Scale Invariant Feature Transform

Jenn-Jier James Lien (連 震 杰)
Professor

Computer Science and Information Engineering
National Cheng Kung University

(O) (06) 2757575 ext. 62540
jjlien@csie.ncku.edu.tw
http://robotics.csie.ncku.edu.tw

Major Issues

1. SIFT: Scale Invariant Feature Transform

2. Detection of Scale-Space Extrema
   1) Difference of Gaussian (DoG) filter Vs. Laplacian of Gaussian (LoG)
   2) Non-maxima/minima suppression (Local Maximum)

3. Keypoint Localization (or Feature Selection)
   1) Keypoint localization in sub-pixel accuracy
   2) Remove low contrast keypoints
   3) Eliminating edge responses by Hessian matrix

4. Keypoint/Feature Orientation Assignment

5. Keypoint/Feature Descriptor

Content

Content

Motivation

- This paper presents a method for extracting distinctive invariant features from images.
- The features are invariant to image scale and rotation.
- The features are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination.

Content

$L(\sigma) = G(\sigma) * I$ (Scale space)

$D(\sigma) = (G(k^{(n-1)}\sigma) - G(k^{n}\sigma)) * I$ (DoG images)

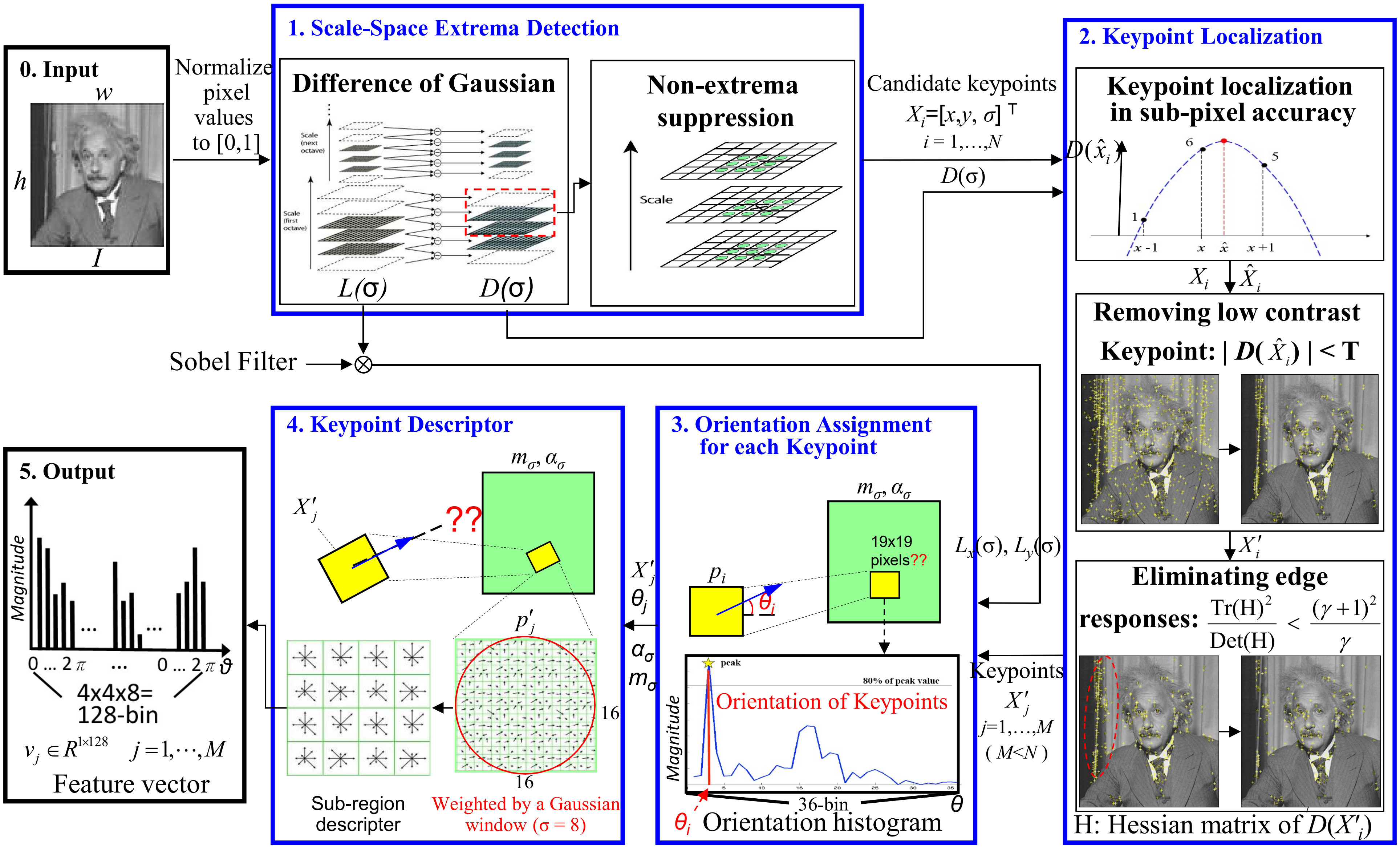
α_σ (Gradient orientation of $L(\sigma)$)

m_σ (Gradient magnitude of $L(\sigma)$)

X'_i (Accurate keypoint localization)

θj (Orientation of i-th keypoint)

2. System Flowchart

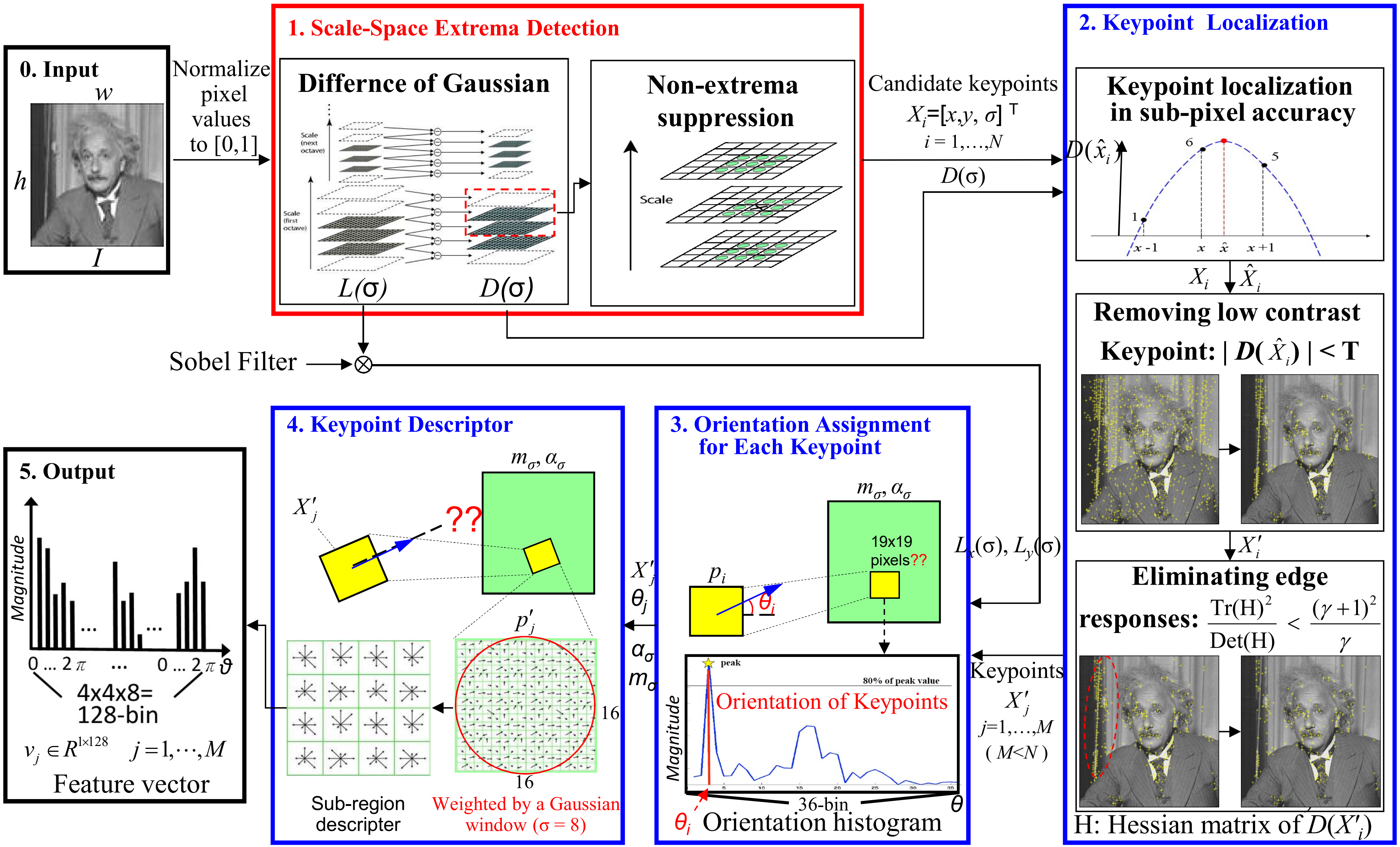


Content

System Flowchart

$L(\sigma) = G(\sigma) * I$ (Scale space)
$D(\sigma) = (G(k^{(n-1)}\sigma) - G(k^{n}\sigma)) * I$ (DoG images)

α_σ (Gradient orientation of $L(\sigma)$)
m_σ (Gradient magnitude of $L(\sigma)$)
X'_i (Accurate keypoint localization)
θj (Orientation of i-th keypoint)

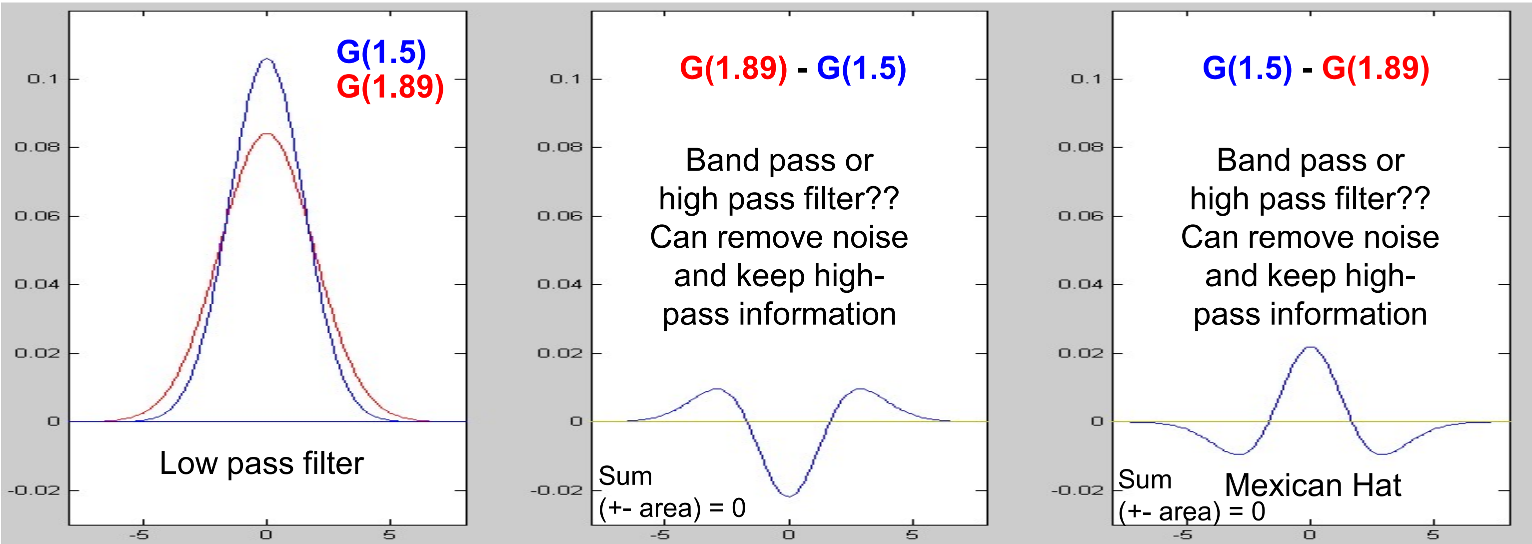


3. Detection of Scale-Space Extrema

◆ Difference of Gaussian (DoG)

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

Gaussian **Difference of Gaussian (DoG)**



低通與高通濾波器

- 高頻(high-frequency components)：短距離內灰階值變化值大，intensity gradient 大。(例：texture、邊緣或雜訊) 。
- 低頻(low-frequency components) ：短距離內灰階值變化值小 ，intensity gradient 小。(例：textureless)
- 高通濾波器（high-pass filter） ：保持高頻部分，減少低頻部分。(係數總和為0)(例如 Sobel 或Canny)
- 低通濾波器（low-pass filter） ：保持低頻部分，減少高頻部分。 Smooth (係數總和為1)(例如 Gaussian Filter)

Example

high-pass filter

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

low-pass filter

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Q&A. DoG vs. LoG (1/2)

- Difference of Gaussian operation provides an efficient approximation to the Laplacian.

- In reality, Laplacian has better performance than DoG but slow.

Q&A. DoG vs. LoG (2/2)

✓ LoG (Laplacian of Gaussian)

g(x,y): LoG image.
f(x,y): Input image.
★: Convolution.

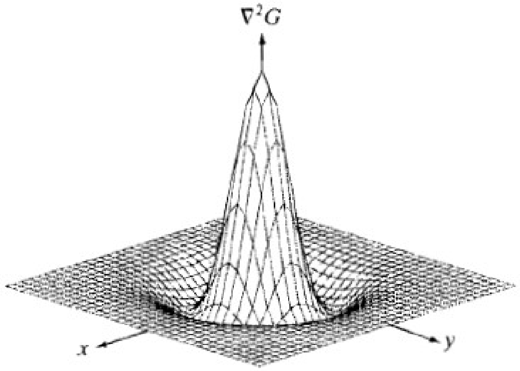
- Laplacian operator: $\nabla^2 = \frac{\partial^2}{\partial^2 x} + \frac{\partial^2}{\partial^2 y}$
- 2-D Gaussian function: $G(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$
- $g(x,y) = [\nabla^2 G(x,y)] \bigstar f(x,y)$
  $g(x,y) = \nabla^2 [G(x,y) \bigstar f(x,y)]$
  finding the zero crossings of *g(x, y)* to determine the locations of edges in *f(x, y)*.

$$\nabla^2 G(x,y) = \frac{\partial^2 G(x,y)}{\partial^2 x} + \frac{\partial^2 G(x,y)}{\partial^2 y}$$

$$= \frac{\partial}{\partial x}\left[\frac{-x}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right] + \frac{\partial}{\partial y}\left[\frac{-y}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}\right]$$

$$= \left[\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} + \left[\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$= \left[\frac{x^2 + y^2 - 2\sigma^2}{\sigma^4}\right] e^{-\frac{x^2+y^2}{2\sigma^2}}$$



LoG的三維模型

	-2	-1	0	1	2
-2	0	0	-1	0	0
-1	0	-1	-2	-1	0
0	-1	-2	16	-2	-1
1	0	-1	-2	-1	0
2	0	0	-1	0	0

lo=2

近似的5*5 mask

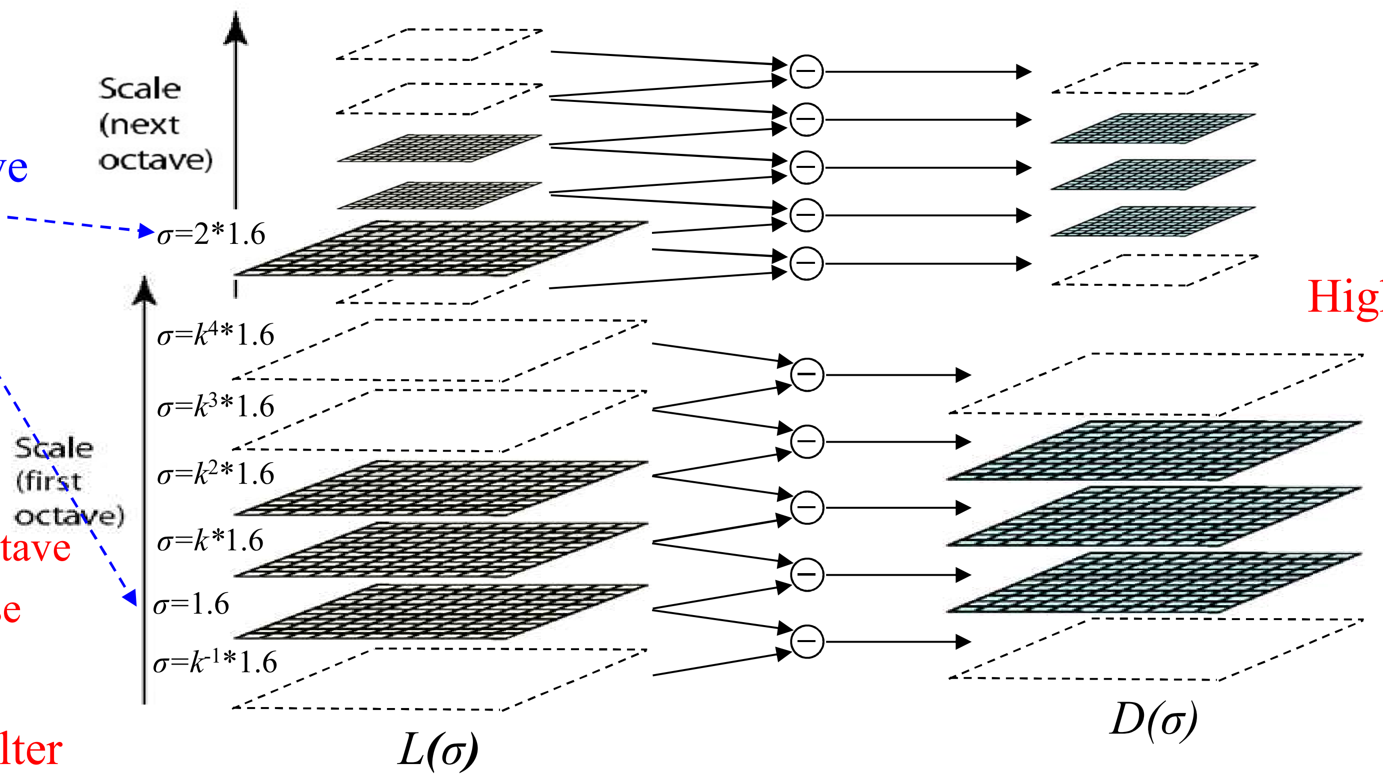
s is incorrect!!

3.1 Difference of Gaussian (DoG) filter (1/2)

Dividing into octave is for efficiency only

Until image size < 32x32

σ doubles for the next octave



High frequency

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

s: Image per octave

s = 3 in this case

Gaussian filter

DoG filter

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

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

$$\begin{aligned} &D(x,y,k^{(n-1)}\sigma) \qquad n = 1,2,\cdots,s \\ &= \boxed{(G(x,y,k^{n}\sigma) - G(x,y,k^{(n-1)}\sigma))} * I(x,y) \\ &= L(x,y,k^{n}\sigma) - L(x,y,k^{(n-1)}\sigma) \end{aligned}$$

W: Correct s!!

3.1 Difference of Gaussian (DoG) filter (1/2)

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

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

G(x,y,σ): Gaussian filter.

