Day 10

深度學習與電腦視覺

學習馬拉松

cupoy 陪跑專家：楊鎮銘

CUPOY

基礎影像處理

SIFT 的應用

重要知識點

CUPOY

- 了解圖片抽取特徵後的泛用性
- 了解如何透過特徵去配對
  - 透過距離來判斷特徵的相似程度
    - 配對演算法

特徵應用

CUPOY

當我們取得特徵之後，就代表我們已經有能力去識別關鍵點的特殊性

在這之後可以接到許多電腦視覺的任務

- 配對：判斷兩張圖片上相同物體的位置
- 辨識：判斷兩張圖片上是否有相同物體
- 全景圖：尋找兩張圖片的相同視角，再經過轉換合成全景圖
- ...

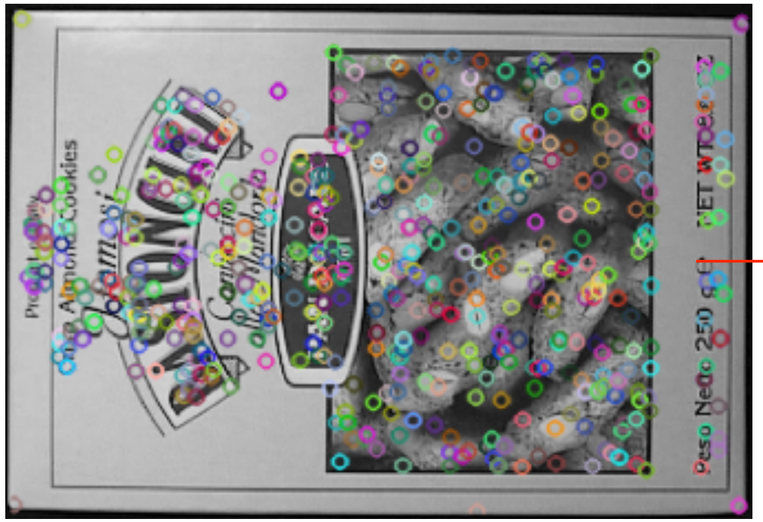
廣泛的說，SIFT 只是其中一種抽取特徵的方式

這邊會延續上一章節以 SIFT 為例介紹配對的應用

特徵配對 Feature Matching - 任務目標

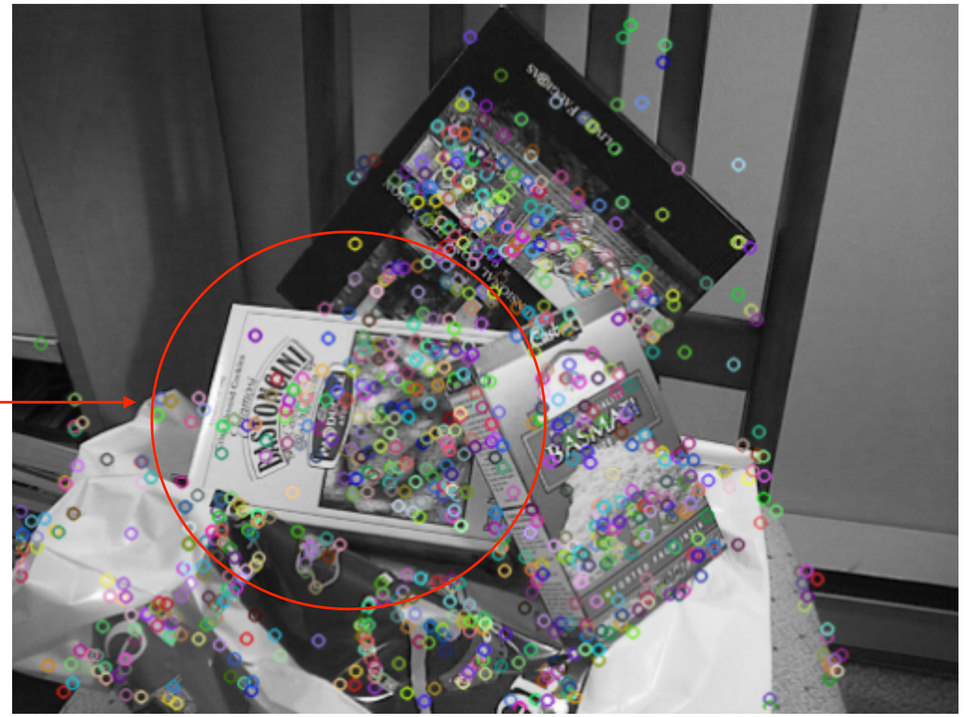
CUPOY

首先配對會有兩張圖片，其中一張是 query image
我們會各自在兩張上面抽取 SIFT 特徵



Query image

目標



SIFT 特徵 - 尺度不變性

CUPOY

配對會從兩張圖片中的關鍵點中，透過計算其特徵空間上的距離，
若小於一個設定的閥值就視為是相同的特徵
在 SIFT 特徵的配對任務中，通常會使用 L2 norm 的方式計算

