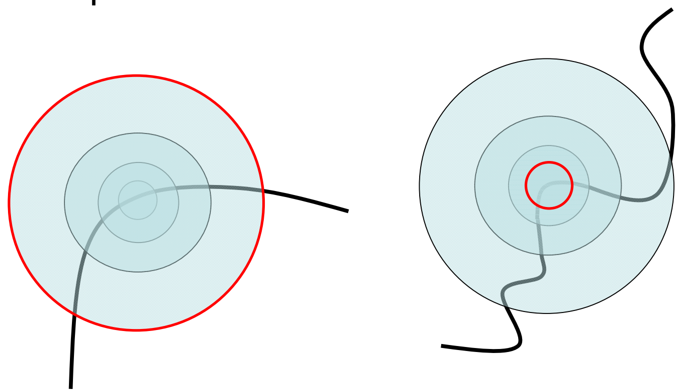
## Scale invariant detection

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



## Scale invariant detection

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



# SIFT
# (Scale Invariant Feature Transform)

## SIFT

- SIFT is an carefully designed procedure with empirically determined parameters for the invariant and distinctive features.

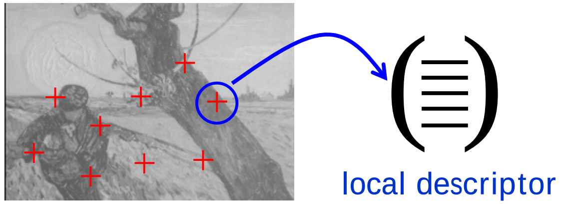
## SIFT stages:

- Scale-space extrema detection
- Keypoint localization

**detector**

- Orientation assignment
- Keypoint descriptor

**descriptor**




**A 500x500 image gives about 2000 features**

## 1. Detection of scale-space extrema

- For scale invariance, search for stable features across all possible scales using a continuous function of scale, scale space.
- SIFT uses DoG filter for scale space because it is efficient and as stable as scale-normalized Laplacian of Gaussian.

## DoG filtering

Convolution with a variable-scale Gaussian

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

$$G(x, y, \sigma) = 1/(2\pi\sigma^2)\exp^{-(x^2+y^2)/\sigma^2}$$

Difference-of-Gaussian (DoG) filter

$$G(x, y, k\sigma) - G(x, y, \sigma)$$

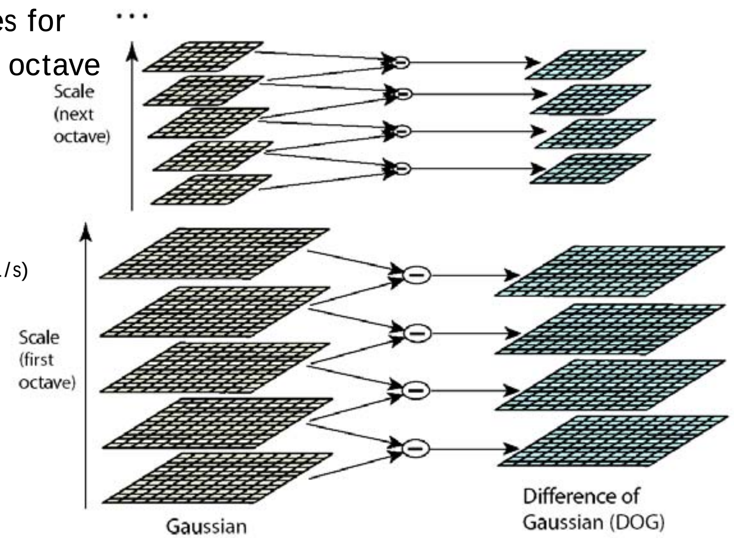
Convolution with the DoG filter

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned}$$

## Scale space

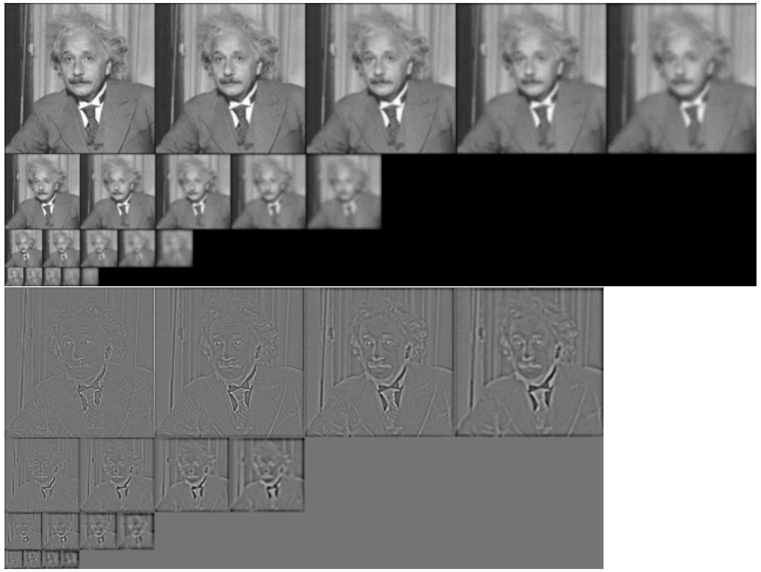
$\sigma$ doubles for the next octave

$K=2^{(1/s)}$

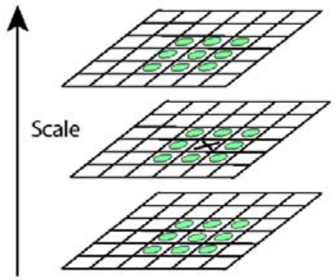



Dividing into octave is for efficiency only.

