Day 09

深度學習與電腦視覺

學習馬拉松

cupoy 陪跑專家：楊鎮銘

基礎影像處理

SIFT 的概念與實作

重要知識點

CUPOY

- 了解 SIFT 是基於甚麼觀念進行改進
- 了解 SIFT 演算法的物理意義

何謂特徵？

前面我們介紹了 Filter 的操作，了解計算圖片的導數就會取得邊緣
在電腦視覺中我們也可以把這個過程稱為「抽取特徵」

所謂特徵，可以非常直覺的解釋為「圖片中最特別的地方」
可以是邊緣，輪廓，紋理等資訊

SIFT 特徵 - 概念

本章節要介紹的 SIFT 就是其中一種表徵 (appearance feature)
基於局部的外觀特徵，進一步考慮到圖片特徵的狀況

- 縮放不變性，旋轉不變性
- 光線與雜訊容忍度高

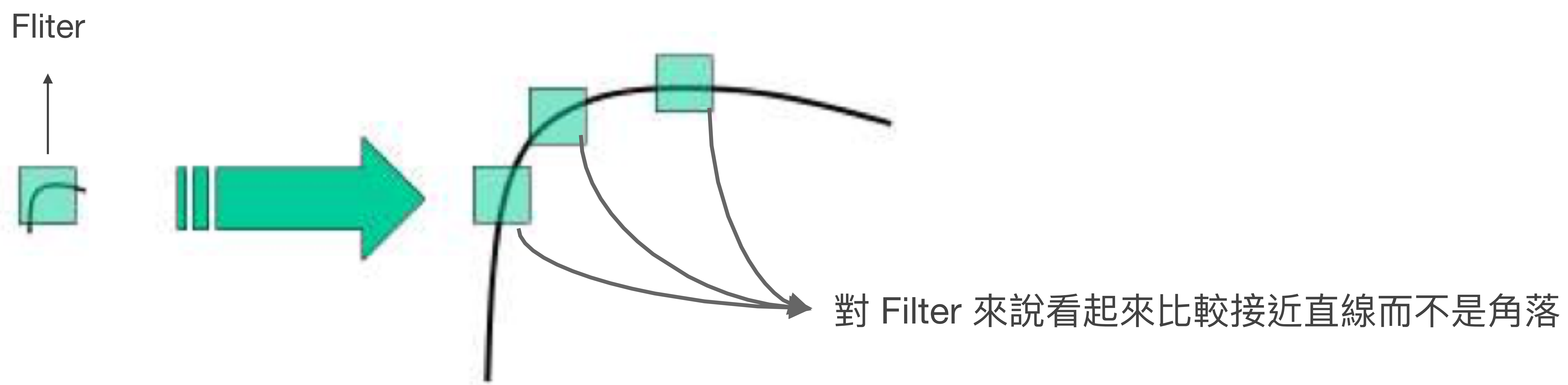
在 SIFT 演算法中可以了解到如何做關鍵點偵測
並如何抽出 SIFT 特徵來敘述關鍵點

右圖：示意該圖片的 SIFT 特徵

SIFT 特徵 - 尺度不變性

SIFT 主要考慮到的問題之一是尺度

以 corner detector (e.g. Harris) 為例，Filter 可以偵測到範圍以內的角落點，但是同樣的 pattern 放大後以同樣的 Filter 去偵測就會失敗