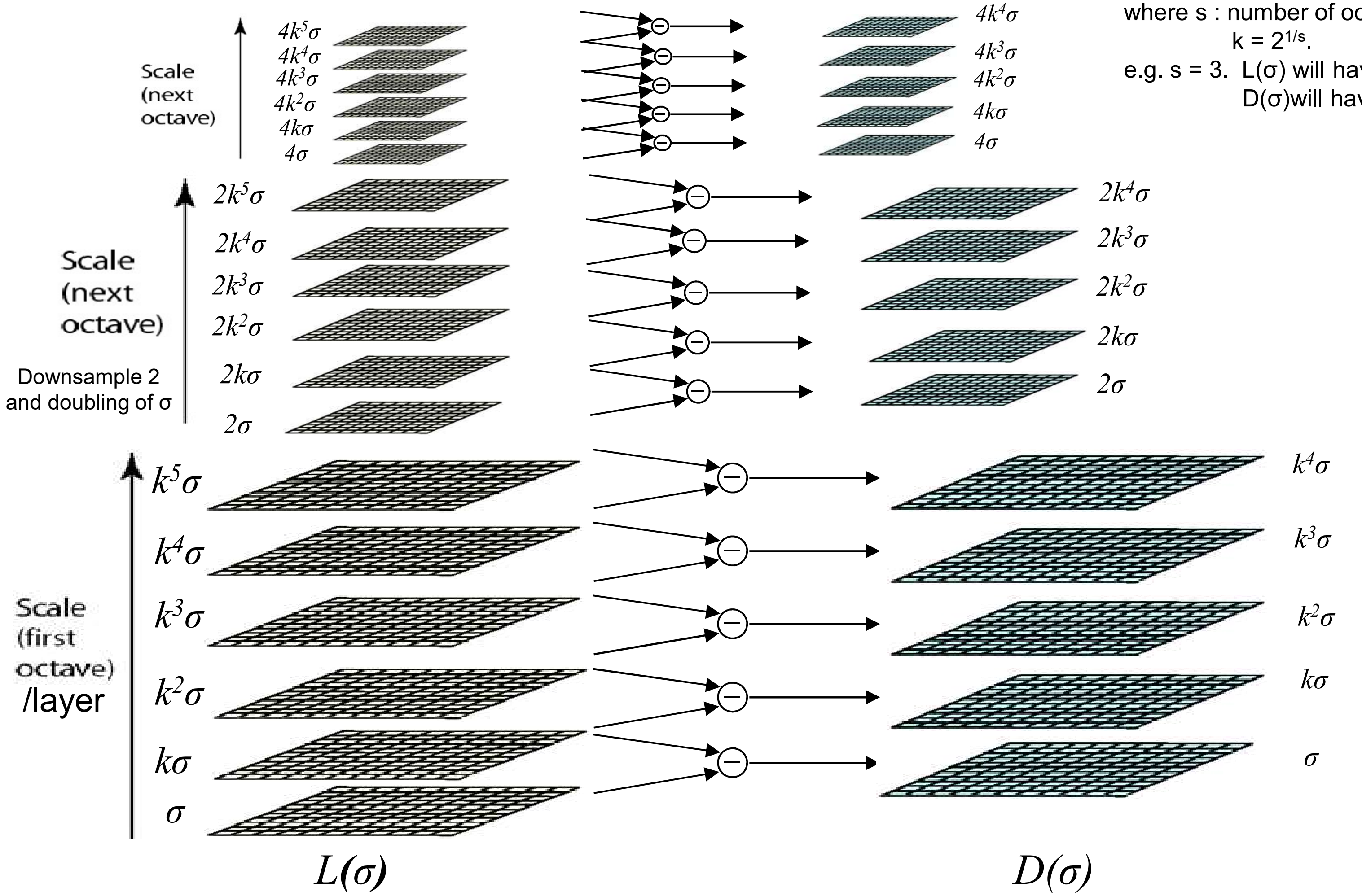
I(x,y) : Input Image.

$$D(x,y,k^{(n-1)}\sigma)=(G(x,y,k^{n}\sigma)-G(x,y,k^{(n-1)}\sigma))*I(x,y)$$

$$n=1,2,\cdots,s+2 \qquad =L(x,y,k^{n}\sigma)-L(x,y,k^{(n-1)}\sigma)$$

How many images in each octave /layer?

L(σ) will have s+3 images for each octave.
D(σ) will have s+2 images for each octave.
where s : number of octave /layer (3 ~ 5).
$k = 2^{1/s}$.
e.g. s = 3. L(σ) will have 6 images for each octave and D(σ)will have 5 images for each octave.



E.g.
s = 3, $k = 2^{1/3}$

$\boxed{k^3\sigma = 2\sigma \quad 2k^3\sigma = 4\sigma}$

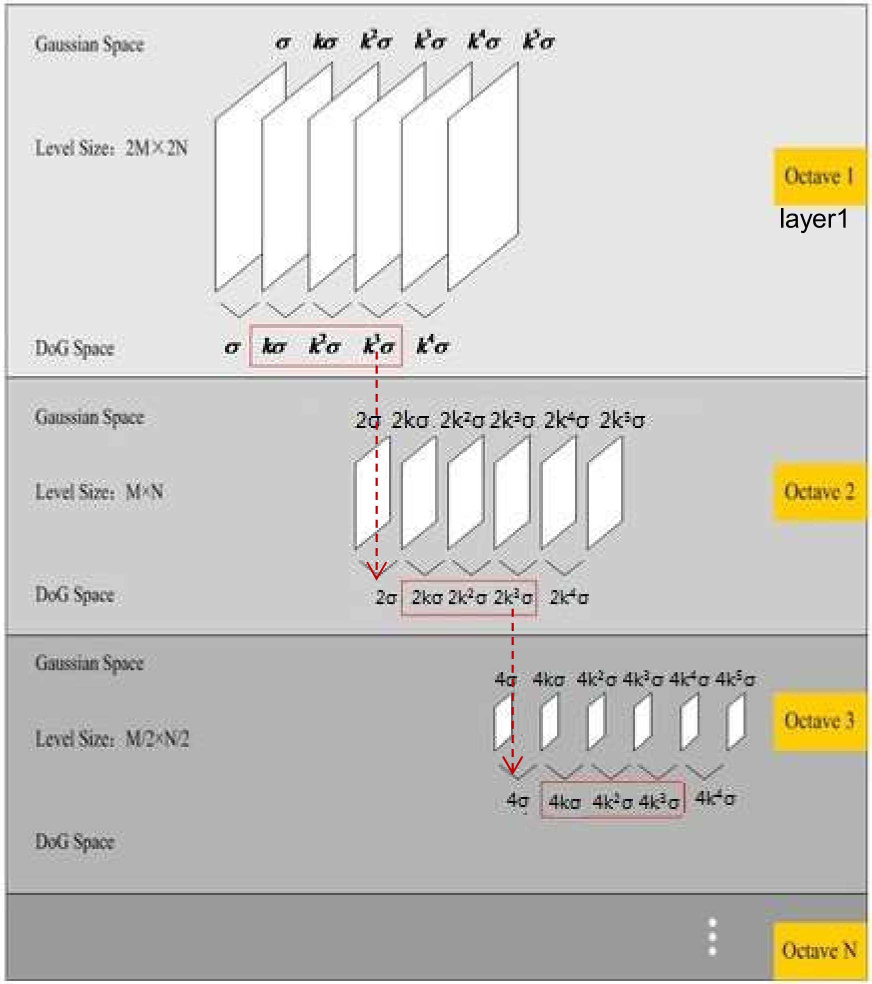
3.1 Difference of Gaussian (DoG) filter (2/2)

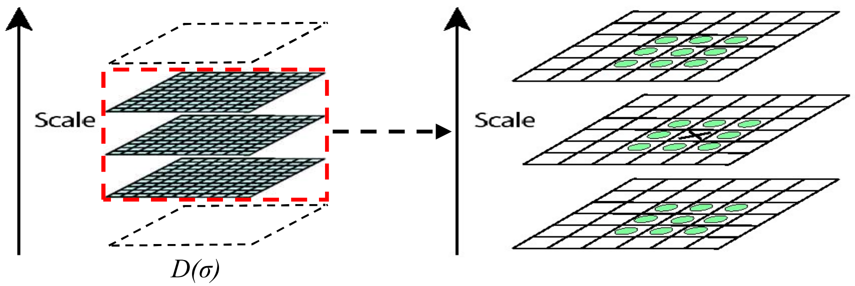
L(σ)

D(σ)

W: Correct s!!

3.2 Non-maxima/minima suppression





- $X_i = (x, y, \sigma)^T$ is selected as a candidate keypoint if it is larger or smaller than all 26 neighbors.
- Local maximum/minimum: After local extrema detection, we find extrema point as: $X_i = (x, y, \sigma)^T$

E.g.

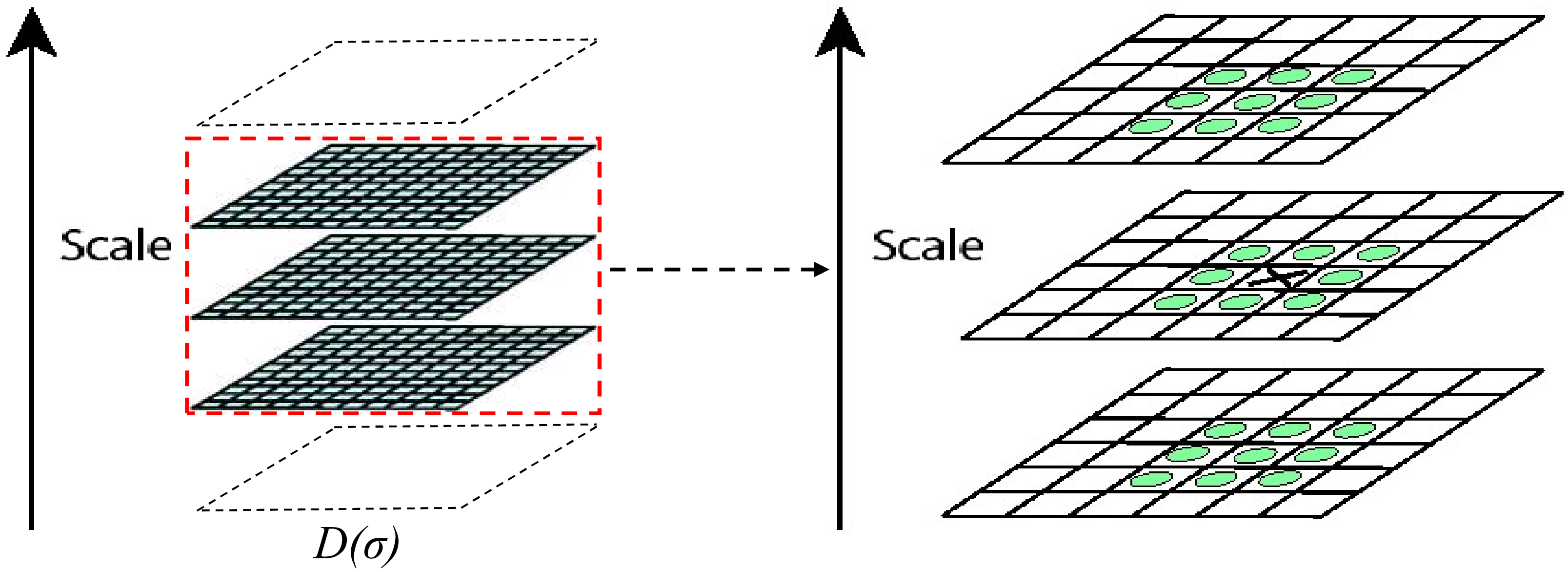
s = 3, k = $2^{1/3}$

$k^3\sigma = 2\sigma$

$2k^3\sigma = 4\sigma$

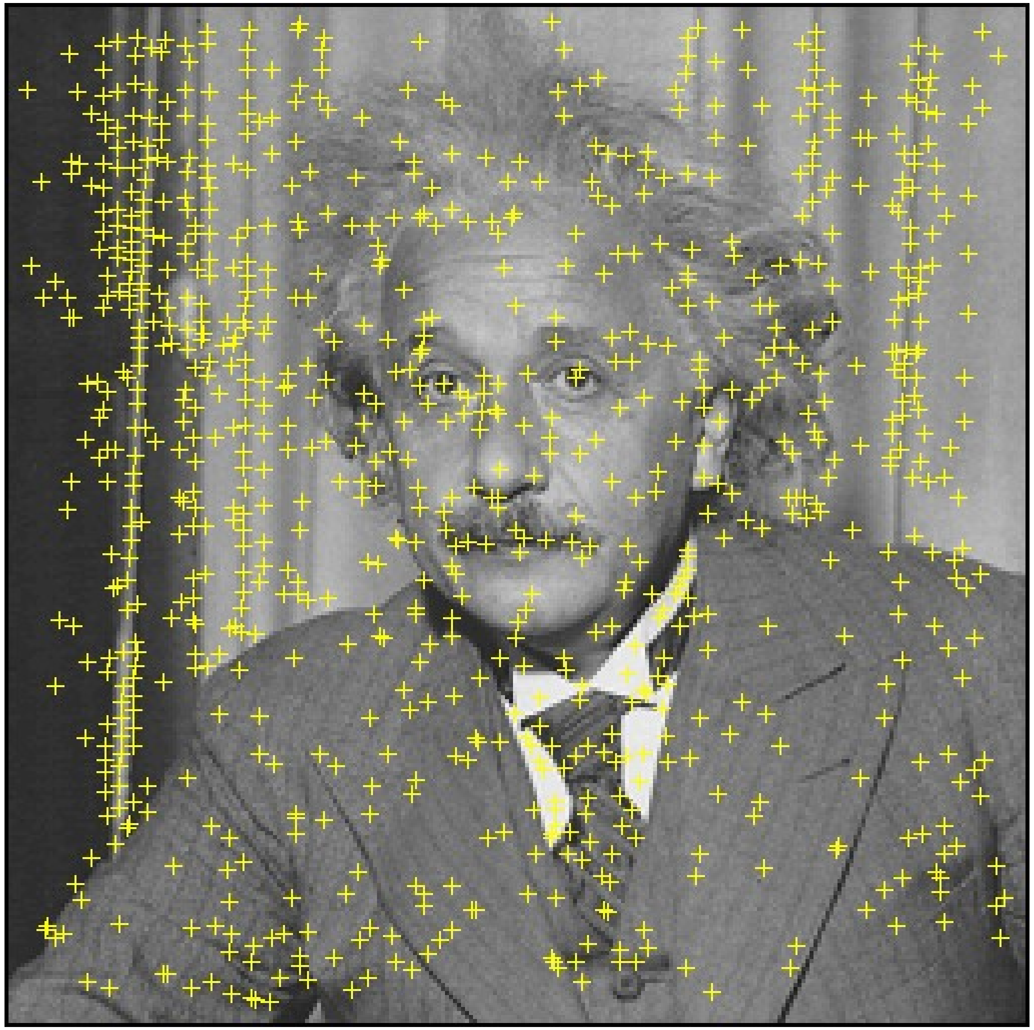
Reference :
http://blog.csdn.net/abcjennifer/article/details/7639681

3.2 Non-maxima/minima suppression (1/2) ??



- x is selected as a candidate keypoint if it is larger or smaller than all 26 neighbors.
- Local maximum/minimum
- Q? So does this candidate keypoint exist any 8-connected component keypoints? Probably Not

3.2 Non-maxima/minima suppression (2/2)



Extrema in *D* (Candidate keypoints)

Content

$L(\sigma) = G(\sigma) * I$ (Scale space)

$D(\sigma) = (G(k^{(n-1)}\sigma) - G(k^{n}\sigma)) * I$ (DoG images)

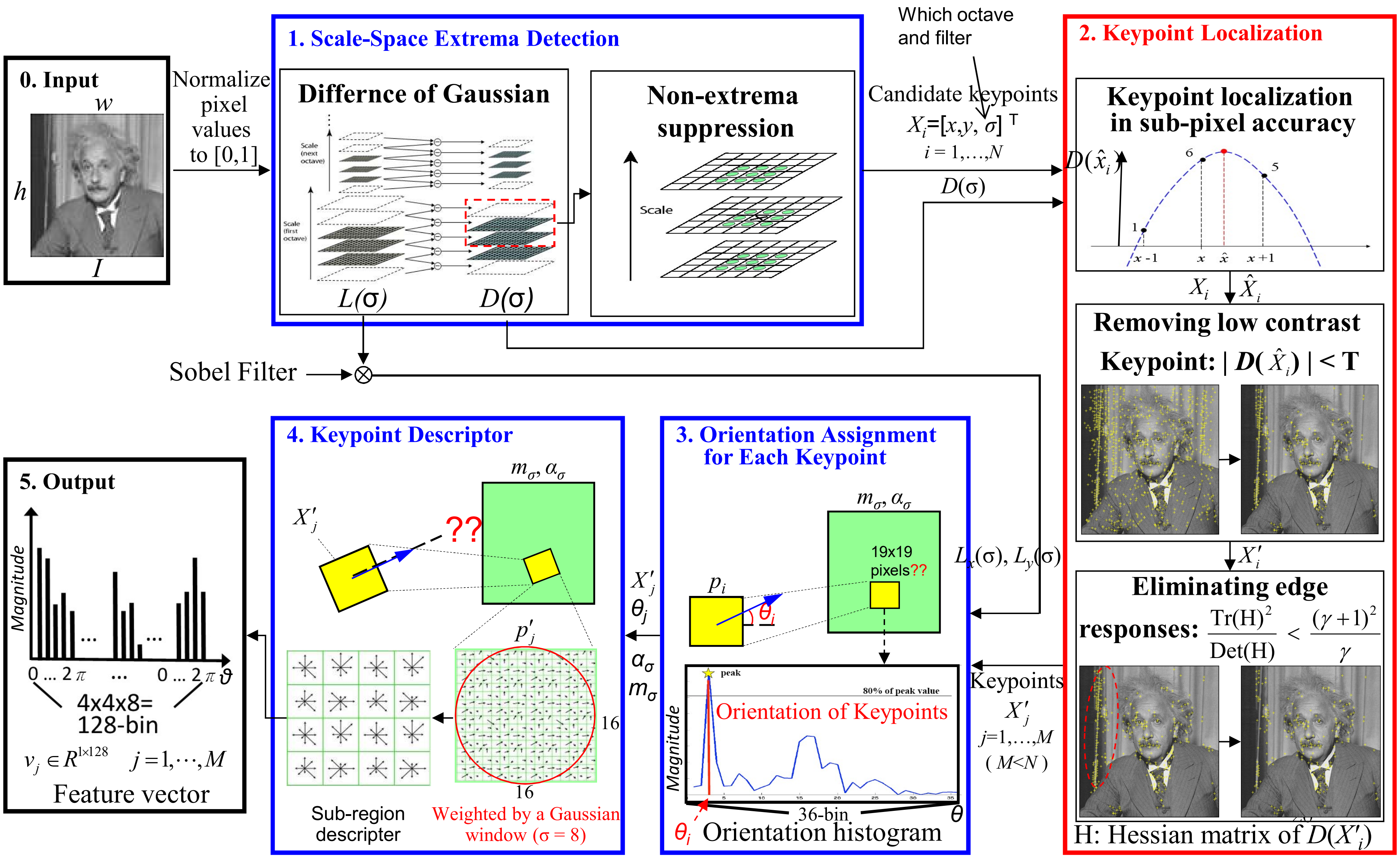
α_σ (Gradient orientation of $L(\sigma)$)

m_σ (Gradient magnitude of $L(\sigma)$)

X'_i (Accurate keypoint localization)

θj (Orientation of i-th keypoint)

System Flowchart



4. Accurate Keypoint Localization

Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This information allows points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge.

The initial implementation of this approach (Lowe, 1999) simply located keypoints at the location and scale of the central sample point. However, recently Brown has developed a method (Brown and Lowe, 2002) for fitting a 3D quadratic function to the local sample points to determine the interpolated location of the maximum, and his experiments showed that this provides a substantial improvement to matching and stability. His approach uses the Taylor expansion (up to the quadratic terms) of the scale-space function, $D(x, y, \sigma)$, shifted so that the origin is at the sample point:

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

where D and its derivatives are evaluated at the sample point and $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum, $\hat{\mathbf{x}}$, is determined by taking the derivative of this function with respect to $\mathbf{x}$ and setting it to zero, giving

$$\hat{\mathbf{x}} = -\frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1} \frac{\partial D}{\partial \mathbf{x}}. \tag{3}$$

As suggested by Brown, the Hessian and derivative of D are approximated by using differences of neighboring sample points. The resulting 3×3 linear system can be solved with minimal cost. If the offset $\hat{\mathbf{x}}$ is larger than 0.5 in any dimension, then it means that the extremum lies closer to a different sample point. In this case, the sample point is changed and the interpolation performed instead about that point. The final offset $\hat{\mathbf{x}}$ is added to the location of its sample point to get the interpolated estimate for the location of the extremum.