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2}$$

兩個 128 維向量根據上面公式計算可以得到一個距離

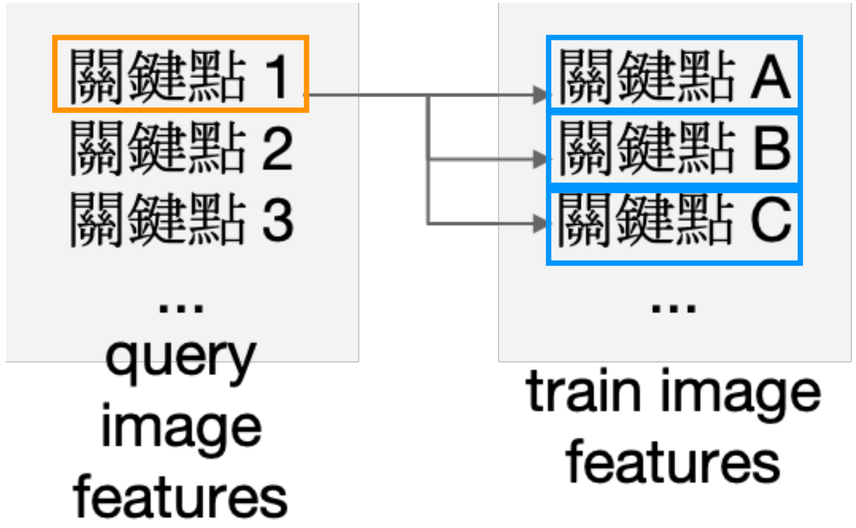
SIFT 特徵 - 尺度空間極值偵測

CUPOY

簡單暴力的配對方法是逐一針對 query image 的關鍵點
對每個 train image 的關鍵點計算 L2 距離

- 取得距離最小的配對
- 取得 k 個最適合的配對

這邊為了確保配對的合適性
可以先在計算時取得 k 個配對
在根據距離去過濾不適合的配對

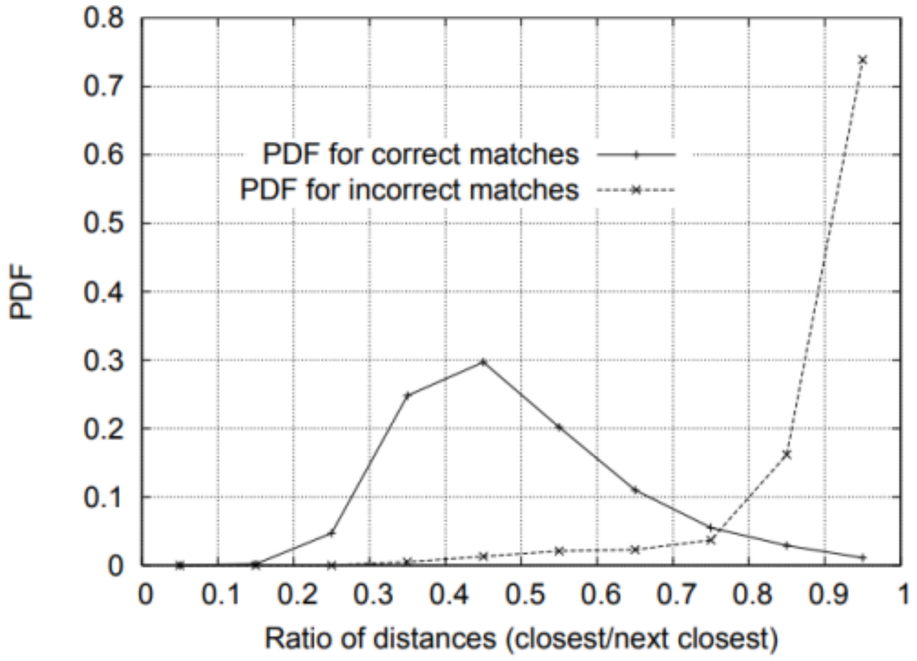


特徵配對 Feature Matching - ratio test

CUPOY

我們可以尋找 k=2 個最好的 match 方式，透過 ratio test 的方式來過濾一些不適當的配對，因為有時候 query 的關鍵點並不會出現在 train image

根據補充資料的論文提到
建議比值設定在 0.7~0.8 比較好



特徵配對 Feature Matching - 實作

CUPOY

```
# 另外一種宣告 SIFT 的方式
sift = cv2.xfeature2d_SIFT.create()

# 取得關鍵點的同時也計算 128 維敘述子向量
kp1, des1 = sift.detectAndCompute(img_query, None)
kp2, des2 = …

# 建立 Brute-Force Matching 物件
bf = cv2.BFMatcher(cv2.NORM_L2)
```

mask 參數，如果有設定的話可以
針對部份圖片計算 SIFT

第一個是 query image
第二個是 train image

宣告要使用 L2 norm 計算距離

特徵配對 Feature Matching - 實作

CUPOY

