國立臺灣師範大學機電工程學系

碩士論文

指導教授:陳美勇博士

使用邊緣偵測和特徵偵測結合之移動物體偵測

Using Edge Detection Combined with Feature Detection for Moving Object Detection

研究生 ： 邱敬棋 撰

中 華 民 國 1 0 6 年 8 月

摘要

本文是針對影像作移動物體偵測。現今有非常多的方式對視訊監控影像作移動偵測的方法,在物體移動中,大部分最常見的方法是對物體找出特定的特徵點，並在兩張影像中計算此特徵點的移動，但有時這些特徵點有時候會較難被定義清楚，因為物體移動的時候容易使影像模糊，特別是在影像無法事先得知的情況下更為困難。

在本文主要是使用加速穩健特徵(SURF)演算法來定義移動物件的特徵點，因為 SURF 偵測特徵的速度相對於 SIFT 來說會比較快,但是不管是 SIFT 還是 SURF，在檢測的物體移動時，匹配結果則不如預期中良好，因為物體在移動時可能存在著不正確的特徵點，所以本文提出了邊緣和特徵偵測去作結合，以此提高特徵匹配的情況，除此之外本研究中我們各種不同的移動方式做偵測去計算正確的特徵點並做分析。在實驗中，我們可以進一步的了解此方法相較於傳統的方法上，能有更良好的辨識能力。

關鍵字：SURF、SIFT、邊緣偵測

Abstract

This thesis is detecting object for moving images. Nowadays, there are many methods for moving object detection on surveillance, and the method used is to find features and then to use the motion of those features between images to calculate features points moving. But the feature points sometimes are more difficult to define because the objects moving are easy to make images blur. Especially, when the objects may not be known in advance.

In this thesis, using SURF algorithm defines the features of motional images because it detecting speed is faster than SIFT. But whether it is SIFT or SURF when the detected object moves, the matching result is not as good as expected because the objects may have incorrect feature points on moving. In the thesis, we provide edge and feature detection to combine for increasing the feature matching. In addition, this study we use a lot of different detection to detect and calculate the correct feature points to analyze. In experiment, we can further understand our methods getting the better ability to identify compared to the traditional methods.

Keywords: SURF, SIFT, edge detection

誌謝

經過這幾年的碩士生涯將要結束了，首先，我要感謝我的指導教授陳美勇教授，在進入碩士後學到了不僅是在研究上，甚至是在品德上都可以有很多的幫助，也謝謝教授可以讓我這樣研究上可以不斷嘗試新的東西，也在我迷惘的時候給我一些方向讓我去努力，當然也要感謝口試委員鍾國亮博士、黃正民博士以及陳彥霖博士，針對論文描述不足之處給予指正，讓我論文本身不足的地方再去做糾正和加強。

此外，也要謝謝實驗室的成員秉剛、智翔、玠虢、昀翰、威任、維斌、廷恩、岱桓、梓瑄和庭剛，在做實驗的時候給了我一些幫助和鼓勵我，當然也要感謝我們最努力上班工作的 JuJu 了，一天到晚都跟聽我說抱怨的話，也適時給我建議解決一些研究以外的事情，還有也要感謝文彬，從大學的時候就幫助我很多的學弟，謝謝你。

最後，我還要感謝我的家人，爸爸、媽媽和姐姐，給我在煩惱的時候可以適時的鼓勵我且解決我生活各式各樣的煩惱，讓我沒有後顧之憂專心做研究，還要謝謝一些陪我玩攝影的好夥伴，謝謝你們陪我散心，也謝謝 Christina Nana 每天都帶給我歡笑。沒有大家的幫助可能沒有現在的結果，謝謝大家。

邱敬棋　謹誌

民國一零六年八月

精密運動實驗室

目錄

圖目錄

表目錄

第一章 緒論

1.1 前言

現代的科技蓬勃發展，機器人逐漸地開始取代人力。而在電腦視覺上，也隨著電腦的運行速度越來越快，讓複雜的運算時間可以縮短，使攝影機在偵測物體影像方面有了極大的突破，所以近幾年來，越來越多的學者提出了新的演算法來提升偵測物體的速度和精確度。

在目前的科技上，攝影機已經安裝在諸多如輪式移動機器人，機器人手臂和控制平台等多種機器人上，以便捕捉物體或避障。因此物體檢測變得越來越重要了，而在本論文提出了一個邊緣強化和演算法的結合用於偵測動態影像，因為在物體移動的時候相對來說特徵的數量會減少，所以本文藉由邊緣強化的方法讓動態的影像邊緣更加清晰以達到增加特徵數量的目標。

1.2 文獻回顧

本小節將介紹與本文相關的文獻敘述,包括:平滑影像、邊緣偵測、紋理偵測、銳化影像、特徵偵測……等

1.2.1 平滑影像

這個作用主要是用來處理雜訊，在一些物理現象或是解析度的限制情況下，容易讓影像出現雜訊，造成研究的結果不易分析，所以需使用濾波器來濾除多餘的雜訊，已達後續的分析方便，本文之後將會分別介紹均值濾波、中值濾波和高斯濾波。

首先我們要介紹卷積(Convolution)，對一個影像進行濾波需透過卷積，這方法廣泛運用在訊號處理上，但計算過程較為複雜，而在影像上的邏輯計算相對簡單，而且這不只使用在濾波上面，還使用包含邊緣偵測、角點偵測都會使用的硬像處理方法。

卷積計算的方式需要提供一個遮罩，使用這個遮罩在影像中由上往下、左往

右分別計算出每個像素卷積後的結果，且卷積的結果會儲存進另一個影像裡，不會覆寫在原始的影像上。

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

圖 1-1 卷積遮罩

1.2.2 邊緣檢測

邊緣檢測主要是基於影像的一階和二階導數，但二階導數通常對雜訊會比較敏感，因此在做邊緣檢測的時候，必須先用濾波器來改善與雜訊有關的干擾。本篇論文會初步介紹 Canny[5]偵測、Sobel 偵測、Scharr 運算元，至於本篇所以用的 Laplace 偵測則會在後面的原理做詳細的介紹。

Canny 邊緣檢測是由 John Canny 所提出了一個邊緣檢測法[1]。Canny 對於邊緣的檢測非常的敏感，只要有輕微的變化就會被當作邊緣偵測出來。

Canny 的檢測步驟如下:

a. 首先去除雜訊:使用高斯濾波，如 1-1 式

$$G_\sigma = \frac{e^{-(x^2+y^2)/4\sigma}}{\sqrt{\sigma}} \tag{1-1}$$

其中 σ 為濾波平滑因子。

b. 計算梯度大小和方向:以 Sobel 運算計算垂直與水平梯度。

c. 非極大值刪除: 假設邊緣點在其他梯度方向相比應該是最大值，但如果此點不是最大值則視為非邊緣點。

d. 閾值邊界:Canny 需要使用兩個閾值分別稱為高閾值T_H和低閾值T_L，假設影像像素為 I，若滿足$I > T_H$則該點視為邊緣點，若該像素滿足$T_H > I > T_L$則該點也可視為邊緣點。

(a)

(b)

(c)

圖 1-2 (a)原圖 (b)canny 偵測 (c)灰階化 canny 偵測

Sobel 運算元主要適用於邊緣檢測的離散微分運算元，有分成水平和垂直方向(G_x與G_y)，G_x是來檢測垂直邊緣，G_y是來檢測水平邊緣，如圖 1-4。

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{1-2}$$

$$G_y = \begin{bmatrix} -1 & -2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{1-3}$$

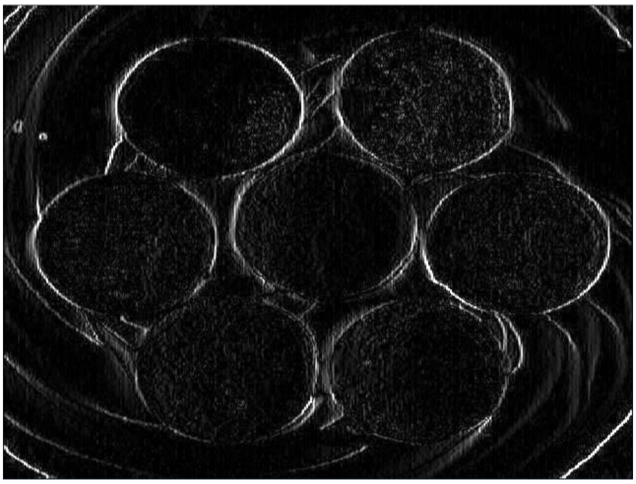
最後綜合以上兩個結果求出近似梯度:

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{1-4}$$

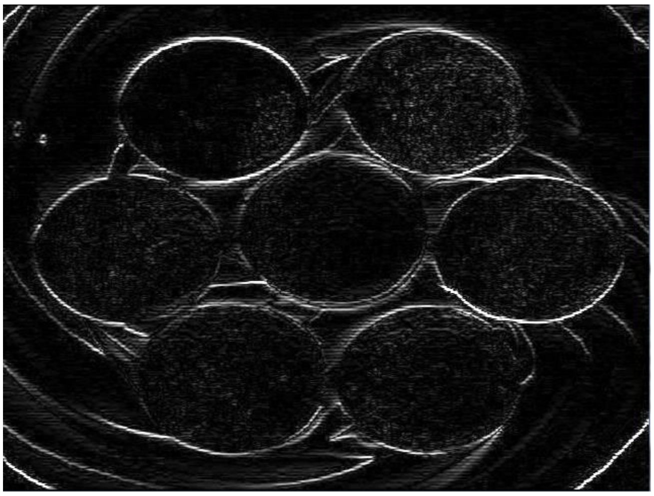
因為用(1-4)式計算太繁雜，所以提出了(1-5)式

$$G = |G_x| + |G_y| \tag{1-5}$$

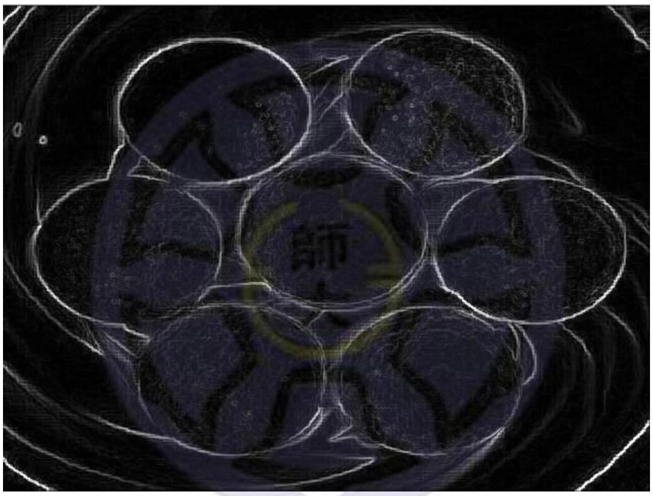
(a)



(b)



(c)



(d)

圖 1-3 (a)原圖 (b)x 方向 sobel (c)y 方向 sobel (d)整理 sobel

1.2.3 特徵偵測

在電腦視覺中，識別匹配、影像檢索和三維重建等都是很重要的部分。本小節主要是介紹識別部分的應用。

Penghui Jiang, Shengjie Zhao, Samuel Cheng [2]，提出了一種改進的 Haar 小波和旋轉不變局部二進制模式(LBP)[9-12]的圖像匹配算法結合。首先，應用 Hessian 矩陣和積分圖像，從圖像中提取關鍵點進行匹配。第二，對於每個關鍵點由旋轉不變的 LBP 模式和 Haar 小波來描述，它們以關鍵點為中心去做計算。最後，通

過使用匹配策略取最近鄰距離來確定兩組關鍵點之間的匹配。

這裡比較跟一般 SURF 的差異,就是在描述子經過 Harr 小波後多加上了旋轉 LBP，下列介紹一下旋轉 LBP 特徵計算步驟:

第一步: 每個像素以 P_j 為中心 ，並計算在 8×8 範圍內的旋轉不變 LBP 特徵，如圖 1-5。

第二步:像素 P_j 離中心 P_i 越遠，所貢獻出的描述子 P_i 越小。因此需要權重加值。

$$w_j = \frac{\exp\left(-\left[(r_j - r_i)^2 + (c_j - c_i)^2\right]\right)}{2\sigma^2 2\pi\sigma^2} \tag{1-15}$$