## Detection of scale-space extrema



## Keypoint localization

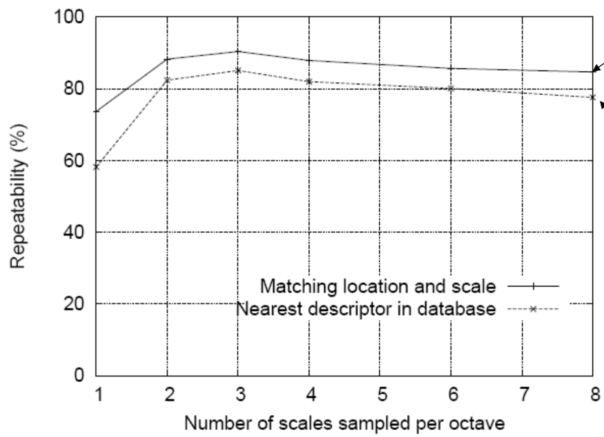



X is selected if it is larger or smaller than all 26 neighbors

## Decide scale sampling frequency

- It is impossible to sample the whole space, tradeoff efficiency with completeness.
- Decide the best sampling frequency by experimenting on 32 real image subject to synthetic transformations. (rotation, scaling, affine stretch, brightness and contrast change, adding noise…)

## Decide scale sampling frequency

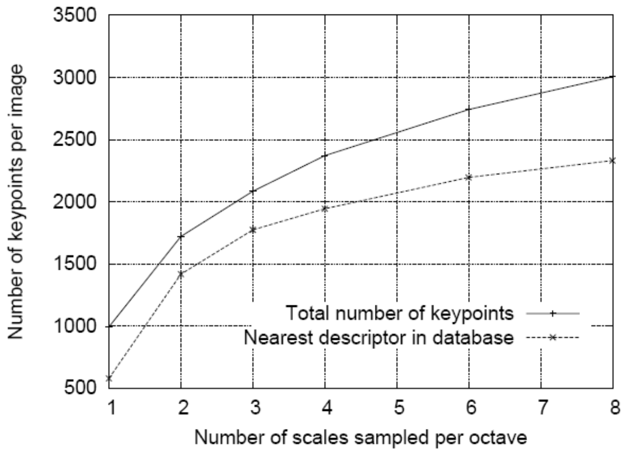



for detector, repeatability

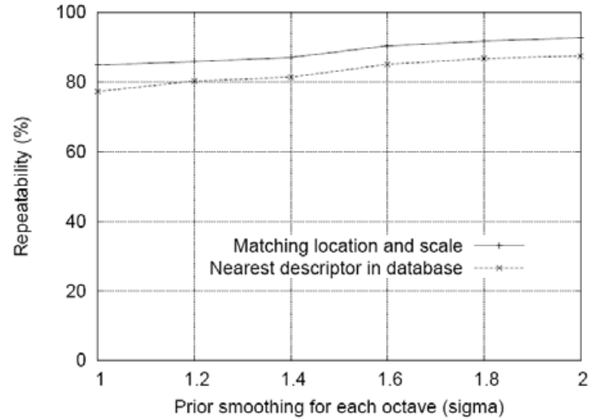
for descriptor, distinctiveness

s=3 is the best, for larger s, too many unstable features

## Decide scale sampling frequency

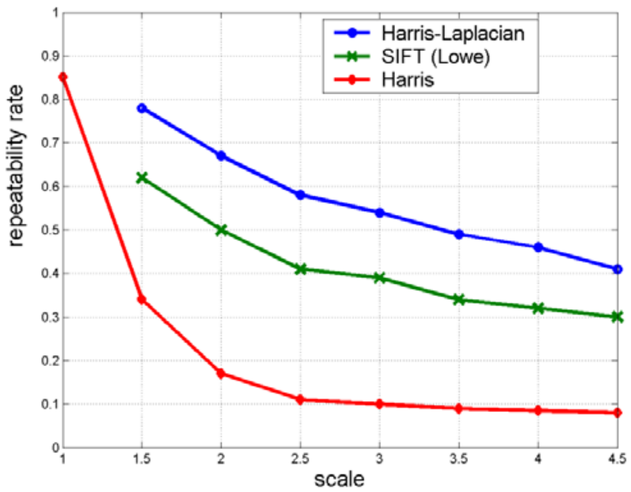



## Pre-smoothing



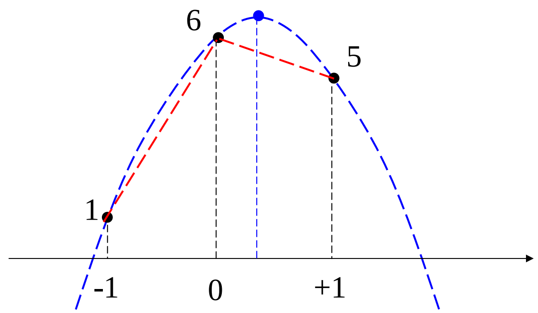

$\sigma$ =1.6, plus a double expansion

## Scale invariance



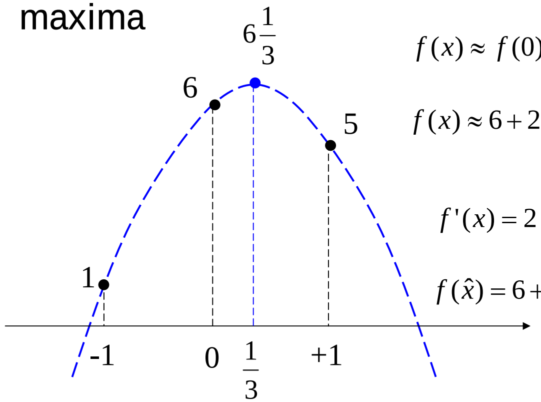

## 2. Accurate keypoint localization

- Reject points with low contrast (flat) and poorly localized along an edge (edge)
- Fit a 3D quadratic function for sub-pixel maxima



## 2. Accurate keypoint localization

- Reject points with low contrast and poorly localized along an edge
- Fit a 3D quadratic function for sub-pixel maxima




$$f(x) \approx f(0) + f'(0)x + \frac{f''(0)}{2}x^2$$

$$f(x) \approx 6 + 2x + \frac{-6}{2}x^2 = 6 + 2x - 3x^2$$

$$f'(x) = 2 - 6x = 0 \longrightarrow \hat{x} = \frac{1}{3}$$

$$f(\hat{x}) = 6 + 2\cdot\frac{1}{3} - 3\cdot\left(\frac{1}{3}\right)^2 = 6\frac{1}{3}$$

## 2. Accurate keypoint localization

- Taylor series of several variables

$$T(x_1,\cdots,x_d) = \sum_{n_1=0}^{\infty}\cdots\sum_{n_d=0}^{\infty}\frac{\partial^{n_1}}{\partial x_1^{n_1}}\cdots\frac{\partial^{n_d}}{\partial x_d^{n_d}}\frac{f(a_1,\cdots,a_d)}{n_1!\cdots n_d!}(x_1-a_1)^{n_1}\cdots(x_d-a_d)^{n_d}$$

- Two variables

$$f(x,y) \approx f(0,0) + \left(\frac{\partial f}{\partial x}x + \frac{\partial f}{\partial y}y\right) + \frac{1}{2}\left(\frac{\partial^2 f}{\partial x\partial x}x^2 + 2\frac{\partial^2 f}{\partial x\partial y}xy + \frac{\partial^2 f}{\partial y\partial y}y^2\right)$$

