Applied Computer Vision
H.W.1

Name: 洪翊誠 / Student ID: 310653005
應用數學系數學建模與科學計算

November 13, 2022

SIFT (Scale-invariant feature transform)

Figure. The original input image

- **Question 1**

  What do we do to get the scale-invariant feature ?

Answer1:

Algorithm SIFT Algorithm

Input: Image
Scale-space extrema detection
Keypoint localization
Orientation assignment
Keypoint descriptor

- generateBaseImage

  生成影像

- computeNumberOfOctaves

  根據搜尋範圍決定 octave 最多個數，確保最高層的大小至少 3×3，才可被搜索

$$N_{Octave} = \lfloor \frac{\log_e(\min\{W, H\})}{\log 2} - 1 \rfloor$$

- generateGaussianImages
  生成高斯模糊影像

$$G_{blur} = G(x,y,\sigma) * I(x,y),\ \text{where } G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2}).$$

- generateDoGImages
  用於計算 *Laplacian operator* 二次微分的近似解
  (Note: $(G(x,y,k\sigma) - G(x,y,\sigma) \approx (k-1)\sigma^2\nabla^2 G)$

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

- findScaleSpaceExtrema
  1. 將每個 octave 中的 scale space 堆疊起，檢查以 (i,j) 為中心 $3\times3\times3$ 中的數值是否 $v_{i,j} = \max v_*$, $*$ means the neighbor of (i,j).
  2. 利用泰勒展開式做二次近似 $D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T\frac{\partial^2 D}{\partial^2 \mathbf{x}}\mathbf{x}$, where $\mathbf{x} = (x,y,\sigma)^T$，解出一次微分為 0 之解 (i.e. $\widehat{\mathbf{x}} = \frac{\partial^2 D}{\partial \mathbf{x}^2}^{-1}\frac{\partial D}{\partial \mathbf{x}}$)，帶入泰勒展式中得 $D(\widehat{\mathbf{x}}) = D + \frac{1}{2}\frac{\partial D^T}{\partial \mathbf{x}}\widehat{\mathbf{x}}$.，計算 Gradient 和 Hessian 矩陣解的 $\widehat{\mathbf{x}}$ 帶入 $D(\widehat{\mathbf{x}})$ 得到向量並以閾值通過 0.03 決定要保留或捨棄該點。

- removeDuplicateKeypoints
  根據... 去除重複的關鍵特徵點，減少後面配對不必要的重複計算量

- convertKeypointsToInputImageSize
  將關鍵點尺度、影像資訊都拉回至與原始輸入影像一致。

- generateDescriptors

  1. 蒐集所有有效的關鍵點，透過這些點來解 affine transfrom 需要的拉伸推移矩陣，以及平移向量如下: 透過 affine transformation 就可得知兩圖之間對應仿射變換關係。

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

  上面待求的未知數為 $a_{i,j}, d_i$ for $i., j \in \{1,2\}$. 解 Normal Equation(或是 Least-

Square minimization) 得最優變換方法。

$$\begin{bmatrix} x_1 & y_1 & 0 & 0 & 1 & 0 \\ 0 & 0 & x_1 & y_1 & 0 & 1 \\ & & \cdots & & & \\ x_k & y_k & 0 & 0 & 1 & 0 \\ 0 & 0 & x_k & y_k & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \\ d_1 \\ d_2 \end{bmatrix} = \begin{bmatrix} u_1 \\ v_1 \\ \vdots \\ u_1 \\ v_1 \end{bmatrix}$$

Rewrite as $\min_x \|Ax - b\|^2$ problem.

- **Question 2**

  Refer to the implement of SIFT, there is a function called `localizeExtremumViaQuadraticFit()`, which is used to fine-tune the local extremum. Please briefly describe why we need to do the fine-tuning .