(r_j, c_j) 和 (r_i, c_i) 是指在圖片中對像素 P_j 和中心點 P_i 的點，π 是關鍵點，σ 是關鍵點的方向並會選擇使用常數。

第三步:計算權重 LBP 特徵在一維向量 T_i

$$T_i = \left[w_1 l_1, w_2 l_2, w_3 l_3, ..., w_n l_n\right] \tag{1-15}$$

第四步: 為了消除照明變化的影響，讓 T_i 正規化

$$\frac{T_i}{\|T_i\|} \to T_i \tag{1-15}$$

向量 T_i 使用不變尺度的關鍵點的方向，得到描述子旋轉不變性，且使用描述向量正規化來獲得對照明變化的強健性。

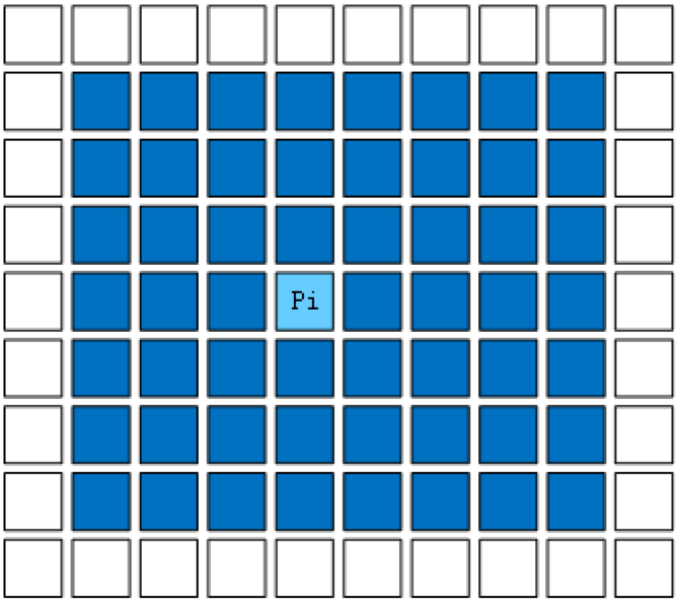
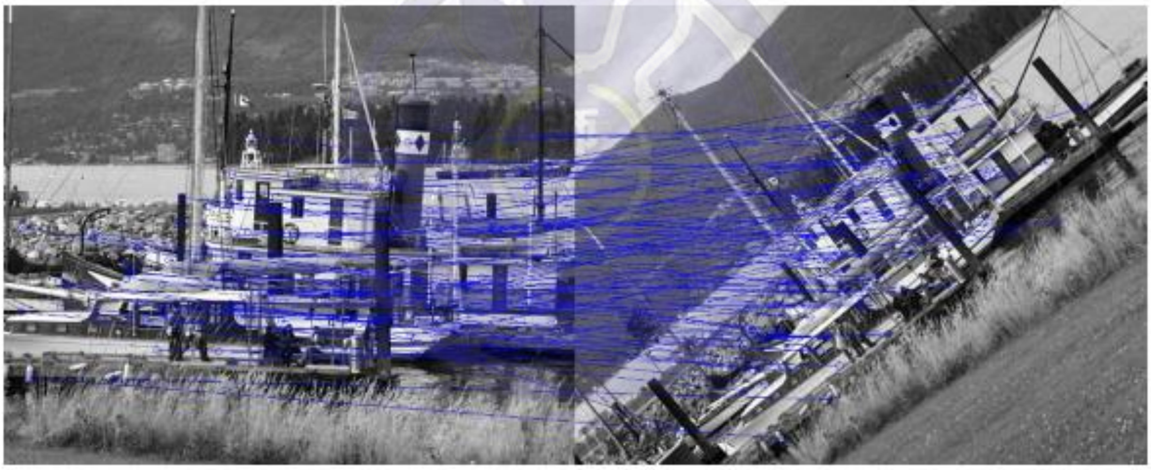
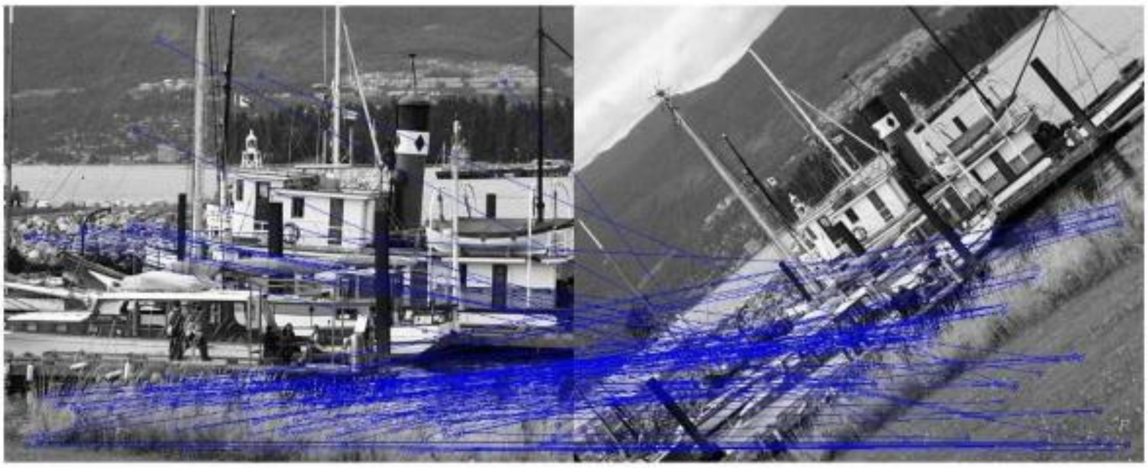


圖 1-4 關鍵點的區域



（a）



(b)

圖 1-5 (a)SURF (b)它的方法

Luo Juan 和 Oubong Gwun[3]是將 SIFT、SURF 和 PCA-SIFT 使用 KNN(K-Nearest Neighbor)進行特徵點匹配，再利用 RANSAC 進行過濾離群值的點。主要是在比較這三種方法的效果，SIFT 在旋轉和尺度改變的情況下表現最好，SURF 則是三個中最快且在有光暗變化的情況下有較高的匹配程度。

表 1-1 比較這三種方法所需時間

比較內容	SIFT	PCA-SIFT	SURF
總匹配數	271	18	186
總時數(ms)	21537800	21396900	3362.86
10 個匹配所需時間(ms)	21480600	20969600	3304.97

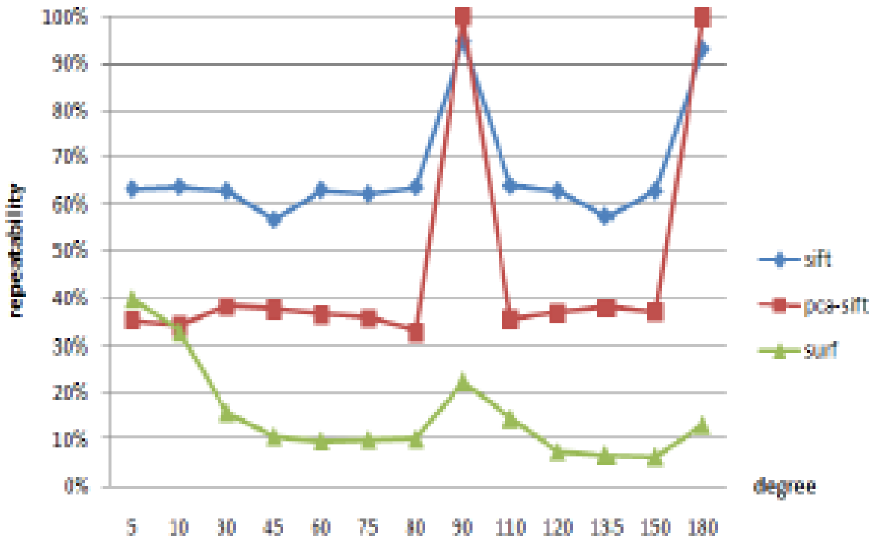


圖 1-6 旋轉比較結果

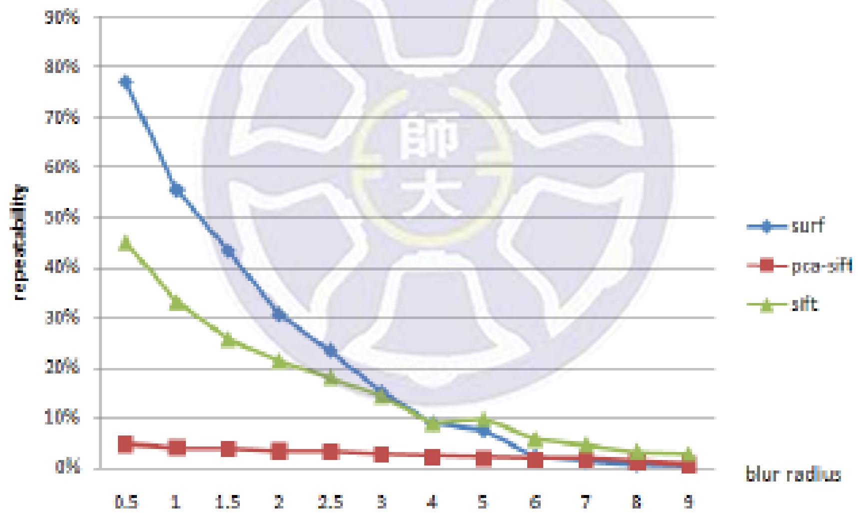


圖 1-7 模糊比較的結果

Ondrej Miksik 和 Krystian Mikolajczyk[4]則是提出了將 SIFT、SURF、FAST、BRIEF、BIRSK 和 ORB 等方法使用 KNN(K-Nearest Neighbor)進行匹配，匹配準確度使用重複率(Repeatablility score)和全準率(Precision_Recall)作為評估。結果顯示在速度上由 BRIEF 和 ORB 在特徵描述子會比 SIFT 和 SURF 來得快，但是 SIFT 和 SURF 在重複率和全準率上仍優於其他方法。

1-3 研究動機與目的

1. 本論文期望能實現影像在移動模糊的情況下，也能確實地去做出比對的效果。
2. 物體在靜態的時候往往都可以偵測出不錯的特徵點去做配對，但如果影像在動態中模糊則會讓匹配的效果變差。
3. 嘗試去使用不能的邊緣偵測去結合，來確認哪一個方法的配對組合效果會比較好。

1.4 本論文之貢獻

1. 提供在動態或是模糊的影像中也能去作出正確的偵測配對的方式。
2. 使用邊緣偵測去結合特徵偵測去分析哪一種結合的方式會比較好。
3. 改善 SURF 在模糊影像中會有偵測錯誤的情況。
4. 分析各式各樣的特徵偵測來的效果。

1.5 本論文之架構

本論文區分為六個章節，各章節之標題及內容說明如下：

[第一章]: 緒論

說明研究背景與目標,並藉由參考文獻回顧,得知在過去相關領域研究中，物體的邊緣和特徵偵測的方式，以及對其效能之評估。

[第二章]: 理論基礎

闡述本研究使用的演算法原理與基本介紹，包含 SIFT (Scale-invariant feature transform)、SURF (speeded up robust features)、LBP (Local binary patterns)、Laplacian of Gaussian。

[第三章]: 影像處理應用

以第二章之理論基礎,使用邊緣偵測和特徵偵測演算法結合,包含 Laplacian of Gaussian、SIFT (Scale-invariant feature transform)與 SURF (Speeded Up

Robust Features)。

[第四章]: 實驗設備

介紹攝影機與開發環境。

[第五章]: 實驗結果與討論

針對實作出來所改良部分進行比較，並展現建立機制之進行成果。

[第六章]: 結論與未來展望

根據整體研究的效果及實驗的目標達成程度進行結論，並說明未來目標。

第二章 理論基礎與應用

2.1 SIFT (Scale-invariant feature transform)

SIFT 全名為尺度不變特徵轉換(Scale-invariant feature transform)是由 Lowe[5]所提出，此演算法可以擷取出關鍵點。首先要將原圖進行縮減取樣 n 次(通常取 n=2)，再進一步用 m 個不同的標準差的高斯函數區分不同尺度，如(2-1)式。