$$f\left(\begin{bmatrix}x\\y\end{bmatrix}\right) \approx f\left(\begin{bmatrix}0\\0\end{bmatrix}\right) + \begin{bmatrix}\frac{\partial f}{\partial x} & \frac{\partial f}{\partial y}\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} + \frac{1}{2}\begin{bmatrix}x & y\end{bmatrix}\begin{bmatrix}\frac{\partial^2 f}{\partial x\partial x} & \frac{\partial^2 f}{\partial x\partial y}\\ \frac{\partial^2 f}{\partial x\partial y} & \frac{\partial^2 f}{\partial y\partial y}\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$$

$$f(\mathbf{x}) \approx f(\mathbf{0}) + \frac{\partial f}{\partial \mathbf{x}}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

## Accurate keypoint localization

- Taylor expansion in a matrix form, $\mathbf{x}$ is a vector, $f$ maps $\mathbf{x}$ to a scalar

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

Hessian matrix (often symmetric)

gradient $\begin{pmatrix}\frac{\partial f}{\partial x_1}\\ \frac{\partial f}{\partial x_1}\\ \vdots\\ \frac{\partial f}{\partial x_n}\end{pmatrix}$

$$\begin{pmatrix}\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\partial x_n}\\ \frac{\partial^2 f}{\partial x_2\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2 f}{\partial x_n\partial x_1} & \frac{\partial^2 f}{\partial x_n\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}\end{pmatrix}$$

## 2D illustration

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

| | | |
|---|---|---|
| $f_{-1,1}$ | $f_{0,1}$ | $f_{1,1}$ |
| $f_{-1,0}$ | $f_{0,0}$ | $f_{1,0}$ |
| $f_{-1,-1}$ | $f_{0,-1}$ | $f_{1,-1}$ |

$$\begin{aligned}\frac{\partial f}{\partial x} &= (f_{1,0} - f_{-1,0})/2\\ \frac{\partial f}{\partial y} &= (f_{0,1} - f_{0,-1})/2\\ \frac{\partial^2 f}{\partial x^2} &= f_{1,0} - 2f_{0,0} + f_{-1,0}\\ \frac{\partial^2 f}{\partial y^2} &= f_{0,1} - 2f_{0,0} + f_{0,-1}\\ \frac{\partial^2 f}{\partial x\partial y} &= (f_{-1,-1} - f_{-1,1} - f_{1,-1} + f_{1,1})/4\end{aligned}$$

## 2D example

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

| | | |
|---|---|---|
| -17 | -1 | -1 |
| -9 | 7 | 7 |
| -9 | 7 | 7 |

## Derivation of matrix form

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

$$h(\mathbf{x}) = \mathbf{g}^{\mathrm{T}}\mathbf{x} = \begin{pmatrix} g_1 & \cdots & g_n \end{pmatrix}\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^{n} g_i x_i$$

$$\frac{\partial h}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial h}{\partial x_1} \\ \vdots \\ \frac{\partial h}{\partial x_n} \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_n \end{pmatrix} = \mathbf{g}$$