SIFT 特徵 - 尺度空間極值偵測

SIFT 會基於邊緣檢測抽取特徵，但不是使用前面提過的 Sobel 概念上是 LoG 但是考慮到計算量使用 DoG 做邊緣檢測

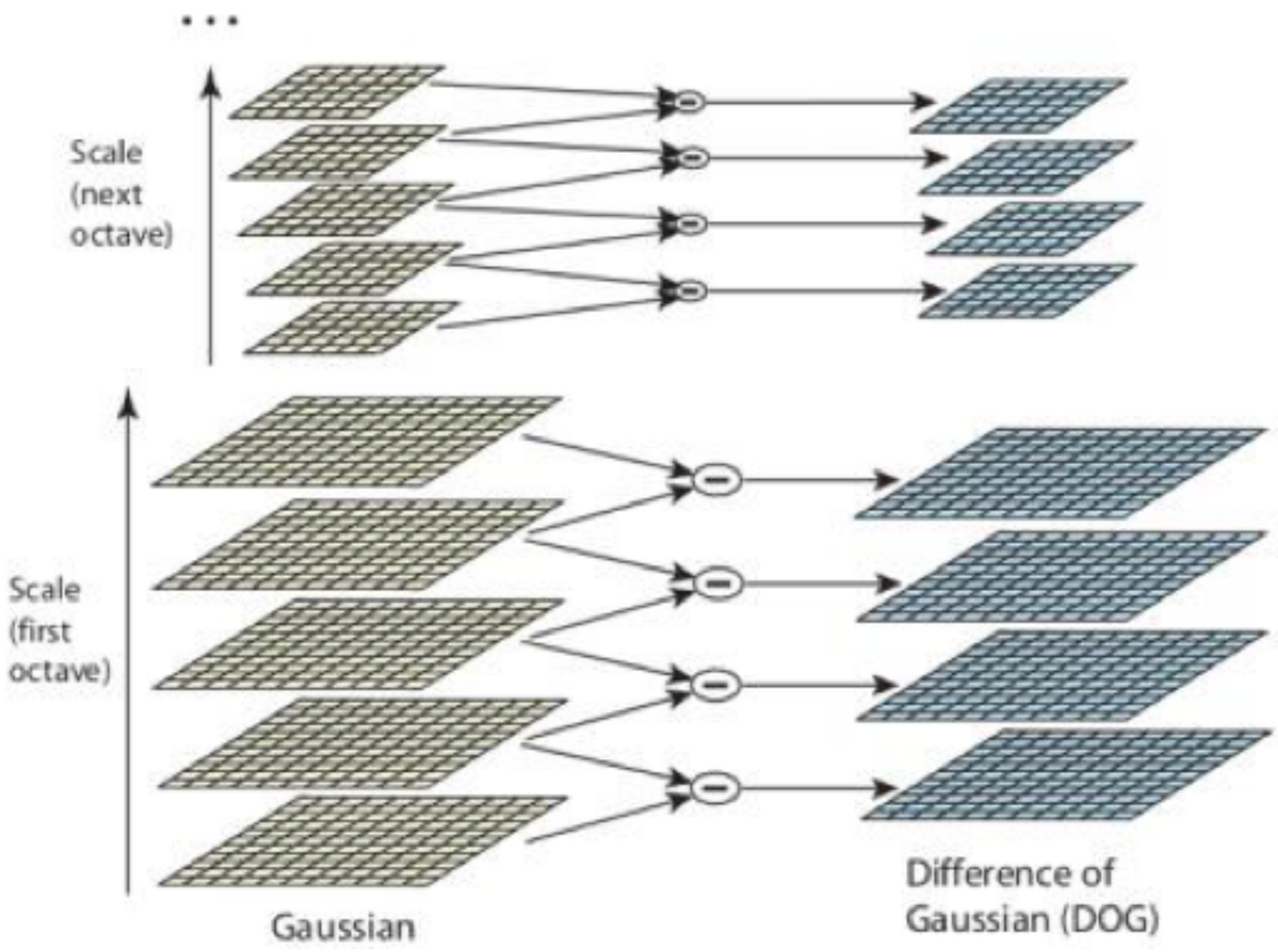
- Laplacian of Gaussian (LoG)
  - 先對圖片做 Gaussian Blur 再算二階導數取得邊緣
- Difference of Gaussian (DoG)
  - 圖片經過不同程度的縮放後計算出不同程度的 Gaussian Blur 最後合併得到一個 Gaussian Pyramid，其差值即為 DoG
  - 結果可以視為 LoG 的約略值 (沒有做二階導數)

這邊討論的特徵主要是物體的邊緣
而二階導數是個適合的工具來找出邊緣，因此這邊才會以此討論 LoG 與 DoG

SIFT 特徵 - 尺度空間極值偵測 (DoG 尺度)

DoG 詳解 (如右圖所示)

- 圖片的一種 scale 稱為一個 octave
- 每種 scale 的圖片經過不同程度的 Gaussian Blur 在計算其差值
- 最後會得到右圖最後的 DoG (Gaussian Pyramid)