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2-1}$$

在 m 有多少次數，則就有 m 個尺度的高斯函數標準差，假設我們把 m=4，這 4 個標準差分別為$\sigma,k\sigma,k^2\sigma,k^3\sigma$。將這些不同尺度的高斯函數與走向尺度做卷積運算:

$$L(x,y,k^t\sigma)=G(x,y,k^t\sigma)*I(x,y) \tag{2-2}$$

接著對同尺度且相鄰影像做 DOG

$$\begin{aligned} D(x,y,k^t\sigma) &= [G(x,y,k^{t+1}\sigma)-G(x,y,k^t\sigma)]*I(x,y) \\ &= L(x,y,k^{t+1}\sigma)-L(x,y,k^t\sigma) \end{aligned} \tag{2-3}$$

如圖 2-1 的 DOG 金字塔

利用 DOG 將每一個像素周圍 8 個像素與上下層的 9 個點總共 26 個點去做比較，如圖 2-2，該像素如果為極值，則為候選點。

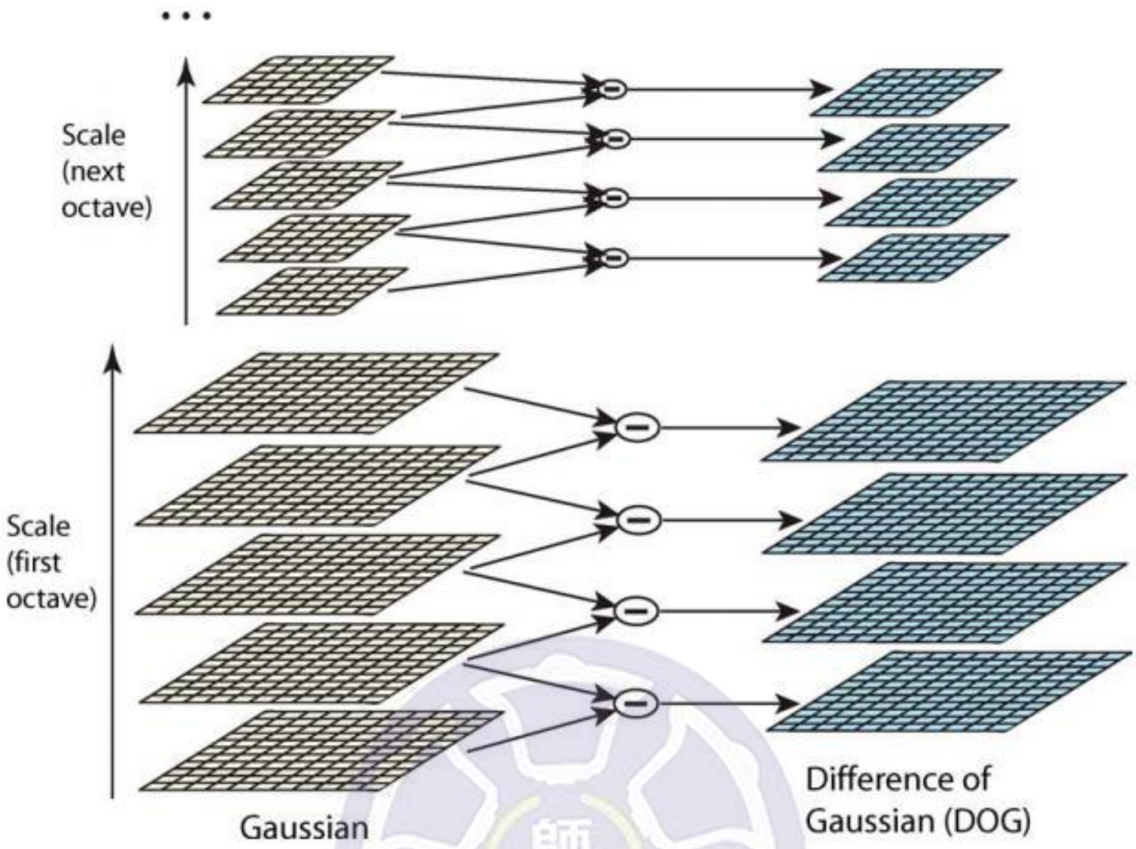


圖 2-1 DOG 金字塔

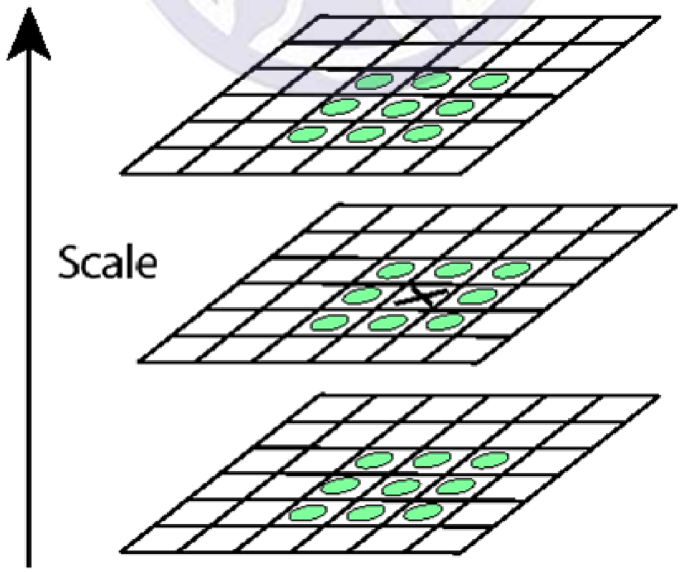


圖 2-2 DOG 找極值

在不同的尺度空間中，可能有太多關鍵點可以識別，有些關鍵點可能相對不易被識別或易受雜訊干擾，所以接下來為了要讓我們選出更好的關鍵點，將移除對比較低的候選點，並挑選每一個為原點，計算二階泰勒展開式求得連續函數 X

$$D(X) = D(0) + \frac{\partial D(0)}{\partial X} + \frac{1}{2} X^T \frac{\partial^2 D(0)}{\partial X^2} X \tag{2-4}$$

其中 D 與其偏微分的值由關鍵點的位置決定，$X = (\mathrm{x,y},\sigma)$ 為關鍵點的偏移量

對泰勒展開式 *D(X)*對 X 的導數為:

$$\frac{\partial D(0)}{\partial X} + \frac{\partial^2 D(0)}{\partial X^2} X = 0 \tag{2-5}$$

$$X = -\left(\frac{\partial^2 D(0)}{\partial X^2} \right)^{-1} \frac{\partial D(0)}{\partial X} \tag{2-6}$$

X 內的座標值如果都小於 0.5，會再進一步檢查 $\left| D(X) \right| < 0.03$，若此值小於 0.03，則捨棄此關鍵點。

取出每個關鍵點的四個二階導數 $D_{xx}, D_{xy}, D_{yx}, D_{yy}$ ，以 Hessian 矩陣表示

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{2-7}$$

計算特徵值 λ_1 和 λ_2:

$$\begin{aligned} \det(H - \lambda I) &= \det\left[\begin{vmatrix} D_{xx} - \lambda & D_{xy} \\ D_{xy} & D_{yy} - \lambda \end{vmatrix} \right] \\ &= D_{xx}D_{yy} - D_{xx}\lambda - D_{yy}\lambda + \lambda^2 - D_{xy}^2 \\ &= \lambda^2 - (D_{xx} + D_{yy})\lambda + (D_{xx}D_{yy} - D_{xy}^2) \\ &= 0 \end{aligned} \tag{2-8}$$

利用兩個特徵值 λ_1 和 λ_2 來計算出 H 行列式(det)與跡數(trace)

$$trace(H) = D_{xx} + D_{yy} = \lambda_1 + \lambda_2 \tag{2-9}$$

$$\det(H) = D_{xx}D_{yy} - D_{xy}^2 = \lambda_1\lambda_2 \tag{2-10}$$

令 $\lambda_1 \geq \lambda_2$ 且 $\lambda_1 = a\lambda_2$ 得到

$$\frac{(trace(H))^2}{\det(H)} = \frac{(a+1)^2}{a} \tag{2-11}$$

求得關鍵點不在邊上判別式

$$\frac{(trace(H))^2}{\det(H)} < \frac{(a^{'}+1)^2}{a^{'}} \tag{2-12}$$

在鍾國亮教授在影像處理與電腦視覺裡[18]表示通常會令 $a^{'} = 10$，此式如果成立代表關鍵點不在邊上。

計算區域內每個像素的梯度值 $m(\alpha, y)$ 與梯度方向 $\theta(x, y)$

$$m(x, y) = \sqrt{\left[L(x+1, y) - L(x-1, y)\right]^2 + \left[L(x, y+1) - L(x, y-1)\right]^2} \tag{2-13}$$

$$\theta(x, y) = \tan^{-1}\{[L(x, y+1) - L(x, y-1)] / [L(x+1, y) - L(x-1, y)]\} \tag{2-14}$$

最後將關鍵點中心訂出一個在關鍵點周圍 16x16 的區域中共 16 個子區域，子區域的大小為 4x4，利用八個方向來統計子區域內的梯度值及梯度向量，會得到 128 維的特徵向量，如圖 2-3。

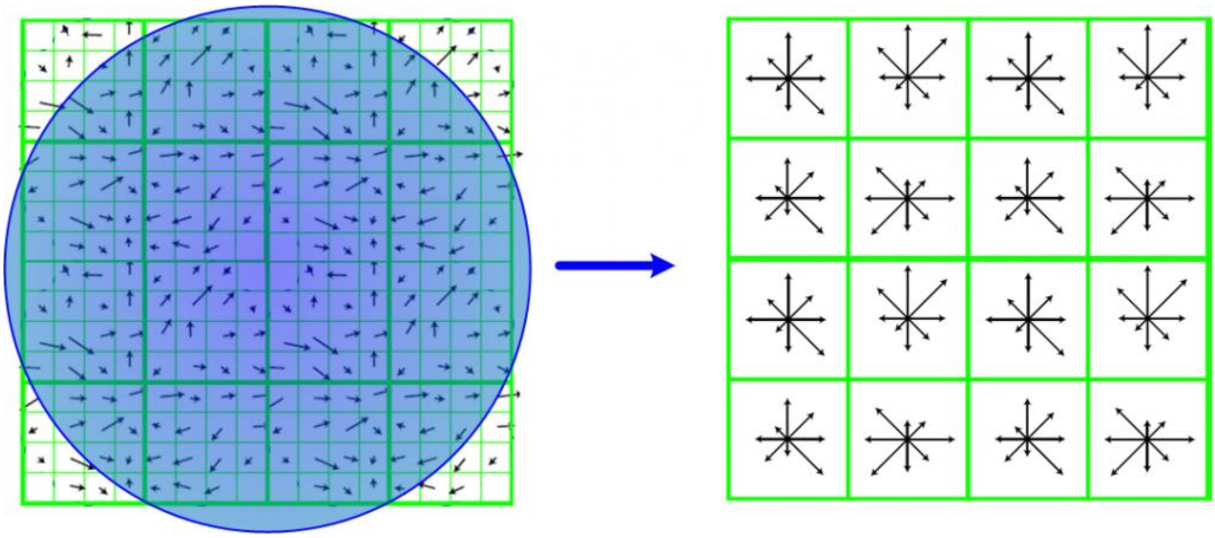


圖 2-3 特徵向量

（a）

(b)

圖 2-4 (a)原圖 (b)SIFT 偵測

2.2 SURF 演算法

加速穩健特徵[6](Speeded Up Robust Features, SURF)是由 Bay 在 2006 年提出了一個強健的圖像辨識和強健演算法。它是去做改良 SIFT 演算法，也可以說是 SIFT 的加速版。一般而言，SURF 會比 SIFT 快上好幾倍，並且在多個圖片的情況下具有更好的穩定性。主要包含了興趣點的偵測(Interest Point Detector)及特徵描述子(Feature Descriptor)。SURF 最大的特點是在於它使用 harr 演算法及積分影像的概念，這樣可以加速程式的執行的時間。