## Derivation of matrix form

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

$$h(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \begin{pmatrix} x_1 & \cdots & x_n \end{pmatrix}^T \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix}\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_i x_j$$

$$\frac{\partial h}{\partial \mathbf{x}} = \begin{pmatrix} \frac{\partial h}{\partial x_1} \\ \vdots \\ \frac{\partial h}{\partial x_n} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} a_{i1}x_i + \sum_{j=1}^{n} a_{1j}x_j \\ \vdots \\ \sum_{i=1}^{n} a_{in}x_i + \sum_{j=1}^{n} a_{nj}x_j \end{pmatrix} = \mathbf{A}^{\mathrm{T}}\mathbf{x} + \mathbf{A}\mathbf{x} = (\mathbf{A}^{\mathrm{T}} + \mathbf{A})\mathbf{x}$$

## Derivation of matrix form

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

$$\frac{\partial h}{\partial \mathbf{x}} = \frac{\partial f}{\partial \mathbf{x}}^T + \frac{1}{2}\left(\frac{\partial^2 f}{\partial \mathbf{x}^2} + \frac{\partial^2 f}{\partial \mathbf{x}^2}^T\right)x = \frac{\partial f}{\partial \mathbf{x}}^T + \frac{\partial^2 f}{\partial \mathbf{x}^2}x$$

$$\mathbf{x}_m = -\frac{\partial^2 f}{\partial \mathbf{x}^2}^{-1}\frac{\partial f}{\partial \mathbf{x}}$$

## Accurate keypoint localization

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

- $\mathbf{x}$ is a 3-vector
- Change sample point if offset is larger than 0.5
- Throw out low contrast (<0.03)

## Accurate keypoint localization

- Throw out low contrast $|D(\hat{\mathbf{x}})| < 0.03$

$$D(\hat{\mathbf{x}}) = D + \frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}} + \frac{1}{2}\hat{\mathbf{x}}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\hat{\mathbf{x}}$$

$$= D + \frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}} + \frac{1}{2}\left(-\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}\right)^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\left(-\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}\right)$$

$$= D + \frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}} + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}^{-T} \frac{\partial^2 D}{\partial \mathbf{x}^2}\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}$$

$$= D + \frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}} + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}$$

$$= D + \frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}} + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T(-\hat{\mathbf{x}})$$

$$= D + \frac{1}{2}\frac{\partial D}{\partial \mathbf{x}}^T \hat{\mathbf{x}}$$

## Eliminating edge responses

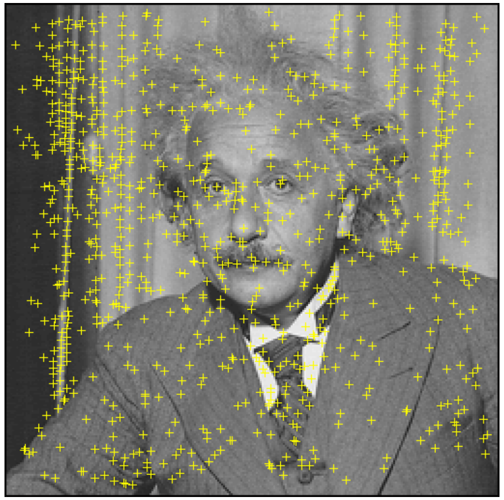
$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$ Hessian matrix at keypoint location

$$\mathrm{Tr}(\mathbf{H}) = D_{xx} + D_{yy} = \alpha + \beta,$$
$$\mathrm{Det}(\mathbf{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta.$$

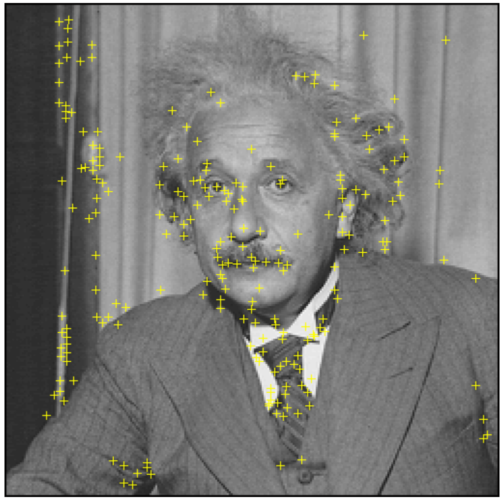
Let $\alpha = r\beta$ $\quad \frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(r\beta+\beta)^2}{r\beta^2} = \frac{(r+1)^2}{r}$

Keep the points with $\frac{\mathrm{Tr}(\mathbf{H})^2}{\mathrm{Det}(\mathbf{H})} < \frac{(r+1)^2}{r}$. r=10

