

بینایی کامپیوتر

فصل اول

نمایش و توصیف شی

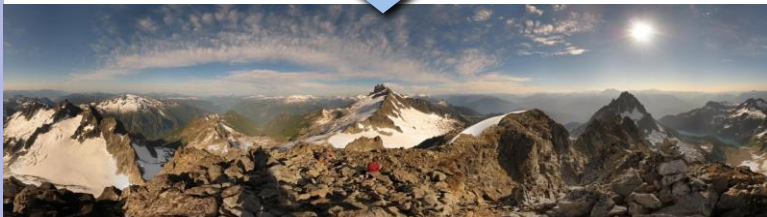
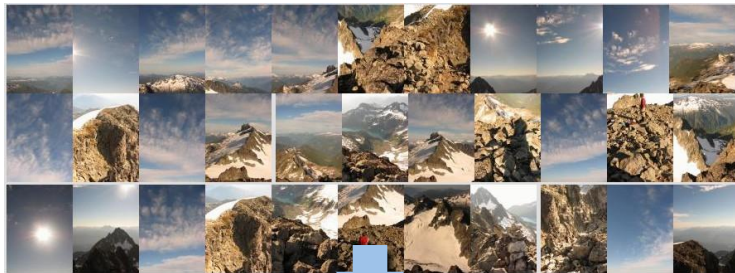
ویژگی هریس و سیفت

HARRIS CORNER & SIFT

محمدجواد فدائی اسلام

ویرایش فروردین ۱۴۰۴

اهمیت استخراج ویژگی کاربرد تولید پانوراما

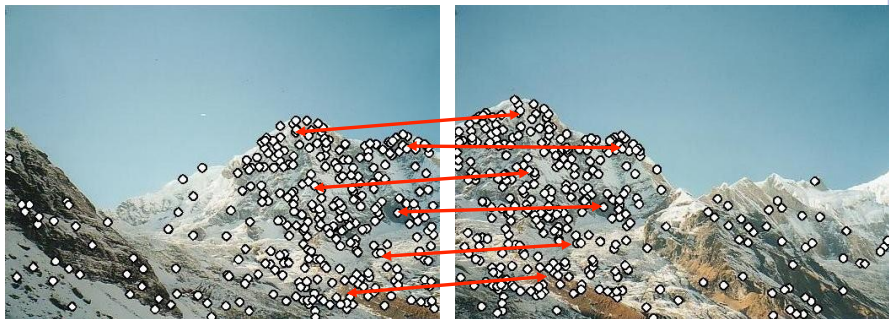


credit: Matt Brown

چگونه دو تصویر برای پانوراما باهم ترکیب شوند؟



چگونه دو تصویر برای پانوراما باهم ترکیب شوند؟
چگونه ویژگی مناسب استخراج نماییم؟



چگونه دو تصویر برای پانوراما باهم ترکیب شوند؟
چگونه دو تصویر ترکیب شوند؟



Figure: Image alignment

پیچیدگی‌های تطبیق (تغییر رنگ)



by [Diva Sian](#)

[Source: N. Snavely]



by [swashford](#)

پیچیدگی‌های تطبیق ... (وجود مانع)



by [Diva Sian](#)



by [scgbt](#)

[Source: N. Snavey]