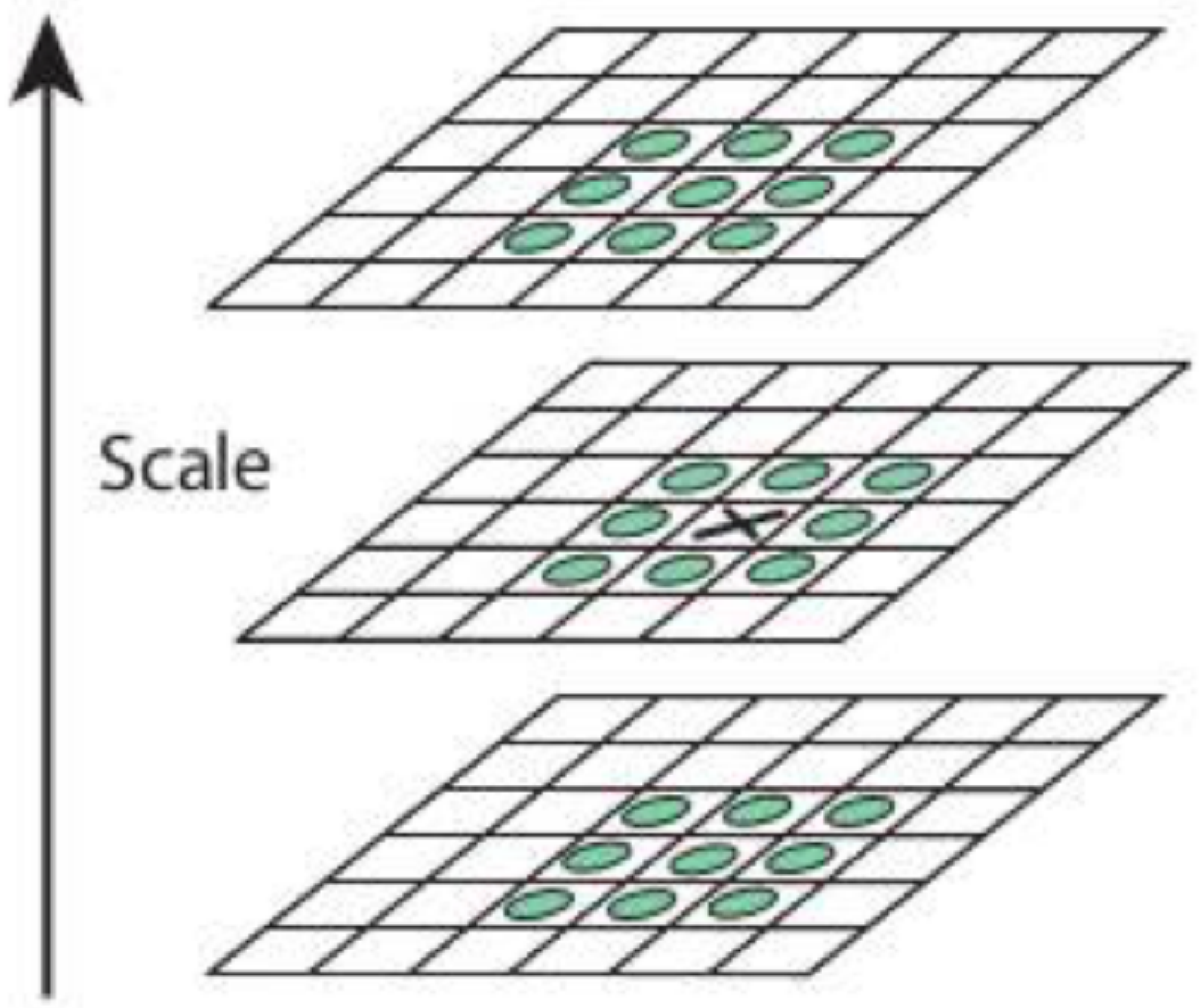
SIFT 特徵 - 尺度空間極值偵測 (極值偵測)

CUPOY

前面提到的 DoG 影像包含多種尺度
接著要針對每個 pixel 判斷是否為極值
判斷範圍 8+18 = 26

- 自己本身周遭的 8 個 pixel
- 同一個 scale 圖片但不同模糊尺度
  相鄰位置共 9*2=18 個 pixel

假如該 pixel 為判斷範圍內的最大 / 最小值
則將其設為有興趣的關鍵點



SIFT 特徵 - 關鍵點定位

經過多尺度極值偵測之後，會得到許多候選的關鍵點
其中也包含許多噪音跟邊的關鍵點
需要更進一步根據周遭資訊來修正並過濾關鍵點

- 鄰近資料差補
  - 主要根據相鄰資訊來修正極值的位置
- 過濾不明顯關鍵點
  - 根據計算曲率來判斷是否為不明顯的關鍵點
- 過濾邊緣關鍵點
  - 根據計算曲率來判斷是否為不明顯的關鍵點

