

東海大學資訊工程學系研究所  
碩士論文

指導教授：林祝興 博士

Dr. Chu-Hsing Lin

應用於監視系統中自動化目標追蹤與辨  
識之探討

On the Automatic Target Tracking and  
Recognition for Surveillance Systems

研究生：蔡祥霖

中華民國一〇四年六月



# 東海大學碩士學位論文考試審定書

東海大學資訊工程學系 研究所

研究生 蔡祥霖 所提之論文

應用於監視系統中自動化目標追蹤與辨識之  
探討

經本委員會審查，符合碩士學位論文標準。

學位考試委員會  
召集人

范進科

簽章

委員

許永年

江意坤

劉榮春

指導教授

林炳志

簽章

中華民國 104 年 6 月 24 日

## 中文摘要

本篇論文是利用 Viola's method 與 OpenTLD 兩種方法實作一個可根據目標追蹤的自動運鏡系統，並比較及測試兩種方法的優缺點。自動運鏡系統可應用在安全監控、減少攝影機死角且自動化的安全監控上。

自動運鏡系統流程是從攝影機輸入接收影像，預測目標即將超出可視範圍時，會轉動自身底盤，讓鏡頭持續追蹤目標。以人臉辨識作追蹤為範例，我們利用 Viola's method 與 OpenTLD 結合 Arduino 實作自動追蹤目標的運鏡系統。從攝影機接收到影像後由 Viola's method 與 OpenTLD 進行影像處理，若目標即將超出可視範圍就會根據目標位置來控制位於攝影機底部的馬達轉動。

我們將 Viola's method 與 OpenTLD 兩種方法進行實驗比較，實驗結果顯示：在準備期間給予足夠的訓練樣本 Viola's method 就可以穩定的辨識臉型，但在影像過度曝光或是曝光不足時容易造成誤判，會使用的 CPU 資源較多。

至於 OpenTLD 則是使用一張訓練樣本就可以追蹤目標，有過濾低對比背景能力避免影像過度曝光或是曝光不足產生誤判。而動態新增訓練集時會增加運算量造成 FPS 隨著時間減少的情形。

關鍵字：OpenTLD、Haar-like 特徵值、OpenCV、Arduino、監視系統、目標追蹤、機器學習

# ABSTRACT

The aim of the thesis is to implement an auto tracking target system with Viola's method and OpenTLD, and to compare the advantages and disadvantages of both methods. This system can be used for security surveillance and eliminating camera blind spot.

The proposed camera system can detect targets by capturing images, and when the target is beyond the viewable range, the camera will try to keep tracking of the target by the rotation of the chassis. For the facial feature tracking, we propose to adopt Arduino and combine it with Viola's method and OpenTLD to handle long-term tracking of unknown objects. When the tracked target is about to exceed the viewable range of the camera, the location of the target is used to control the Arduino motor to adjust the camera lens orientation.

Moreover, this research compare Viola's method with OpenTLD by making an experiment. The experimental results show that Viola's method can stably detect objects with enough training frames on prep stage. It may make an erroneous detection when the image is

overexposed or underexposed, and use more CPU resource. OpenTLD can track targets with only one frame. OpenTLD, as it has ability to avoid low contrast ratio, can escape making erroneous detections. We also found that creating a dynamic training set can increase computation efficiency and decrease FPS over time.

**Keyword:** OpenTLD, Haar-like feature values, OpenCV, Arduino, surveillance systems, target tracking, machine learning



# 目錄

中文摘要 .....	2
ABSTRACT.....	4
目錄 .....	7
圖目錄 .....	9
表目錄 .....	10
第一章 簡介 .....	11
第二章 背景知識 .....	13
2-1 數位影像處理 .....	13
2-1-1 Bitmap .....	13
2-1-2 Alpha Channel .....	14
2-2 OpenCV .....	15
2-3 Viola's method.....	16
2-3-1 Haar-like 特徵 .....	16
2-3-2 積分圖快速運算(Integral Image).....	18
2-4 AdaBoost.....	19