```
# 以 knn 方式暴力比對特徵
matches = bf.knnMatch(des1, des2, k=2)

# 透過 D.Lowe ratio test 排除不適合的配對
candidate = []
for m, n in matches:
    if m.distance < 0.75*n.distance:
        candidate.append([m])
```

ratio test 設定比值為 0.75

特徵配對 Feature Matching - 實作

CUPOY

```
# 顯示配對結果
img_show = cv2.drawMatchesKnn(
    img_query,
    kp1,
    img_train,
    kp2,
    candidate,
    None,
    flags=2)
```

如果看 OpenCV 官方文件這邊可以不用輸入，
但是在 Python 版本沒有預設值，
沒有給定 None 會有錯誤

畫圖的配置，flags=2 代表沒有配對成功的關鍵點不會被畫出來，可以看文件了解其他設置

SIFT 在機器學習上的應用 (optional)

CUPOY

許多機器學習的任務一開始都要先經過抽取特徵的步驟
諸如 SIFT 等傳統電腦視覺的特徵只是其中一種方式
而近期非常熱門的深度學習則是另外一種抽取特徵的方式

SIFT → 分類器 → 結果

MODEL → 分類器 → 結果

SIFT 在機器學習上的應用 (optional)

CUPOY

後面再根據任務類型選擇要對特徵做甚麼處理
所以傳統特徵跟 model 抽的特徵，使用上是差不多的
e.g. 分類任務，我們就把抽完的特徵當作 input 輸入分類器

SIFT → 分類器 → 結果

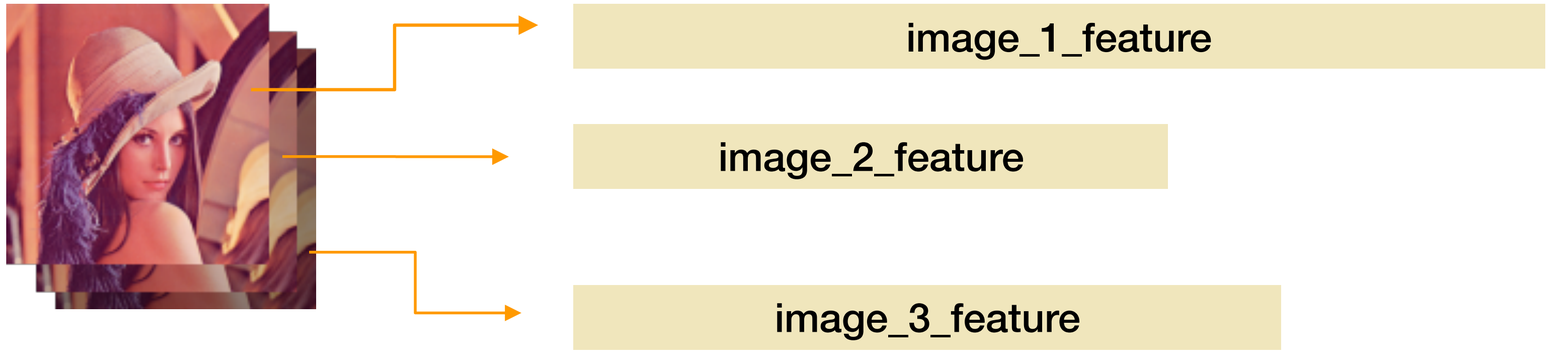
MODEL → 分類器 → 結果

SIFT 在機器學習上的應用 (optional)

