王　兆東

マルチメディア信号解析

2020年5月24日

Report of Third Assignment

In this report I will discuss how I design my program to complete the assigned task, it include the procedure of how I understand the SIFT algorithm and implemented on Python.

1. Environment Introduction

· Programming language: Python3.6

· library would be used: PIL as Pillow, KNeighborsClassifier, matplotlib, numpy

· text book： jupyter

2. Implementation

It has been proved that the only possible scale-space kernel is Gaussian function, to build the scale space we use the $L(x, y, \sigma)$to define the space:

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

The $G(x,y,\sigma)$ represent the convolution of a variable-scale Gaussian, with a input image $I(x,y)$.

Then we use $D(x, y, \sigma)$ to compute from the difference of two nearby scales separated by a constant multiplicative factor *k*:

$$D(x,y,\sigma) = (G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y)$$

$$= L(x,y,k\sigma) - L(x,y,\sigma).$$

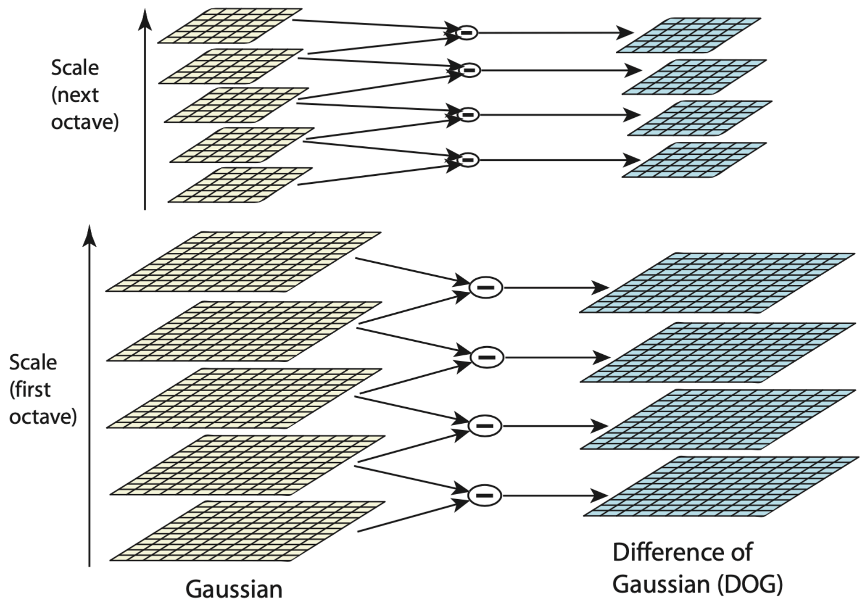
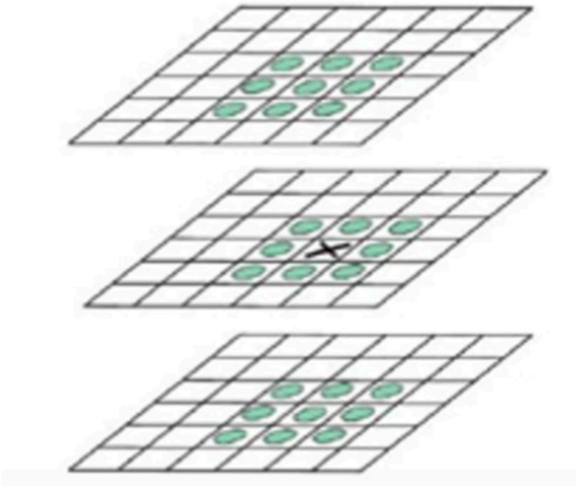


figure1: scale space for each layer

For the image be given we first to construct the first scale space with different Gaussian parameter, from the image effect it can be seen as different degree of Gaussian Blur for fixed size of image, which also called one octave. In the next octave, reduce the image size and do the same procedure, loop this procedure we can have a set of image which seems like a pyramid. In the next procedure calculate each layer and its neighbor's distance, which be given as the Difference of Gaussian. For the number of octave, the function recommended is using the log function of the images' wide or height, and minus 3.