2-5 OpenTLD .....	22
第三章 研究方法 .....	25
3-1 實驗實作 .....	25
3-2 FPS 效能測試 .....	28
3-3 對比度測試 .....	30
3-4 CPU 資源使用測試 .....	32
第四章 研究結果 .....	33
4-1 FPS 效能測試 .....	33
4-2 對比度測試結果 .....	34
4-3 CPU 資源消耗測試結果 .....	36
4-4 綜合比較結果 .....	38
第五章 結論 .....	39
參考文獻 .....	41

## 圖目錄

圖 1 系統主架構 .....	1
圖 2 Haar-like 特徵模板 .....	1
圖 3 AdaBoost 示意圖 .....	1
圖 4 OpenTLD 系統流程圖 .....	1
圖 5 實作硬體架構 .....	1
圖 6 清晨強光誤判圖例 .....	1
圖 7 低對比度誤判圖 .....	1
圖 8 CPU 負載範例圖 .....	33
圖 9 兩種方法 0 到 180 秒 FPS 取樣 .....	1
圖 10 CPU 資源統計圖 .....	1
圖 11 縮小影像後 CPU 資源統計圖 .....	37

## 表目錄

表 1 名稱與動作指令描述表 .....	27
表 2 處理事件的方法表 .....	28
表 3 Viola's method 與 OpenTLD 比較表 .....	38

# 第一章 簡介

由於人們無法記憶所有的事情，為了記錄與回憶，因此發明了多媒體設備記錄無法重現的過往。隨著資訊多媒體技術日新月異，攝影機器的成本愈來愈低，穩定性漸漸提高之後，人們渴望更多美好的影像，因此帶來更多更高難度的攝影需求。

許多攝影師為了捕捉稍縱即逝的珍貴影像，花費了大量的時間與人力在設置拍攝的鏡頭。如果有一個方法可以使鏡頭自動追蹤目標，便可以減少人為操作的時間。現今可以應用的各種情況例如拍攝電影、記錄、教學、運動場景、安全監控等。因此對於多媒體資訊研究的重要性也因此凸顯。

本論文主要的目標為製作一個可以根據影像自行轉動的系統，以減少人力與攝影機的死角。此種自動運鏡系統可以應用在全自動監視攝影機上，進行犯罪追查及記錄動物動向等。

實作一個全自動以臉部辨識為主的追蹤系統，目前有幾種代表性的臉部檢測法。例如類神經網路為基礎的方法[1, 2]、以基因演算法為基礎的檢測法[3]、以 Haar-like 特徵值進行 AdaBoost 演算法[4]、OpenTLD[5]等。

本論文實作一個可以自動追蹤目標的系統，即時性的對接收到的影像做出反饋[6]，此系統流程分成三個部分（如圖 1）：

- (1) 接收影像：透過 OpenCV 提供的函式從攝影機接收影像。
  - (2) 辨識影像：用 Viola's method 或 OpenTLD 處理辨識影像。
  - (3) 控制馬達轉向：根據目標位置判斷是否要控制馬達轉向。
- 我們 使用兩種方法來實作這個系統：

- (1) Viola's method：使用 haar-like 特徵值訓練出來的分類器來進行物件辨識的方法。
- (2) OpenTLD：動態新增訓練樣本追蹤目標的方法。

本論文將這兩種方法進行比較與探討，第二章是介紹相關背景知識，第三章是研究方法及環境配置，第四章是實驗結果及分析，最後則是結論與未來展望。

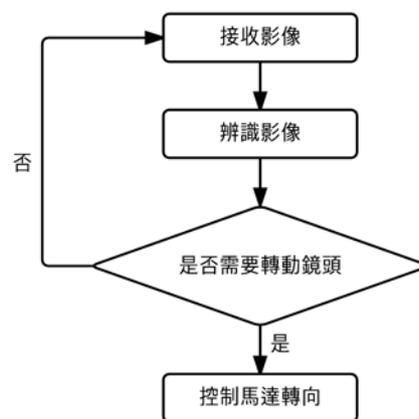


圖 1 系統主架構

## 第二章 背景知識

### 2-1 數位影像處理

從古至今，人類利用文字、圖畫記錄歷史。目的及為健忘的自己留下美好的回憶並且與其他人分享；記錄過錯以便提醒自己或後人不要再犯同樣的錯誤。

在膠片發明前，人們使用繪圖來記錄人、事、物，直到伊士曼柯達公司(Eastman Kodak Company，一般簡稱柯達)發明了膠片，利用感光方式來記錄當時的影像、彩色沖印等等，從此帶動光學攝影的發展。