The function value at the extremum, $D(\hat{\mathbf{x}})$, is useful for rejecting unstable extrema with low contrast. This can be obtained by substituting Eqs (3) into (2), giving

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

For the experiments in this paper, all extrema with a value of $|D(\hat{\mathbf{x}})|$ less than 0.03 were discarded (as before, we assume image pixel values in the range [0, 1]).

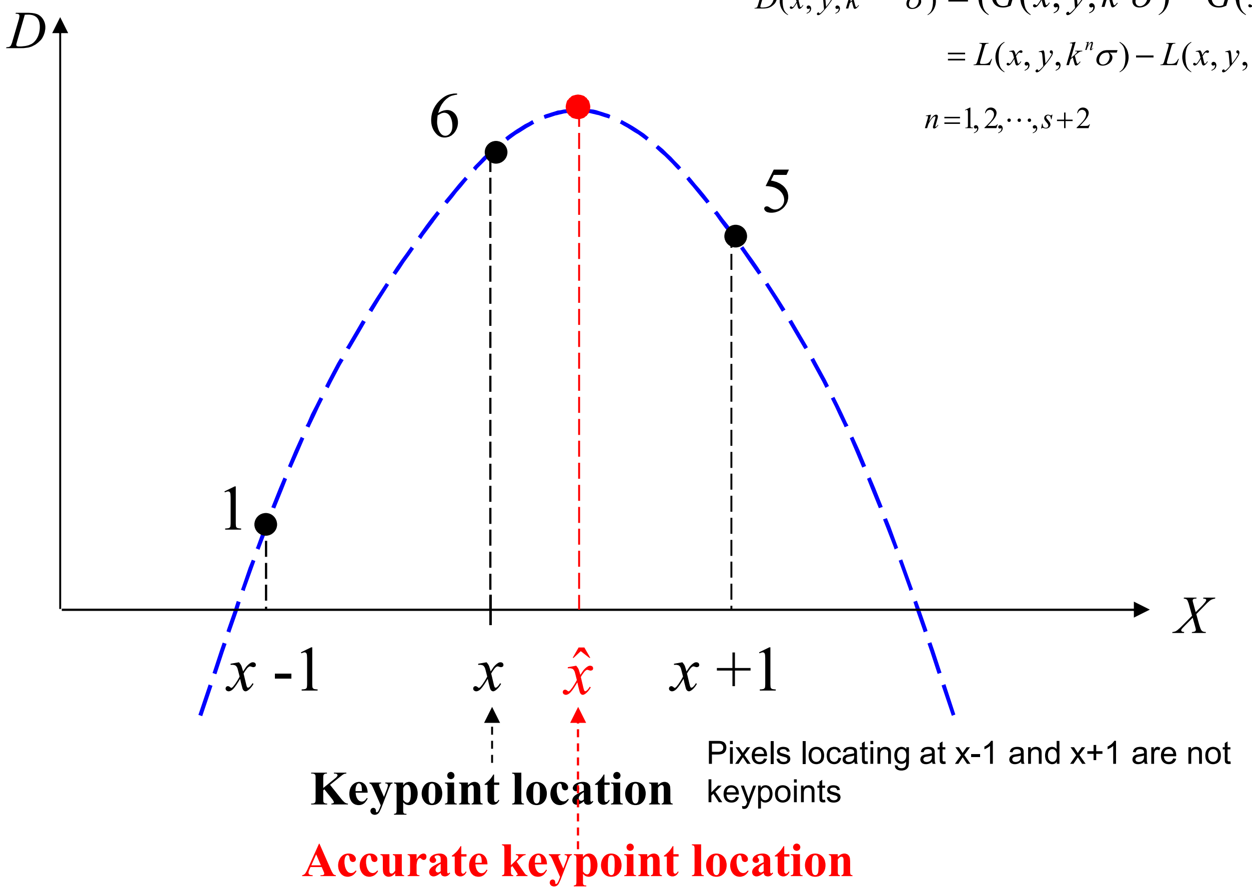
4.1 Accurate keypoint localization (2/7)

- Perform a detailed fit to the nearby data around the keypoint for accurate location by Taylor expansion (up to the quadratic terms).

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

$$D(x,y,k^{(n-1)}\sigma)=(G(x,y,k^{n}\sigma)-G(x,y,k^{(n-1)}\sigma))*I(x,y)$$
$$=L(x,y,k^{n}\sigma)-L(x,y,k^{(n-1)}\sigma)$$

$$n=1,2,\cdots,s+2$$



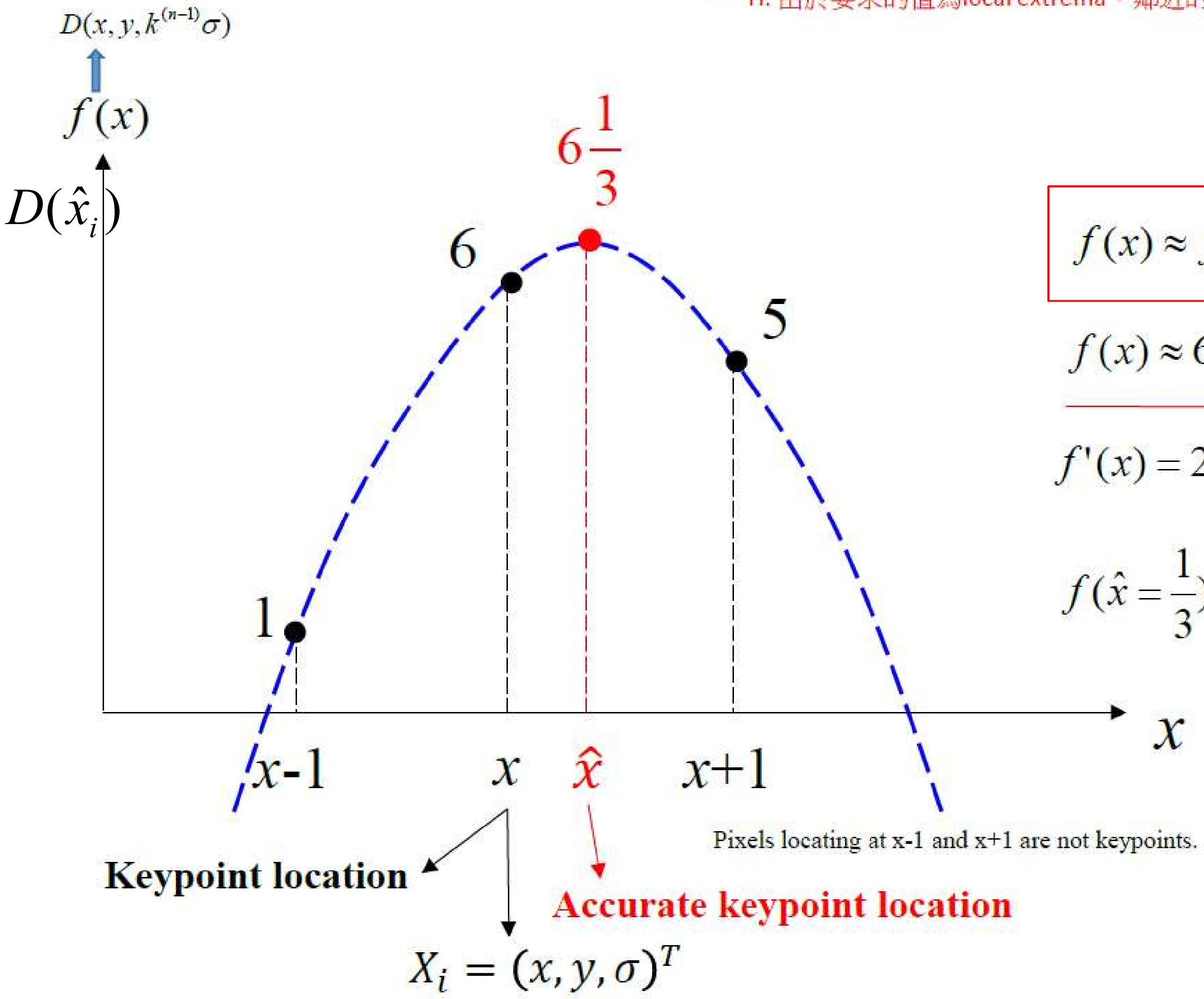
Nonlinear Interpolation!!

4.1 Accurate keypoint localization (3/7)

2.1) Accurate Feature Localization by Taylor expansion J: Ideal solution but slow

Perform a detailed fit to the nearby data around the keypoint for accurate location by Taylor expansion (up to the quadratic terms). Why quadratic, not linear interpolation??

H: 由於要求的值為local extrema，鄰近的點分布會是曲線，因此要用二階來求方程式。



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

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

$$f'(x) = 2 - 6x = 0, \text{ so } \boxed{x = \frac{1}{3} = \hat{x}}$$

(Next page)

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

4.1 Accurate keypoint localization (4/7)

求 $f(x)$

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

3 unknown parameters?
Need 3 1D points => 3 equations

$$if\ x = 0, \quad f(x) = f(0) + f'(0) * 0 + \frac{f''(0)}{2} * 0^2 = \boxed{6}$$

$$if\ x = -1, \quad f(x) = f(0) + f'(0) * (-1) + \frac{f''(0)}{2} * (-1)^2$$

$$= 6 - f'(0) + \frac{f''(0)}{2} = \boxed{1}$$

$$f'(0) - \frac{f''(0)}{2} = 5 \ \ldots\ldots\ldots\ldots\ldots\ldots (1)$$

$$if\ x = 1, \quad f(x) = f(0) + f'(0) * 1 + \frac{f''(0)}{2} * 1^2$$

$$= 6 + f'(0) + \frac{f''(0)}{2} = \boxed{5}$$

$$f'(0) + \frac{f''(0)}{2} = -1 \ \ldots\ldots\ldots\ldots\ldots (2)$$

(1)+(2) $\quad 2f'(0) = 4 \quad f'(0) = 2$

代回(2) $\quad \frac{f''(0)}{2} = -3 \quad f''(0) = -6$

$$f(x) = 6 + 2x - 3x^2$$

4.1 Accurate keypoint localization (5/7)

- Taylor series of several variables

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

- Taylor expansion in matrix form, **x** is a nx1 vector, *f* maps **x** to a scalar.

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

$$\text{gradient}\quad\begin{pmatrix}\frac{\partial f}{\partial x_1}\\ \frac{\partial f}{\partial x_1}\\ \vdots\\ \frac{\partial f}{\partial x_n}\end{pmatrix}\qquad\begin{pmatrix}\frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1\partial x_n}\\ \frac{\partial^2 f}{\partial x_2\partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2\partial x_n}\\ \vdots & \vdots & \ddots & \vdots\\ \frac{\partial^2 f}{\partial x_n\partial x_1} & \frac{\partial^2 f}{\partial x_n\partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2}\end{pmatrix}\quad\text{Hessian}$$

4.1 Accurate keypoint localization (6/7) ??

DoG

- Fit D(x,y,σ) quadratic function for sub-pixel extrema.

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

- x is a 3x1 vector $[x\ y\ \sigma]^T$

??

$$\frac{\partial D(\mathbf{x})}{\partial \mathbf{x}} = \frac{\partial D^T}{\partial \mathbf{x}} + \frac{1}{2}\left(\frac{\partial^2 D}{\partial \mathbf{x}^2} + \frac{\partial^2 D}{\partial \mathbf{x}^2}^T\right)\mathbf{x} = \frac{\partial D^T}{\partial \mathbf{x}} + \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} = 0$$

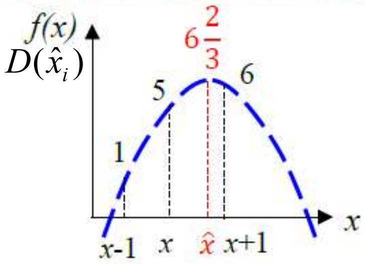
Sub-pixel extrema position

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

J: Similar to optical flow

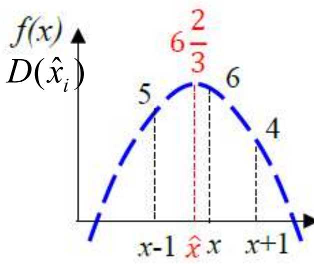
- If the offset $\hat{x}$ is larger than 0.5 in any dimension, then it means that the extrema lies closer to a different sample point. In this case, the sample point is changed and the interpolation performed instead about that point. The final offset $\hat{x}$ is added to the location of its sample point to get the interpolated estimate for the location of the extrema.

Example



$\hat{x} = \frac{2}{3} > 0.5$

Sample point change to x+1



$\hat{x} = -\frac{1}{3}$

J: I think both x_head value and D(x_head) at right side are not as left site. And the quadratic curve won't be the same, either

4.2 Remove low contrast keypoints (7/7)

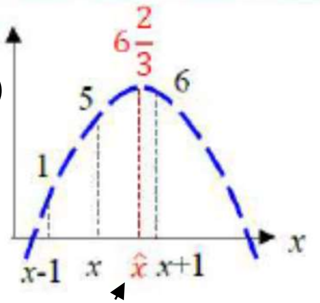
2.2) Removing Low Contrast Feature

- Throw out low contrast features which is $|D(\hat{x}_i)| < 0.03$

Original value additional value

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

$D(\hat{x}_i)$



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

derive:

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

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

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

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

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

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

J: If Alfa and Beta are eigenvalues, then probably the equations are trying to solve both eigenvalues

4.3 Eliminating edge responses (1/2) ??

2.3) Eliminating Edge Responses

- Compute 2x2 Hessian matrix at the location and scale of the feature:

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

$$\mathrm{Tr}(H) = D_{xx} + D_{yy} = \alpha + \beta$$ trace

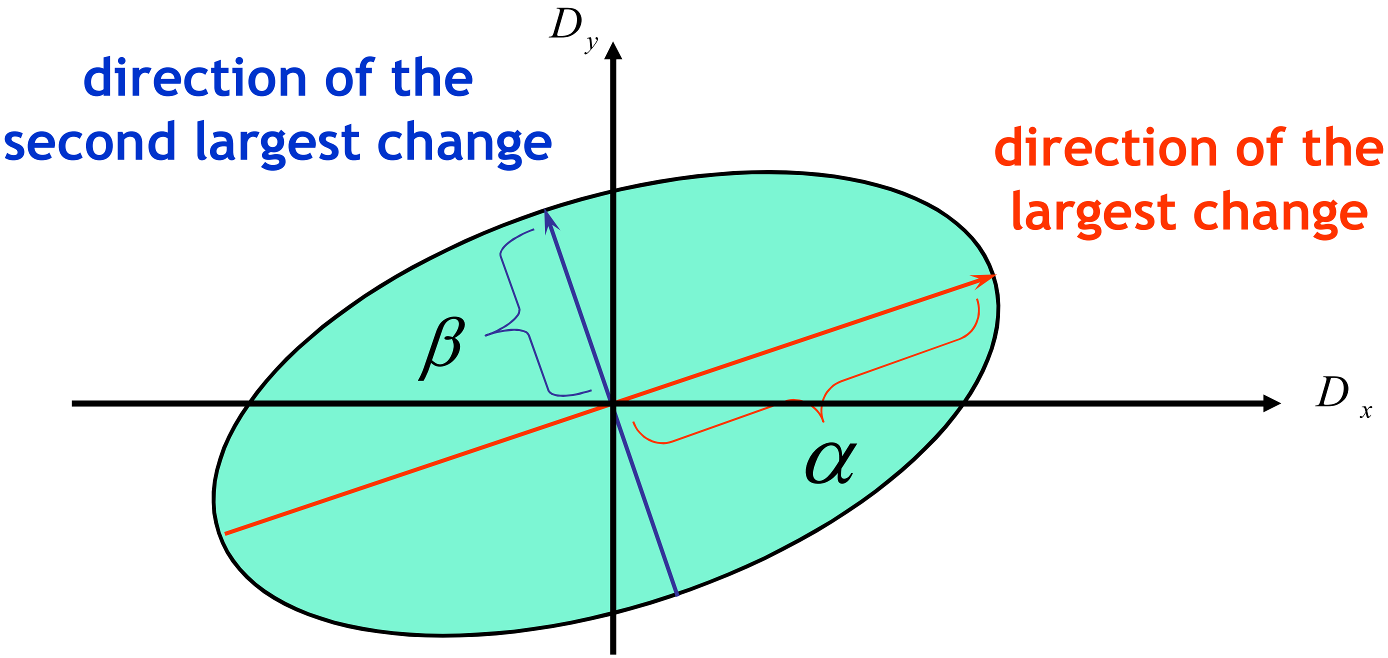
$$\mathrm{Det}(H) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$ determine

Hessian matrix at keypoint location

α : **The larger eigenvalue of H.**

β : **The smaller eigenvalue of H.**

$$\begin{bmatrix} D_{xx} - \alpha & D_{xy} \\ D_{xy} & D_{yy} - \beta \end{bmatrix} = 0$$

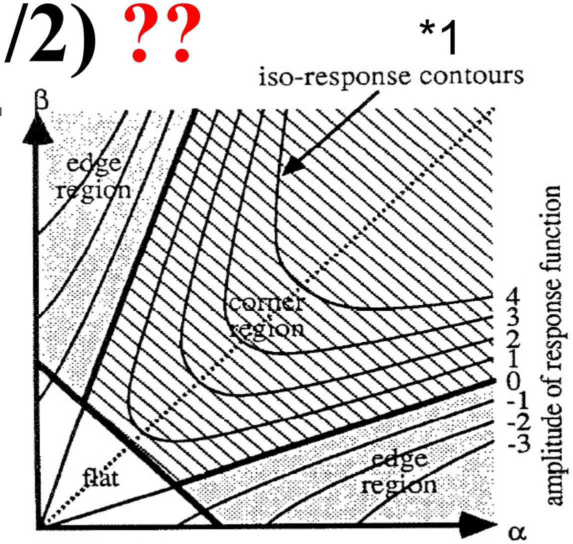


4.3 Eliminating edge responses (2/2) ??

J: Hope to keep the corner points?