پیچیدگی‌های تطبیق

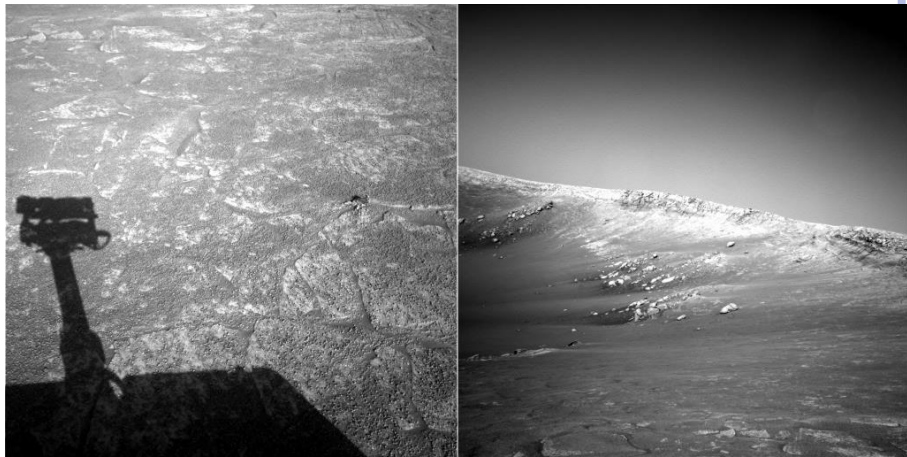


Figure: NASA Mars Rover images

پیچیدگی‌های تطبیق تطبیق ویژگی‌های سیفت

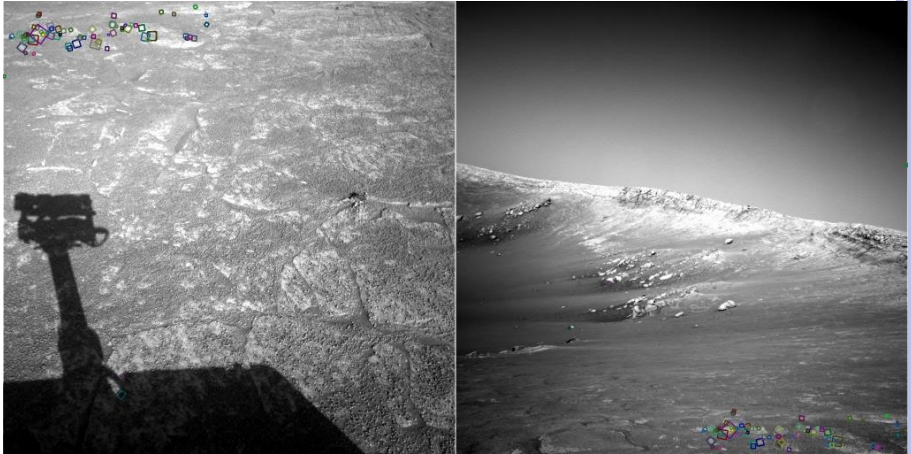


Figure: NASA Mars Rover images with SIFT feature matches

[Source: N. Snavely]

مراحل تطبیق

○ شناسایی نقاط مناسب

Detection: Identify the interest points.

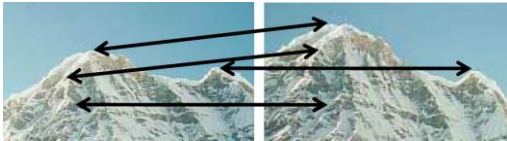
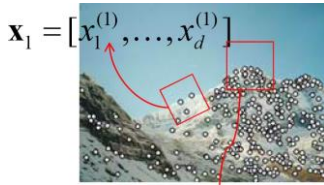
○ استخراج ویژگی برای نقاط مناسب (توصیف نقاط)

Description: Extract vector feature descriptor around each interest point.

○ انطباق (تطبیق)

Matching: Determine correspondence between descriptors in two views.

مراحل تطبيق



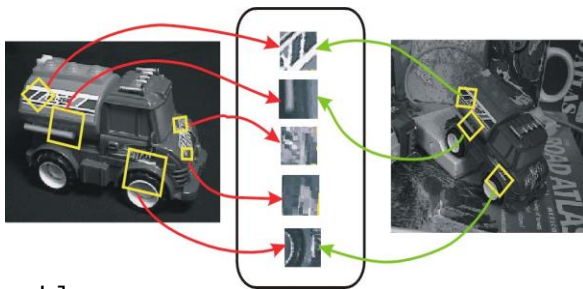
[Source: K. Grauman]

شناسایی نقاط مناسب

- باید حداقل تعدادی نقاط مشابه در هر دو تصویر شناسایی شوند.
- رویه شناسایی باید به طور مستقل در هر دو تصویر اجرا شود.
- نقاط باید نسبت به تغییرات هندسی و تصویربرداری مقاوم باشند.
- نقاط باید انطباق قابل اعتمادی را ایجاد نمایند.
- به دنبال مناطق از تصویر باشیم که «غیرعادی - غیرمعمول» باشند تا منجر به تطابق بدون ابهام در تصاویر دیگر شود. چگونه «غیر معمول (unusual)» را تعریف کنیم؟