電腦發明後，科技技術不斷改進，可以為人類減少繁雜的步驟並減少人為失誤以及擁有更好的體驗。然而在電腦的二進制世界裡，如何定義現實生活繽紛的色彩是一個困難的問題。

#### 2-1-1 Bitmap

在電腦的二進制世界與現實世界中該如何定義一張畫裡面的所有顏色與所有位置是一項考驗。所以基本的 Bitmap 被發明出來，

一般稱為點陣圖。主要是利用二維矩陣記錄每個點的位置，使用二進制來記錄每個點的顏色。每一個點的顏色用一個二進制的單位表示。若這個二進制的位數愈大，則可以使用的顏色就愈多，也就需要更大的儲存空間。

舉例來說，由光的三原色紅綠藍 RGB 之光學強度來定義一個顏色，一般螢幕上都是使用這種色彩編碼方式顯現，若使用八位元來記錄三原色的光學強度，則每一個顏色都有 種變化，此時利用光的三原色混和就可以產生 種變化。

一般影印會使用 CMYK 來處理影印工作，即印刷四分色：

C：Cyan

M：Magenta

Y：Yellow

K：Key(black)

印刷四分色與光的三原色不同的一個特性是，光的三原色之光學強度越強時，亮度會越接近白色，而印刷四分色則相反。

## 2-1-2 Alpha Channel

一般的影像編碼都會包含一個 Alpha Channel，主要是用來處理影像透明度。舉例來說，RGB 各佔了八個位元，而 Alpha Channel

也佔八個位元，由 Alpha 的數值來決定透明度，有的表示方法來表示透明程度。

## 2-2 OpenCV

OpenCV(Open Source Computer Vision Library) [7]是一個提供電腦影像處理和機器學習的開源函數庫。由 Intel 發起並開發，以 C++為基底編寫，主要介面也是 C++。許多學術上的影像處理演算法都在 OpenCV 上實作。OpenCV 提供了基本的電腦影像處理函式來推廣機器視覺的研究，是一套開源且最佳化的基礎函數庫，使開發人員的程式更容易閱讀和使用。其目的是為了促進知識的傳播，加快電腦影像應用程式的開發與商品化，使程式更快更簡單的完成。

OpenCV 包含許多模組，簡單的介紹以下幾個基本的模組：

Core：基礎資料結構，有一些基本的函數讓其他模組使用。

Imgproc：處理影像的模組，包含線性和非線性濾鏡、影像幾何轉換、顏色空間轉換、直方圖均衡化之類的處理。

Video：分析影片的模組，提供運動預估、去除背景及物件追蹤的方法。

calib3d：包含基礎幾何多視點演算法、攝影機校準、物件姿勢猜測及 3D 元素重建。

features2d：特徵辨識與特徵比較器。

Objdetect：辨識物件和已定義的類別。

Highgui：包含可簡單擷取影片或圖片的介面和基本的 UI 定義。

GPU：用 GPU 加速的演算法模組。

因此，利用 OpenCV 可以很方便的處理許多影像工作，同時也提供 Java 及 Python 介面，本論文中使用的方法都有使用到 OpenCV 的函數庫來對影像做基本的操作。

## 2-3 Viola's method

Papageorgiou 提出了 Haar-like 特徵值運算方法，用於識別物件或檢測，使電腦自動從影像中區分真實世界中的物件與背景，用來區分混亂的背景與目標。Haar-like 特徵是用來檢測影像中指定位置的相鄰矩形的特徵值，計算每一個矩形的像素灰階值總和相減並取其差值，接著利用這些差值對圖像的子區域進行分類，Viola 以 Haar-like 特徵值製作了一個快速物件辨識系統。[4, 8, 9]