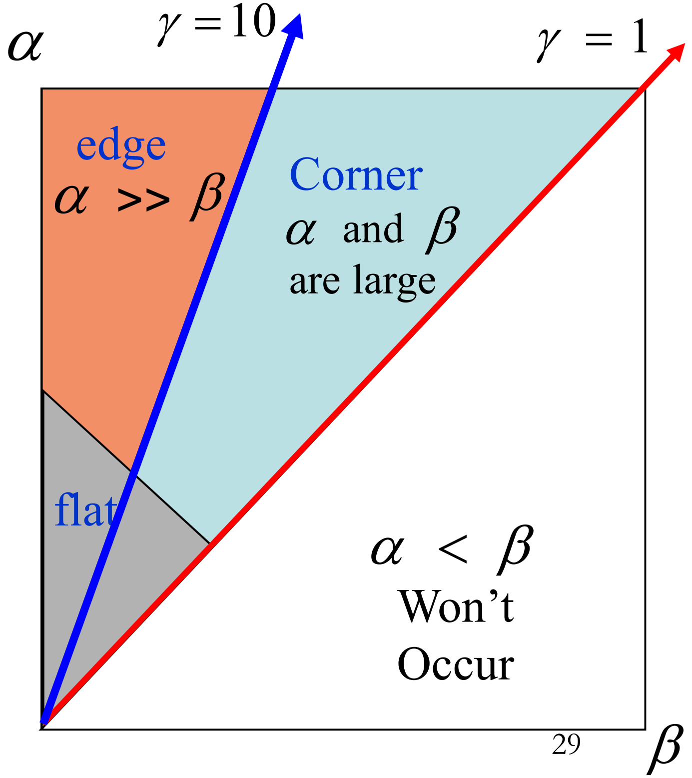
$$\mathrm{Tr}(\mathrm{H}) = D_{xx} + D_{yy} = \alpha + \beta$$

$$\mathrm{Det}(\mathrm{H}) = D_{xx}D_{yy} - (D_{xy})^2 = \alpha\beta$$



Let $\alpha = \gamma\beta$

$$\frac{\mathrm{Tr}(\mathrm{H})^2}{\mathrm{Det}(\mathrm{H})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(\gamma\beta+\beta)^2}{\gamma\beta^2} = \frac{(\gamma+1)^2}{\gamma}$$

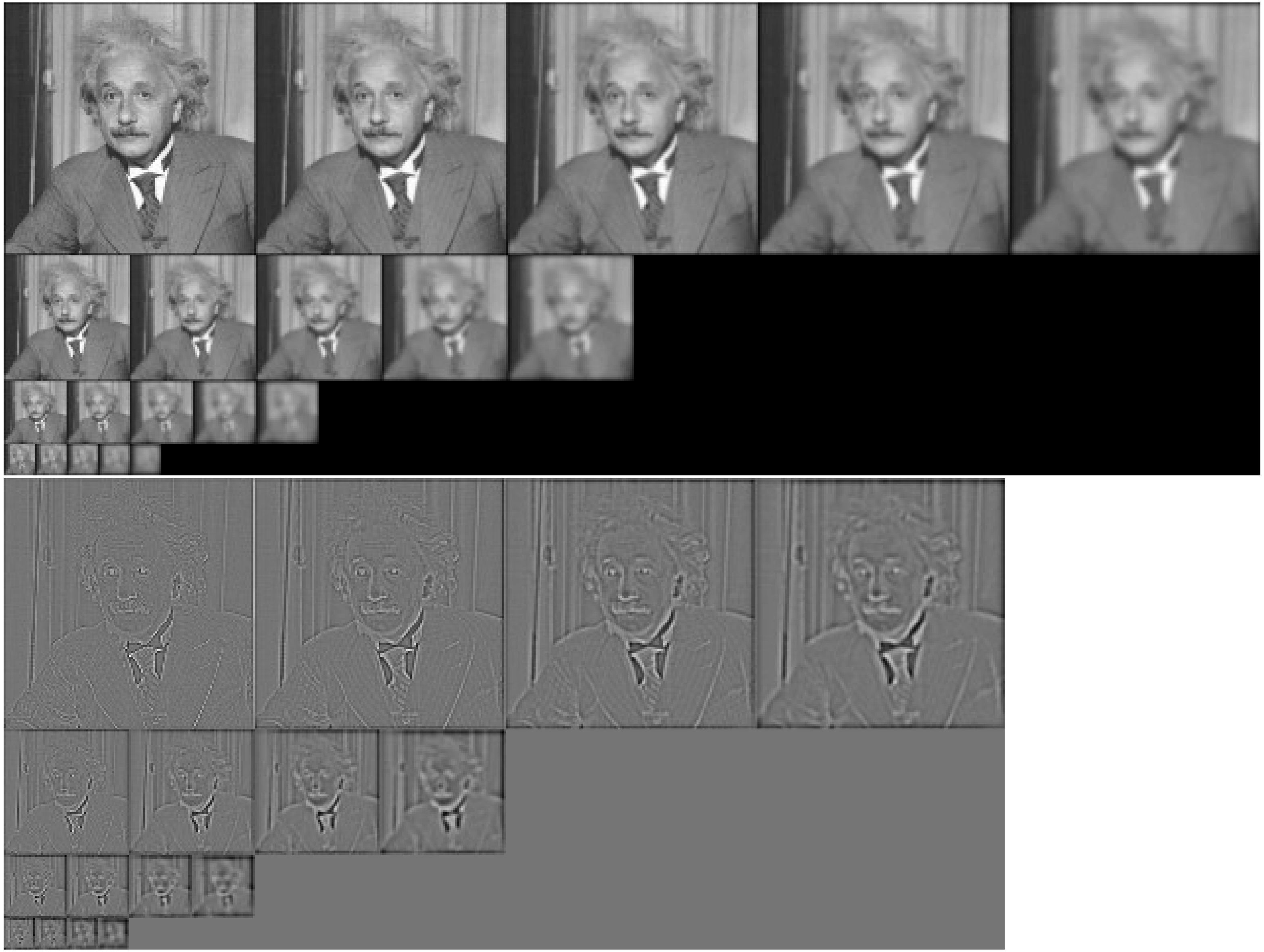


Keep the (corner) points if $\dfrac{\mathrm{Tr}(\mathrm{H})^2}{\mathrm{Det}(\mathrm{H})} < \dfrac{(\gamma+1)^2}{\gamma}$

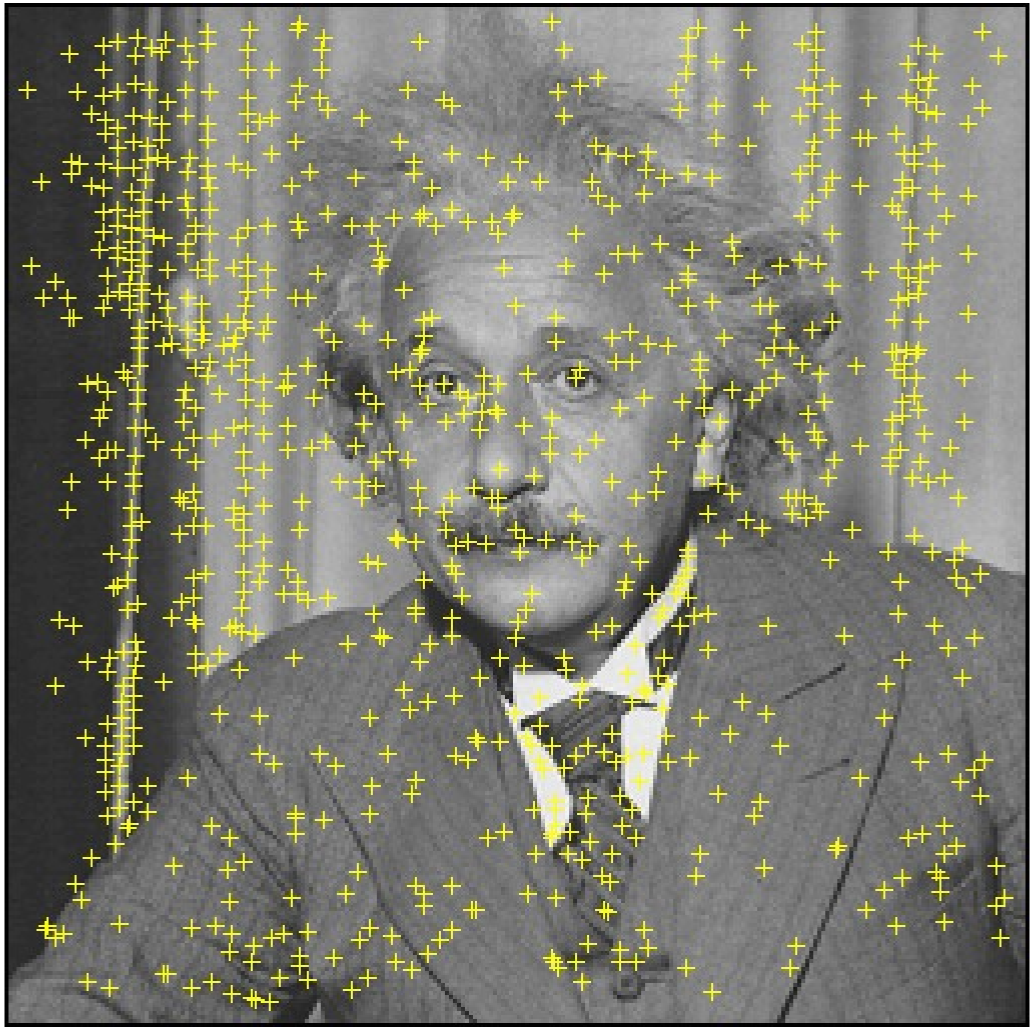
$\gamma = 10$

*1 C. Harris and M. Stephens, "A Combined Corner and Edge Detector," 4th Alvey Vision Conference, pp147-151, 1988.

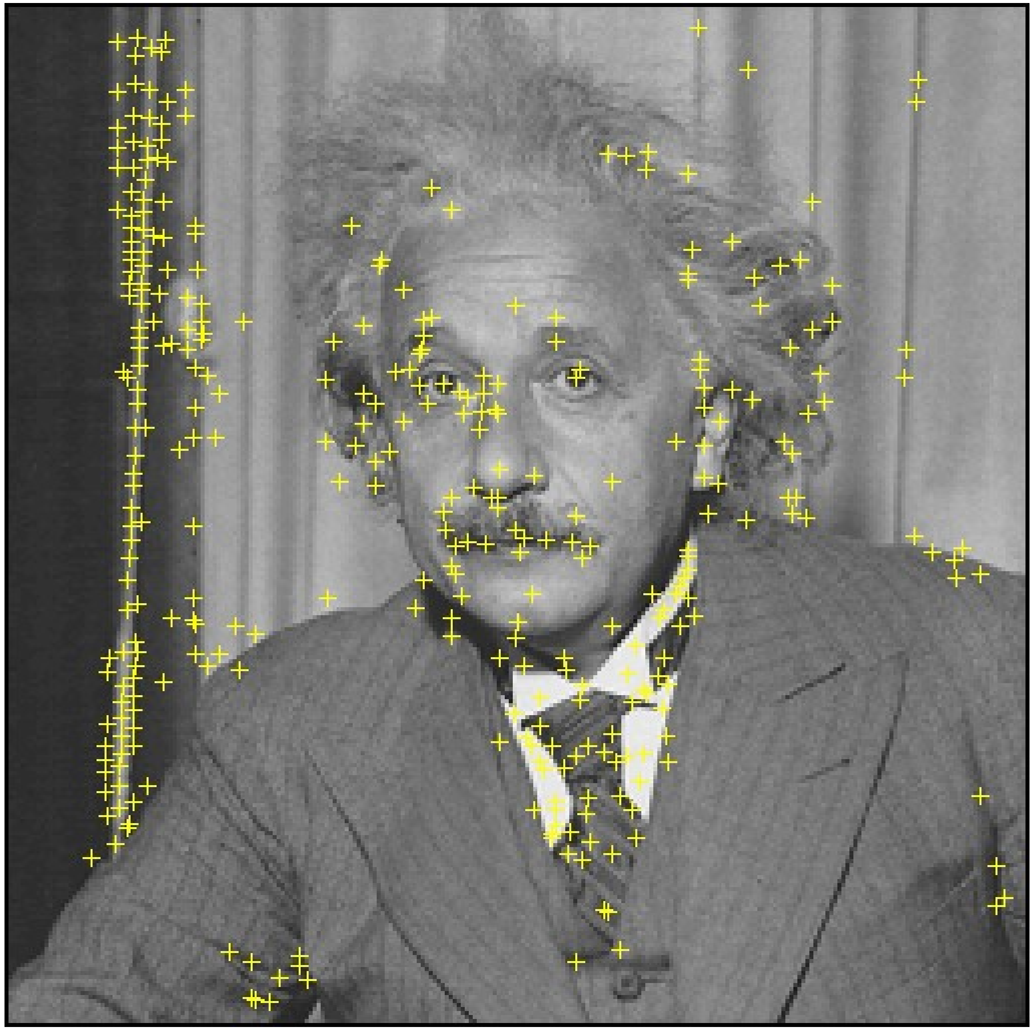
SIFT flow



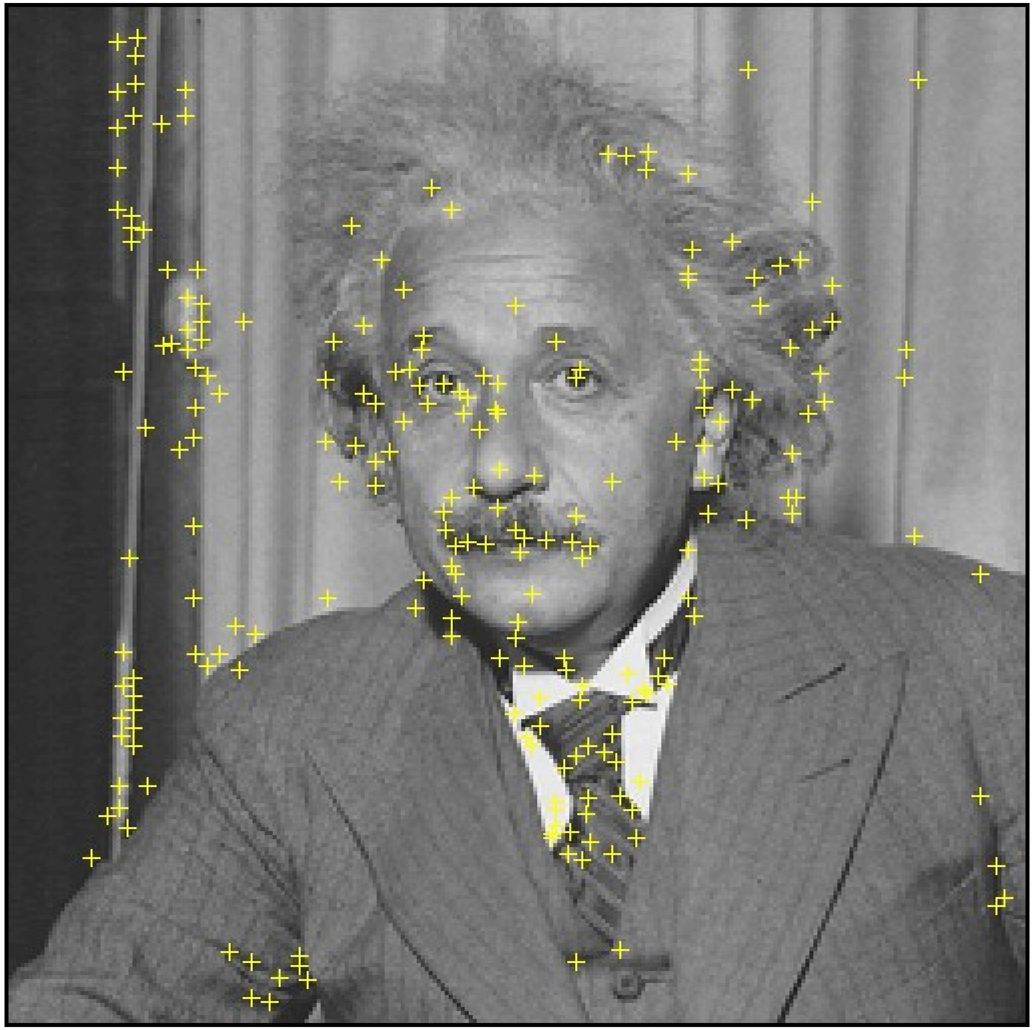
Extrema in *D*



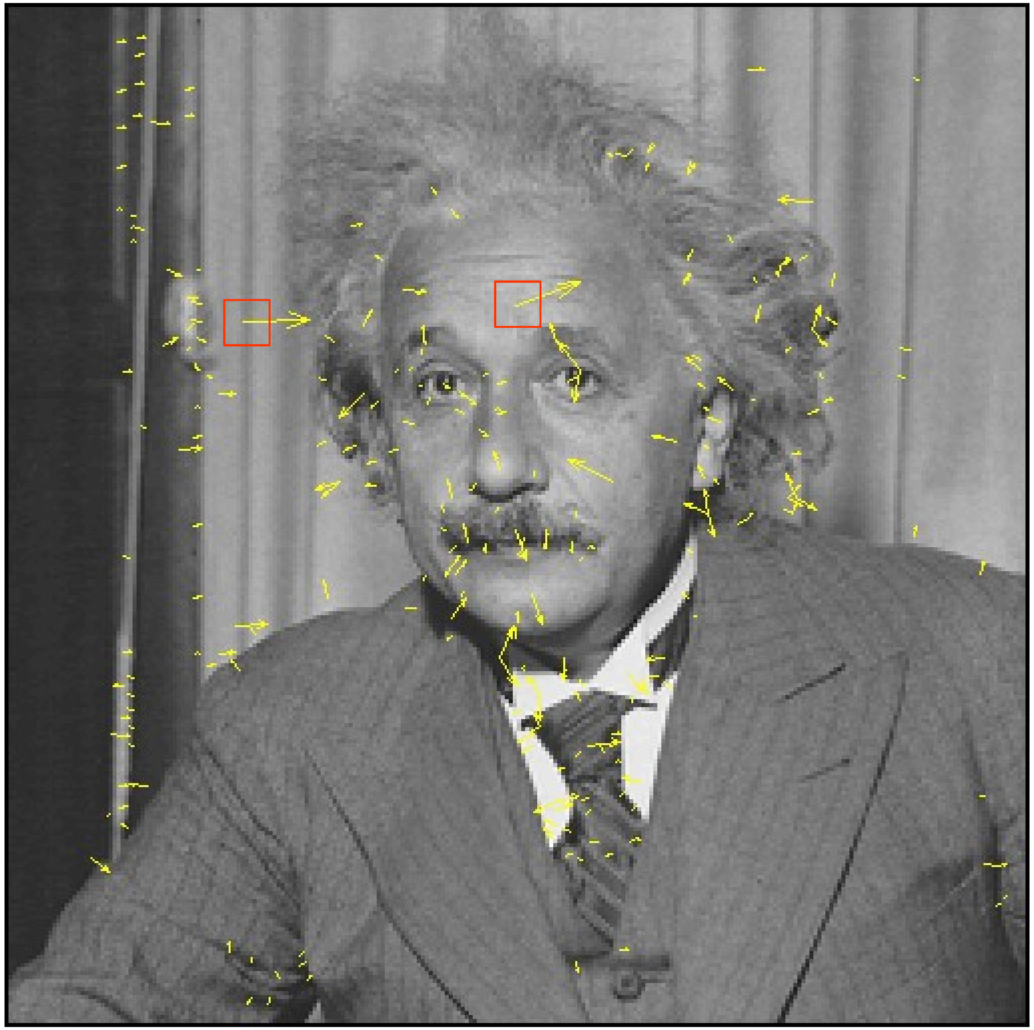
Remove low contrast



Remove edges response



SIFT descriptor



Content

System Flowchart

$L(\sigma) = G(\sigma) * I$ (Scale space)