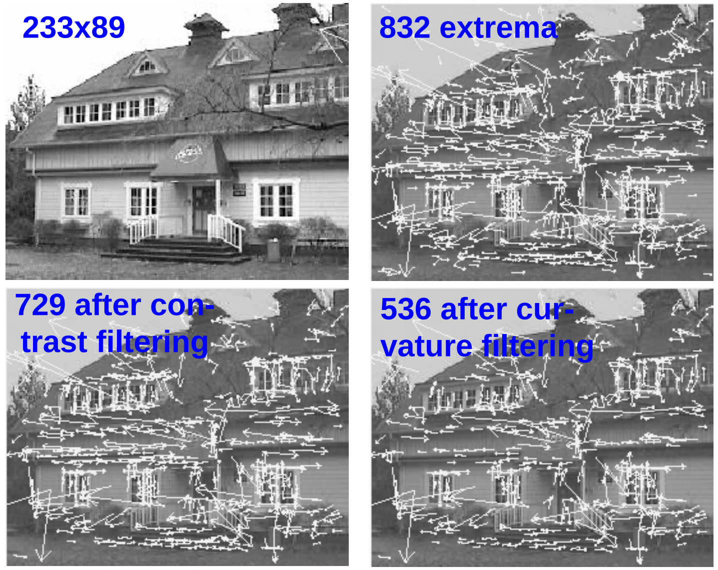
## Maxima in D



## Remove low contrast and edges



## Keypoint detector

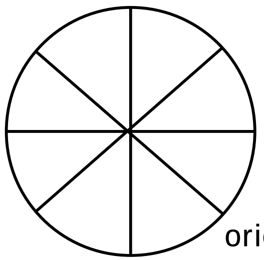


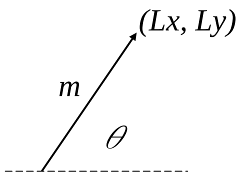

## 3. Orientation assignment

- By assigning a consistent orientation, the keypoint descriptor can be orientation invariant.
- For a keypoint, L is the Gaussian-smoothed image with the closest scale,

$$m(x,y) = \sqrt{(L(x+1,y)-L(x-1,y))^2 + (L(x,y+1)-L(x,y-1))^2}$$

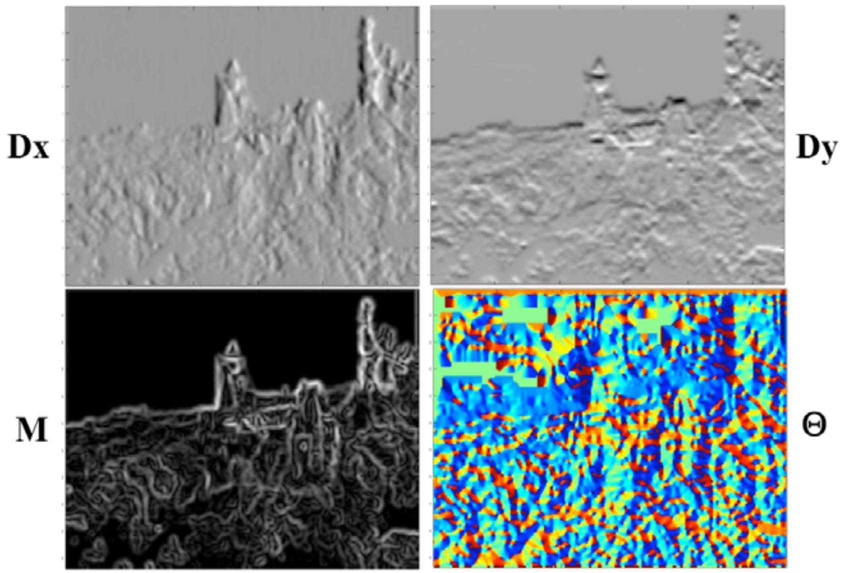
$$\theta(x,y) = \tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y)))$$





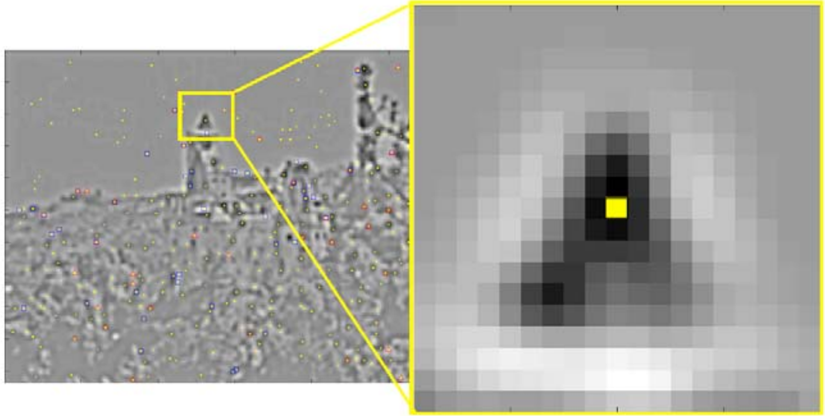

orientation histogram (36 bins)