### 2-3-1 Haar-like 特徵

一個矩形 Haar-like 特徵可以定義為矩形中幾個區域的像素灰階總和的差值，可以任意縮放尺寸並放在影像上滑動檢測。這種特性也被稱為 2 矩形特徵（2-rectangle feature）。圖 2 為特徵模板範

例，相鄰兩個矩陣中的灰階像素差值為特徵值，a, b 為基礎的邊緣特徵模板，c 為對角線特徵模板。

後續有人提出更多特徵檢測模板，如邊緣特徵、線性特徵、中心特徵和對角線特徵。特徵模板內有白色和黑色兩種矩形，該模板的特徵值為白色矩形像素和減去黑色矩形像素和。Haar 特徵值反映了圖像的灰度變化情況，例如：臉部的特徵能由矩形特徵簡單的描述，如：眼睛要比臉頰顏色要深，鼻樑兩側比鼻樑顏色要深，嘴巴比周圍顏色要深等。但矩形特徵只對一些簡單的圖形結構做辨識分類，例如邊緣、線段較敏感，所以只能描述特定走向（水平、垂直、對角）的結構。

圖 2 為特徵值計算的範例，計算方法為黑色矩形內的像素加總減白色矩形內的像素加總。

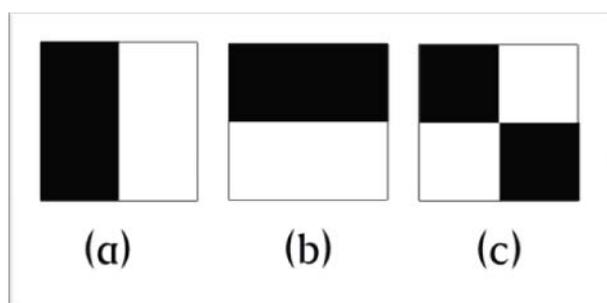


圖 2 Haar-like 特徵模板

## 2-3-2 積分圖快速運算(Integral Image)

積分圖是用來快速運算 Haar-like 特徵值的方法。Haar-like 特徵模板是一個可以任意縮放的矩形，可在影像上滑動檢測。當影像的像素增加之後，整個影像像素的排列組合也會增加，於是 Viola 提出了積分圖算法，積分圖快速運算只要走訪一次圖像就可以求出圖像中所有區域像素的總和，可提高特徵值計算的效率。積分圖主要的概念是將圖像從起點開始到各個點所形成的矩形區域之像素總和作為一個矩陣保存在快取記憶體中。要計算某個區域的像素總和時可以直接利用索引來計算積分圖，不用再重新計算這個區域的像素和。積分圖能夠在多種尺度下，使用相同的時間來計算不同的特徵值，因此提高了檢測速度，積分圖快速運算公式如下：

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y')$$

點  $(x, y)$  的  $ii(x, y)$  是原圖像  $(x, y)$  左上角方向所有像素的和。

這邊  $ii$  就是積分圖(Integral Image)， $i$  則是原始圖，

使用下面公式就可以重複使用之前已經運算過的矩形總和：

$$s^m(x^m, y^m) = s(x^m, y | \mathbf{1}^m) + t^m$$

$$u^m(x^m, y^m) = u^m(x | \mathbf{1}^m, y)$$

## 2-4 AdaBoost

有了檢測圖形的特徵模板跟處理運算圖形的積分圖之後，再利用 AdaBoost 機器學習將訓練樣本集產生一個分類器。AdaBoost 是由 Yoav Freund 和 Robert Schapire 提出的一個機器學習的方法，全文 Adaptive Boosting，簡稱 AdaBoost。

AdaBoost 方法中使用的分類器都是弱分類器，錯誤率較高。但只要弱分類器的分類效果成功率大於 50%，就能在組合分類器的階段提升辨識成功率。錯誤率高於 50% 的弱分類器可以在最終組合多分類器的階段中，給予負數權重同樣也能提升分類效果。[10-12]

AdaBoost 方法是一種疊代算法，在每一輪中加入一個新的弱分類器，直到達到預定足夠小的錯誤率。每一個訓練樣本都被賦予一個權重，表明它被分類器選入訓練集的機率。如果某個樣本點已經被準確地分類，那麼在構造下一個訓練集中，它的權重就被降低；相反，如果某個樣本點沒有被準確地分類，那麼它的權重就得到提