2.2.1 積分影像

積分影像是可以用來加速影像處理的方法。假設 $I(i, j)$ 為一個影像，其中 i 和 j 為影像的位子，然後我們假設積分影像為 $II(x)$ 便可定義成(2-15)

$$II(x) = \sum_{i=0}^{i\le x} \sum_{j=0}^{j\le y} I(i, j) \tag{2-15}$$

如圖 2-5 假設為一張灰階的影像圖，每一個數字分別代表他們的灰階值，我們標記紅色灰階值的總和，經過積分影像處理過後會變成圖 2-6

1	3	1	2
3	2	3	3
5	4	2	1

圖 2-5 影像灰階圖

1	4	5	7
4	8	13	20
5	13	26	46

圖 2-6 積分影像

2.2.2 構建 Hessian 矩陣

在進行 Hessian 矩陣構建前，需要先進行高斯濾波 $G(x,y,\sigma)$ 與原始圖 $I(x,y)$:

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \tag{2-16}$$

其中，高斯濾波 $G(x,y,\sigma)$ 的定義如下:

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{\frac{-(x^2+y^2)}{2\sigma^2}} \tag{2-17}$$

Hessian 矩陣是二階偏導數函數矩陣。假設 Hessian 矩陣為 $H(x,\sigma)$，影像 I 的點為 $\mathbf{x}=(x,y)$，σ 定義為 $H(x,\sigma)$ 在 x 處的尺度，如圖 2-7。

$$H(x,\sigma)=\begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \tag{2-18}$$

矩陣裡元素的定義如下:

$$L_{xx}=\frac{\partial^2}{\partial x^2}G(\sigma)*I \tag{2-19}$$

$$L_{yy}=\frac{\partial^2}{\partial y^2}G(\sigma)*I \tag{2-20}$$

$$L_{xy} = \frac{\partial^2}{\partial x \partial y} G(\sigma) * I \qquad (2\text{-}21)$$

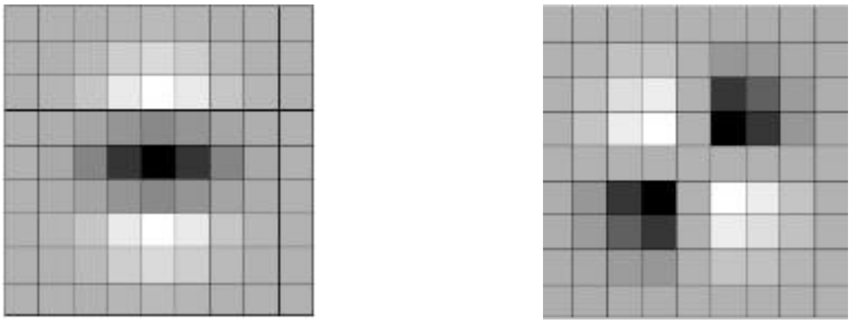


圖 2-7 高斯二階偏導數

SURF 的作者 Bay 為了快速計算積分影像，所以提出了一種 Box Filter 來近似高斯二階導數，近似高斯函數的尺度σ=1.2，運用三種結構的 box filter 於不同的尺度空間，得到近似高斯二階導數在 x、y 與 xy 的尺度空間，再結合前面所做的運算可得近似值:D_{xx} 、D_{yy} 及 D_{xy} ，最後可得近似 Hessian 行列式值(2-22)式，如圖 2-8 就是二階導數濾波。

$$\det(H_{approx}) = D_{xx} D_{yy} - (w D_{xy})^2 \qquad (2\text{-}22)$$

其中 w 是為了平衡準確值與近似值間的誤差所以引入了權重，權重會隨著尺度變化，其定義為:

$$w = \frac{\left|L_{xy}(1.2)\right|_F \left|D_{yy}(9)\right|_F}{\left|L_{yy}(1.2)\right|_F \left|D_{xy}(9)\right|_F} = 0.912 \approx 0.9 \qquad (2\text{-}23)$$

雖然權重會隨這尺度而變，但作者經過實驗得到一個經驗的值。

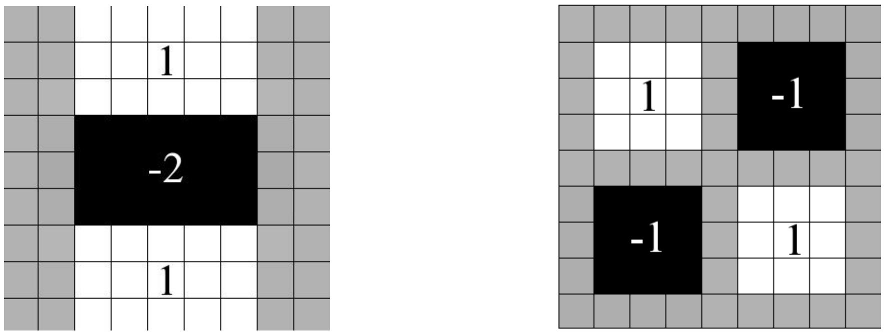
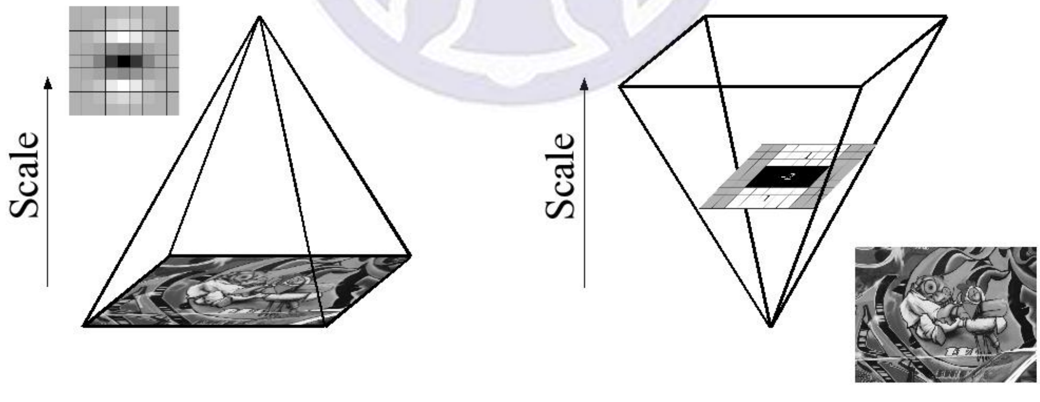


圖 2-8 近似高斯二階偏導數

2.2.3 尺度空間

在 SIFT 演算法，同一個 octave 層中的圖片尺寸相同，不同的 octave 尺寸也會不同，因為要從上一層圖片採樣得到。而在進行高斯濾波時，SIFT 的高斯濾波大小不會變，只是在不同的 octave 之間改變圖片大小。有別於傳統的，SURF 使用積分影像與 box filter，改變濾波器的大小來建立多尺度空間。圖 2-9 傳統和 SURF 的尺度空間示意圖。

在 SURF 的定義，濾波器用於尺度空間的最底層，其對應的σ=1.2，因為濾波器尺寸不同能要維持相同比例。



(a) (b)

圖 2-9 (a)傳統的方法(b)SURF 的方法

2.2.4 定位特徵點

要定位特徵點大致分為三個步驟:第一，找出在 Hessian 行列式值大於門檻值的點。第二，使用 3x3x3 非最大抑制法比較候選點與其它相鄰的 26 個點，如果

該候選點的 Hessian 行列式值大於其他點的話，代表此點為區域性的極值，如圖 2-10。第三，利用 Hessian 行列式值以內插法求得次像素精度。

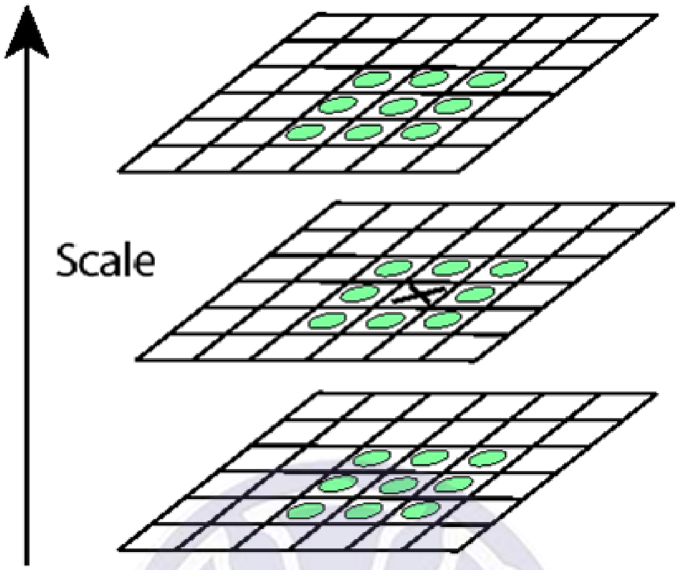


圖 2-10 尺度層圖像

2.2.5 特徵點主方向

為了決定主方向可以達到尺度不變的特性，以興趣點上的尺度間距為 s，使用 Haar 濾波器的邊長為 4s，興趣點為圓心計算在半徑 6s 的範圍內取樣點的 x、y 方向，同時乘上標準差 $\sigma = 2s$ 高斯係數。以圓心角為 $\pi/3$ 扇形區域，以一個微小的小角度旋轉一周掃描興趣點，在每一個角度分別去對 x、y 方向的 Harr 小波轉換而得到了一個新的向量，圖 2-11。最後，把所有的向量去做比較，最大者就是興趣點的主方向。

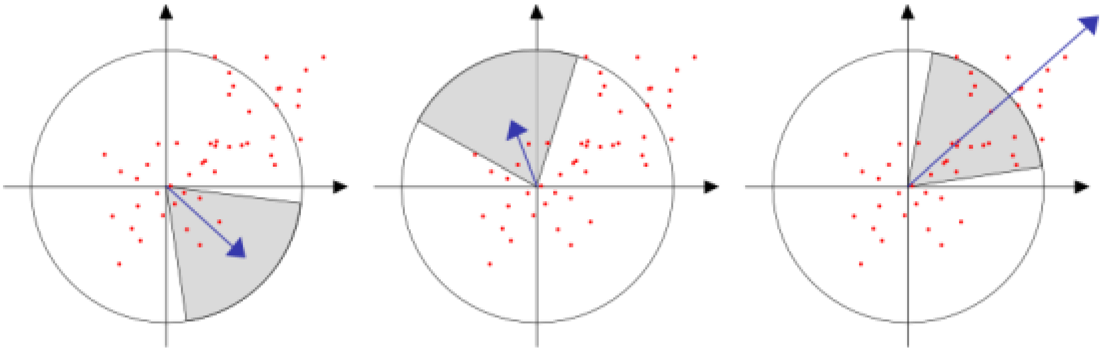


圖 2-11 主方向的搜尋

2.2.6 特徵點描述子

以興趣點為中心，建立邊長為20s的方塊作為 Harr 小波響應的範圍，再將方塊分成 4x4 個方形方塊。將每個方塊的響應值和絕對值相加，形成了四個分量 $\sum dx$ 、$\sum|dx|$ 、$\sum dy$ 、$\sum|dy|$ 的總和為特徵向量 v ，如 2-24 式，dx 代表水平方向的小波響應，dy 則是代表垂直方向的小波響應，如圖 2-12。這整個方塊總計為一個 4x4x4=64 為的向量。

$$v=(\sum dy,\sum dx,\sum|dy|,\sum|dx|) \tag{2-24}$$

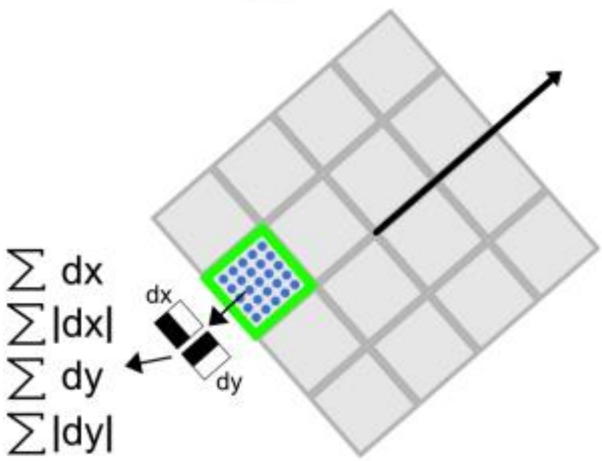


圖 2-12 特徵向量描述子

(a)