ویژگی‌های محلی مستقل

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

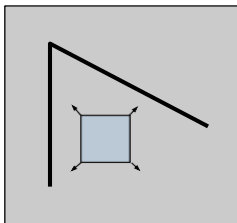


Feature Descriptors

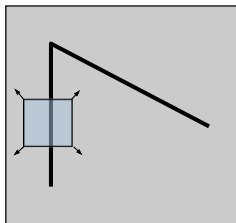
[Source: N. Snavely]

شناسایی نقاط غیر معمول

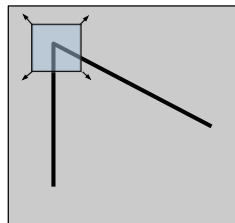
- مکان‌یابی نواحی بدون بافت تقریباً غیرممکن است.
- وصله‌هایی با تغییرات کنتراست (گرادیان) بزرگ آسان‌تر قابل مکان‌یابی هستند.
- خط مستقیم مناسب نیست زیرا حرکت در یک جهت هیچ تغییری به دست نمی‌دهد.
- نقاطی مناسبند که در دو جهت دارای گرادیان بالا باشند، مانند گوشه‌ها



“flat” region:
no change in all
directions



“edge”:
no change along the
edge direction

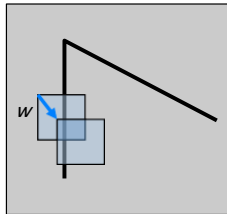


“corner”:
significant change in
all directions

طراحی یک معیار ساده انطباق

- با جابه‌جایی پنجره W به اندازه (u, v) مقادیر پیکسل‌ها به چه میزان تغییر می‌کند؟
- محاسبه با استفاده از جمع مربعات اختلاف

summing up the squared differences (SSD)



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

طراحی یک معیار ساده انطباق

AUTO-CORRELATION FUNCTION

- ما نمی‌دانیم که این ویژگی در نهایت با کدام مکان تصویر دیگر مطابقت دارد.
- ما فقط می‌توانیم میزان پایداری این معیار را با توجه به تغییرات کوچک در موقعیت u با مقایسه یک وصله تصویر با خودش محاسبه کنیم.
- این تابع همبستگی خودکار است.

- Auto-correlation function

$$E_{AC}(\Delta u) = \sum_i [I_0(p_i + \Delta u) - I_0(p_i)]^2$$

طراحی یک معیار ساده انطباق ...

• با فرض حرکت کوچک و با استفاده از بسط سری تیلور داریم:

$$I_0(p_i + \Delta u) \approx I_0(p_i) + \nabla I_0(p_i) \Delta u,$$

$$\nabla I_0(p_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y} \right) p_i$$

$$E_{AC}(\Delta u) = \sum_i [I_0(p_i + \Delta u) - I_0(p_i)]^2 =$$

$$\sum_i [I_0(p_i) + \nabla I_0(p_i) \Delta u - I_0(p_i)]^2 = \sum_i [\nabla I_0(p_i) \Delta u]^2 = \Delta u^T A \Delta u$$

ساده سازی ریاضی

$$\left([a \quad b] \begin{bmatrix} c \\ d \end{bmatrix}\right)^2 = [a \quad b] \begin{bmatrix} c \\ d \end{bmatrix} [a \quad b] \begin{bmatrix} c \\ d \end{bmatrix} =$$

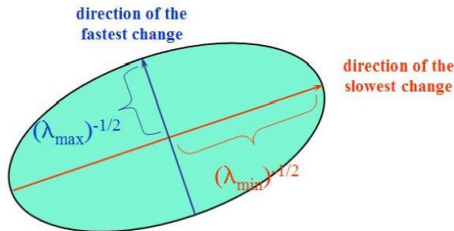
$$[c \quad d] \begin{bmatrix} a \\ b \end{bmatrix} [a \quad b] \begin{bmatrix} c \\ d \end{bmatrix} = [c \quad d] \begin{bmatrix} a^2 & ab \\ ab & b^2 \end{bmatrix} \begin{bmatrix} c \\ d \end{bmatrix}$$

طراحی یک معیار ساده انطباق ...