## Orientation assignment



## Orientation assignment

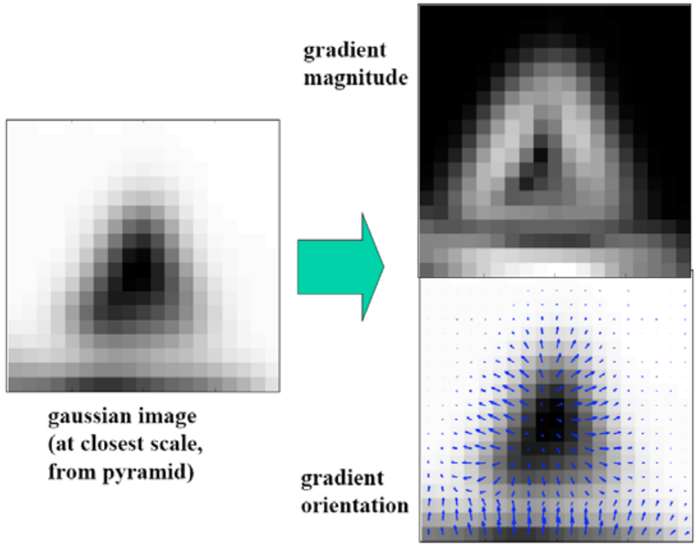




•Keypoint location = extrema location
•Keypoint scale is scale of the DOG image

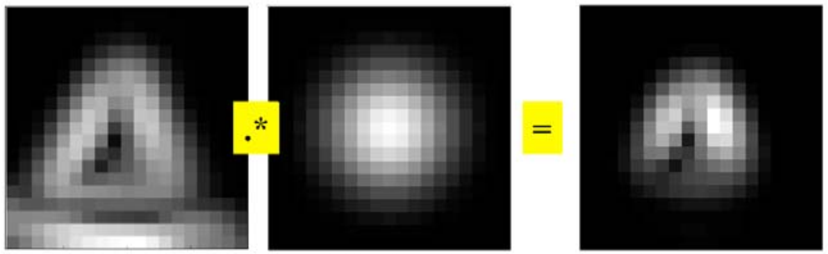
## Orientation assignment






## Orientation assignment






gradient magnitude

weighted by 2D gaussian kernel

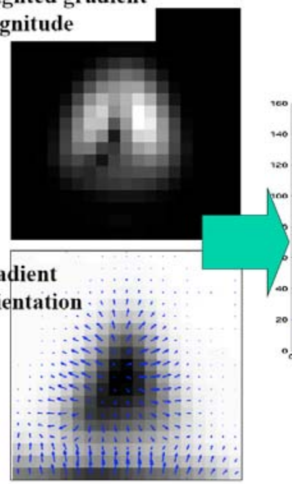
weighted gradient magnitude

$\sigma$ =1.5*scale of the keypoint

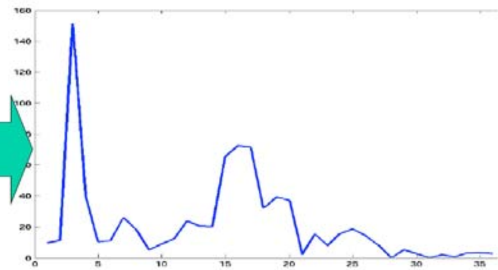
## Orientation assignment



weighted gradient magnitude

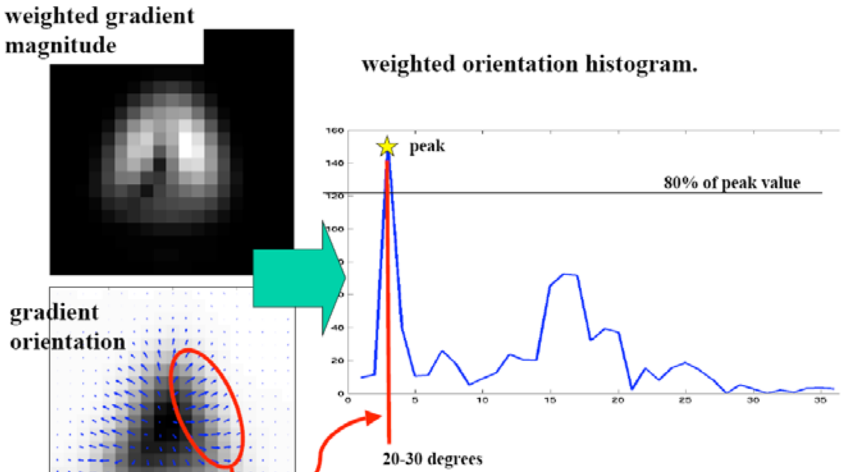



weighted orientation histogram.
Each bucket contains sum of weighted gradient magnitudes corresponding to angles that fall within that bucket.



36 buckets
10 degree range of angles in each bucket, i.e.
0 <=ang<10 : bucket 1
10<=ang<20 : bucket 2
20<=ang<30 : bucket 3 …

## Orientation assignment

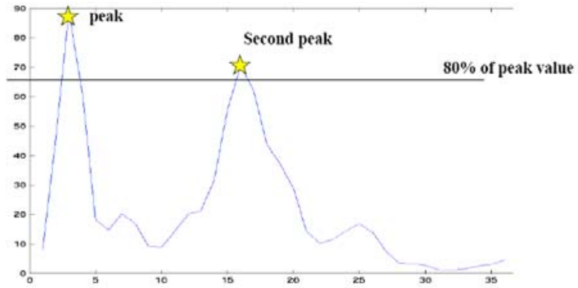



Orientation of keypoint
is approximately 25 degrees

## Orientation assignment

There may be multiple orientations.

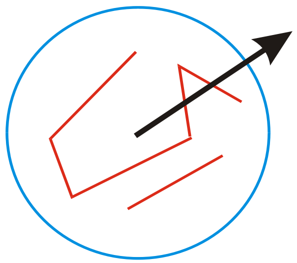
accurate peak position is determined by fitting

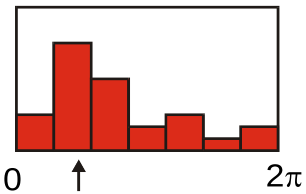



In this case, generate duplicate keypoints, one with orientation at 25 degrees, one at 155 degrees.