CUPOY

SIFT 雖然可以做機器學習任務，但是實作上存在一些問題
因為演算法的關係，不保證所有圖片都會產生一樣維度的特徵

一般機器學習任務的 input 都要是同樣的維度
因此 SIFT 特徵必須做前處理



SIFT 在應用上的問題 (optional)

CUPOY

- SIFT 每一個特徵點的維度其實是一樣的，但每張圖片產生的特徵點個數不同，才會導致圖片的特徵維度不同
- 其中一種作法是做 Clustering，每一張圖片都取 n 個特徵點來固定圖片的特徵維度

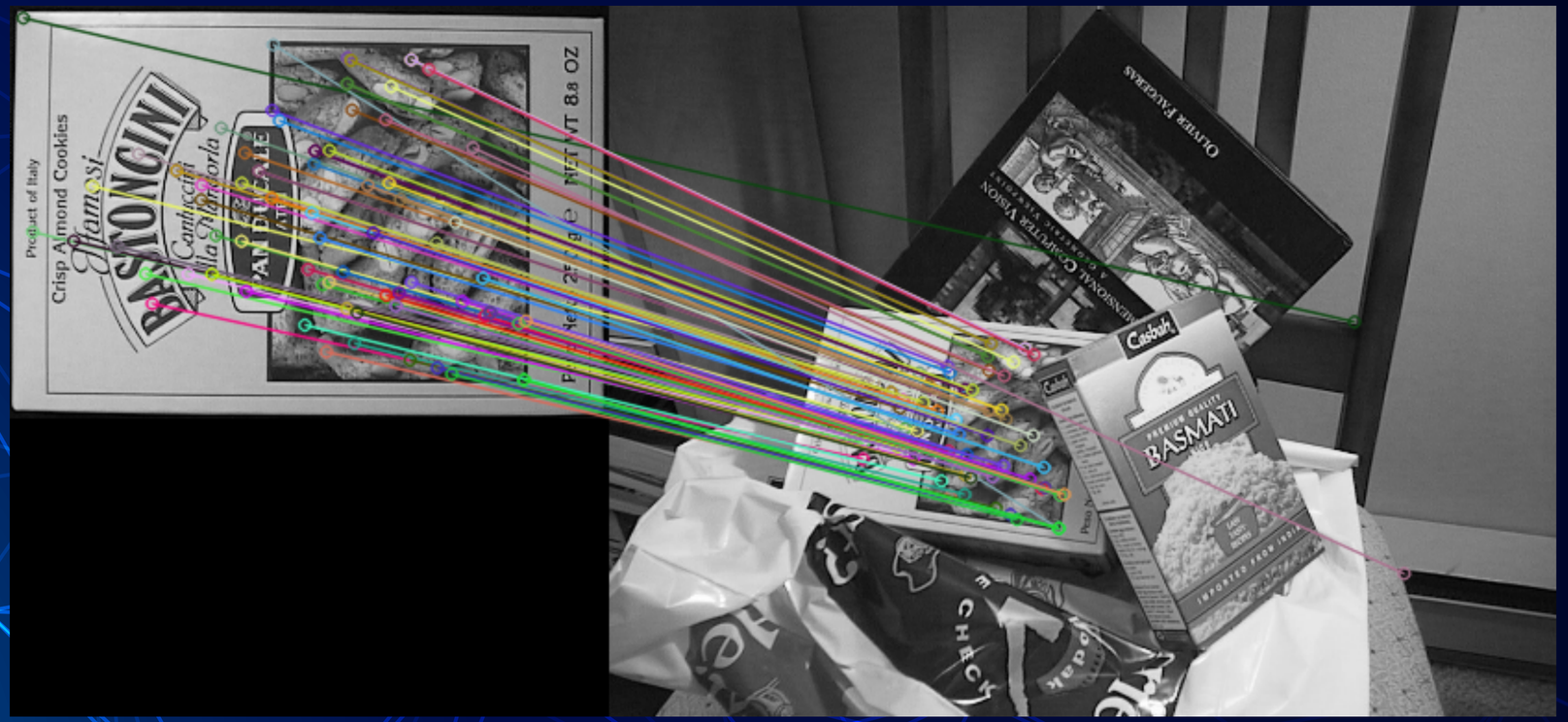
缺點：如果圖片太簡單導致部份圖片特徵太少就會失效，所以類似 MNIST 或是 CIFAR 等簡單的資料集就不太適合