The next part is find the key points, the extreme value we define as bigger than the 26 points around, this is the most hard part so I can only explain it with draft paper. It could be seen as image:



When find the key point, define it as *X(x,y,δ) , and use the Taylor series to transfer it.*

$$f\left(\begin{bmatrix}x\\y\\\sigma\end{bmatrix}\right)=f\left(\begin{bmatrix}x_0\\y_0\\\sigma_0\end{bmatrix}\right)+\left[\frac{\partial f}{\partial x},\frac{\partial f}{\partial y},\frac{\partial f}{\partial \sigma}\right]\left(\begin{bmatrix}x\\y\\\sigma\end{bmatrix}-\begin{bmatrix}x_0\\y_0\\\sigma_0\end{bmatrix}\right)$$

$$+\frac{1}{2}\left(\begin{bmatrix}x\\y\\\sigma\end{bmatrix}-\begin{bmatrix}x_0\\y_0\\\sigma_0\end{bmatrix}\right)^T\begin{bmatrix}\frac{\partial^2 f}{\partial x\partial x},\frac{\partial^2 f}{\partial x\partial y},\frac{\partial^2 f}{\partial x\partial \sigma}\\ \frac{\partial^2 f}{\partial x\partial y},\frac{\partial^2 f}{\partial y\partial y},\frac{\partial^2 f}{\partial y\partial \sigma}\\ \frac{\partial^2 f}{\partial x\partial \sigma},\frac{\partial^2 f}{\partial y\partial \sigma},\frac{\partial^2 f}{\partial \sigma\partial \sigma}\end{bmatrix}\left(\begin{bmatrix}x\\y\\\sigma\end{bmatrix}-\begin{bmatrix}x_0\\y_0\\\sigma_0\end{bmatrix}\right)$$

For the formula how to transfer see the draft below, the parameter T is variable which default given the value 0.04:

$$f(x) = f(x_0) + \frac{\partial f^T}{\partial x}\hat{x} + \frac{1}{2}\hat{x}^T\frac{\partial^2 f}{\partial x^2}\hat{x}$$

$$\frac{\partial f'(x)}{\partial x} = \frac{\partial f^T}{\partial x} + \frac{1}{2}\left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f^T}{\partial x^2}\right)\hat{x} = \frac{\partial f^T}{\partial x} + \frac{\partial^2 f}{\partial x^2}\hat{x}$$

let $\frac{\partial f(x)}{\partial x} = 0 \Rightarrow \hat{x} = \frac{\partial^2 f^{-1}}{\partial x^2}\frac{\partial f}{\partial x}$

$$f(x) = f(x_0) + \frac{\partial f^T}{\partial x}\hat{x} + \frac{1}{2}\left(-\frac{\partial^2 f^{-1}}{\partial x^2}\frac{\partial f}{\partial x}\right)^T\frac{\partial^2 f}{\partial x^2}\left(-\frac{\partial^2 f^{-1}}{\partial x^2}\frac{\partial f}{\partial x}\right)$$

$$= f(x_0) + \frac{\partial f^T}{\partial x}\hat{x} + \frac{1}{2}\frac{\partial f^T}{\partial x}\cdot\frac{\partial^2 f^{-T}}{\partial x^2}\cdot\frac{\partial^2 f}{\partial x^2}\cdot\frac{\partial^2 f^{-1}}{\partial x^2}\cdot\frac{\partial f}{\partial x}$$

$$= f(x_0) + \frac{1}{2}\frac{\partial f^T}{\partial x}\cdot\hat{x}$$

if $|f(x)| < \frac{T}{n}$ then split the point.

Then considering to reduce the edge effect, use the rule as follows, H(x, y) is define as Hessian matrix.

Reduce the Edge Effect

$$H(x,y) = \begin{bmatrix} D_{xx}(x,y) & D_{xy}(x,y) \\ D_{xy}(x,y) & D_{yy}(x,y) \end{bmatrix}$$

$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta$ when $\alpha > \beta$ and $\alpha = \gamma\beta$