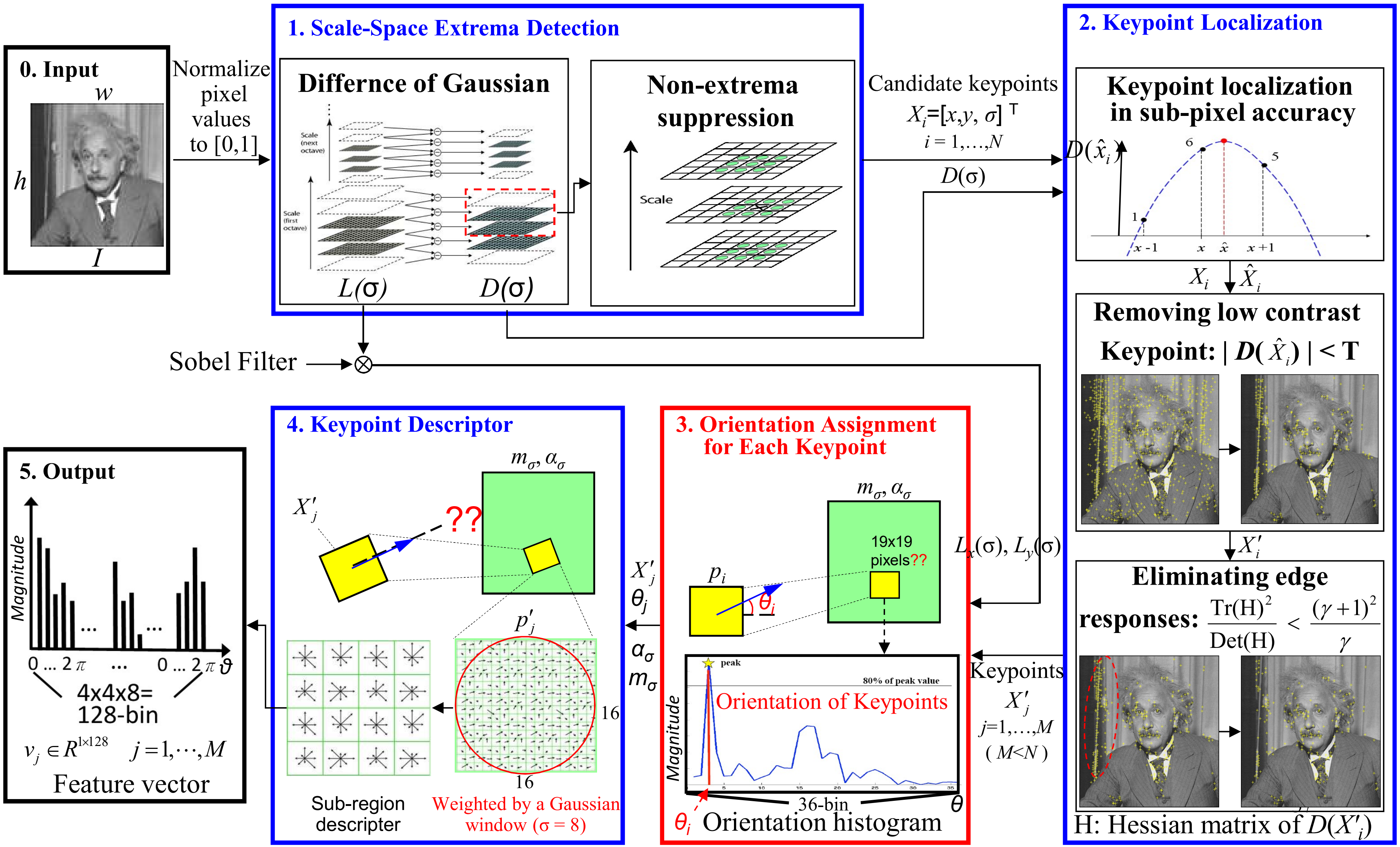
$D(\sigma) = (G(k^{(n-1)}\sigma) - G(k^{n}\sigma)) * I$ (DoG images)

α_σ (Gradient orientation of $L(\sigma)$)

m_σ (Gradient magnitude of $L(\sigma)$)

X'_i (Accurate keypoint localization)

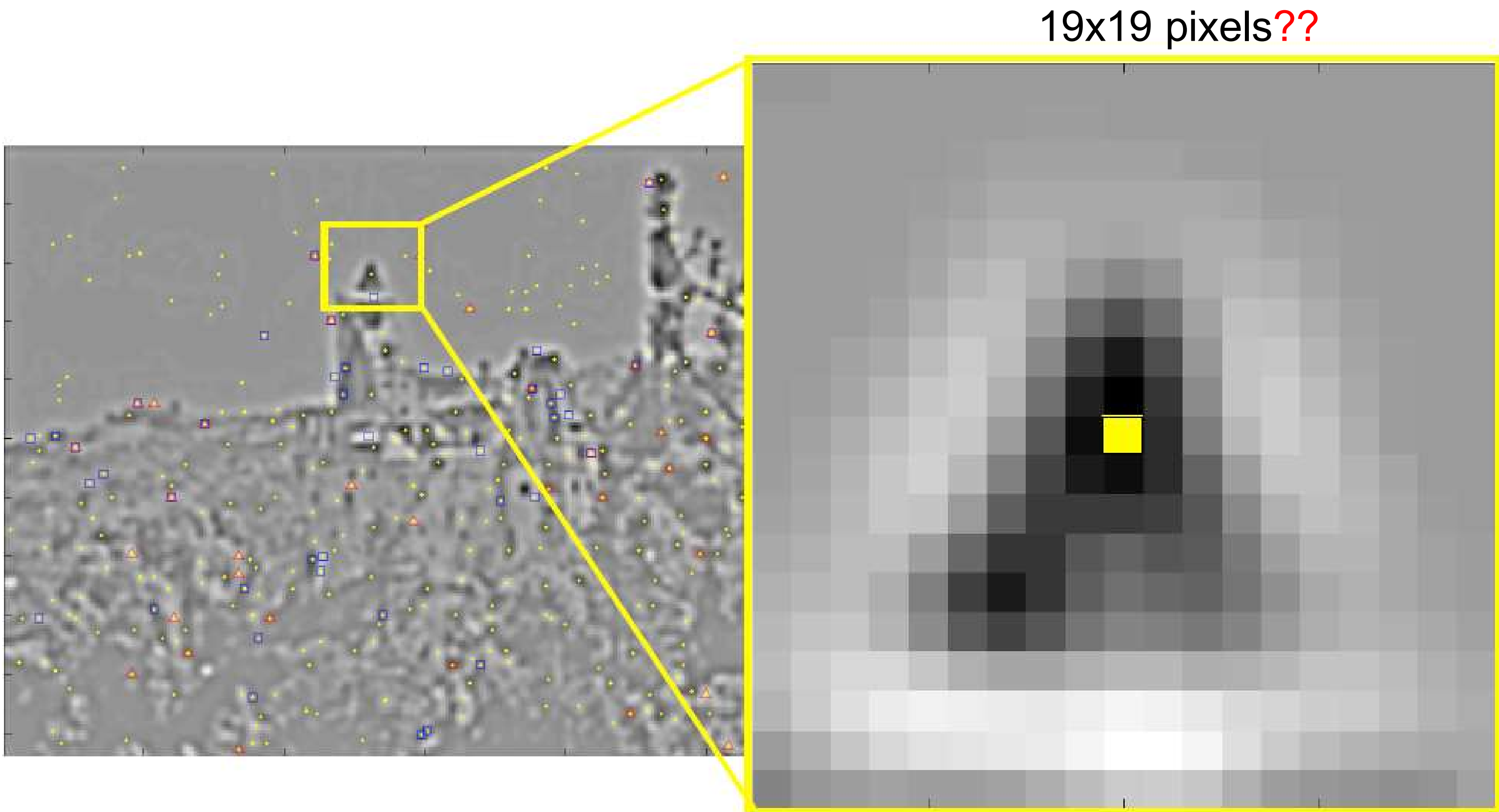
θj (Orientation of i-th keypoint)



5. Orientation Assignment (1/5)

- An orientation histogram is formed from the gradient orientations of sample points within a region around the keypoint X_j' (patch p_j) at $L(\sigma)$.
- By assigning a consistent orientation, the keypoint descriptor can be represented relative to this orientation and therefore achieve invariance to image rotation.
- Solve the rotation invariance problem.

5. Orientation Assignment (2/5)



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

Previously, L is I after Gaussian smooth:
$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y)$

Window size??,
tan_Theta = 180

5. Orientation Assignment (3/5) ??

P_j : j-th patch
m(x, y): gradient magnitude
α(x, y) : orientation
P_j^m : Gradient magnitude of P_j
P_j^α : Gradient orientation of P_j
$L(\sigma)$: Scale space of image.

- Each sample added to the histogram is weighted by its gradient magnitude and by a Gaussian-weighted circular window.

$$L_x(x, y, \sigma) = L(x+1, y, \sigma) - L(x-1, y, \sigma) \Rightarrow m_\sigma(x, y) = \sqrt{L_x(x, y, \sigma)^2 + L_y(x, y, \sigma)^2}$$

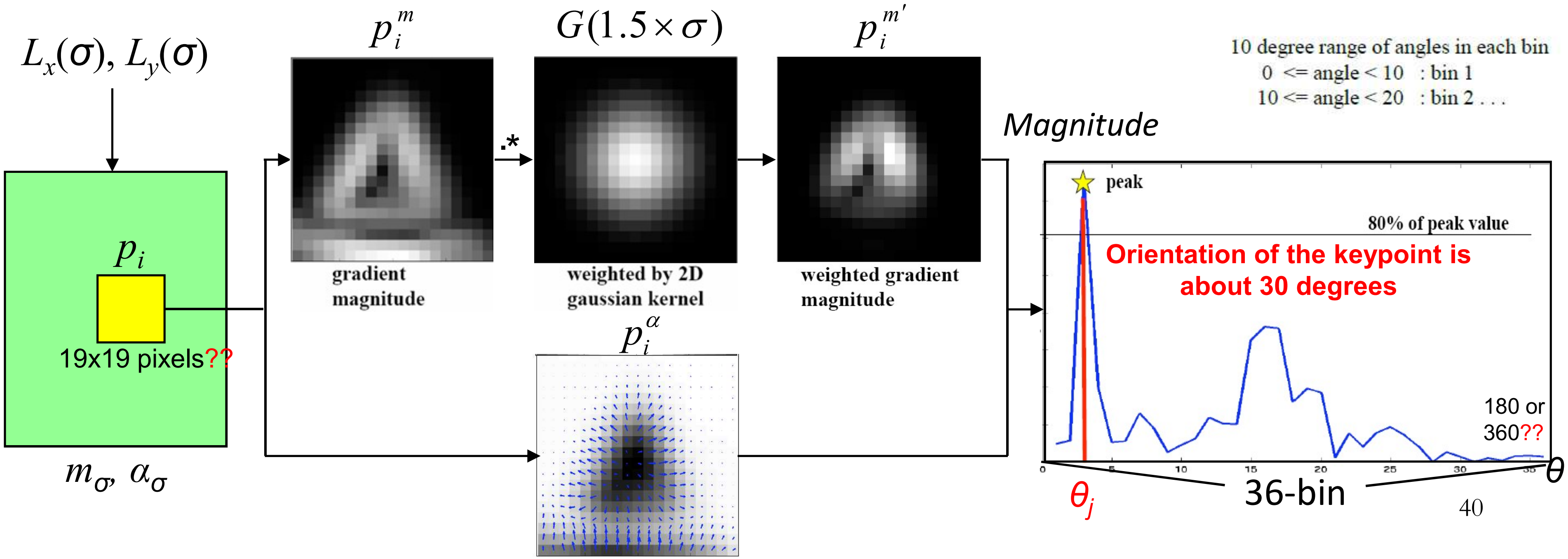
$$L_y(x, y, \sigma) = L(x, y+1, \sigma) - L(x, y-1, \sigma)$$

$$\alpha_\sigma(x, y) = \tan^{-1}\left(\frac{L_y(x, y, \sigma)}{L_x(x, y, \sigma)}\right)$$

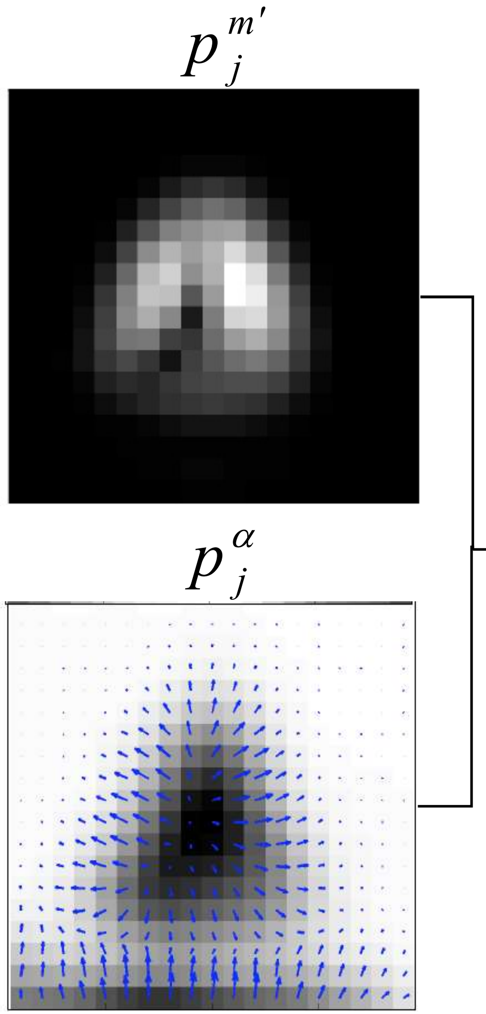
$$L_x(\sigma) = \begin{array}{ccc} \boxed{-1} & \boxed{0} & \boxed{+1} \\ x-1 & x & x+1 \end{array} * L \qquad L_y(\sigma) = \begin{array}{cc} \boxed{-1} & y-1 \\ \boxed{0} & y \\ \boxed{+1} & y+1 \end{array} * L$$

- Consistent orientation for each feature is θ_j.

$$p_j^{m'}(x, y) = p_j^m(x, y) \times G(x, y, 1.5 \times \sigma)$$

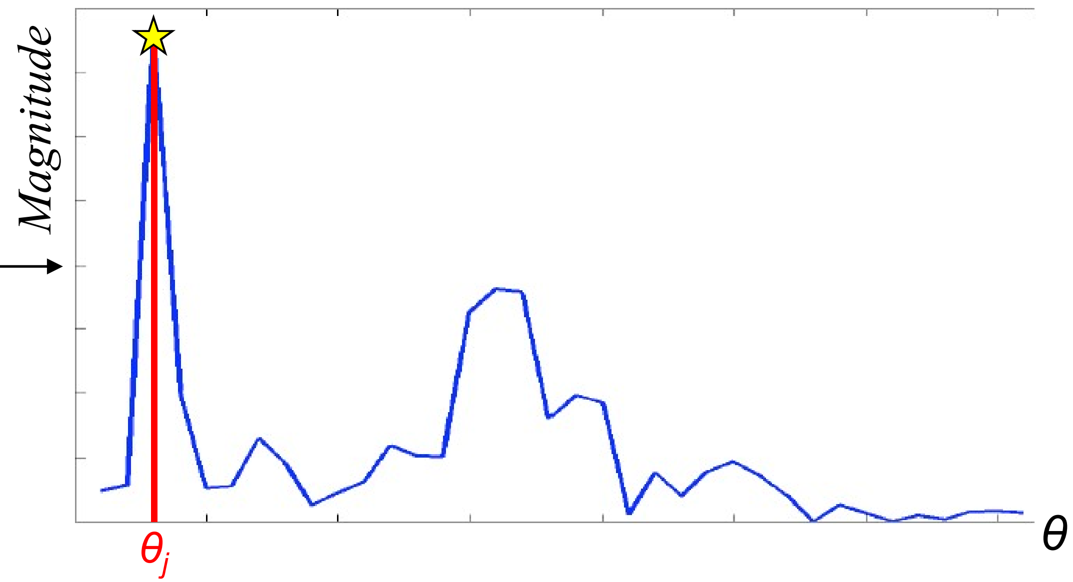


5. Orientation Assignment (4/5)



19x19 pixels??

Weighted orientation histogram.
Each bin contains sum of weighted gradient Magnitude corresponding to angles that fall within that bin.

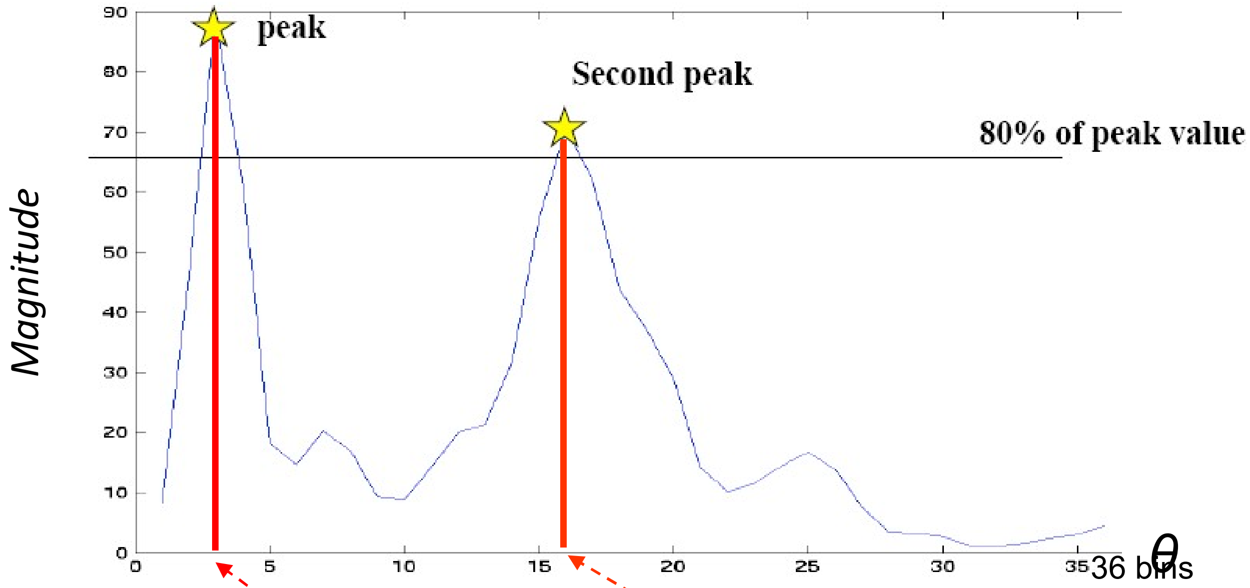


36 bins
10 degree range of angles in each bin

0 <= angle < 10 : bin 1
10 <= angle < 20 : bin 2
20 <= angle < 30 : bin 3 …

5. Orientation Assignment (5/5)

There may be multiple orientations.



In this case, generate duplicate keypoints, one with orientation at $\boxed{30}$ degrees, one at $\boxed{155}$ degrees.

- Any other local peak that is within 80% of the highest peak is used to also create a keypoint with that orientation.