高。通過這樣的方式，AdaBoost 方法能「聚焦於」那些較難分辨的樣本上。以下簡述其步驟：

步驟一，分類器訓練：利用數個正面訓練樣本與負面訓練樣本透過 Haar-like 特徵值來訓練一個弱分類器。Haar-like 特徵值是由 2~4 個矩形組成透過相鄰矩形內的像素總和相減的特徵值。

步驟二，訓練集分類權重更新：將弱分類器產生分類錯誤的樣本權重增加後進行下個弱分類器訓練，重複執行後得到多個弱分類器。

步驟三，級聯分類器：利用多個弱分類器集合來對影像進行分類，每個分類器都握有不一樣的權重。將訓練出來的眾多分類器集合，成為一個強分類器，如圖 3。

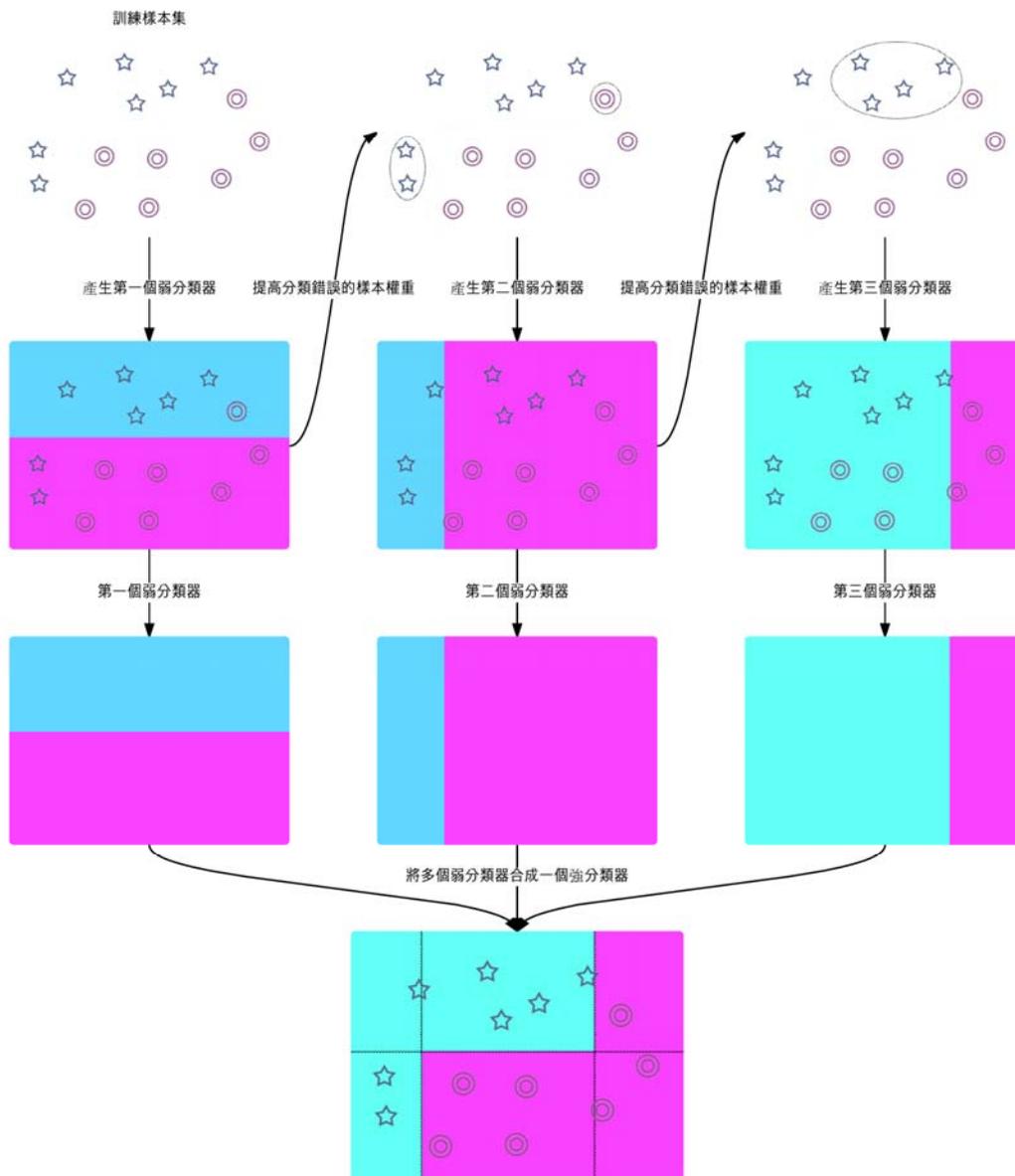


圖 3 AdaBoost 示意圖

## 2-5 OpenTLD

Viola's method 系統利用 Adaboost 演算法輸出分類器來對影像進行分類。此方法必須要有相當大的訓練樣本進行訓練才能趨於準確，若在拍攝時無法提供足夠的訓練樣本，準確率就會降低。由 Kalal, Z., K 提出的 OpenTLD(Open Tracking Learning Detection)則沒有此要求，不需要任何事前的訓練集，只需要一個任意樣本就可以執行。OpenTLD 將追蹤工作分為三個部分 Tracking, Learning, Detection[5]。Tracker 負責追蹤出現在每個 frame 的目標物件；Detector 負責辨識目前整個影像內目標物件的位置；Learning 預測 Detector 的錯誤和更新訓練集以避免未來出錯。圖 4 為 OpenTLD 系統流程圖，透過 Learning 模組不斷新增訓練樣本增加 Detection 準確率，Detection 重新初始化 Tracking 模組使追蹤能力提高，Detection 與 Tracking 輸出的結果會回報給 Learning 模組判斷是否有錯誤，並且視情況更新訓練樣本集使整個系統準確率提高。