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

if Det(H)<0. then split the point X

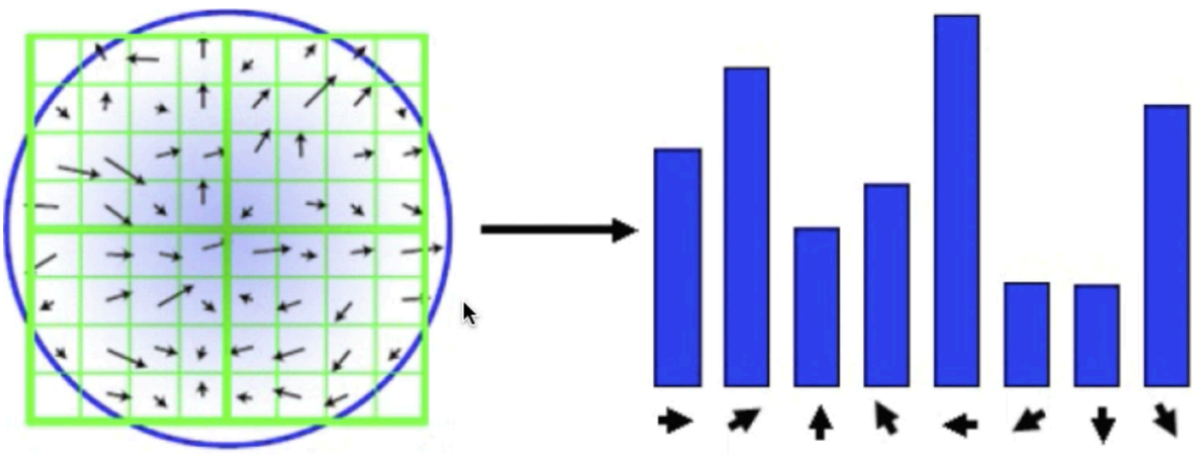
$$\frac{Tr(H)^2}{Det(H)} = \frac{(\alpha+\beta)^2}{\alpha\cdot\beta} = \frac{(\gamma\cdot\beta+\beta)^2}{\gamma\cdot\beta} = \frac{(\gamma+1)^2}{\gamma}$$

if $\frac{Tr(H)}{Det(H)} < \frac{(\gamma+1)^2}{\gamma}$ then split the point X.

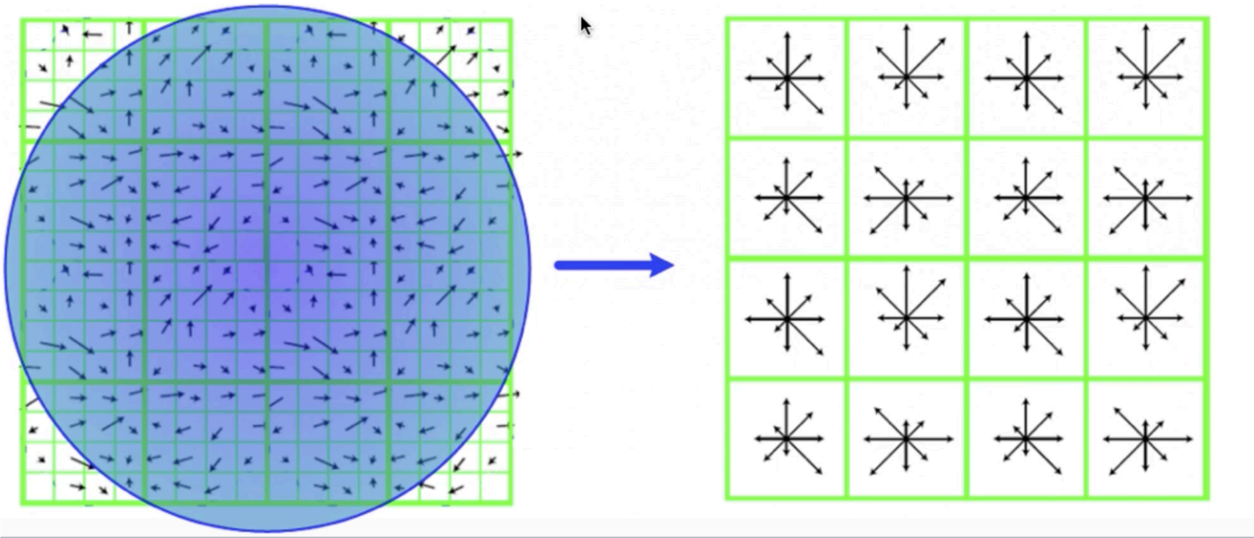
(reccomend $\gamma = 10.0$)

After defined all the key point then we should give the point main direction.

Take the key point as the center of the circle, and count 1.5 times the Gaussian image's every pixels 's gradient direction and amplitude value. For the direction， define 10 degree as one level.