Note: 這邊包含大量數學推導，不詳細解釋，如果有興趣可以參考最後的 reference

SIFT 特徵 - 方位定向

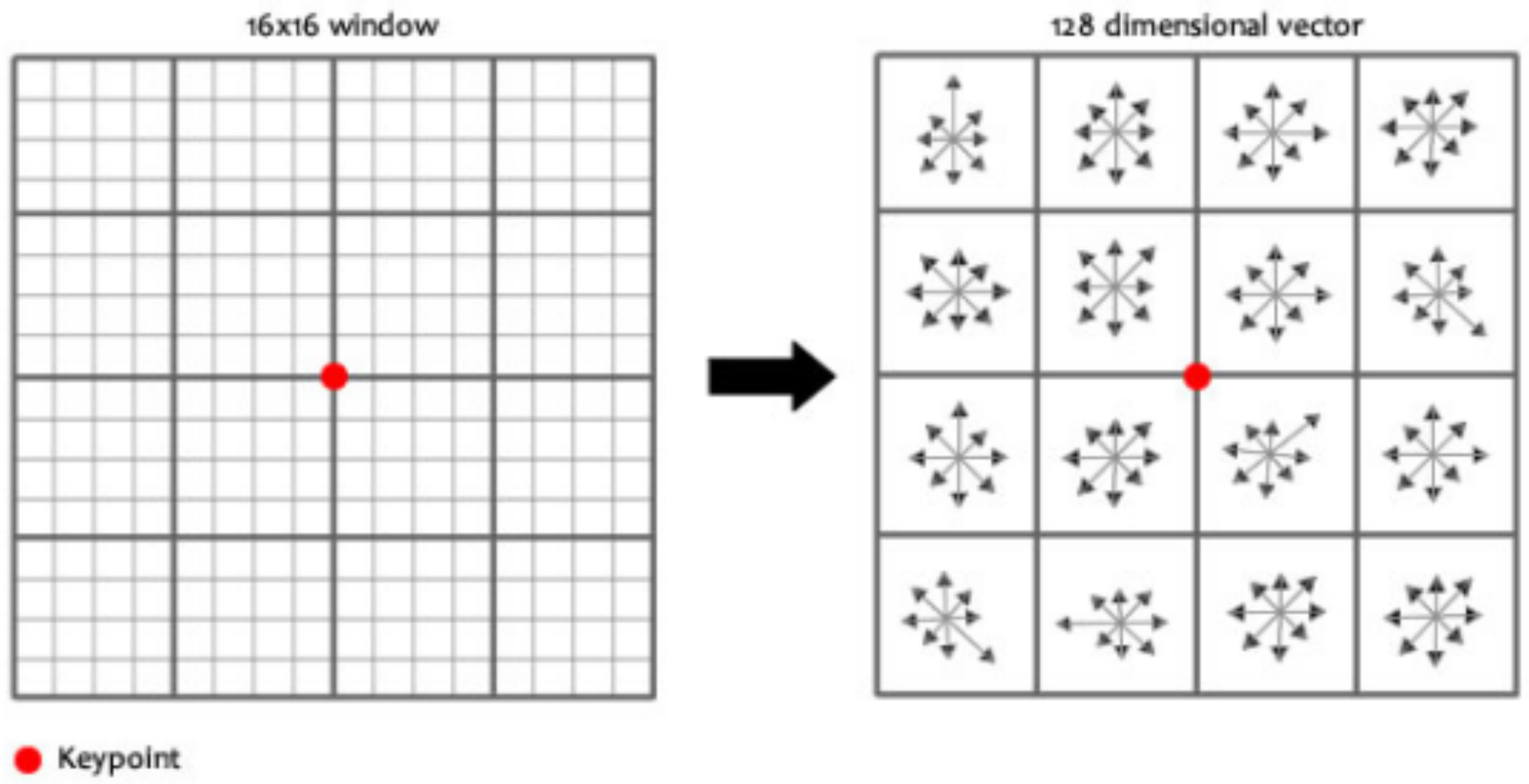
前面我們定義並過濾了許多關鍵點，但是關鍵點只有包含尺度跟位置
SIFT 還想要保有旋轉不變性，因此要給關鍵點定義一個方向