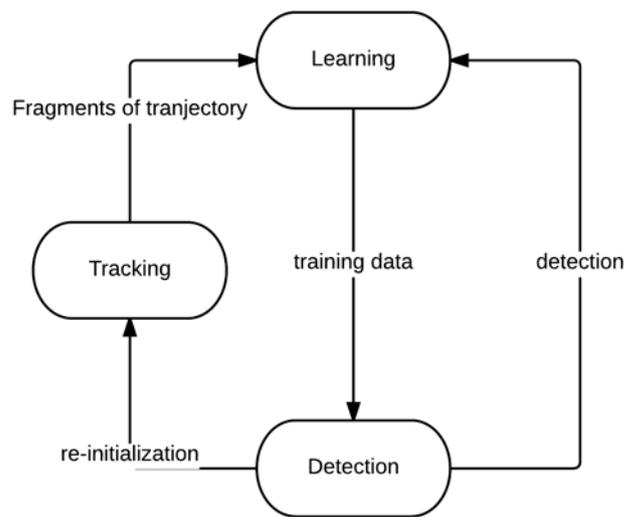


圖 4 OpenTLD 系統流程圖

OpenTLD 的詳細步驟為下：

步驟一，目標樣本輸入：首先從影像選擇一個目標作為追蹤樣本，例如臉或是眼睛給 Tracker 進行追蹤，Tracker 主要根據物體運動的方向追蹤跟隨目標。

步驟二，檢測影像：在 Tracker 追蹤的同時，Detector 會對整個影像做辨識。檢測的方式通常是使用特徵值或是等比例縮放影像辨識[13]。

步驟三，學習與更新樣本：Learning 給定一個目標之後，在接下來的影片串流自我學習，利用半監督式學習方法 semi-supervised learning[14]，使用有被標記正面或負面樣本跟沒有被標記的樣本來做自我學習的工作。使用期望值最大化 (Expectation Maximization)[15]的方法來處理沒有標記正面或負面的樣本達成自我學習(Self-learning)。

步驟四，新增樣本集：從一個被標記正面或負面的樣本初始化開始訓練，對沒有標記的樣本進行評價。將最有信心的反饋加入訓練集之後更新分類器。

步驟五，迴避錯誤：Detector 對整個輸入影像進行掃描檢測，利用過去所學習到的訓練集來尋找目標。Detector 有可能會出現兩種錯誤，即非目標判斷成目標或是目標誤判，步驟四的 Learning 會觀察 Tracker 和 Detector 兩者輸出的關係，判斷 Detector 是否錯誤並且產生訓練樣本，加入訓練集來避免未來產生同樣的錯誤。利用這樣的方法可以降低 Detector 產生錯誤並且避免背景干擾、光源變化等問題。