Answer2:

Quadraticfit，面對實際做影像為離散型問題，我們獲取的 keypoint 不一定為最鄰域極值結果，透過二次擬合的方式近似出極值點的位置藉由插值方式得該點。

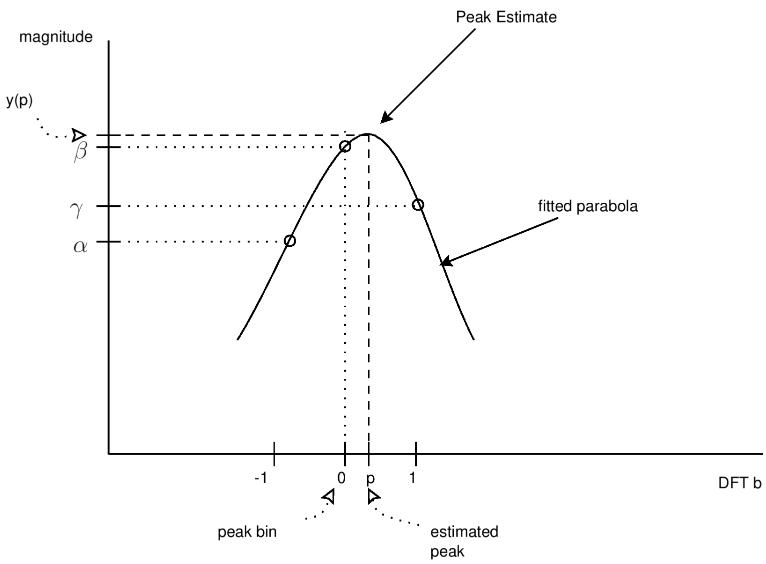

Figure 12.1: Illustration of parabolic peak interpolation using the three samples nearest the peak.

- **Question 3**

  Refer to the implement of SIFT, please list the components of the keypoint.

Answer3:

關鍵點 (keypoint) 資訊提供了

angle : the orientation of gradient.

class_id : -1

octave : 放入 unpackOctave 可轉換回 ocatve 的對應層數位置。(Note: 初始是從-1 開始，也就是影像大小放大 2 倍)。

pt : the position about the image.

response: the value of $|D(\hat{\mathbf{x}})|$ be mentioned before.

size: $\sigma(2^{(idx_I+extremun\_update)}2^{idx_{octave}+1})$ because the input image was doubled

```python
e = kp1[0]
print(f"keypoints angle: {e.angle}")
print(f"keypoints class_id: {e.class_id:}")
print(F"keypoints octave: {e.octave}")
print(f"keypoints repsonse: {e.response}")
print(f"keypoints point: {e.pt}")
print(f"keypoints size: {e.size}")
```

```
keypoints angle: 359.58380126953125
keypoints class_id: -1
keypoints octave: 7209727
keypoints repsonse: 0.023095078766345978
keypoints point: (34.03410720825195, 126.73307037353516)
keypoints size: 5.030222415924072
```