Content

System Flowchart

$L(\sigma) = G(\sigma) * I$ (Scale space)

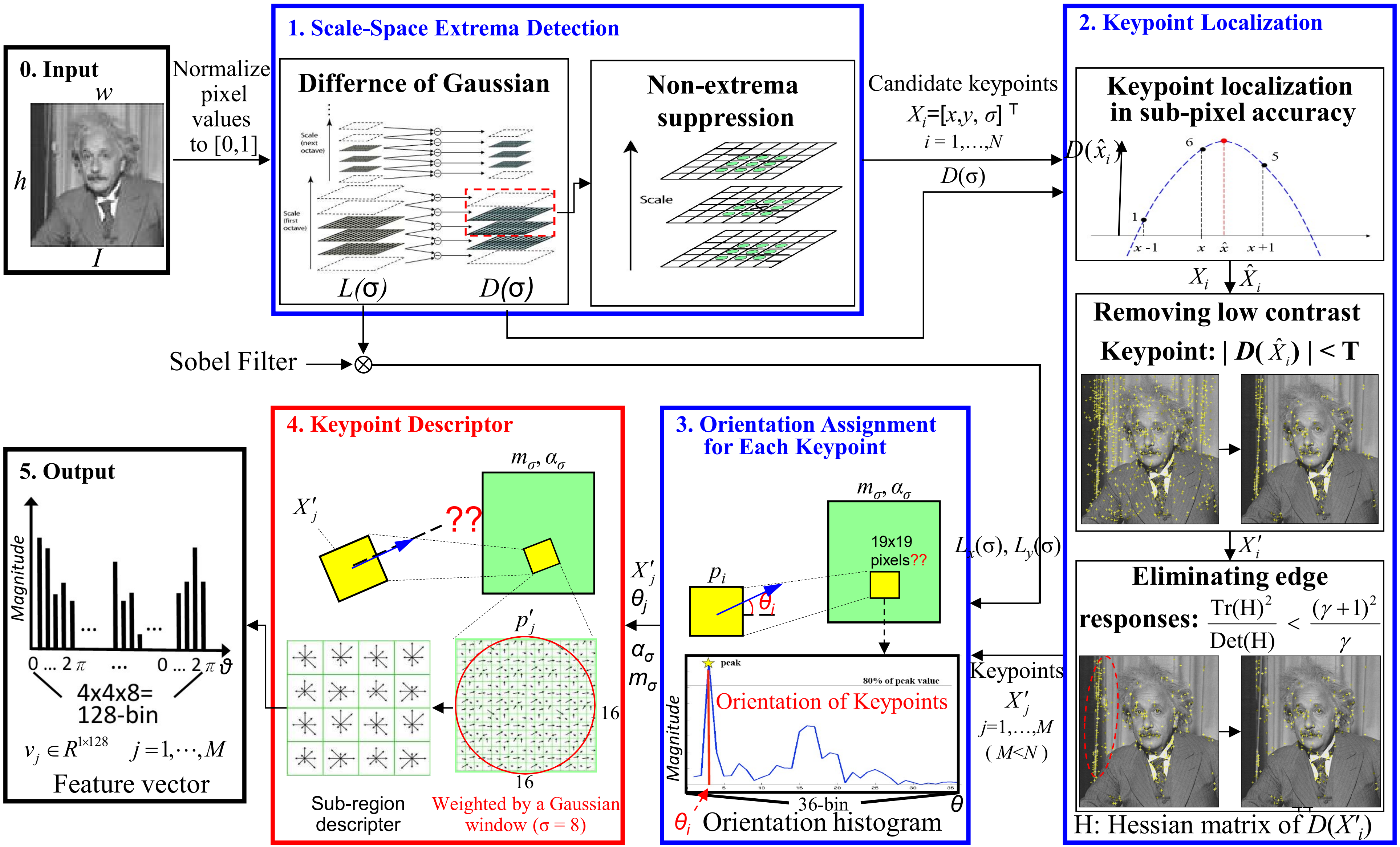
$D(\sigma) = (G(k^{(n-1)}\sigma) - G(k^{n}\sigma)) * I$ (DoG images)

α_σ (Gradient orientation of $L(\sigma)$)

m_σ (Gradient magnitude of $L(\sigma)$)

X'_i (Accurate keypoint localization)

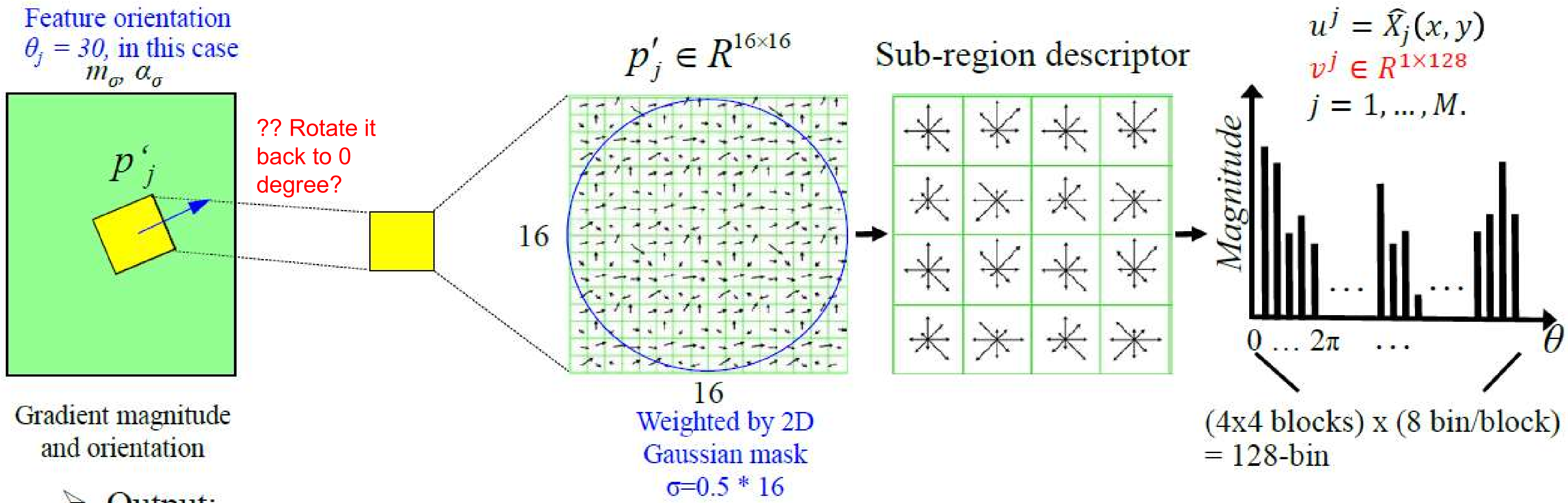
θj (Orientation of i-th keypoint)



6. Keypoint Descriptor ??

4) Compute feature descriptor.

- Image gradients are sampled over 16x16 array of locations in scale space relative to the keypoint orientation.
- The feature vector, v_j *(128 dimensions)*, is normalized to unit length. (Reduce change contrast) $\boxed{v_j \rightarrow \frac{v_j}{\|v_j\|}}$



- Output:
  - has
    Each image have M features.
    - ✓ has
      Each feature have 128 dimension feature vector.
    - ✓ The center position of each feature.

$u_n^j = [x, y]$: The center position of feature j in image n.

$v_n^j \in R^{1\times128}$: Feature vector j in image n (Feature Descriptor).

k = 1, …,M.

Content

7. Parameter Selection (1/2)

- We use a collection of 32 real images drawn from a diverse range, including (image domain) outdoor scenes, human faces, aerial photographs, and industrial images.
- The image domain was found to have almost no influence on any of the results.
- Each image was then subject to a range of transformations, including rotation (random), scaling (0.2 ~ 0.9), change in brightness and contrast, and addition of image noise (1%).

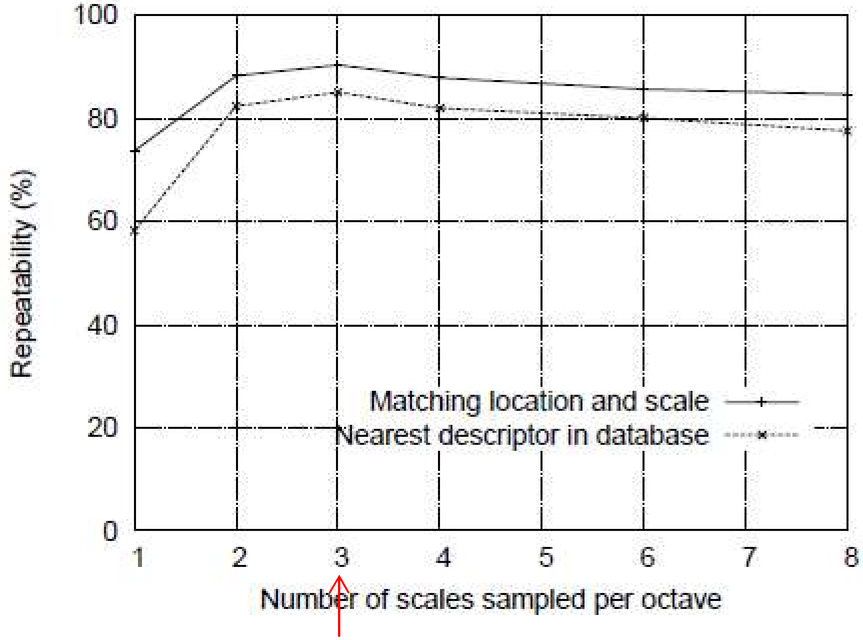
7. Parameter Selection (2/2)

- We define a matching scale as being within a factor of $\sqrt{2}$ of the correct scale, and a matching location as being within σ pixels, where σ is the scale of the keypoint.

- Distinguishable:
  - For a feature x, we found the closest feature x_1 and the second closest feature x_2.
  - If the distance ratio of $d(x, x_1)$ and $d(x, x_2)$ is smaller than 0.8, then it is accepted as a match (x and x_1 are the corresponding point pair).

Merge to one??

7. Parameter Selection: s



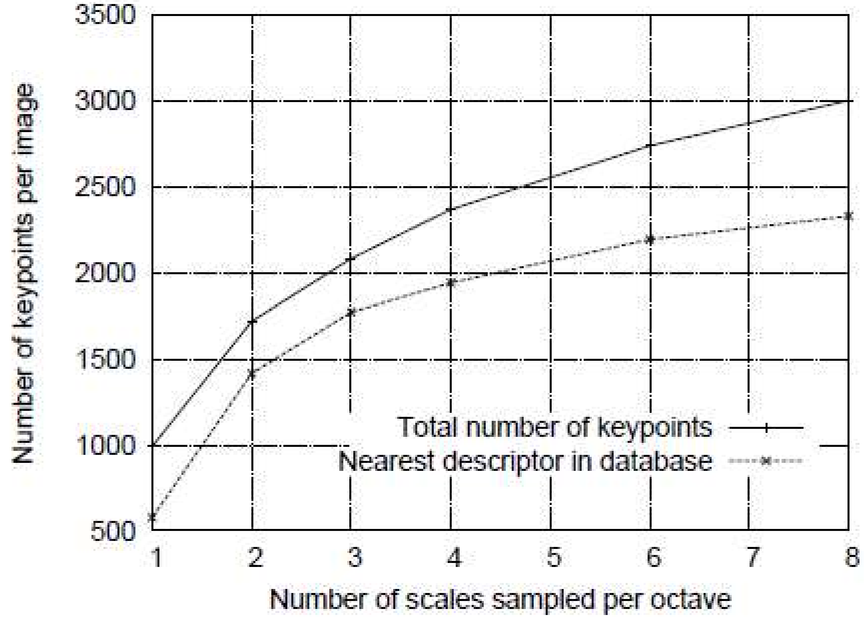


Figure 3:
The top line of the first graph shows the percent of keypoints that are repeatably detected at the same location and scale in a transformed image as a function of the number of scales sampled per octave. The lower line shows the percent of keypoints that have their descriptors correctly matched to a large database. The second graph shows the total number of keypoints detected in a typical image as a function of the number of scale samples.

7. Parameter Selection: Pre-smoothing

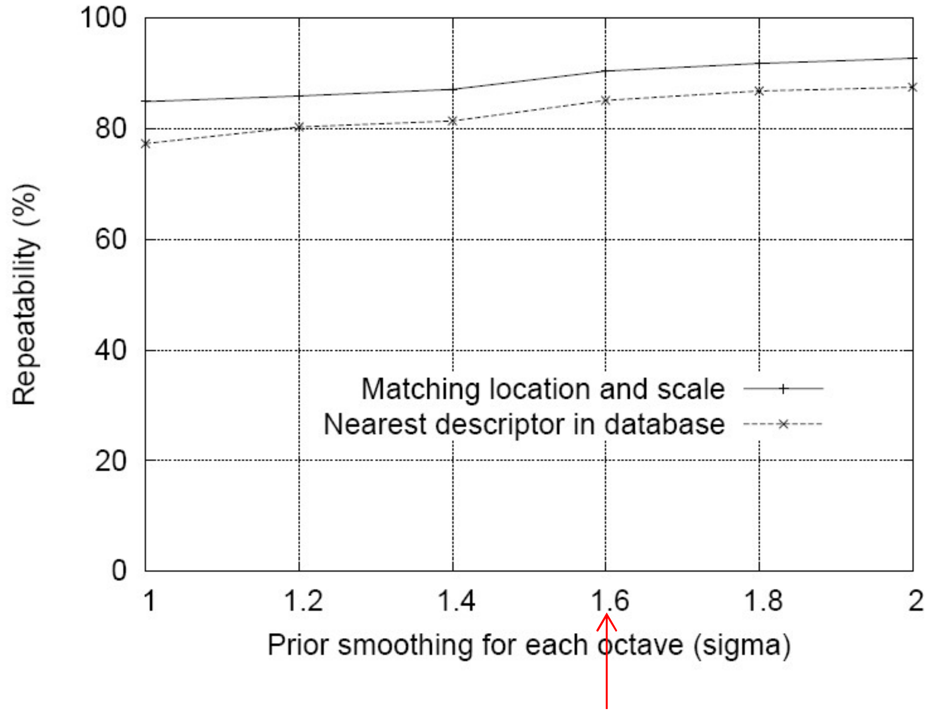


Figure 4: The top line in the graph shows the percent of keypoint locations that are repeatably detected in a transformed image as a function of the prior image smoothing for the first level of each octave. The lower line shows the percent of descriptors correctly matched against a large database.

There is a cost to using a large σ in terms of efficiency.

We have chosen to use $\sigma = 1.6$, which provides close to optimal repeatability

7. Parameter Selection: Why 4x4x8?

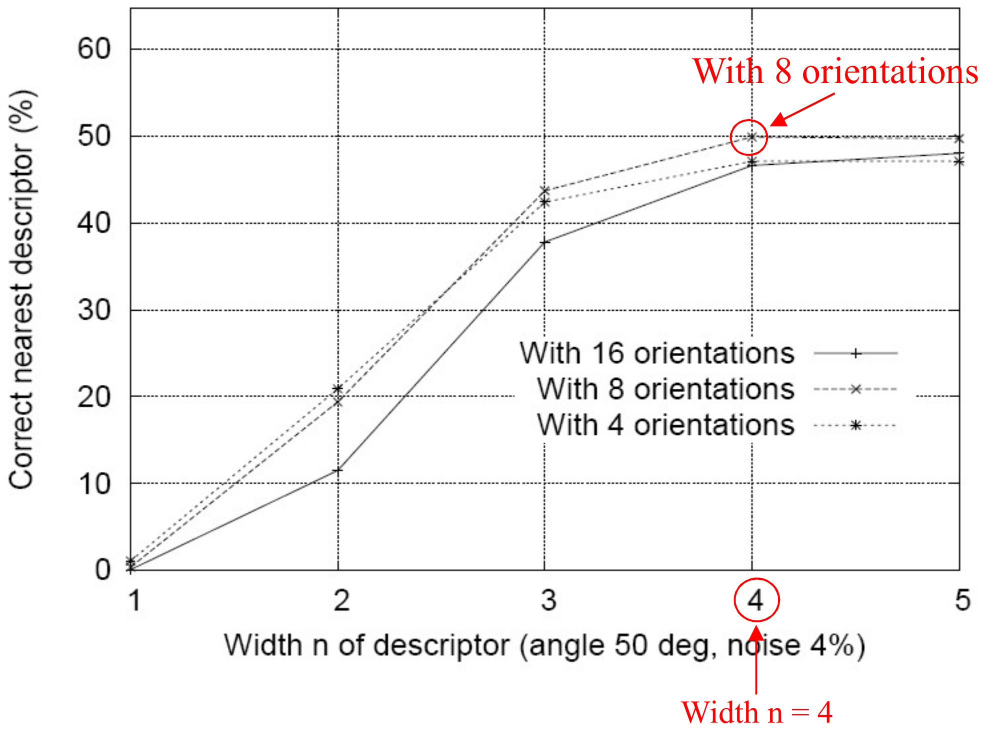


Figure 8: This graph shows the percent of keypoints giving the correct match to a database of 40,000 keypoints as a function of width of the n x n keypoint descriptor and the number of orientations in each histogram. The graph is computed for images with affine viewpoint change of 50 degrees and addition of 4% noise.

The graph shows that a single orientation histogram (n = 1) is very poor at discriminating, but the results continue to improve up to a 4x4 array of histograms with 8 orientations.

7. Sensitivity to affine change

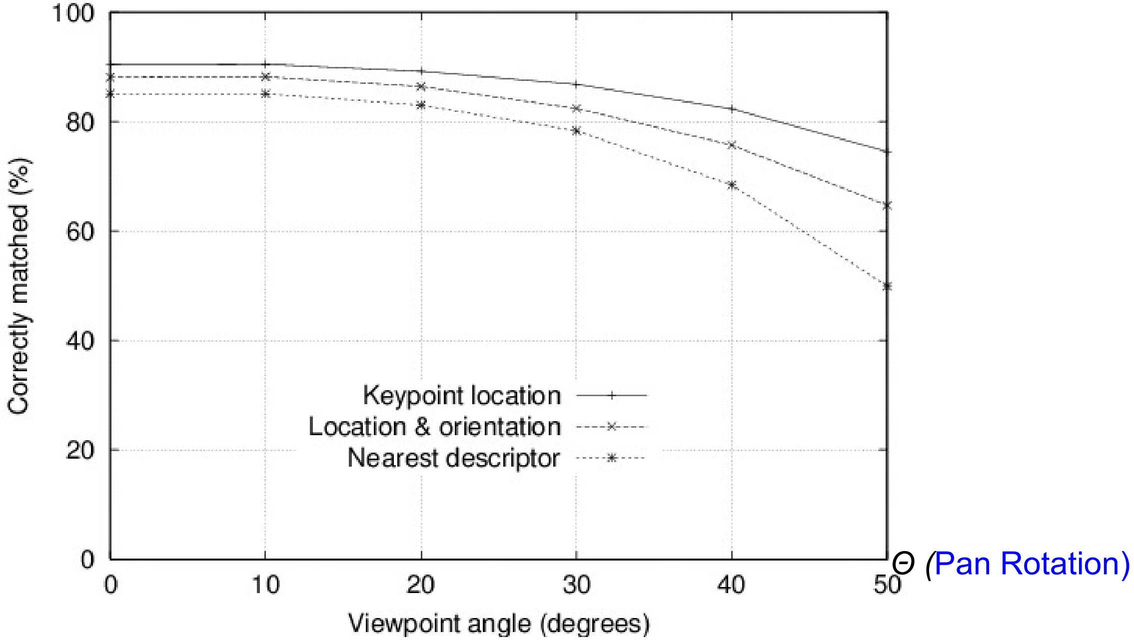
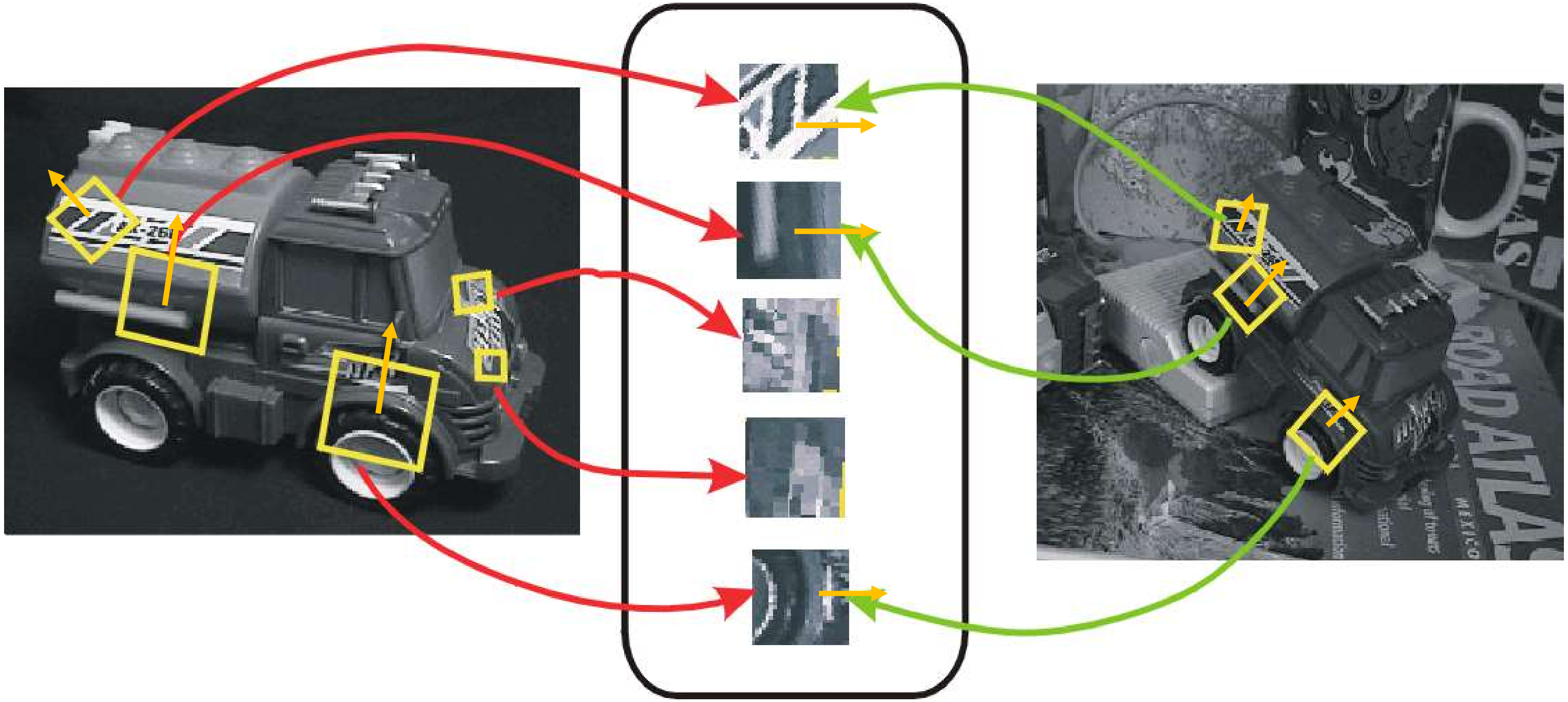
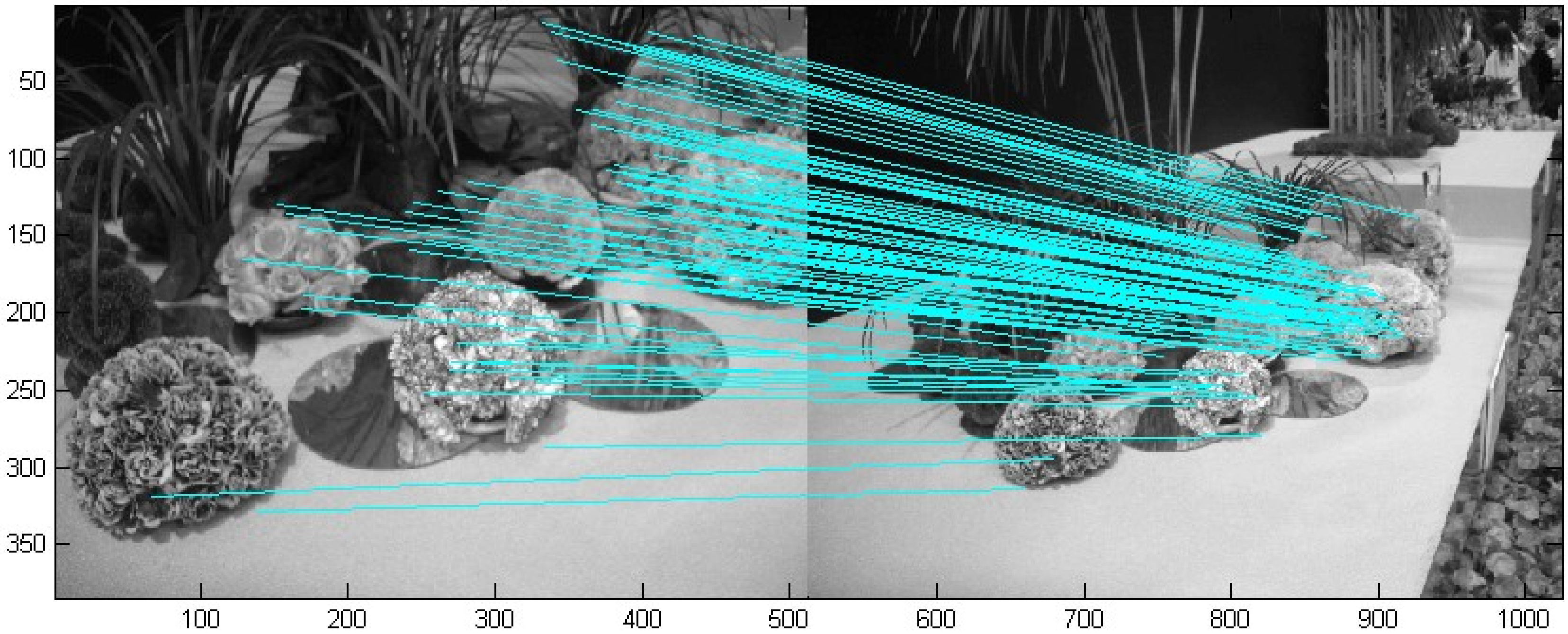