(b)

圖 2-13 (a)原圖(b)SURF 檢測

2.3 Laplacian of Gaussian

Laplacian of Gaussian 主要是利用高斯模糊的結果，再用拉普拉斯繼續求邊緣，首先我們要先介紹拉普拉斯算子，然後再介紹和高斯模糊再結合後的效果。

2.3.1 拉普拉斯算子

先舉一維的例子來解釋一下，假設 $f(x,y)$ 沿著 x 軸微分

$$\nabla_x f = f(x+1,y) - f(x,y) \tag{2-25}$$

再對 x 軸微分

$$\begin{aligned}\nabla_x^2 f &= f(x+2,y) - f(x+1,y) - [f(x+1,y) - f(x,y)] \\ &= f(x+2,y) - 2f(x+1,y) + f(x,y)\end{aligned} \tag{2-26}$$

令 $x = x+1$ 以進行參數轉換

$$\nabla_x^2 f = f(x+1,y) - 2f(x,y) + f(x-1,y) \tag{2-27}$$

接著對 y 軸做

$$\nabla_x^2 f = f(x, y+1) - 2f(x, y) + f(x, y-1) \tag{2-28}$$

合併兩式

$$\begin{aligned}\nabla^2 f &= \nabla_x^2 f + \nabla_y^2 f \\ &= f(x, y+1) + f(x+1, y) + f(x, y-1) + f(x-1, y) - 4f(x, y)\end{aligned} \tag{2-29}$$

拉普拉斯可以使用面罩圖 2-14，完成偵測邊緣圖 2-15

0	1	0
1	-4	1
0	1	0

圖 2-14 拉普拉斯面罩

(a) (b)

圖 2-15 (a)原圖 (b) 拉普拉斯算子

接下來我加入高斯濾波(2-30)式

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{2-30}$$

我假設原影像為 $f(x,y)$

$$LoG = \nabla^2[G(x,y,\sigma) * f(x,y)] = [\nabla^2 G(x,y,\sigma)] * f(x,y) \qquad (2\text{-}31)$$

證明 LoG 有結合率，如(2-32)式

$$\begin{aligned}
LoG &= \nabla^2[G(x,y,\sigma) * f(x,y)] \\
&= \frac{\partial}{\partial x^2}[G(x,y,\sigma) * f(x,y)] + \frac{\partial}{\partial y^2}[G(x,y,\sigma) * f(x,y)] \\
&= \frac{\partial}{\partial x^2}[G(x,y,\sigma)] * f(x,y) + \frac{\partial}{\partial y^2}[G(x,y,\sigma)] * f(x,y) \\
&= [\frac{\partial}{\partial x^2}G(x,y,\sigma) + \frac{\partial}{\partial y^2}G(x,y,\sigma)] * f(x,y) \\
&= [\nabla^2 G(x,y,\sigma)] * f(x,y)
\end{aligned} \qquad (2\text{-}32)$$

加入高斯濾波的原因是要減少 Laplacian 對於雜訊的敏感度，如圖 2-16 為鍾國亮教授所提出的 5x5 的 LoG 面罩[18]，其中中間值遠大於周圍係數。

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

圖 2-16 LoG 的面罩

2.4 隨機抽樣一致性(RANSAC)

RANSAC[7]方法用來去除不在對比影像範圍的來提升精確度。RANSAC 方法是在圖片的特徵點中隨機取出 n 個點計算模型參數，如果有圖片的特徵點和模型誤差小於該模型的門檻值，將其列入 Consensus set，則代表此點為內群。之後重複多次後，在 Consensus set 中有超過一定數量的情況下，則代表此模型為可行

的，最後重複上述的步驟到一定的次數後，找出誤差最小的模型為 RANSAC 的結果。

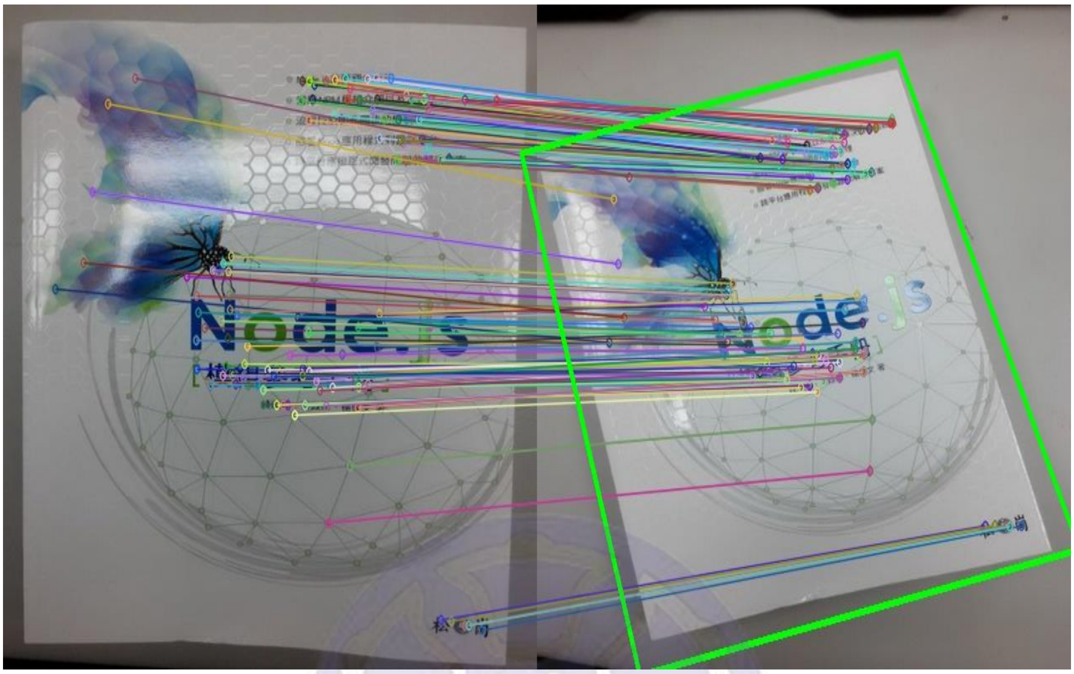


圖 2-17 RANSAC 結果示意圖[8]

2.5 比較 SIFT 和 SURF 差異

本論文主要是利用 SURF 來偵測特徵點，而不是使用傳統的 SIFT，因為在運算的速度來說 SURF 會比 SIFT 來得快，以下會介紹一下兩者的差異:

表 2-1　SIFT 和 SURF 比較圖

	SIFT	SURF
特徵點偵測	用不同尺度的圖片與高斯函數做卷積。	用不同大小的 box filter 與原始圖像做卷積。
方向	特徵點鄰接矩形區域內，利用梯度直方圖計算。	特徵點鄰接圓域內，計算 x、y 方向上的 Haar 小波響應。
特徵描述子	16*16 區域劃分為 4*4(或 2*2)的子區域，每個子域計算 8bin 直方圖。	20*20 區域劃分為 4*4 子域，每個子域計算 5*5 個採樣點的 Haar 小波響應，記錄 $\sum dx, \sum dy, \sum \lvert dx \rvert, \sum \lvert dy \rvert$。

2.6 雜訊處理

在偵測物體的情況下，讀出來的影像的資訊不一定是乾淨且單純的，因此消除雜訊是相當重要的步驟。當攝影機在偵測移動平面上時，雜訊處理更為重要，而處理雜訊通常是使用平滑濾波器來處理。平滑濾波器主要分兩種:線性濾波和非線性濾波，線性濾波是有一個固定的參數作為核心如均值濾波和高斯濾波，反之非線性濾波則是沒有固定的參數如中值濾波和雙邊濾波。均值濾波是指將每個像素互相替換相鄰矩形內的平均值。高斯濾波則是改變核心的參數，將越靠近中心的像素權重越大，核心參數是由高斯函數計算得到，核心的各像素相加替換當前的像素值，之後再用卷積處理影像。中值濾波是將濾波範圍內的像素去排序，並用中值去替換當前的像素，模糊會比高斯來得少一些。雙邊濾波是使用像素之間靠近程度，還外加考慮了光線和色彩的差異，這樣能夠得到有效雜訊去除在影像上，但處理時間相對來說會比較久。

(a) (b)

(c) (d)

(e) (f)

圖 2-18　(a)原圖 (b)灰階雜訊圖 (c)均值濾波

(d)高斯濾波 (e)中值濾波 (f)雙邊濾波

2.7 影像銳化

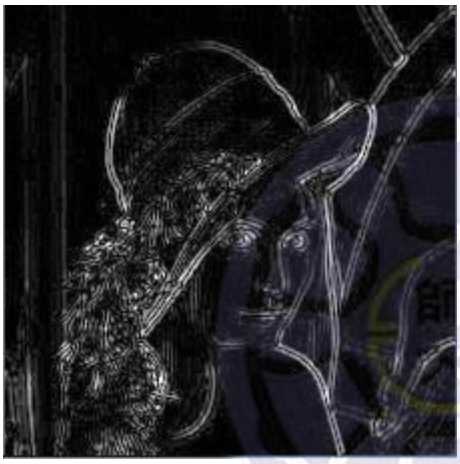
影像銳化最主要的目的就是把模糊的影像變得更加清晰，請可以把細節變得更加清楚。本論文使用邊緣銳化圖像，為了加強邊緣和細節來改善特徵匹配的結果，而 Sobel 和 Laplacian 是最常被用來使用強化邊緣。

(a)

(a) (c)

圖 2-19 (a)原圖 (b)Sobel 偵測 (c)Sobel 銳化

(a)



(a)

(c)

圖 2-20 (a)原圖 (b)Laplacian 偵測 (c) Laplacian 銳化

2.8 特徵擷取結合邊緣銳化

在一般靜態的情況下，我們可以使用不管是 SIFT 或者是 SURF 都可以很準確地抓取到目標的特徵，但如果在動態的情況下，特徵點相對來說會少很多，所以我們使用了邊緣強化可以增加特徵點，這樣不管是在靜態還是動態的情況下，都能去抓到多個特徵點，但是如果比較靜態，加強邊緣的話並不會比較多的，但在動態會好很多。