- 以每 10 度為單位計算周圍的梯度值
- 梯度值最大的方向當作是該關鍵點的主要方向

SIFT 特徵 - 關鍵點描述子

賦與關鍵點位置，尺度，方向確保移動，縮放，旋轉的不變性
還需要額外建立描述子來確保不同光線跟視角也有不變性

- 描述子會正規化成 128 維的特徵向量
- 以關鍵點周圍 16*16 的區域共 4*4 的子區域，計算 8 個方向的直方圖，共 4*4*8 = 128 維的特徵向量



Note：

每個關鍵點都會產生 128 維的特徵
而圖片會產生 N 個關鍵點，也就是會產生 (N, 128) 維度特徵

SIFT 實作

如果要透過 OpenCV 使用 SIFT 的話必須要額外安裝擴充的函式庫
為了避免版本問題，我們會指定安裝版本

```
pip install opencv-contrib-python==3.4.2.16
```

接著才有辦法使用 SIFT 來抽取特徵

```
sift = cv2.xfeatures2d.SIFT_create() # 建立 SIFT 物件
keypoints = sift.detect(img_gray, None) # 抽取關鍵點
img_show = cv2.drawKeypoints(img_gray, keypoints, img)
```

知識點回顧

- 了解 SIFT 演算法的物理意義
  - 從 LoG 變為 DoG 取得多尺度空間
  - 透過極值找候選的關鍵點
  - 透過主曲率過濾關鍵點
  - 透過區域劃分決定方向並正規化成 128 維向量

範例

- 實作取得 SIFT 特徵
  - 需要轉成灰階圖片
  - 需要額外安裝 OpenCV 相關套件

推薦延伸閱讀

原创 **SIFT特征提取分析**

2012-06-06 22:06:09 Rachel-Zhang 阅读数 534765 文章标签: 角点特征 SIFT 角点检测 更多

SIFT（Scale-invariant feature transform）是一种检测局部特征的算法，该算法通过求一幅图中的特征点（interest points,or corner points）及其有关scale 和 orientation 的描述子得到特征并进行图像特征点匹配，获得了良好效果，详细解析如下：

算法描述

SIFT特征不只具有尺度不变性，即使改变旋转角度，图像亮度或拍摄视角，仍然能够得到好的检测效果。整个算法分为以下几个部分：

1. 构建尺度空间

这是一个初始化操作，尺度空间理论目的是模拟图像数据的多尺度特征。

高斯卷积核是实现尺度变换的唯一线性核，于是一副二维图像的尺度空间定义为：

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

其中 G(x,y,σ) 是尺度可变高斯函数 $$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2}.$$

（x，y）是空间坐标，是尺度坐标。σ大小决定图像的平滑程度，大尺度对应图像的概貌特征，小尺度对应图像的细节特征。大的σ值对应粗糙尺度(低分辨率)，反之，对应精细尺度(高分辨率)。为了有效的在尺度空间检测到稳定的关键点，提出了高斯差分尺度空间（DOG scale-space）。利用不同尺度的高斯差分核与图像卷积生成。