Experiment

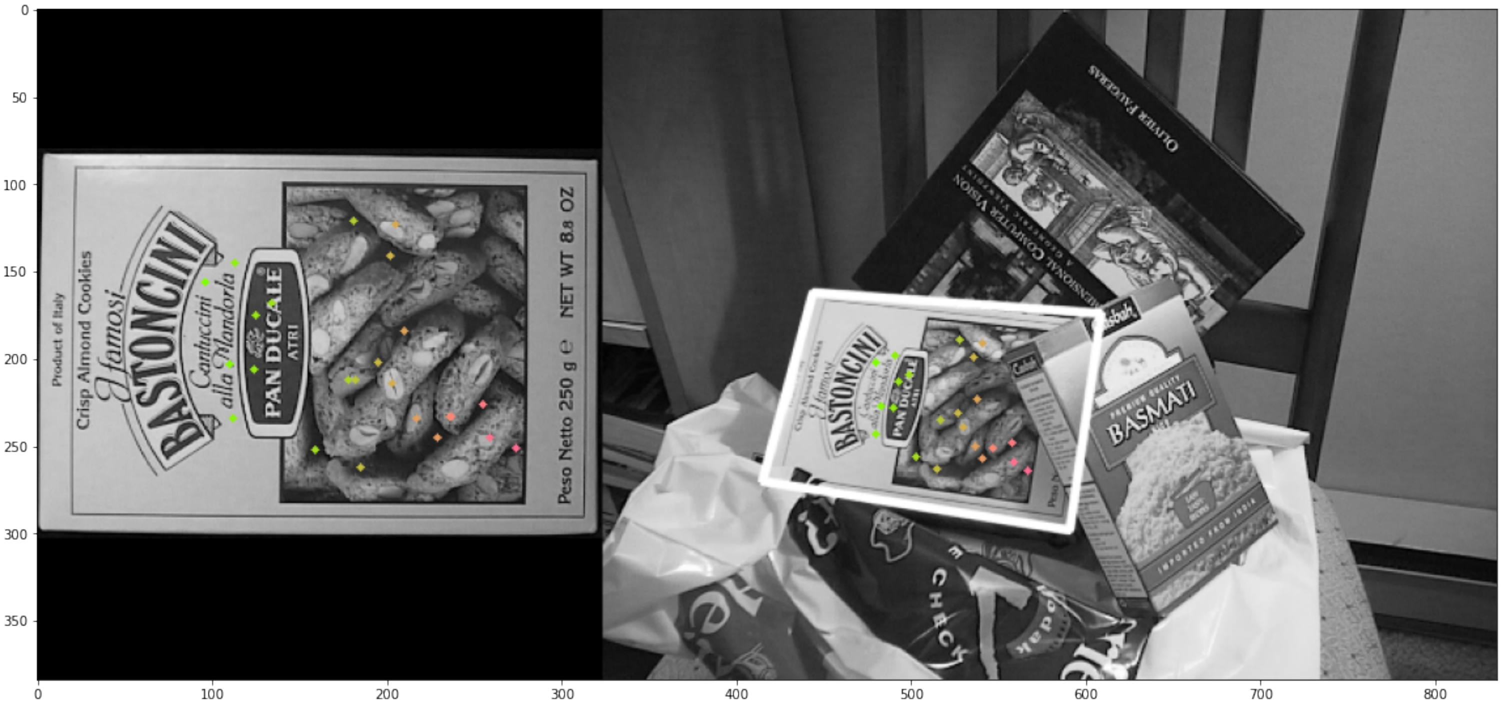

Figure. The good matches key point scatter on the image

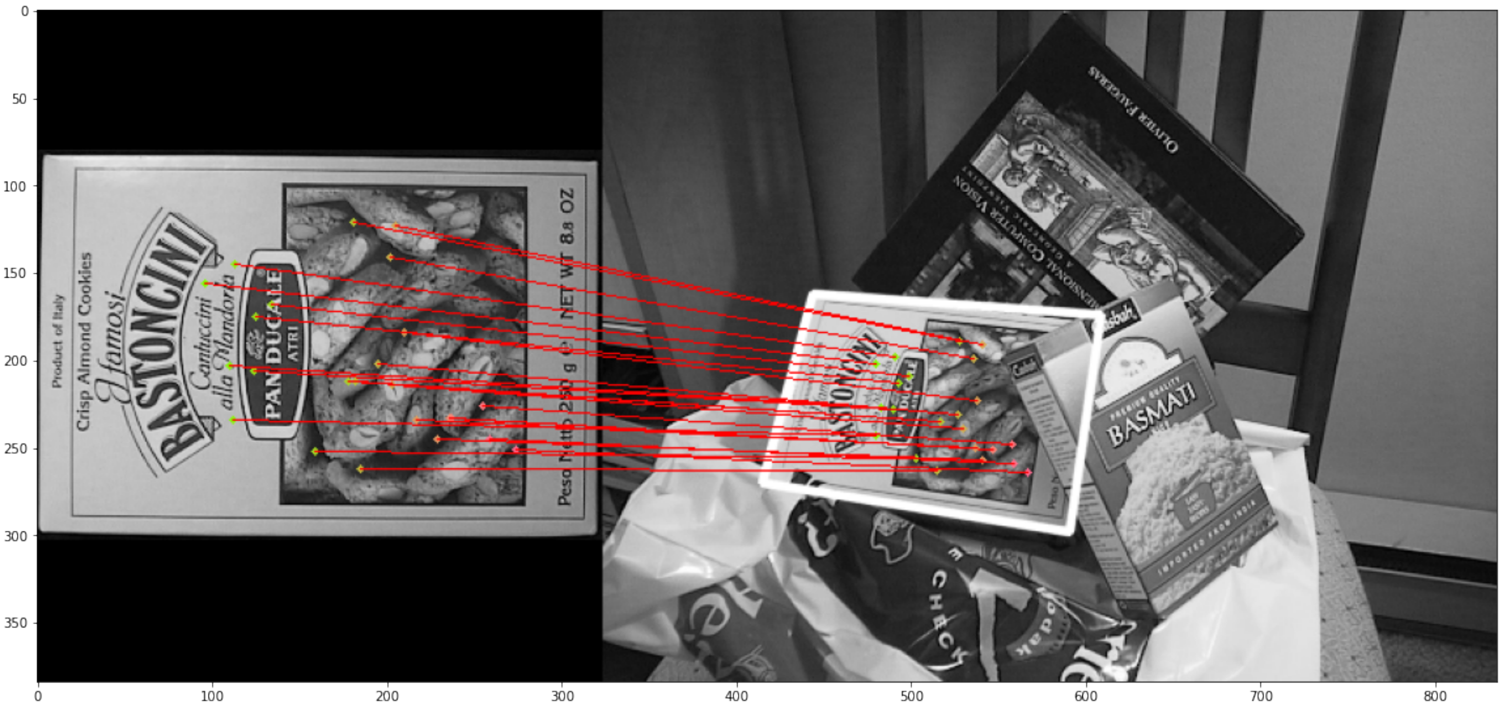