Figure 9: This graph shows the stability of detection for keypoint location, orientation, and final matching to a database as a function of affine distortion. The degree of affine distortion is expressed in terms of the equivalent viewpoint rotation in depth for a planar surface.

8.1 Example

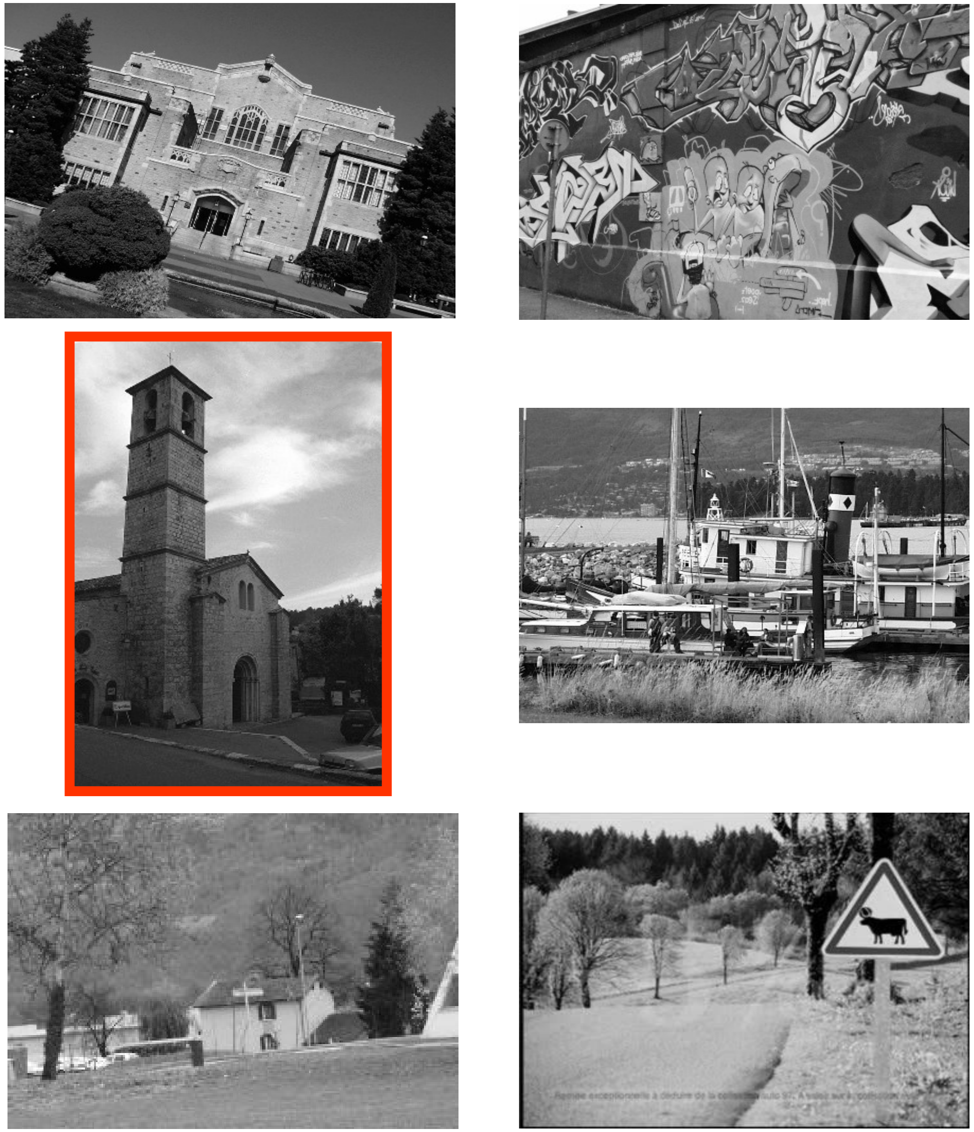


8.1 Example


50
100
150
200
250
300
350
100
200
300
400
500
600
700
800
900
1000

8.2 Image retrieval

change in viewing angle



> 5000 images

8.2 Image retrieval

22 correct matches

8.2 Image retrieval

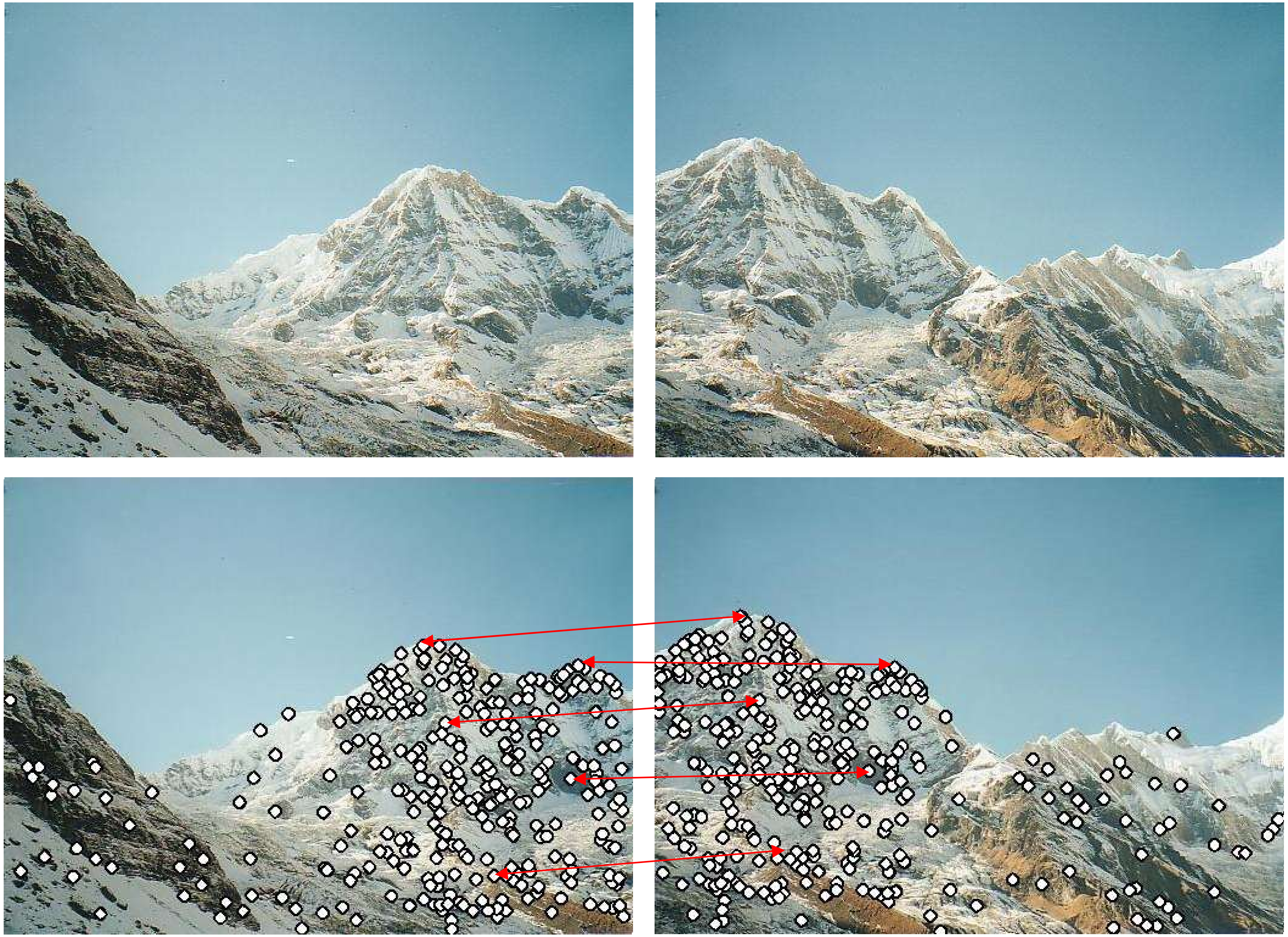
change in viewing angle

+ scale change

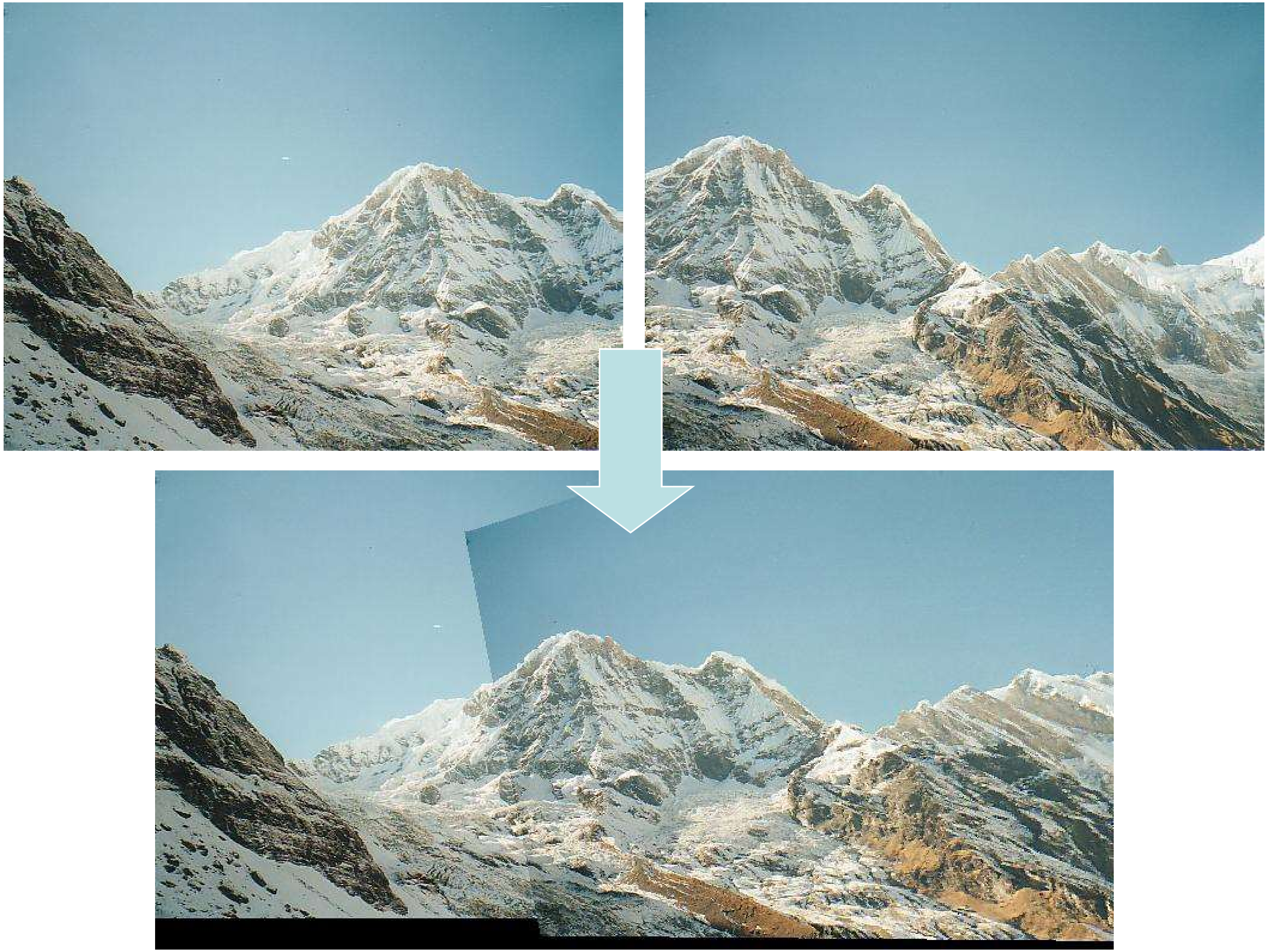
. . .