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

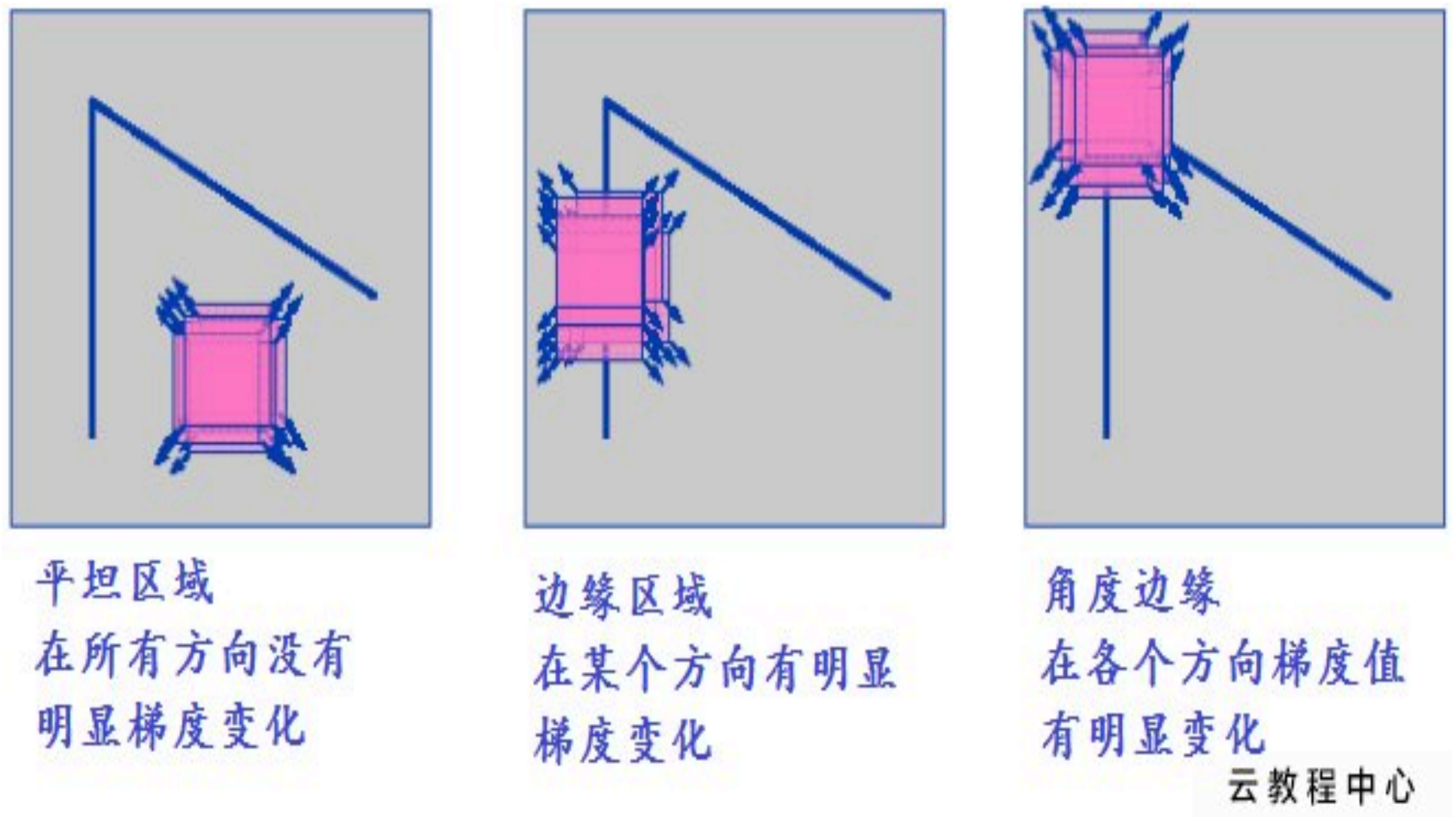
SIFT 特徵分析

部落格文章中有針對 SIFT 詳細數學的推導

連結

基本原理:

角點是一幅影象上最明顯與重要的特徵,對於一階導數而言,角點在各個方向的變化是最大的,而邊緣區域在只是某一方向有明顯變化。一個直觀的圖示如下:

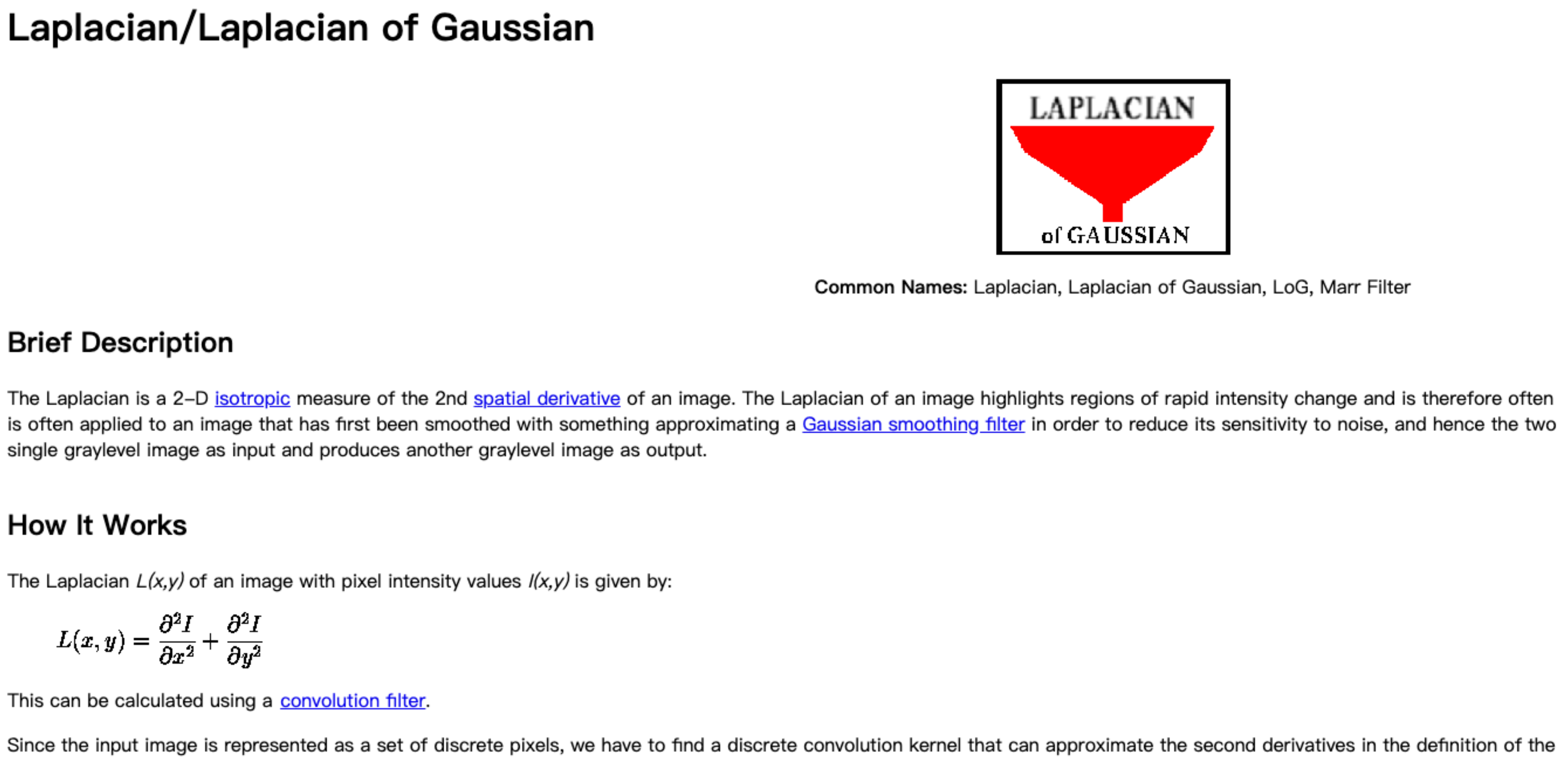


Harris detectXor

本章在討論尺度不變性時有提到用來偵測角落的 Harris detector，可以參考這個中文部落格，裏面包含數學推導

連結

推薦延伸閱讀



Laplacian/Laplacian of Gaussian

LAPLACIAN of GAUSSIAN

Common Names: Laplacian, Laplacian of Gaussian, LoG, Marr Filter

Brief Description

The Laplacian is a 2-D isotropic measure of the 2nd spatial derivative of an image. The Laplacian of an image highlights regions of rapid intensity change and is therefore often applied to an image that has first been smoothed with something approximating a Gaussian smoothing filter in order to reduce its sensitivity to noise, and hence the two single graylevel image as input and produces another graylevel image as output.