Design decision: you may want to limit number of possible multiple peaks to two.

## Orientation assignment





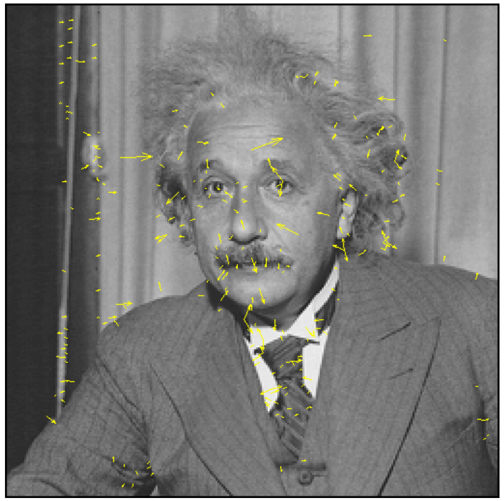

36-bin orientation histogram over 360°, weighted by m and 1.5*scale falloff

Peak is the orientation

Local peak within 80% creates multiple orientations

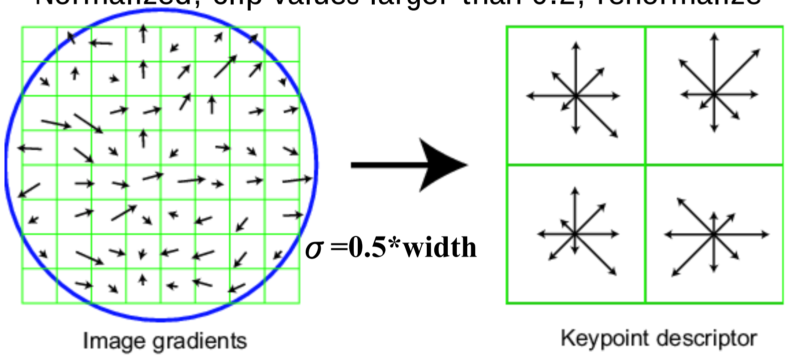
About 15% has multiple orientations and they contribute a lot to stability

## SIFT descriptor

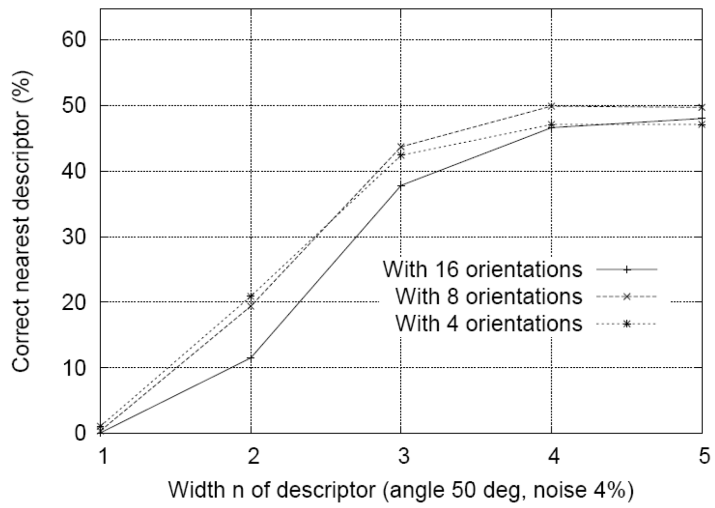


## 4. Local image descriptor

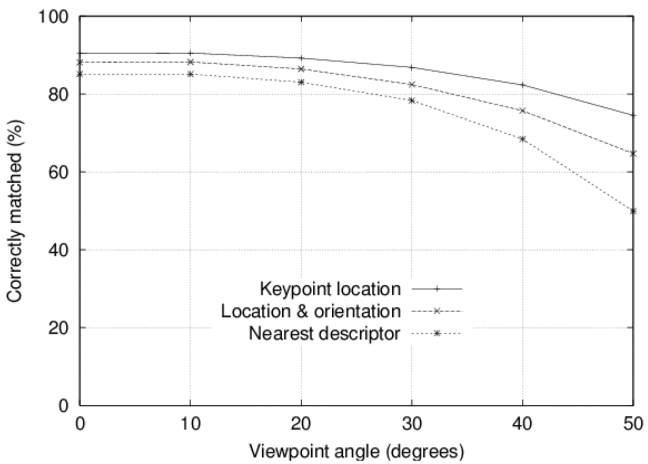
- Thresholded image gradients are sampled over 16x16 array of locations in scale space
- Create array of orientation histograms (w.r.t. key orientation)
- 8 orientations x 4x4 histogram array = 128 dimensions
- Normalized, clip values larger than 0.2, renormalize