在靜態原圖的情況下，使用一般的演算法就可以很清楚地去偵測每一個特徵，但如果在動態的情況可以發現，在使用邊緣偵測再去做特徵擷取的方法會比在靜

態的情況下來得多一些。

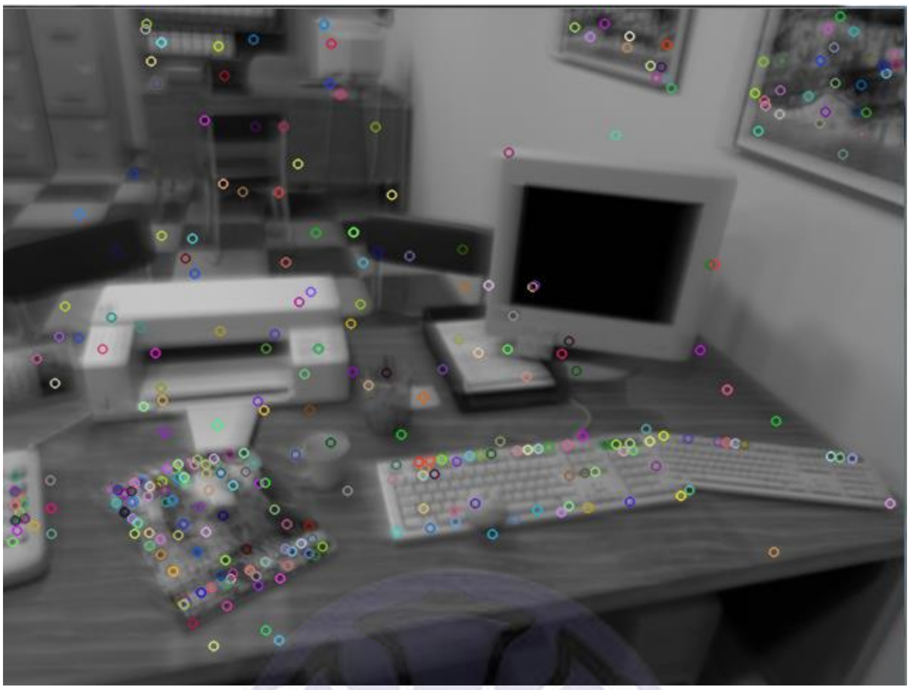
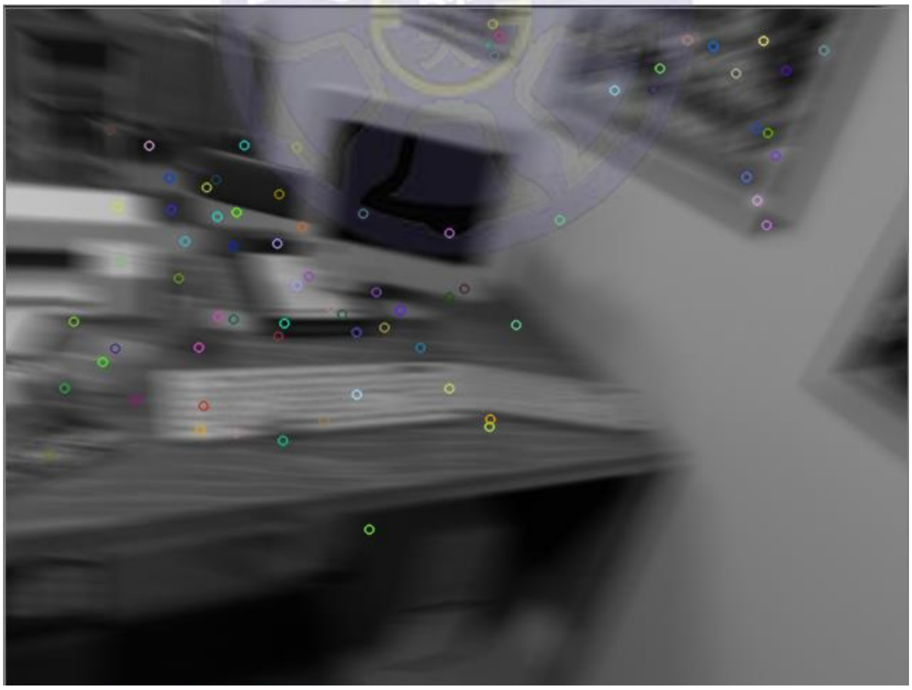
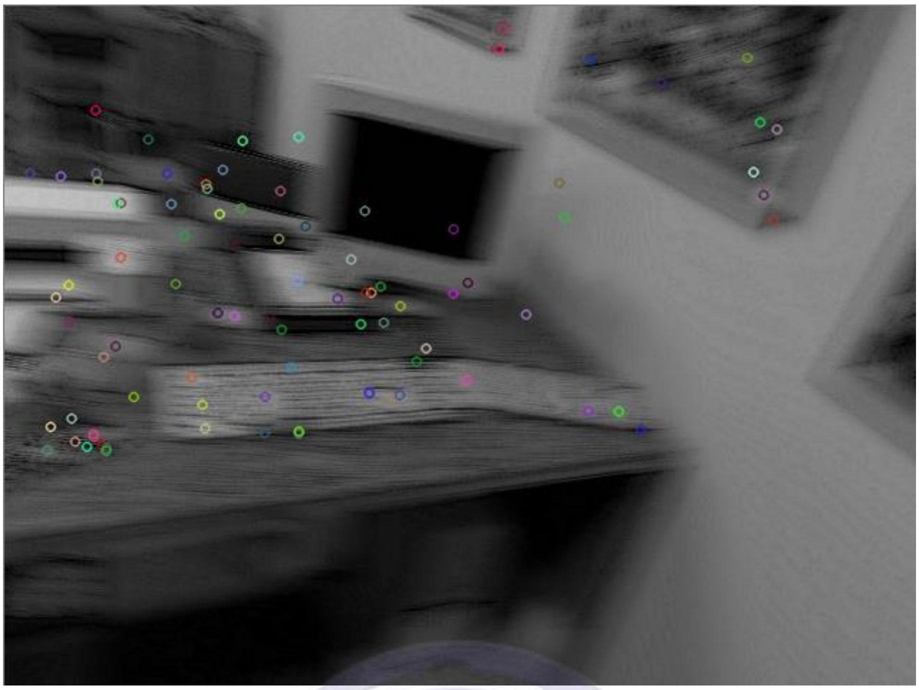


圖 2-21 原圖靜態特徵點[14]



(a)



(b)

圖 2-22 動態特徵圖(a)沒有邊緣銳化 (b)有邊緣銳化

第三章 實驗設備

3.1 個人電腦

使用電腦為 Intel Core i5-4430CPU 3.00GHz，RAM 為 12GB，在 Visual Studio 平台下進行作業，來做一系列的影像處理。規格如下

表 3-1 電腦配備

CPU	Intel Core i5-4430 3.00GHz
RAM	12GB
Operating System	Windows 7
Language	Visual Studio C++

3.2 攝影機

本研究採用 LifeCam Studio

圖 3-1 攝影機

3.3 資料庫

這裡我們使用了 CERTH Image Blur Dataset[13]的資料庫，它提供一個全面的圖像數據集，包括無失真，自然模糊和人為模糊的圖像，用於圖像評估。

CERTH 圖像模糊數據集由 2450 個圖像組成，其中 1850 個是在不同拍攝條件下由各種相機拍攝的照片，在拍攝後未以任何方式進行更改，然後對它們還提供幾種類型的高斯、動態的圖像。這裡我們主要是要使用他們的動態影像來呈現效果。

這裡我們還提供一個資料庫 Trajectory for Variable Frame-Rate[14]，它有一個很大的優點就是它有顯示是多少拍照頻率，且有能供更快的動態影像，可以進一步的去證明我們的方法。

第四章 實驗結果與討論

4.1 偵測物體架構

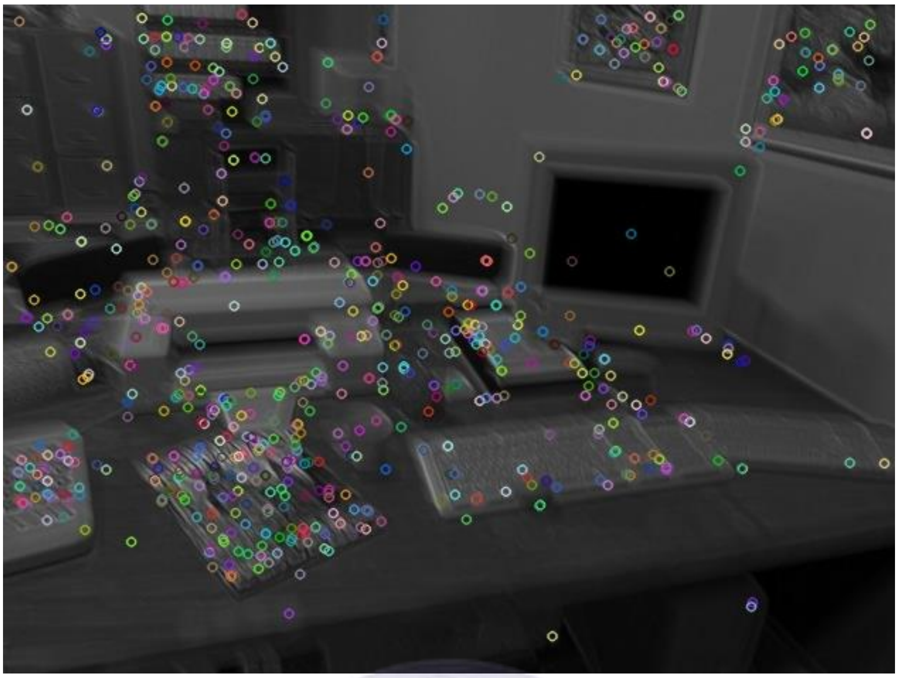
一個完備的偵測物體的架構，需要雜訊處理、邊緣強化以及特徵偵測的概念才能去完成。由於物體偵測的方式必須隨時要掌握特徵資訊，且此特徵資訊不論在靜態的時候或者在移動時皆需要，又因需要多個演算法去做運算才能得到精準的特徵，所以基本設定為循環設定；在偵測物體的程序前則是要先使用於雜訊處理或者是圖片是一個乾淨單純的，且執行一次便能獲得充分資訊，運算時間又相對較短；而在偵測的程序在處理中，但必須持續地去擷取特徵直到完全確認偵測到物體。

4.2 實驗一

首先會去做有使用高斯濾波情況下分別去做 Sobel 偵測如圖 5-1，再來我們會分別去做 SIFT 和 SURF 的特徵點偵測如圖 5-2，再來去做匹配，這裡因為有可能會有匹配錯誤的情形，所以使用 RANSAC 來做篩選，分別顯示當沒有做 RANSAC 和有做 RANSAC 的差別如圖 5-3，最後會和實驗二一起統計在不同的特徵偵測和有無 RANSAC 的匹配點，有做 RANSAC 的情況在本文稱為好的匹配點，而沒有經過 RANSAC 稱為匹配點且我們連續做 20 張動態圖像去做分析去統計。如圖 5-4 為實驗一的流程圖。

圖 4-1 Sobel 偵測銳化的結果

(a)



(b)