> 5000 images

8.3 Automatic image stitching

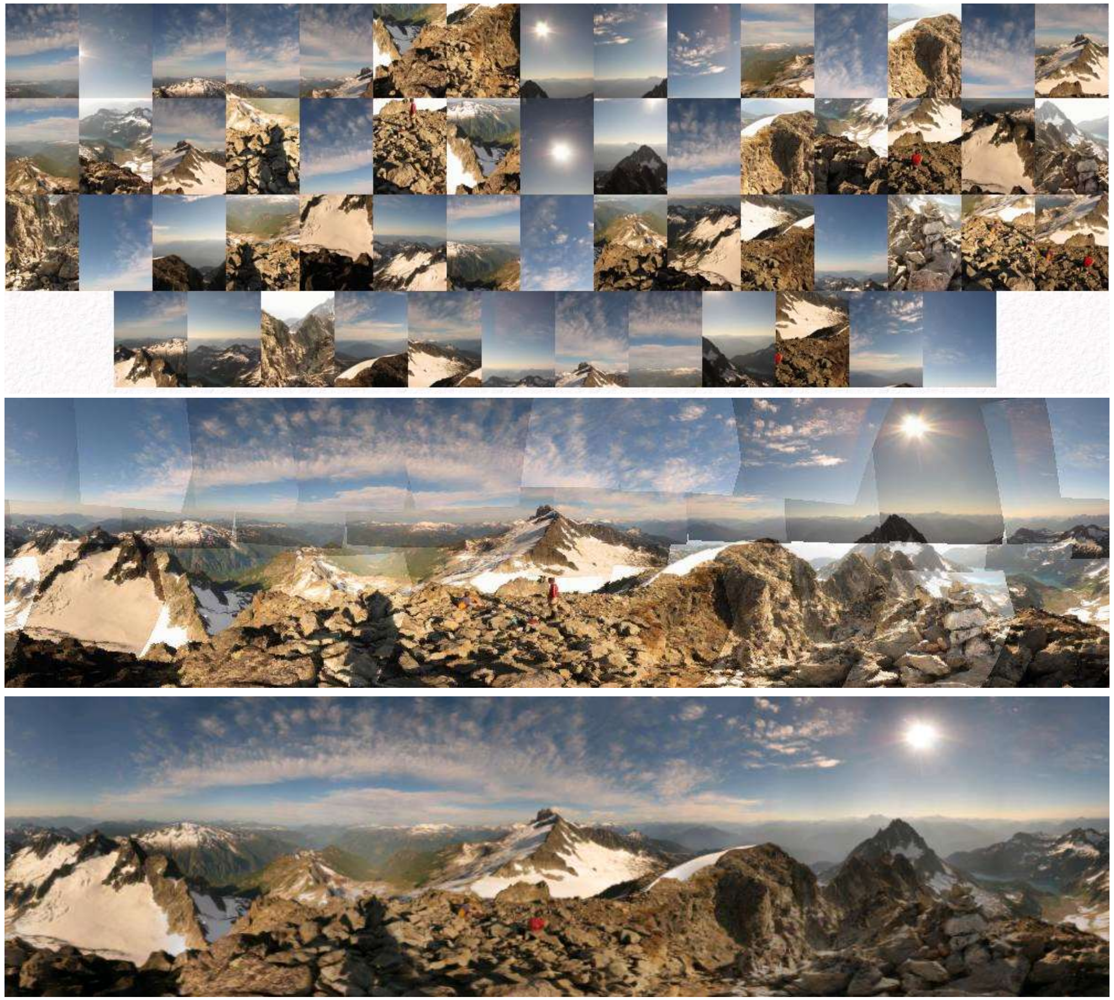


8.3 Automatic image stitching



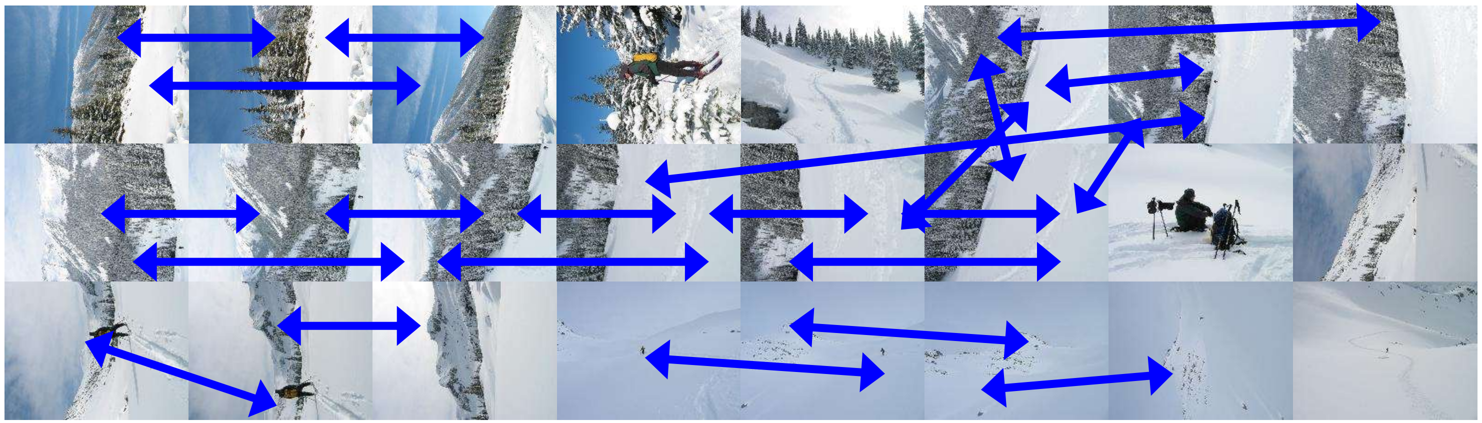
8.3 Image Stitch

- Automatic Mosaicing

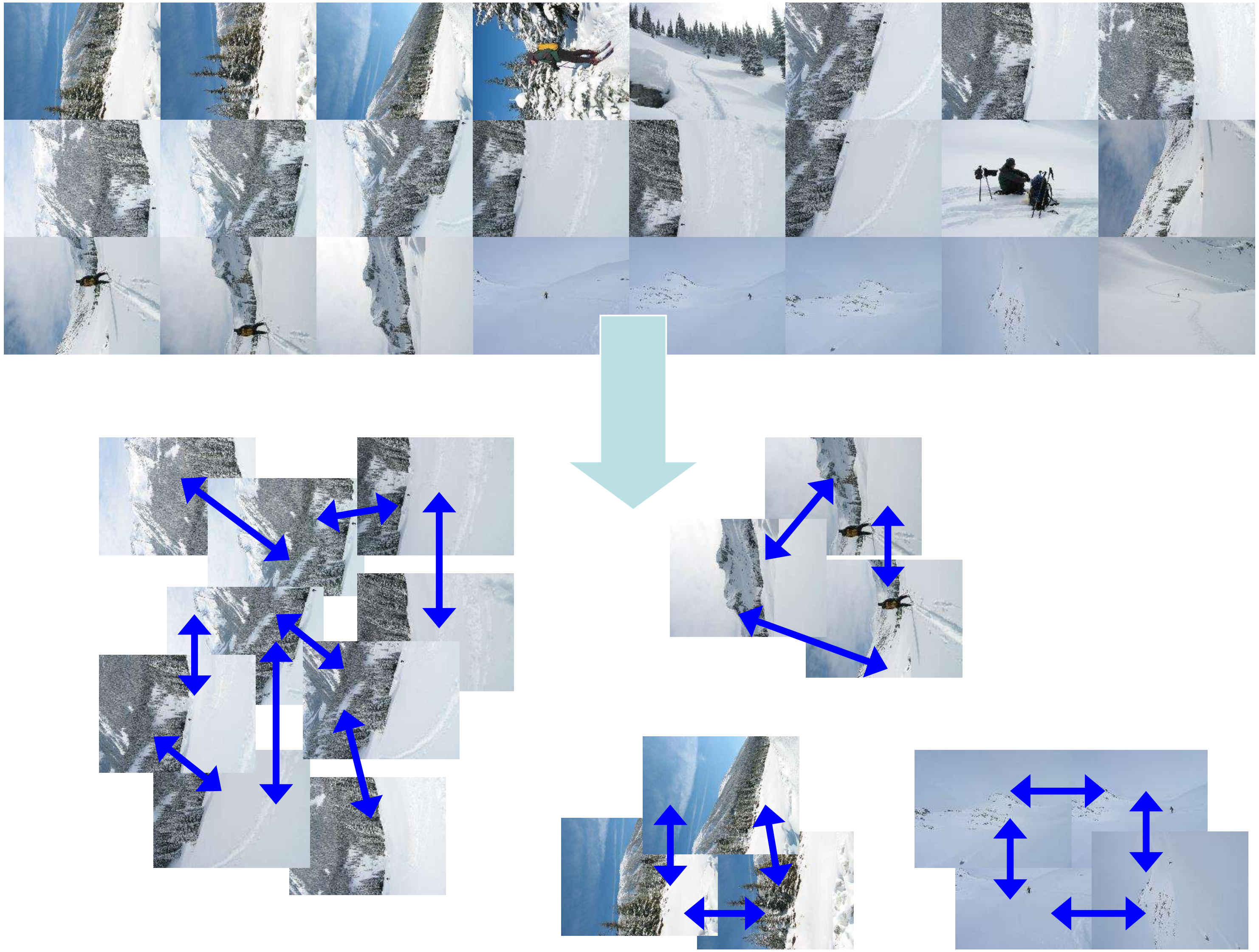


- http://www.cs.ubc.ca/~mbrown/autostitch/autostitch.html

8.3 Automatic image stitching



8.3 Automatic image stitching



8.3 Automatic image stitching

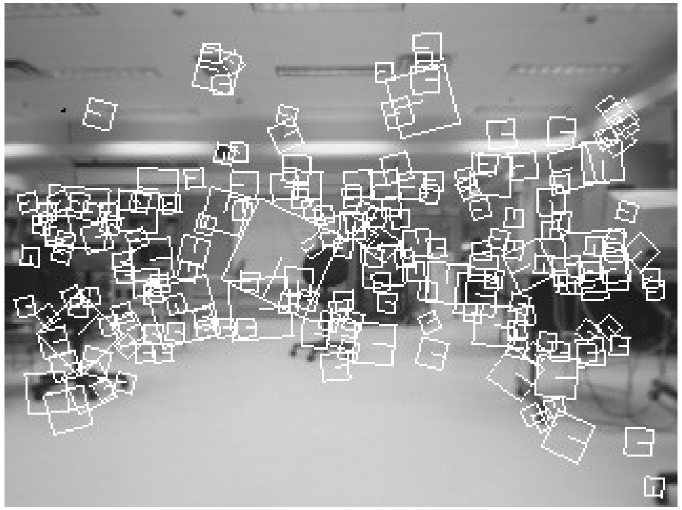
8.4 Robotics - Sony Aibo

SIFT is used for

- Recognizing charging station
- Communicating with visual cards
- Teaching object recognition

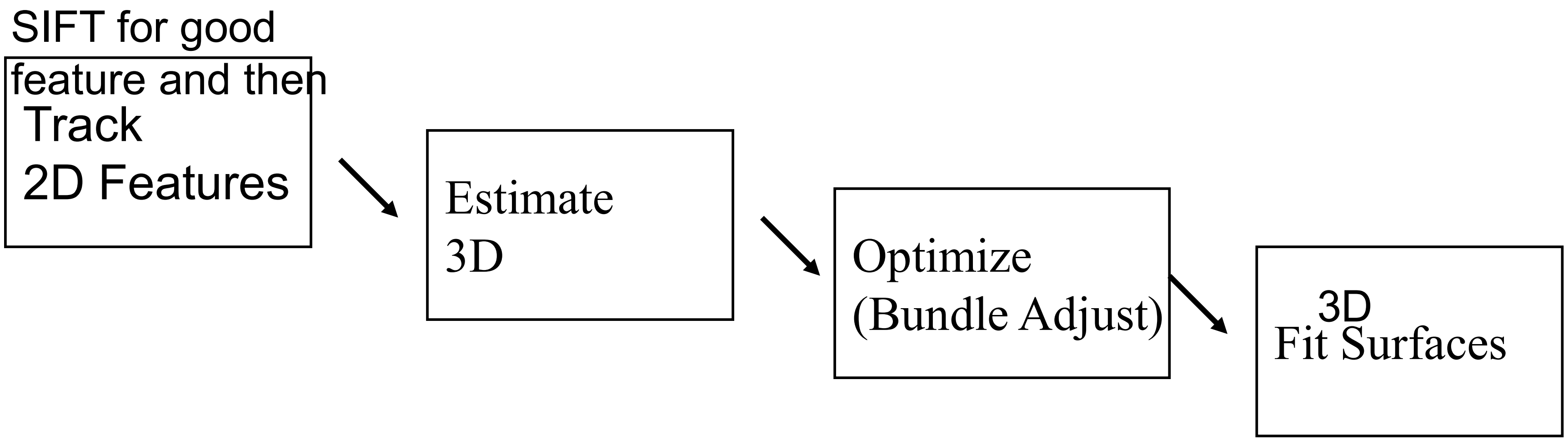
- soccer

8.4 Robot location



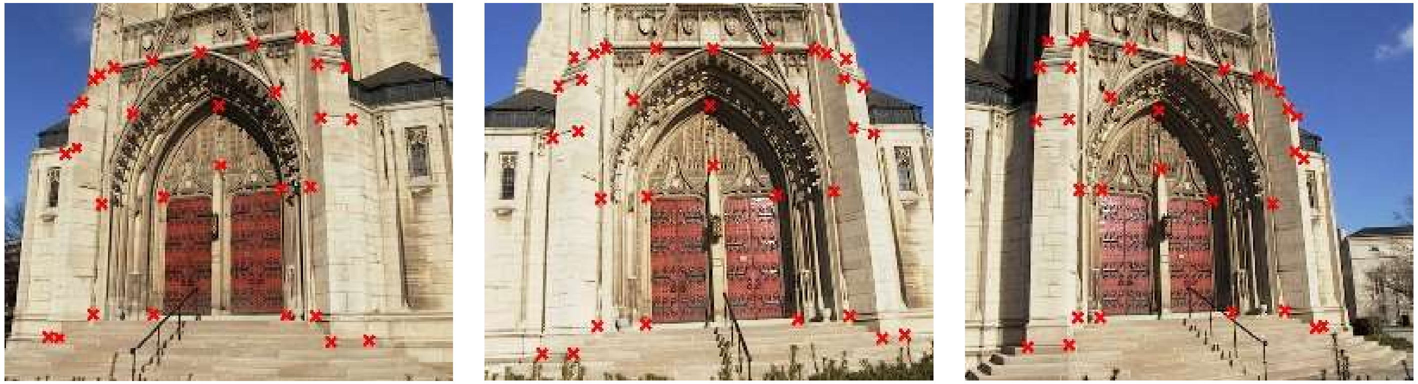
8.5 Structure from Motion (1/2)

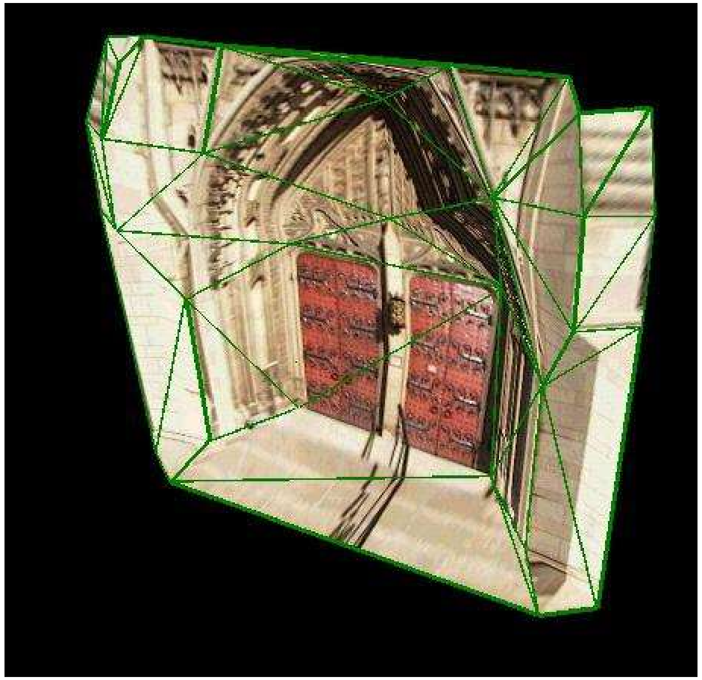
- The SFM Problem
  - Reconstruct scene geometry and camera motion from two or more images



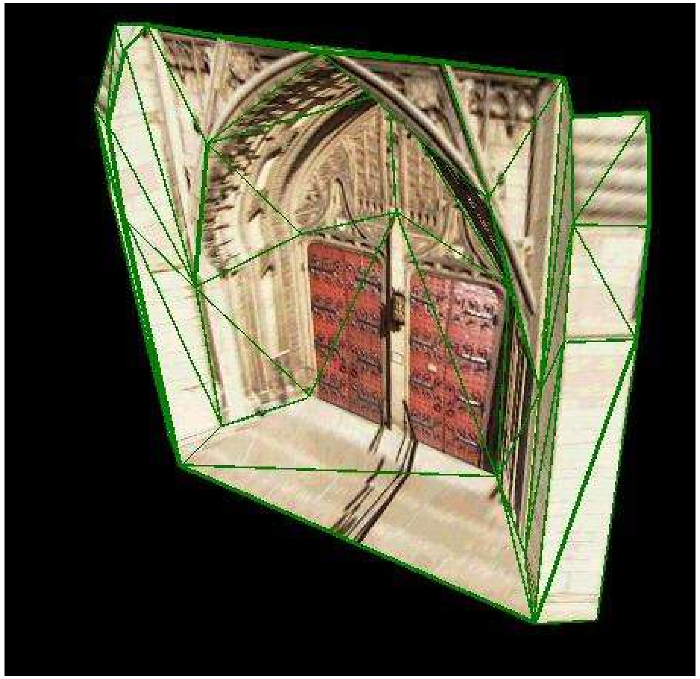
SFM Pipeline

8.5 Structure from Motion (2/2)



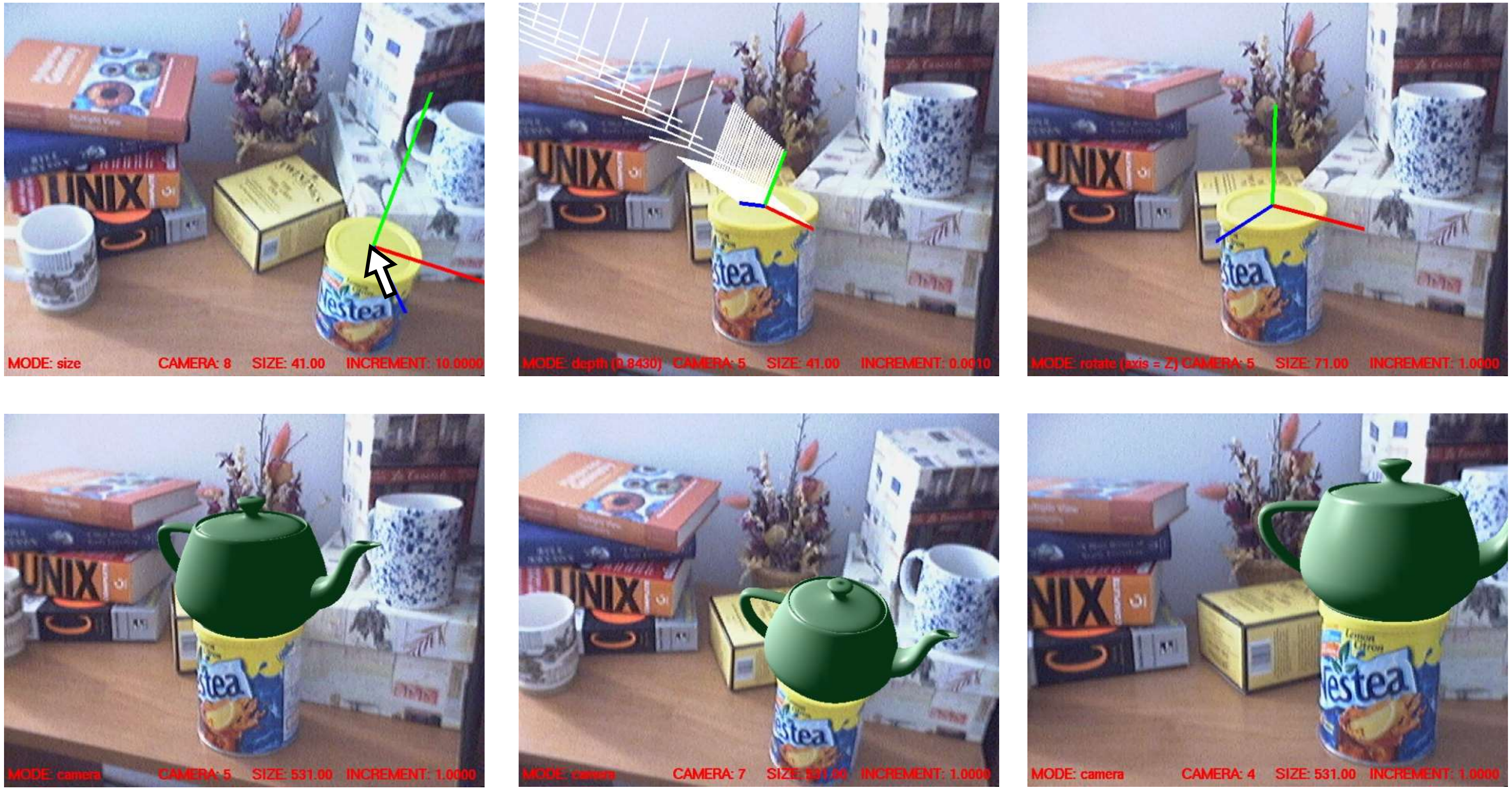


Poor mesh



Good mesh

8.6 Augmented reality



Reference

1. C. Harris and M. Stephens, "A Combined Corner and Edge Detector," 4th Alvey Vision Conference, pp147-151, 1988.
2. D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," IJCV, 60(2), , pp91-110, 2004.

Append A. Gradient & Hessian of D

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

$$\begin{pmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \\ \frac{\partial f}{\partial \sigma} \end{pmatrix} \qquad \begin{pmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial x \partial \sigma} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} & \frac{\partial^2 D}{\partial y \partial \sigma} \\ \frac{\partial^2 D}{\partial x \partial \sigma} & \frac{\partial^2 D}{\partial y \partial \sigma} & \frac{\partial^2 D}{\partial \sigma^2} \end{pmatrix}$$

$$\frac{\partial D}{\partial x} = \frac{D_{x+1,y,\sigma} - D_{x-1,y,\sigma}}{2} \qquad \frac{\partial^2 D}{\partial x^2} = D_{x+1,y,\sigma} - 2D_{x,y,\sigma} + D_{x-1,y,\sigma} \qquad \frac{\partial D}{\partial x \partial y} = \frac{D_{x+1,y+1,\sigma} - D_{x+1,y-1,\sigma} - D_{x-1,y+1,\sigma} + D_{x-1,y-1,\sigma}}{4}$$

$$\frac{\partial D}{\partial y} = \frac{D_{x,y+1,\sigma} - D_{x,y-1,\sigma}}{2} \qquad \frac{\partial^2 D}{\partial y^2} = D_{x,y+1,\sigma} - 2D_{x,y,\sigma} + D_{x,y-1,\sigma} \qquad \frac{\partial D}{\partial y \partial \sigma} = \frac{D_{x,y+1,\sigma+1} - D_{x,y+1,\sigma-1} - D_{x,y-1,\sigma+1} + D_{x,y-1,\sigma-1}}{4}$$

$$\frac{\partial D}{\partial \sigma} = \frac{D_{x,y,\sigma+1} - D_{x,y,\sigma-1}}{2} \qquad \frac{\partial^2 D}{\partial \sigma^2} = D_{x,y,\sigma+1} - 2D_{x,y,\sigma} + D_{x,y,\sigma-1} \qquad \frac{\partial D}{\partial x \partial \sigma} = \frac{D_{x+1,y,\sigma+1} - D_{x+1,y,\sigma-1} - D_{x-1,y,\sigma+1} + D_{x-1,y,\sigma-1}}{4}$$

Append B. Derivation of matrix form (1/2)

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

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

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

Append B. Derivation of matrix form (2/2)

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

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

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

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

$$= (\mathbf{A}^\mathrm{T} + \mathbf{A})\mathbf{x}$$