$$E_{AC}(\Delta u) = \Delta u^T A \Delta u$$

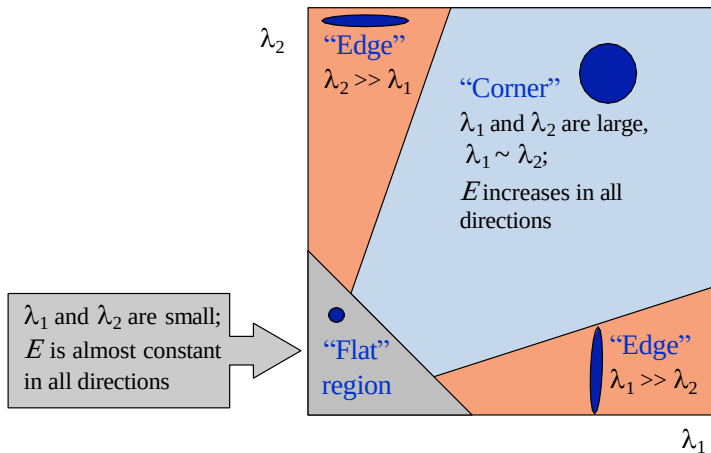
$$A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$$A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = U \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} U^T, Au_i = \lambda_i u_i$$

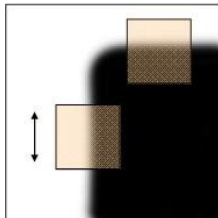


ماتریس A متقارن است.
مقادیر ویژه A شدت تغییر را در دو جهت اصلی گرادیان متعامد در
پنجره نشان می‌دهد.

بزرگی مقادیر ویژه و اثر ماتریس A در مکان‌های مختلف تصویر



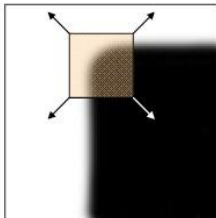
بزرگی مقادیر ویژه ماتریس A در مکان‌های مختلف تصویر



“edge”:

$$\lambda_1 \gg \lambda_2$$

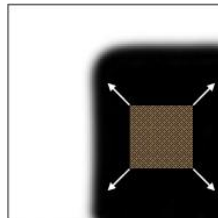
$$\lambda_2 \gg \lambda_1$$



“corner”:

λ_1 and λ_2 are large,

$$\lambda_1 \sim \lambda_2;$$



“flat” region

λ_1 and λ_2 are
small;

الگوریتم هریس

- گرادیان هر نقطه از تصویر در دو جهت محاسبه شود.
- ماتریس A برای هر نقطه محاسبه شود.
- مقادیر ویژه A برای هر نقطه محاسبه شود.
- نقاطی که مقدار آستانه آنها از حد آستانه بیشتر شد، به عنوان ویژگی در نظر گرفته شود.

$$Ax = \lambda x \Rightarrow (A - \lambda I)x = 0$$

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

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, A - \lambda I = \det \begin{bmatrix} a - \lambda & b \\ c & d - \lambda \end{bmatrix} = 0$$

$$(a - \lambda)(d - \lambda) - bc = 0$$

$$\lambda^2 - (a + d)\lambda + ad - bc = 0$$

$$\lambda_1 + \lambda_2 = a + d = \text{trace}(A)$$

$$\lambda_1 \lambda_2 = ad - bc = \det(A)$$

RESPONSE FUNCTION (R)

FIRST APPROACH

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

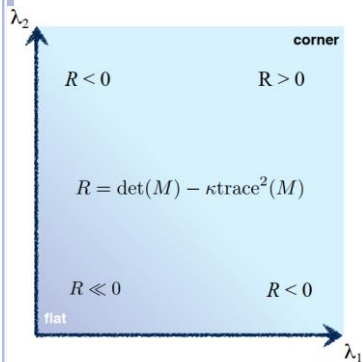
- $x \ll y \Rightarrow \frac{xy}{x+y} = x \left(\frac{1}{\frac{x}{y}+1} \right) \sim x$

- $R = \min(\lambda_1, \lambda_2) = \frac{\det(A)}{\text{trace}(A)}$

RESPONSE FUNCTION (R)

SECOND APPROACH

- $R = \lambda_1 \lambda_2 - \kappa(\lambda_1 + \lambda_2)$
- $R = \det(A) - \kappa \text{trace}(A)^2$,
 $\kappa \in [0.04, 0.06]$