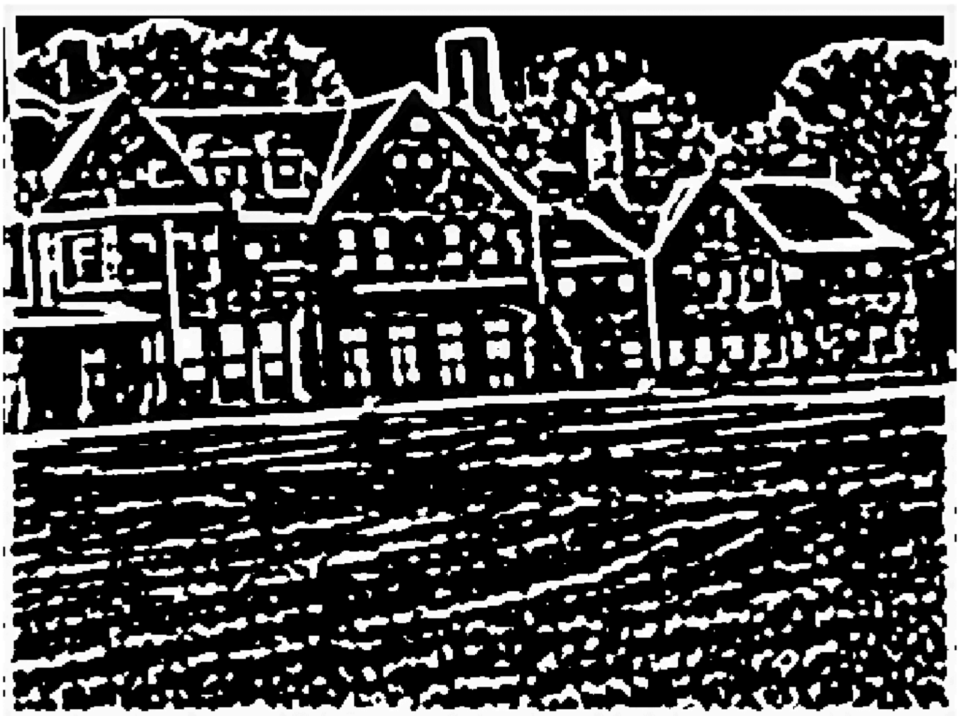
Define the symbol of every key point using KNN, split the image pixels as 4x4 parts, for each part calculating the sum in 8 directions with weight.



Here are all the procedures to find the key point and symbol by SIFT.

3. Running Result

Running the sift.py, with images number 3 and 4 as example, as Image from different octaves



Sift to find the feature point and match.

Appendix

How to calculate the derivative function in coding:

As the formula said the if we want calculate the f1-f3, we find the coordinate on the axis below and calculate with function.

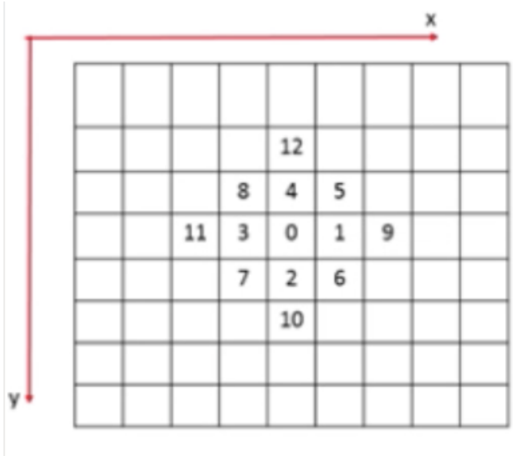
$$\left(\frac{\partial f}{\partial x}\right)=\frac{f_1-f_3}{2h} \qquad (1)$$

$$\left(\frac{\partial f}{\partial y}\right)=\frac{f_2-f_4}{2h} \qquad (2)$$

$$\left(\frac{\partial^2 f}{\partial x^2}\right)=\frac{f_1+f_3-2f_0}{h^2} \qquad (3)$$

$$\left(\frac{\partial^2 f}{\partial y^2}\right)=\frac{f_2+f_4-2f_0}{h^2} \qquad (4)$$

$$\left(\frac{\partial^2 f}{\partial x\partial y}\right)=\frac{(f_8+f_6)-(f_5+f_7)}{4h^2} \qquad (5)$$


x
y
12
8
4
5
11
3
0
1
9
7
2
6
10