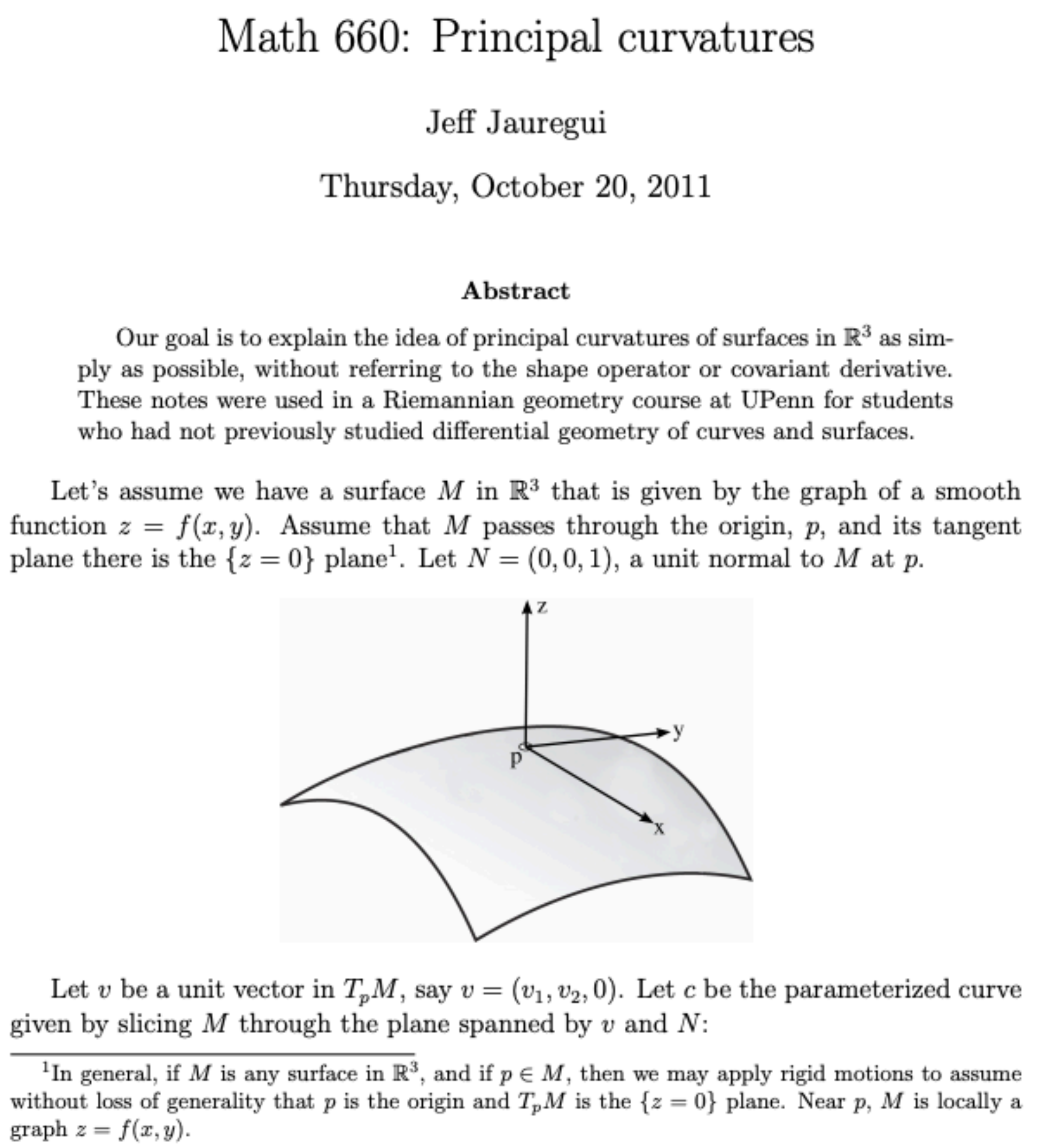
How It Works

The Laplacian *L(x,y)* of an image with pixel intensity values *I(x,y)* is given by:

$$L(x,y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

This can be calculated using a convolution filter.

Since the input image is represented as a set of discrete pixels, we have to find a discrete convolution kernel that can approximate the second derivatives in the definition of the



Math 660: Principal curvatures

Jeff Jauregui

Thursday, October 20, 2011

Abstract

Our goal is to explain the idea of principal curvatures of surfaces in $\mathbb{R}^3$ as simply as possible, without referring to the shape operator or covariant derivative. These notes were used in a Riemannian geometry course at UPenn for students who had not previously studied differential geometry of curves and surfaces.

Let's assume we have a surface M in $\mathbb{R}^3$ that is given by the graph of a smooth function $z = f(x,y)$. Assume that M passes through the origin, p, and its tangent plane there is the $\{z=0\}$ plane[1]. Let $N = (0,0,1)$, a unit normal to M at p.

Let v be a unit vector in T_pM, say $v = (v_1, v_2, 0)$. Let c be the parameterized curve given by slicing M through the plane spanned by v and N:

[1] In general, if M is any surface in $\mathbb{R}^3$, and if $p \in M$, then we may apply rigid motions to assume without loss of generality that p is the origin and T_pM is the $\{z=0\}$ plane. Near p, M is locally a graph $z = f(x,y)$.

Laplacian of Gaussian

本章在介紹 DoG 的時候有提到 LoG，有興趣可以參考這個部落格文章，包含數學推導跟一些圖可以幫助理解

連結

主曲率計算

本章在介紹 DoG 的時候有提到 LoG，有興趣可以參考這個部落格文章，包含數學推導跟一些圖可以幫助理解

連結

解題時間 Let’s Crack It

CUPOY

請跳出 PDF 至官網 Sample Code & 作業開始解題