SIMPLE HARRIS CODE

```
img = cv2.imread(image_path)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)

# مشتق‌ها با فیلتر سوبل
Ix = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
Iy = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)

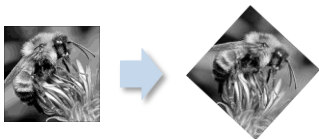
# محاسبه عناصر ماتریس M
Ixx = Ix**2
Iyy = Iy**2
Ixy = Ix*Iy

# محاسبه R
k = 0.04
det_M = Ixx * Iyy - Ixy**2
trace_M = Ixx + Iyy
R = det_M - k * (trace_M**2)

# آستانه‌گذاری برای گوشه‌ها
threshold = 0.01 * R.max()
corners = R > threshold
```

IMAGE TRANSFORMATIONS

Geometric:



Rotation



Scale

Photometric:

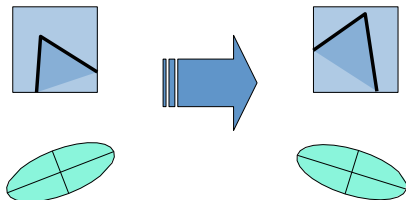


Intensity change

[Source: N. Snavely]

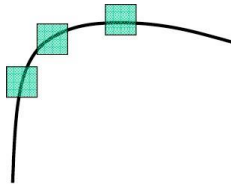
هریس مستقل از چرخش است

$$A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} = U \begin{bmatrix} \lambda_0 & 0 \\ 0 & \lambda_1 \end{bmatrix} U^T, Au_i = \lambda_i u_i$$



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

هریس مستقل از مقیاس نیست

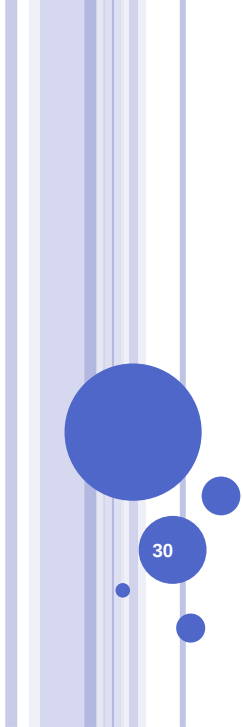


All points will be
classified as **edges**



Corner !

[Source: K. Grauman]



SCALE INVARIANT FEATURE TRANSFORM (SIFT)

30

SIFT BACKGROUND

○ Scale-invariant feature transform

- **SIFT:** to **detect** and **describe** local features in an images.
- Proposed by *David Lowe* in ICCV1999.
- Refined in IJCV 2004.
- Cited more than **65953** times till now(March 2022).
- Wildly used in image search, object recognition, video tracking, gesture recognition, *etc.*

ویژگی‌های مطلوب سift

- Invariant to scale change
- Invariant to rotation change
- Invariant to illumination change
- Robust to addition of noise
- Robust to substantial range of affine transformation
- Robust to 3D view point
- Highly distinctive for discrimination

CLAIMED ADVANTAGES OF SIFT

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

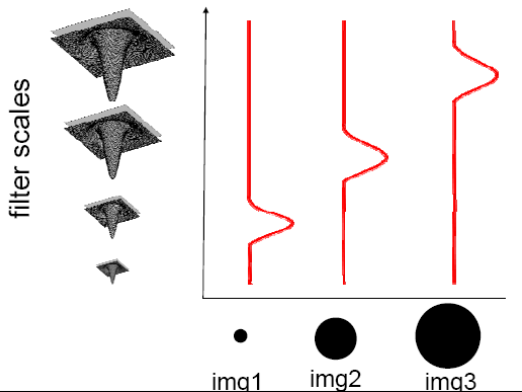


مراحل استخراج ویژگی SIFT

- Scale-space extrema detection
- Keypoint localization
- Orientation assignment
- Keypoint descriptor

اساس ویژگی سفت بر فیلتر لاپلاس استوار است

Laplacian-of-Gaussian = “*blob*” detector

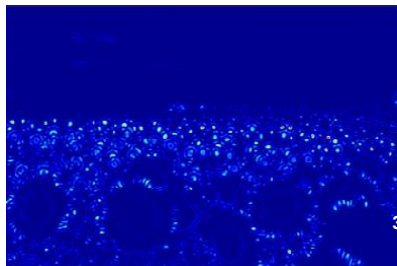
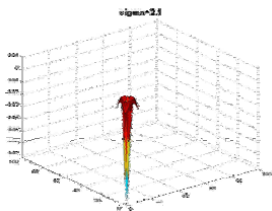