## Why 4x4x8?



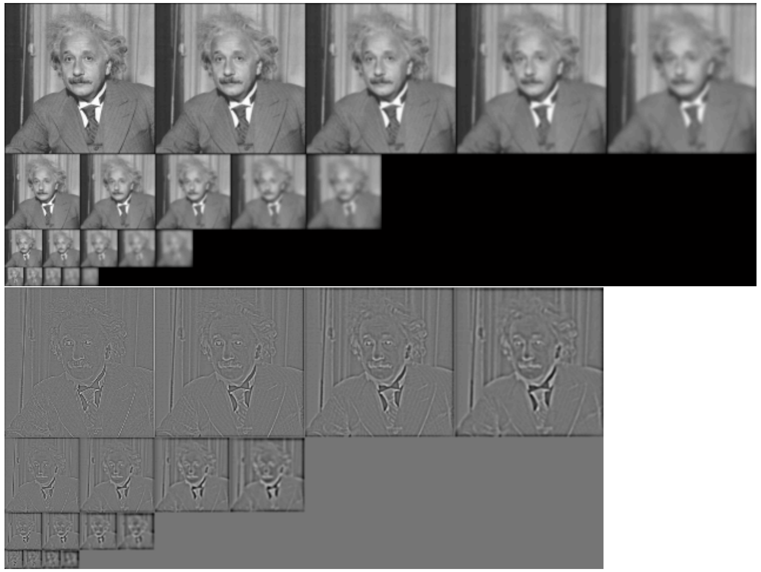

## Sensitivity to affine change



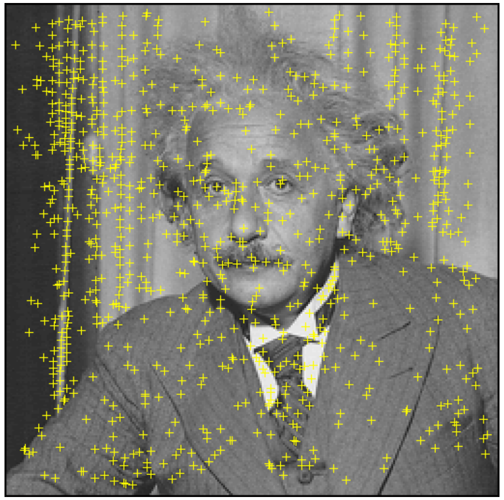

## Feature matching

- for a feature x, he found the closest feature $x_1$ and the second closest feature $x_2$. If the distance ratio of $d(x, x_1)$ and $d(x, x_1)$ is smaller than 0.8, then it is accepted as a match.

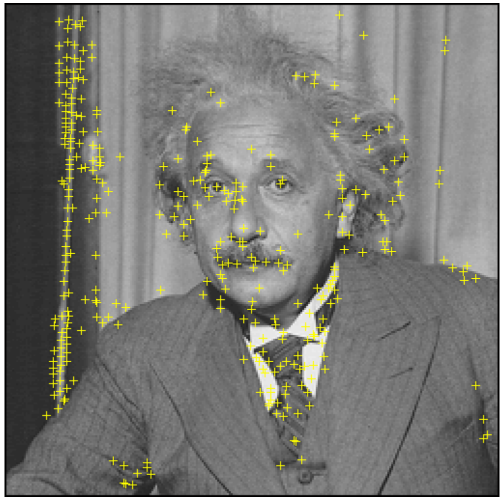
## SIFT flow



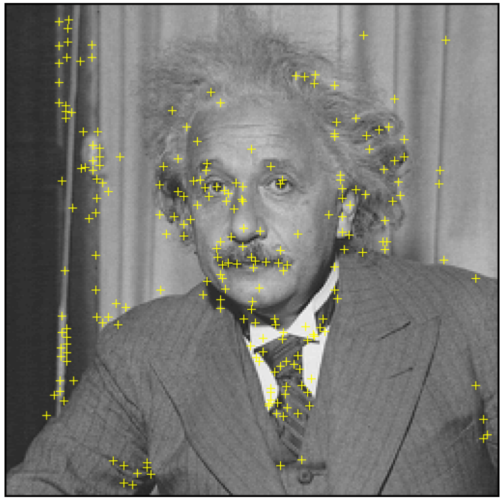
## Maxima in D



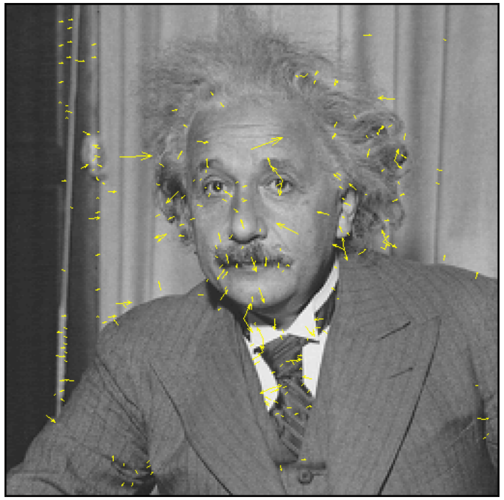
## Remove low contrast



## Remove edges



## SIFT descriptor



## Estimated rotation

- Computed affine transformation from rotated image to original image:

| | | |
|---|---|---|
| 0.7060 | -0.7052 | 128.4230 |
| 0.7057 | 0.7100 | -128.9491 |
| 0 | 0 | 1.0000 |

- Actual transformation from rotated image to original image:

| | | |
|---|---|---|
| 0.7071 | -0.7071 | 128.6934 |
| 0.7071 | 0.7071 | -128.6934 |
| 0 | 0 | 1.0000 |

## Reference


- Chris Harris, Mike Stephens, A Combined Corner and Edge Detector, 4th Alvey Vision Conference, 1988, pp147-151.
- David G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision, 60(2), 2004, pp91-110.
- SIFT Keypoint Detector, David Lowe.
- Matlab SIFT Tutorial, University of Toronto.