Figure. Lines on those matches pair.

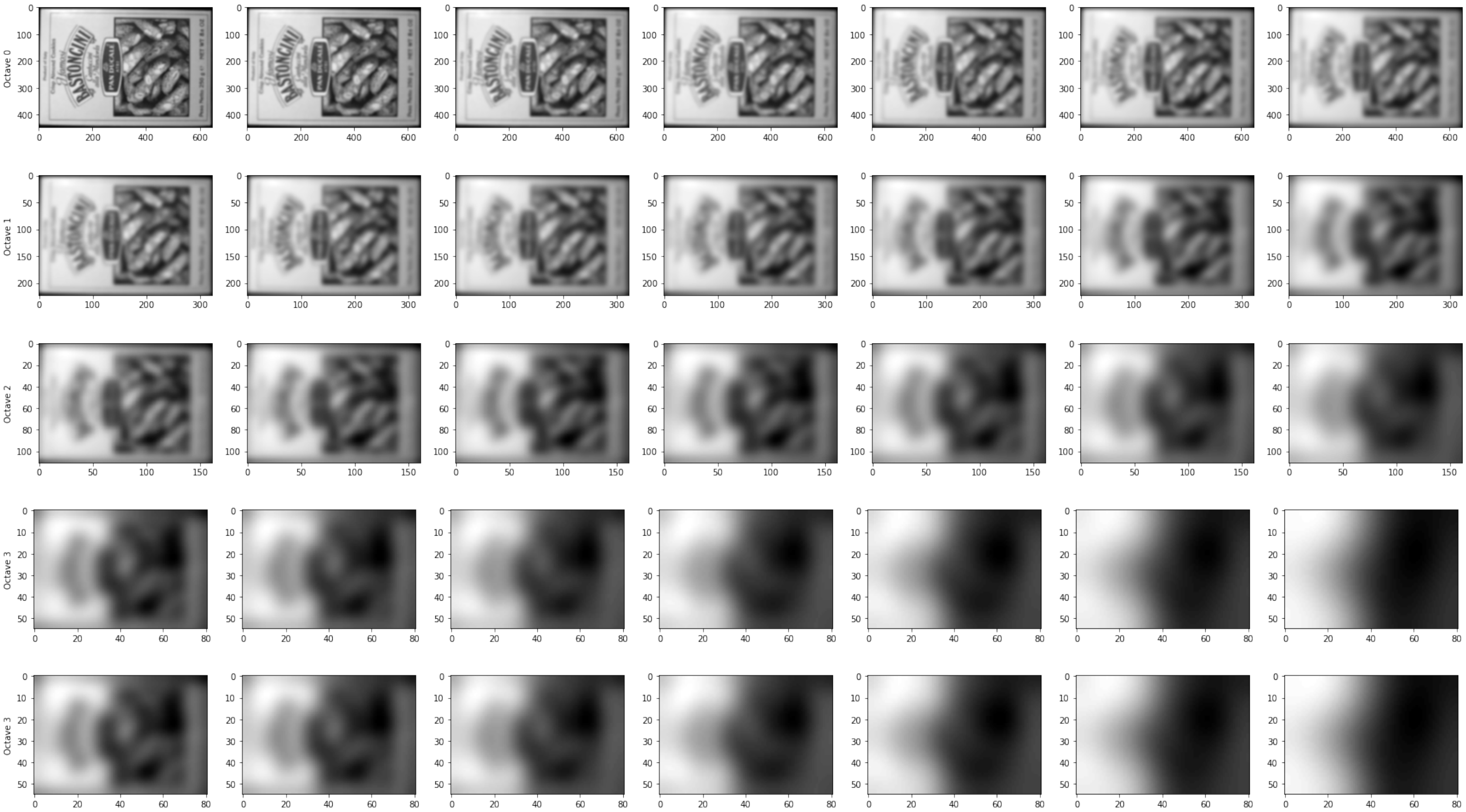

Figure. Visualize octave in each row, and gaussian blur images with the number of scale-space plus three