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

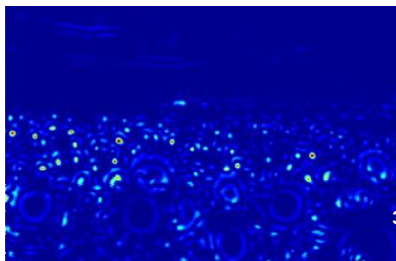
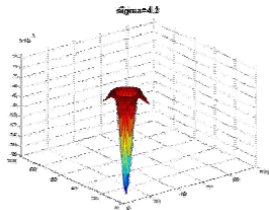
$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 2 & 1 & 0 \\ 1 & 2 & -16 & 2 & 1 \\ 0 & 1 & 2 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

2-D LoG
by a 5×5
convolution kernel

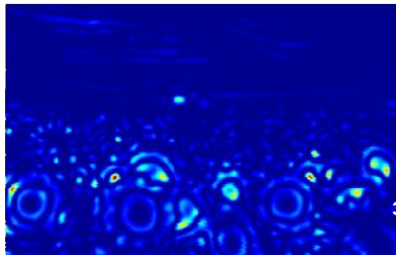
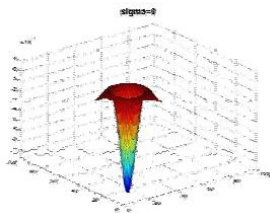
اثر فیلتر لاپلاس در ابعاد مختلف



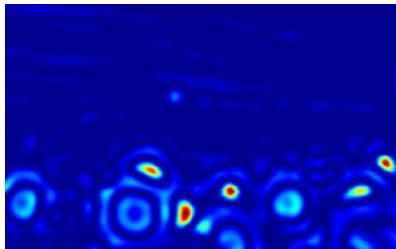
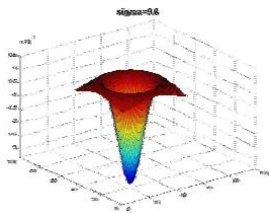
اثر فیلتر لاپلاس در ابعاد مختلف ...



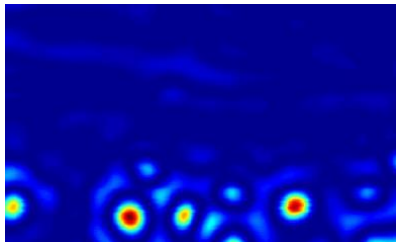
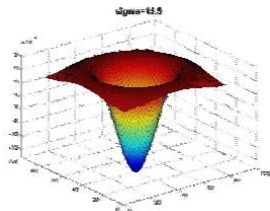
اثر فیلتر لاپلاس در ابعاد مختلف ...



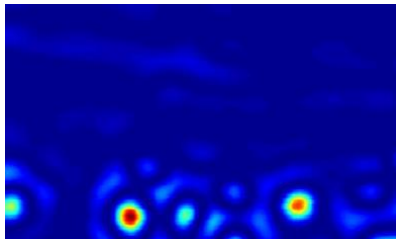
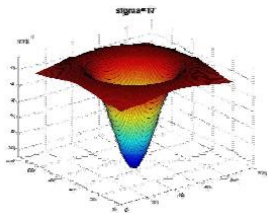
اثر فیلتر لاپلاس در ابعاد مختلف ...



اثر فیلتر لاپلاس در ابعاد مختلف ...



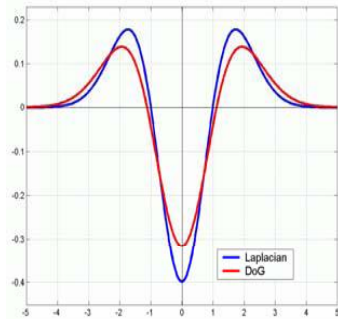
اثر فیلتر لاپلاس در ابعاد مختلف ...