知識點回顧

- 了解特徵在電腦視覺的泛用性
  - 若有更適合的特徵可以替換 SIFT 直接接後面的流程
- 了解特徵在配對任務中扮演的角色
  - 在多種電腦視覺任務中也很常出現透過距離來判斷兩個向量的相似度，這邊的物理意義就是判斷是否為相同特徵

範例

透過 SIFT 特徵實作 Brute-Force Matching



推薦延伸閱讀

CUPOY

Math 660: Principal curvatures

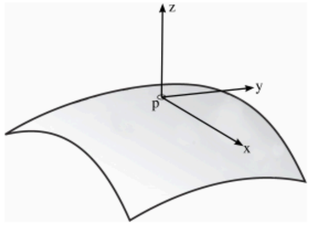
Jeff Jauregui

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 $z = f(x, y)$.

Distinctive Image Features from Scale-Invariant Keypoints

原論文 Section 7.1 裏面提到配對任務中 ratio test 目的

連結

Feature Matching

Goal

In this chapter

- We will see how to match features in one image with others.
- We will use the Brute-Force matcher and FLANN Matcher in OpenCV

Basics of Brute-Force Matcher

Brute-Force matcher is simple. It takes the descriptor of one feature in first set and is matched with all other features in second set using some distance calculation. And the closest one is returned.

For BF matcher, first we have to create the BFMatcher object using **cv2.BFMatcher()**. It takes two optional params. First one is `normType`. It specifies the distance measurement to be used. By default, it is `cv2.NORM_L2`. It is good for SIFT, SURF etc (`cv2.NORM_L1` is also there). For binary string based descriptors like ORB, BRIEF, BRISK etc, `cv2.NORM_HAMMING` should be used, which used Hamming distance as measurement. If ORB is using `WTA_K == 3 or 4`, `cv2.NORM_HAMMING2` should be used.

Second param is boolean variable, `crossCheck` which is false by default. If it is true, Matcher returns only those matches with value (i,j) such that i-th descriptor in set A has j-th descriptor in set B as the best match and vice-versa. That is, the two features in both sets should match each other. It provides consistant result, and is a good alternative to ratio test proposed by D.Lowe in SIFT paper.

Once it is created, two important methods are *BFMatcher.match()* and *BFMatcher.knnMatch()*. First one returns the best match. Second method returns *k* best matches where k is specified by the user. It may be useful when we need to do additional work on that.

Like we used cv2.drawKeypoints() to draw keypoints, **cv2.drawMatches()** helps us to draw the matches. It stacks two images horizontally and draw lines from first image to second image showing best matches. There is also **cv2.drawMatchesKnn** which draws all the k best matches. If k=2, it will draw two match-lines for each keypoint. So we have to pass a mask if we want to selectively draw it.

Let's see one example for each of SURF and ORB (Both use different distance measurements).

Brute-Force Matching with ORB Descriptors

Here, we will see a simple example on how to match features between two images. In this case, I have a queryImage and a trainImage. We will try to find the queryImage in trainImage using feature matching. (The images are `/samples/c/box.png` and `/samples/c/box_in_scene.png`)

We are using SIFT descriptors to match features. So let's start with loading images, finding descriptors etc.

OpenCV

Quick search

Go

Table Of Contents

Previous topic

ORB (Oriented FAST and Rotated BRIEF)

Next topic

Feature Matching + Homography to find Objects

OpenCV - Feature Matching

官方文件有其他 matching 方式的介紹，
也有介紹不同的特徵適用不同的方式來計算距離

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