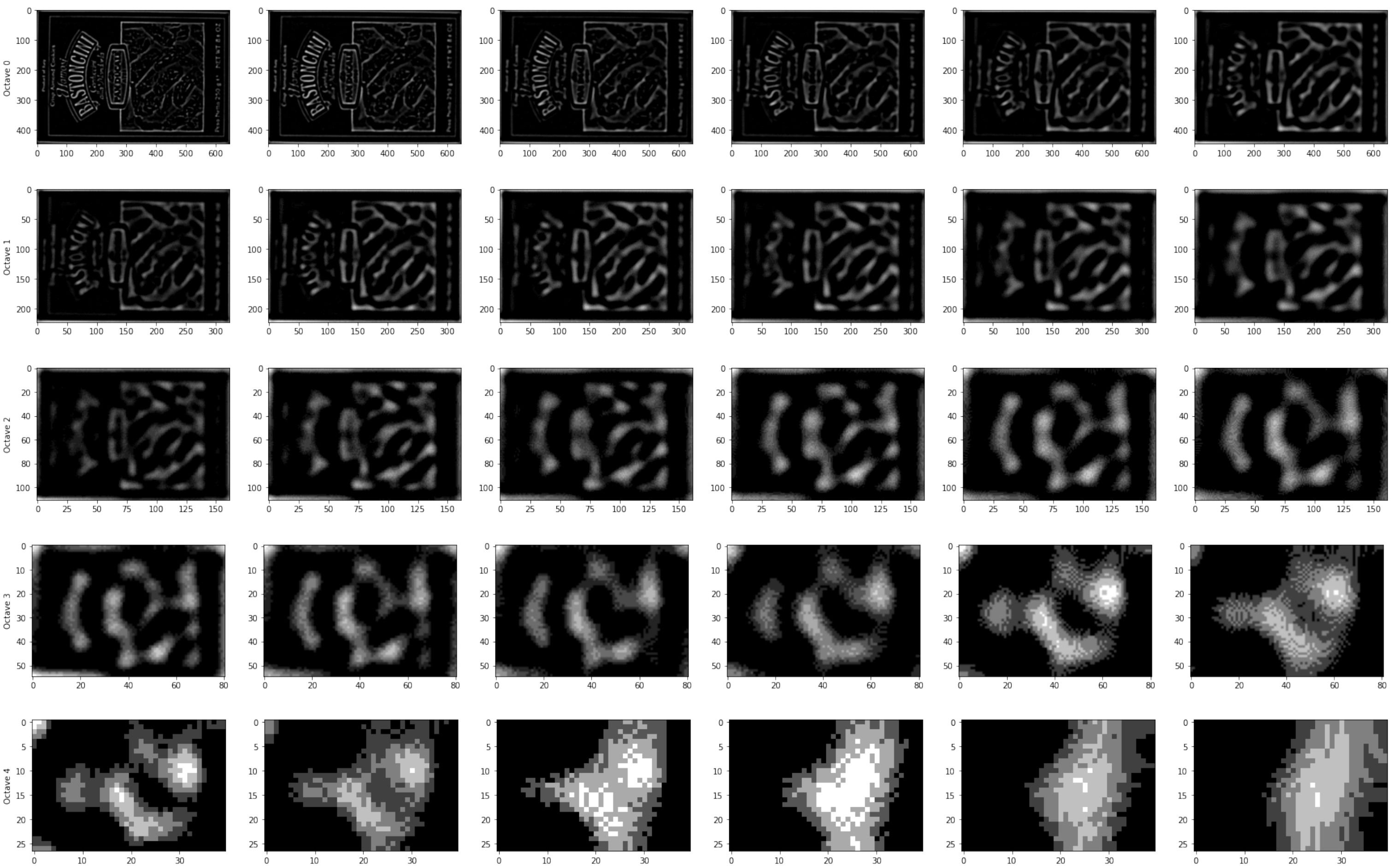

Figure. Visualize the difference of Gaussian (DoG) to approximate the Laplacian