LoG vs. DoG

تقریب لاپلاسین با اختلاف گوسی

$$\nabla^2 G_\sigma(x, y) = \left(\frac{x^2 + y^2}{\sigma^4} - \frac{2}{\sigma^2} \right) G_\sigma(x, y)$$

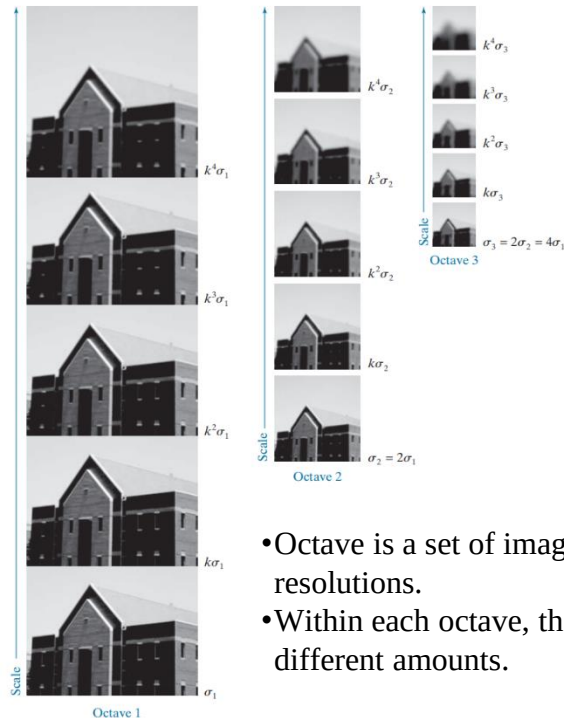


$$\underbrace{G(x, y, k\sigma) - G(x, y, \sigma)}_{\text{DoG}} \approx (k-1)\sigma \frac{\partial G}{\partial \sigma} = (k-1)\sigma^2 \underbrace{\nabla^2 G}_{\text{LoG}}$$



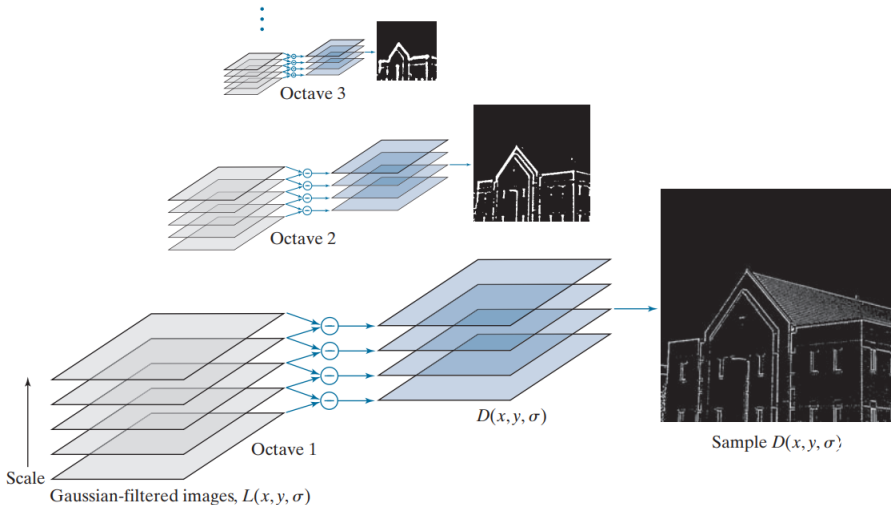
DoG: Difference of Gaussians

OCTAVE & SCALE



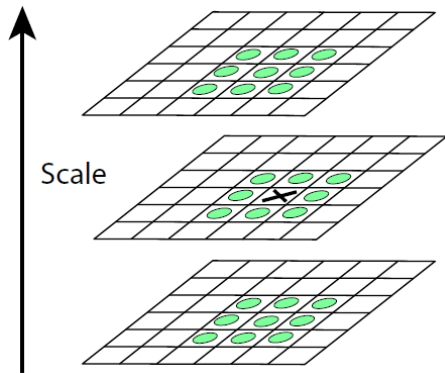
- Octave is a set of images at different resolutions.
- Within each octave, the image is blurred by different amounts.

GENERATING DoG IMAGE BY SUBTRACTING TWO CONSECUTIVE BLURRED IMAGES IN EACH OCTAVE



LOCAL EXTREMA DETECTION

- Detecting extrema (maxima or minima) of the $D(x,y,\sigma)$ images in an octave.
- Compare x with its 26 neighbors at 3 scales



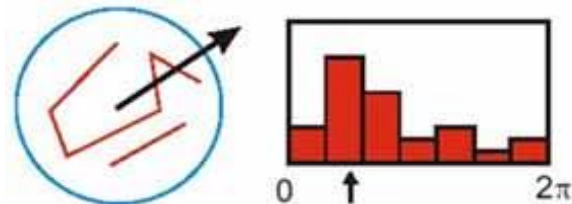
ORIENTATION

Gradient and angle:

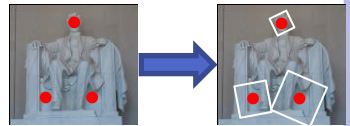
$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$

$$\theta(x, y) = \arctan 2((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y)))$$

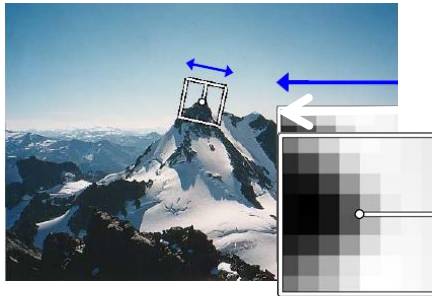
Orientation selection



SIFT DESCRIPTOR

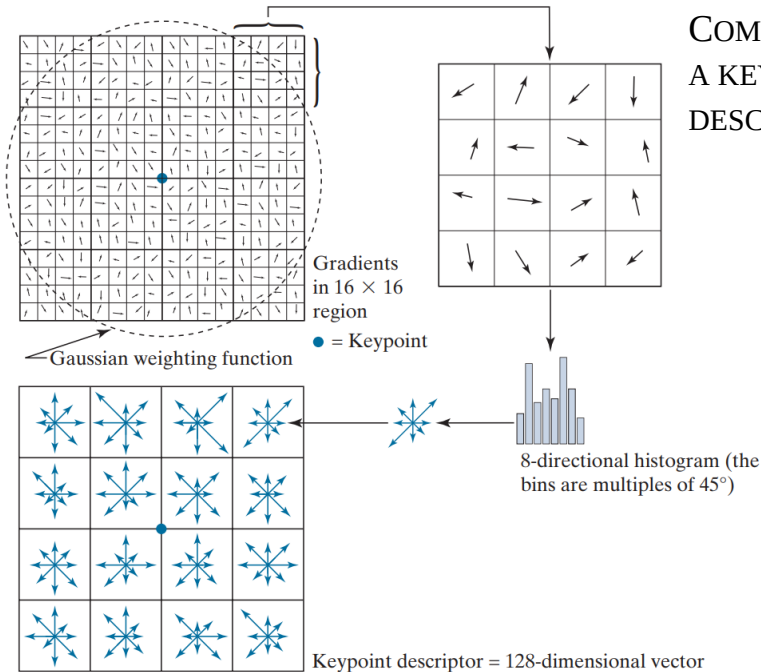


- Making descriptor rotation invariant



- Rotate patch according to its dominant gradient orientation
- This puts the patches into a canonical orientation.

COMPUTING A KEYPOINT DESCRIPTOR



SUMMARY OF SIFT FEATURE

- Descriptor: 128-D

- 4 by 4 patches, each with 8-D gradient angle histogram:
 $4 \times 4 \times 8 = 128$
- Normalized to reduce the effects of illumination change.

- Position: (x, y)

- Where the feature is located at.

- Scale

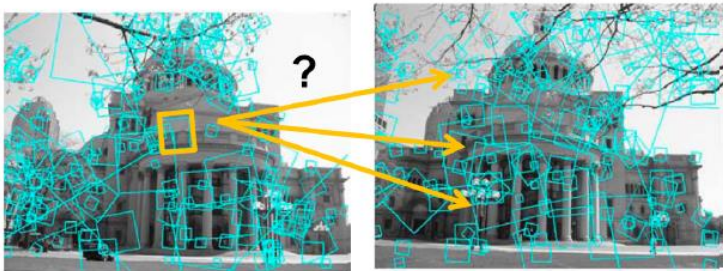
- Control the region size for descriptor extraction.

- Orientation

- To achieve rotation-invariant descriptor.

MATCHING LOCAL FEATURES

- To **generate candidate matches**, find patches that have the most similar appearance (e.g., lowest SSD)
- Simplest approach: **compare them all**, take the closest (or closest k, or within a thresholded distance)



AMBIGUOUS MATCHES

- At what SSD value do we have a good match?
- To add robustness, consider ratio of distance to best match to distance to second best match
 - If low, first match looks good.
 - If high, could be ambiguous match.



$\text{Closest} / \text{NextClosest} < \text{threshold}(0.6 \text{ for example})$