圖 4-2 (a)SIFT 特徵點 (b)SURF 特徵點

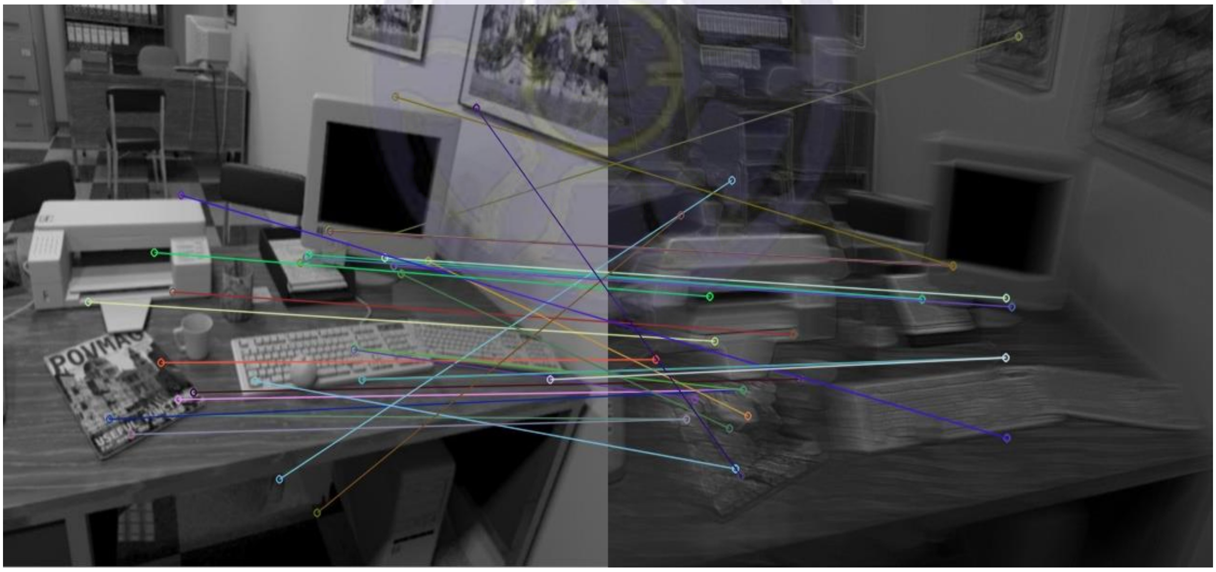


圖 4-3 經過 RANSAC 匹配的結果

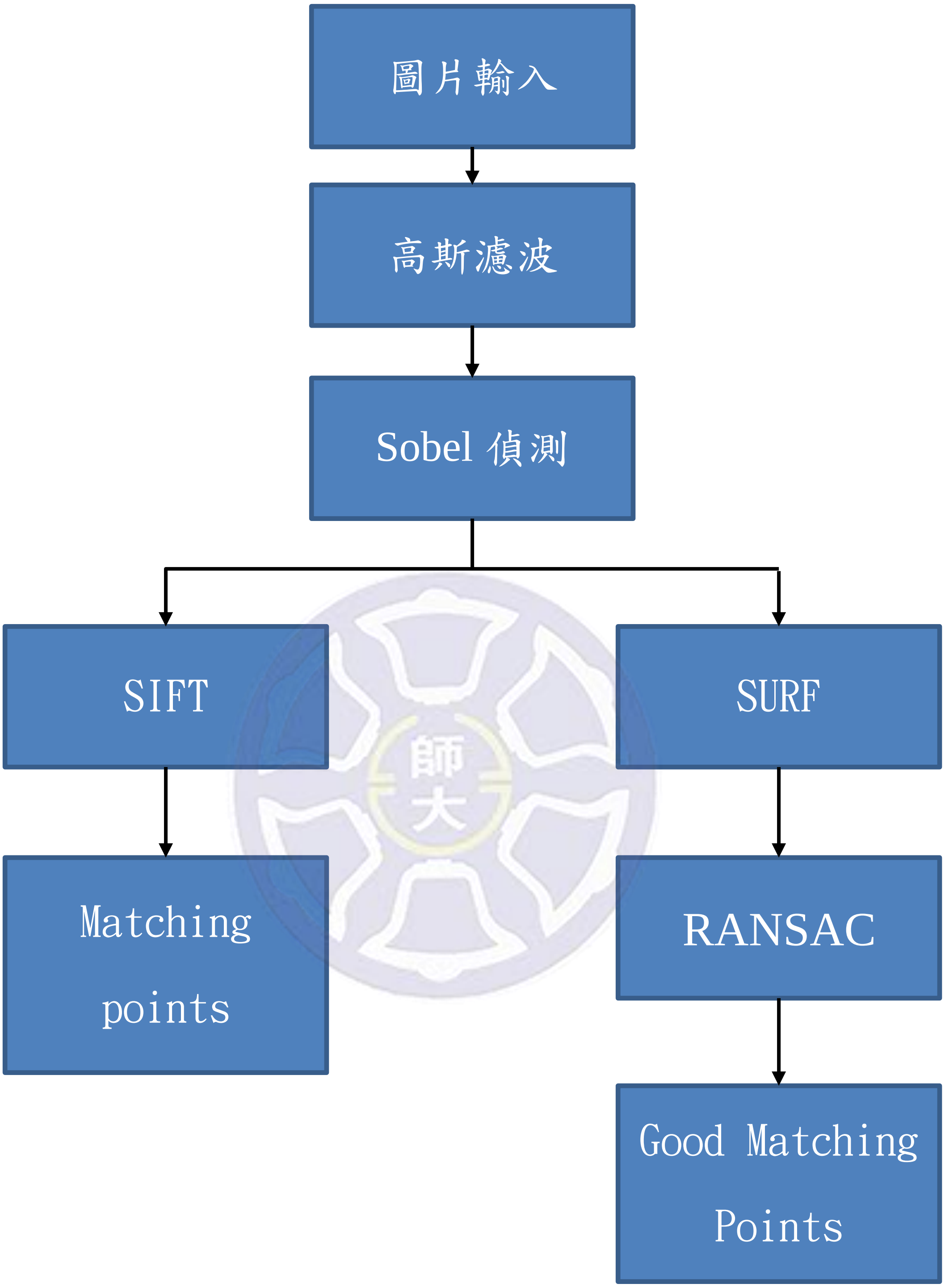
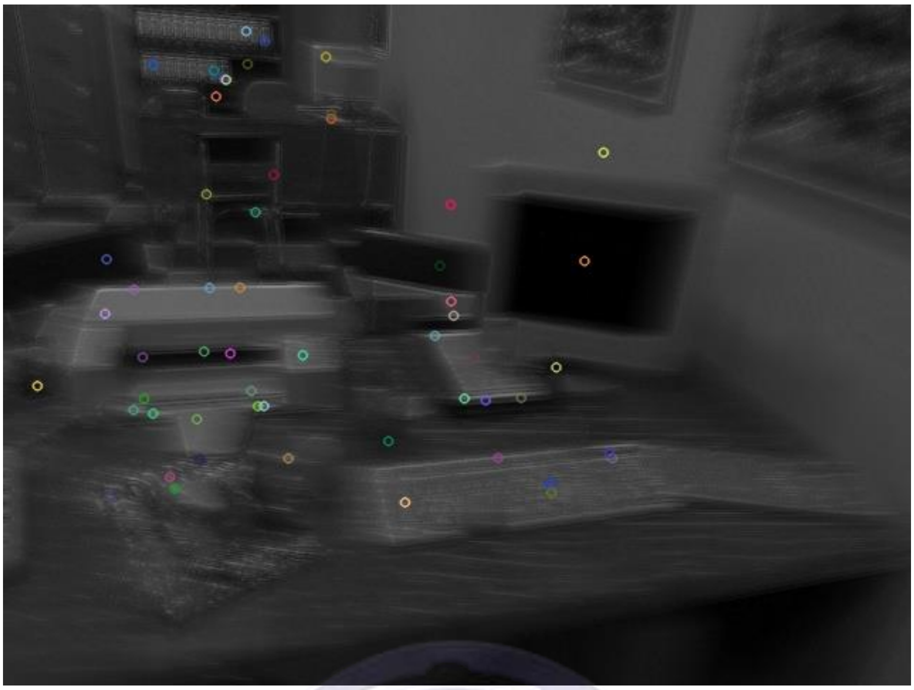


圖 4-4 經過 Sobel 分別結合 SIFT 和 SURF 的流程圖

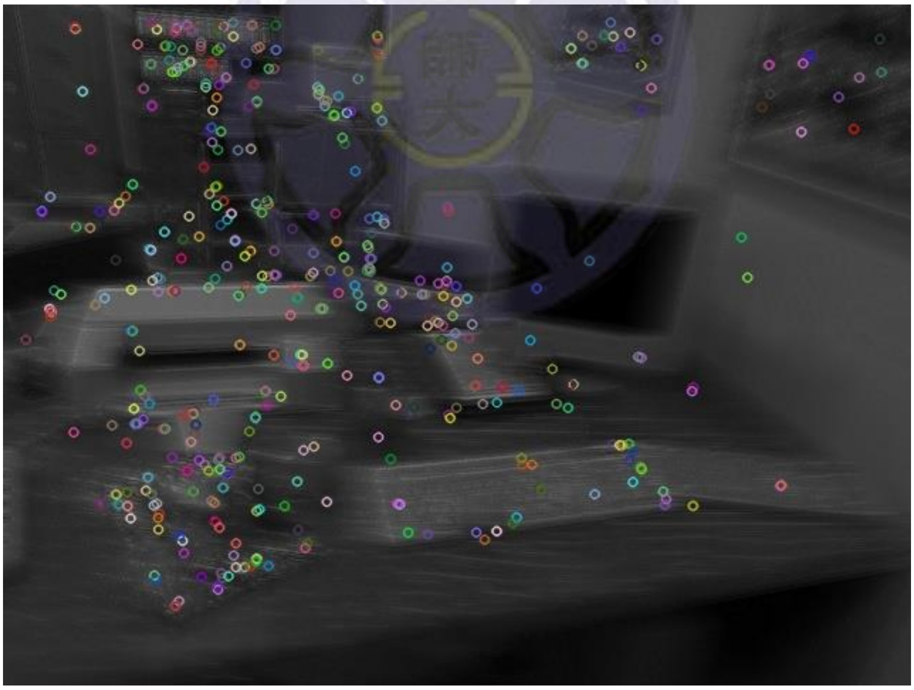
4.3 實驗二

這裡我們使用 LoG 的方法先去做邊緣的銳化如圖 4-5，接下來一樣分別使用 SIFT 和 SURF 偵測特徵點如圖 4-6，之後使用 RANSAC 來做篩選，並分別顯示當沒有做 RANSAC 和有做 RANSAC 的差別如圖 4-7，最後會和實驗一一起統計在不同的特徵偵測和有無 RANSAC 的匹配點，有做 RANSAC 的情況在本文稱為好的匹配點，而沒有經過 RANSAC 稱為匹配點且我們連續做 20 張動態圖像去做分析去統計。如圖 4-8 為實驗二的流程圖。

圖 4-5 LoG 銳化



(a)



(b)

圖 4-6 (a)SIFT 特徵點 (b)SURF 特徵點

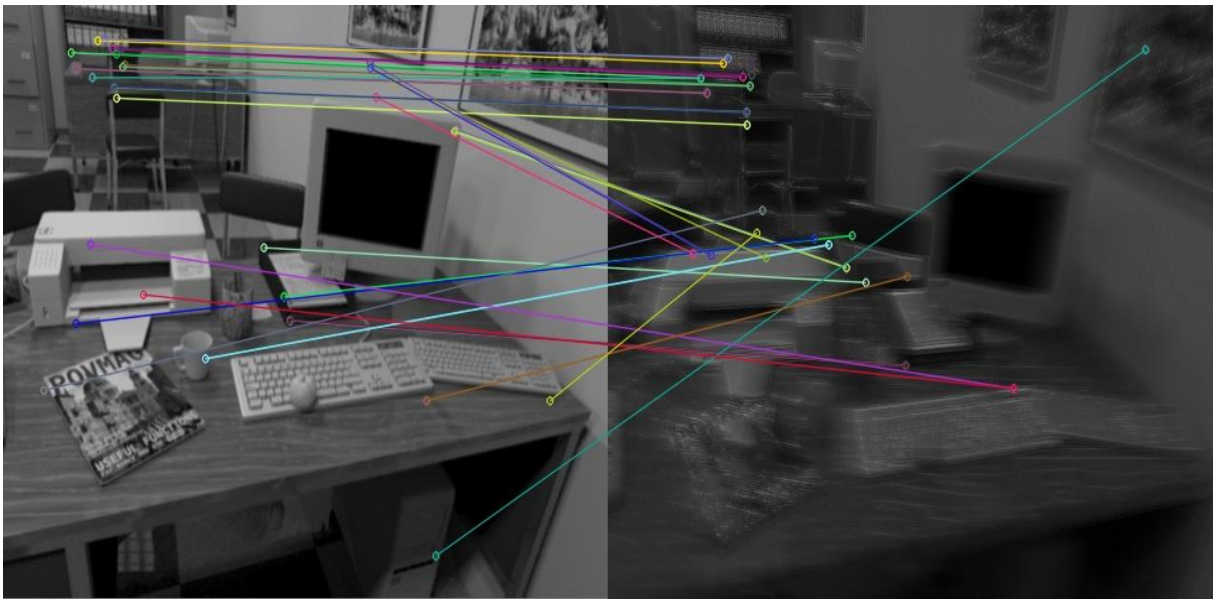


圖 4-7 經過 RANSAC 匹配的結果

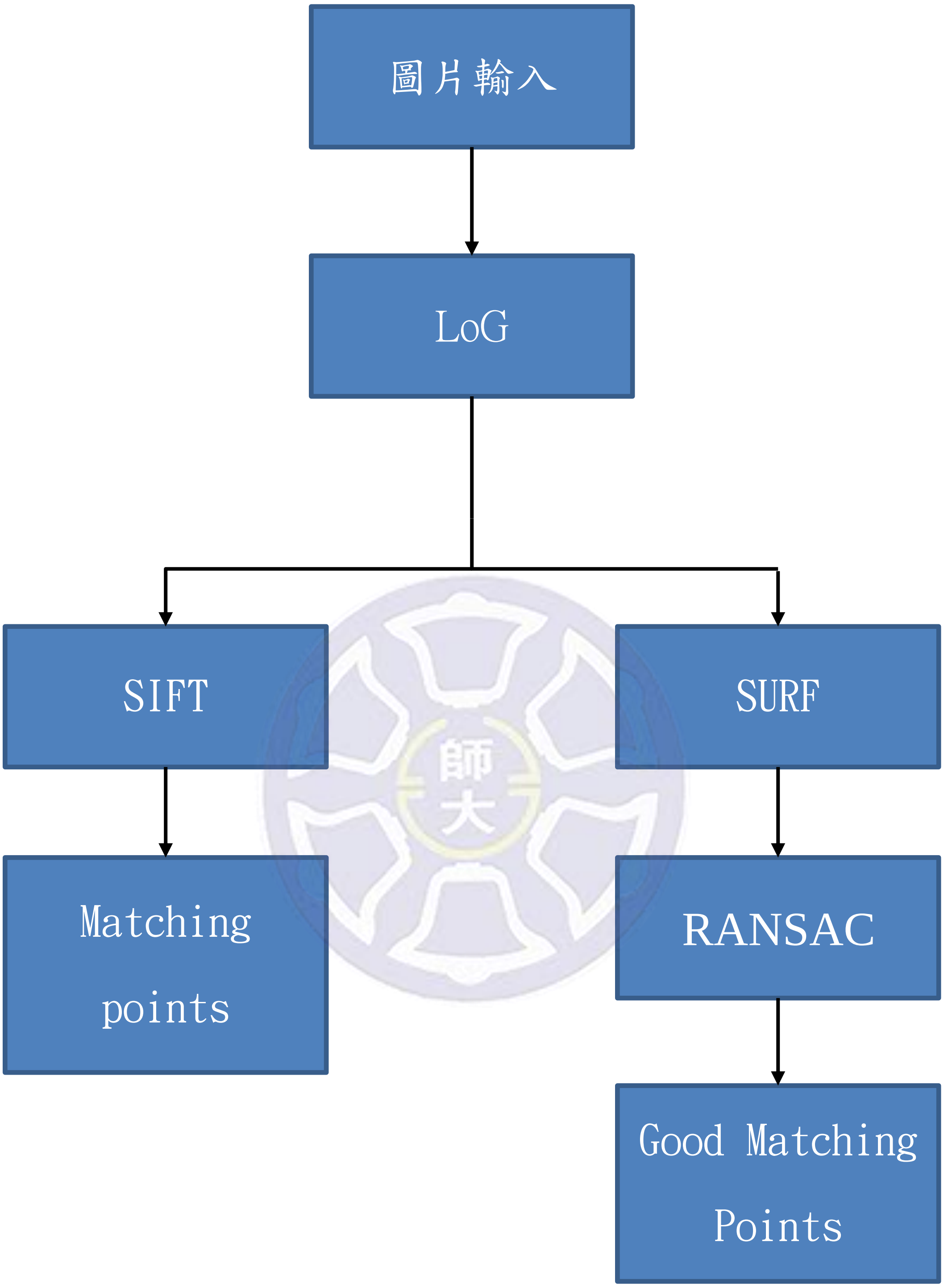


圖 4-8 經過 LoG 分別結合 SIFT 和 SURF 的流程圖

4.4 比較實驗結果

經過一連串的比較之後，我們可以得到四種不同的數據來做比較，這四種方法分別做了 20 張動態影像的結果，而這四種方法分別是 Sobel 結合 SIFT、Sobel 結合 SURF、LoG 結合 SIFT 和 LoG 結合 SURF，四種來比較哪一個方法的匹配點數較多且還會比較有使用 RANSAC 篩選後的匹配點數如表 4-1。

表 4-1 比較四種方法的匹配數

比較方法	RANSAC	NO RANSAC
Sobel 結合 SIFT	10.95	173
Sobel 結合 SURF	38.3	1052.3
LoG 結合 SIFT	8.55	123.95
LoG 結合 SURF	37.55	1061.8

我可以發現不在動態的影像方面，不管是使用 Sobel 或是 LoG 他們分別去跟 SURF 結合的方法都會比和 SIFT 還得好，重點是 SURF 本身的運算的時間就會比 SIFT 來得快，所以進一步可以得知在動態影像去做偵測匹配時，去和 SURF 結合效果會來得更好。

4.5 實驗三

我們使用兩本不同的書籍去做偵測，其中一本書用手去移動的結果，分別使用 SURF 的演算法並且使用 RANSAC 所得出來的結果，差別在於一個有使用 Laplacian of Gaussian 邊緣偵測另一個沒有使用。

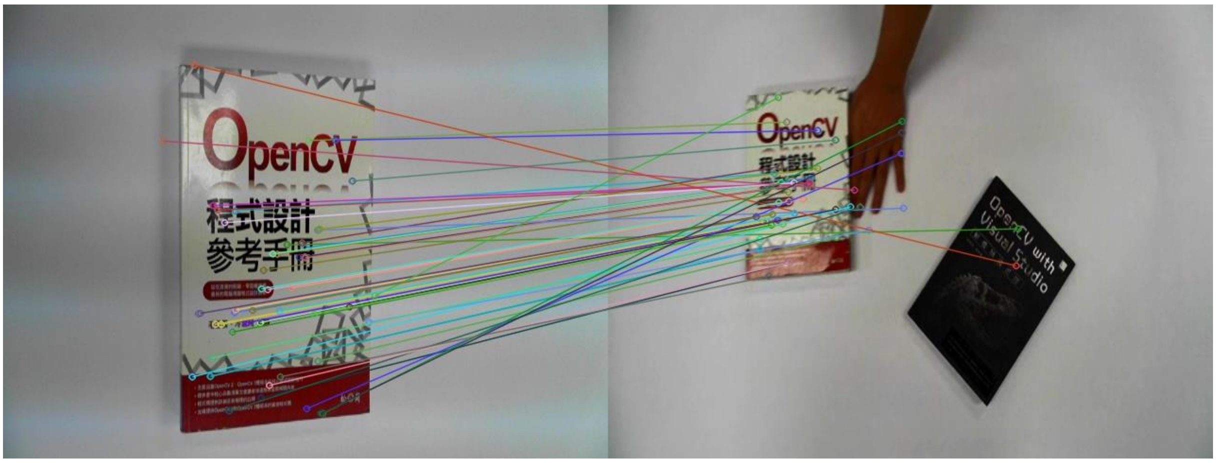


圖 4-9 沒使用邊緣強化的情況(平移)

圖 4-10 使用邊緣強化的情況(平移)

表 4-2 Laplacian of Gaussian 邊緣偵測於有無邊緣強化之比較結果(平移)

比較內容	邊緣強化	無邊緣強化
圖 1 特徵點數	**980**	855
圖 2 特徵點數	**518**	500
匹配個數	**980**	855
好的匹配點個數	**927**	794
運算時間(s)	0.87322	0.877515

可以發現在有邊緣強化的情況下，特徵點的個數、匹配點的個數都會比沒有邊緣強化了情況下來得好。

4.6 實驗四

本文接下來討論如果書本在旋轉的情況下所偵測出來的結果。這裡是用 SURF 演算法和 RANSAC 篩選所偵測出來，一樣比較有無 Laplacian of Gaussian 邊緣偵測強化的結果。

圖 4-11 沒使用邊緣強化的情況(旋轉)

圖 4-12 使用邊緣強化的情況(旋轉)

表 4-3 Laplacian of Gaussian 邊緣偵測於有無邊緣強化之比較結果(旋轉)

比較內容	邊緣強化	無邊緣強化
圖 1 特徵點數	**980**	855
圖 2 特徵點數	**804**	766
匹配個數	**980**	855
好的匹配點個數	**947**	702
運算時間(s)	0.865755	0.919031

在物體旋轉的情況下一樣可以發現，在沒有邊緣強化下特徵點的個數、匹配點的個數都會比沒有邊緣強化的情況下來得好。

從實驗三和實驗四可以發現物體不管是在平移或者是在旋轉的情況下，有使用邊緣強化的情況都會比原本沒有邊緣強化來得好，並可得知邊緣強化可加強特徵點的匹配數。

第五章 結論與未來展望

本文的研究主要提出了基於邊緣強化去結合特徵演算法去對移動物體去做偵測。在邊緣偵測的方式我們使用 Laplacian of Gaussian 和 Sobel 的方法成功的可以改善物體在移動的情況下較難去得到特徵點的問題，並且使用 RANSAC 的演算法去做篩選來選出最符合特徵匹配的結果，最後去比較結合各特徵演算法所需要的時間和特徵點個數，並且我們可以從實驗數據方面看到，在 LoG 和 Sobel 的方法去跟 SURF 結合效果都會來得比 SIFT 好，這可能是因為 SURF 在動態環境的情況下所抓到的特徵點會比 SIFT 來得多，所以結合 SURF 的效果會來得比較好的原因。

本實驗雖然可以成功的偵測到移動物體的特徵去做匹配，但是如果物體超越到某個速度或是數量太多情況，可能會變成偵測效果沒這顯著，所以如何要在移動物體在很快的情況，同時可以偵測且比對或許是未來所需要考慮的因素，除此之外也可使用深度學習的方式，去訓練特徵點讓比較不好的特徵點進一步的去除並且可以得到更精確的匹配。

還有因為計算 RANSAC 的計算所花費時間較長，所以有一部分的時間花費在此步驟中，如何把 RANSAC 運算速度加快便可即時邊是實現，雖然有人提出一種方式是 SURF 降維，但因為降低 SURF 的維度而造成準確率下降，所以必須進一步的去衡量優先順序。最後希望可以透過此研究和機械手臂去做結合達到辨識且能分析夾取點的效果。

參考文獻

[1] John Canny, “A Computational Approach to Edge Detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* , vol.PAMI-8, no.6, Nov, 1986.

[2] Penghui Jiang, Shengjie Zhao, Samuel Cheng, “Rotational invariant LBP-SURF for fast and robust image matching,” *International Conference on Signal Processing and Communication Systems*, Dec. 2015.

[3] L. Juan, O.Gwun, “A Comparison of SIFT, PCA-SIFT and SURF,” International Journal of Image Processing, pp.143-152, 2009.

[4] Ondrej Miksik, Krystian Mikolajczyk, “Evaluation of Local Detectors and Descriptors for Fast Feature Matching,” *Pattern Recognition*, pp.2681-2684, 2012.

[5] Lowe, David G. "Distinctive Image Features from Scale-Invariant Keypoints ". *International Journal of Computer Vision*, pp. 91–110, 2004.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, “SURF: Speeded Up Robust Features,” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[7] Martin A. Fischler, Robert C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Apphcatlons to Image Analysis and Automated Cartography,” *Communications of the ACM*, vol. 24, pp.381-395, 1981.

[8] 錢逢祥，蔡政崇，楊傑文，Node.js 模組參考手冊，松崗出版社，2015

[9] Marko Heikkila, Matti Pietikainen “A Texture-Based Method for Modeling the Background and Detecting Moving Objects,” *IEEE TRANSACTIONS ON*

PATTERN ANALYSIS AND MACHINE INTELLIGENCE, vol. 28, no. 4, Apr. 2006.

[10]T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, July 2002.

[11]Marko Heikkila, Matti Pietiainen, and Cordelia Schmid, “Description of interest regions with local binary patterns,” *Pattern Recognition*, vol. 42, Mar., pp. 425–436, 2009.

[12]Prabhakar C J and Praveen Kumar P U, “LBP-SURF Descriptor with Color Invariant and Texture Based Features for Underwater Images,” *ICVGIP*, Dec. 2012.

[13]E. Mavridaki, V. Mezaris, “No-Reference blur assessment in natural images using Fourier transform and spatial pyramids”, *IEEE International Conference on Image Processing*, Paris, France, Oct., 2014.

[14]Ankur Handa, Richard A. Newcombe, Adrien Angeli and Andrew J. Davison,” Real-Time Camera Tracking: When is High Frame-Rate Best,” *Computer Vision–ECCV 2012*, pp. 222-235, 2012

[15]朱庭葳，以邊緣強化之 SIFT 偵測移動物件， 國立臺灣科技大學電機系，碩士論文，2016 年

[16]鍾秉剛，基於向量模型設計與實現蒙地卡羅定位與路徑規劃於輪型移動機器人，國立臺灣師範大學機電工程系，碩士論文，2015 年

[17]陳希哲，應用運動回復結構技術建立機器手臂夾取點與三維物體模型，

國立臺灣師範大學機電工程系，碩士論文，2013 年

[18]鍾國亮，影像處理與電腦視覺，東華書局，2015 年

[19]Marius Muja and David G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," *VISAPP*, pp.331-340, 2009