## 第三章 研究方法

### 3-1 實驗實作

Arduino [16]是一個開源的跨平台電子控制介面，可以簡單的操作硬體和軟體。Arduino 提供類似 C 及 JAVA 語言的程式介面，適合用來實作系統。將 Arduino 與辨識方法結合之後實作的硬體架構如圖 5，控制的方法是利用 Servo.h 的函數庫來控制馬達。

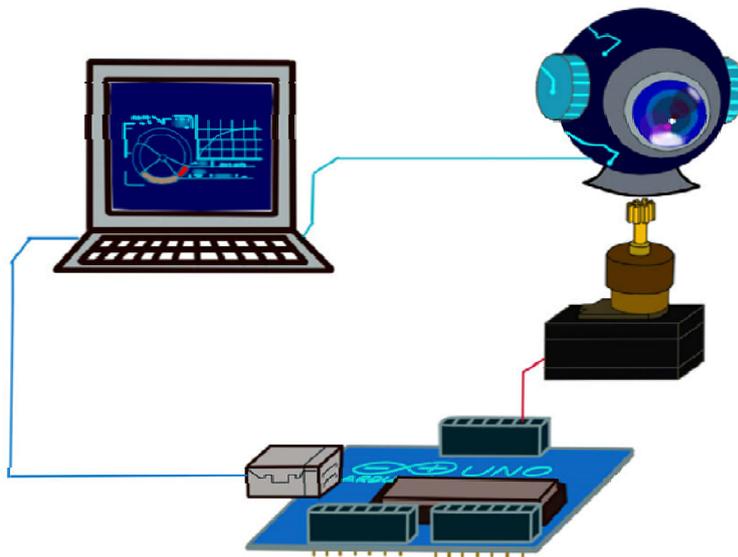


圖 5 實作硬體架構

Arduino 控制步驟如下：

步驟一，啟動連線：啟動 Arduino，開啟 Serial Port。

步驟二，參數初始化：馬達參數初始化、角度校正，程式進入讀取迴圈，從 Serial Port 接收訊號，根據訊號來判斷馬達轉向角度後控制馬達。

步驟三，電腦端控制：電腦從攝影機收到影像後進行處理，判斷該如何轉動馬達後傳訊號給 Serial Port。

預測物件如果要超出螢幕外，就讓鏡頭轉動。選定一個優秀的方法來辨識物件，追求準確率與低反饋時間，如果反饋時間太長就可能失去目標。利用特殊的編碼機制，跳過或是參考 frame 的方法，盡可能降低運算量，來達到更低的反饋時間[9, 17]。從接收處理完影像之後若成功追蹤到目標，則輸出物件座標中心點(x,y)。輸出指令給 Arduino，讓 Arduino 控制攝影機底部的馬達完成轉動工作。實驗的環境是 2010 年末的 MacBook Air, Intel Core 2 Duo 1.86 GHz, 4GB 1067 MHz DDR3, OS X 10.9, OpenCV version 2.4.7。運算量會隨著 pixels 增加而成長，使用 H.264/AVC 做視訊編碼或是縮小影像輸入可以有效降低運算量。

表 1 為名稱與動作指令描述為各屬性定義與描述，表 2 為輸出判斷流程。

表 1 名稱與動作指令描述表

名稱	描述
target	目標物件
tx	物件座標中心點 x 軸
ty	物件座標中心點 y 軸
W	輸入影像寬度
H	輸入影像高度
Left	輸出訊號，馬達左旋
Right	輸出訊號，馬達右旋
Up	輸出訊號，馬達上旋
Down	輸出訊號，馬達下旋
None	不做任何輸出控制訊號
otherwise	不符合任何判斷式

表 2 處理事件的方法表

動作指令	事件
Left	If $tx > W * 3/4$
Right	If $tx < W/4$
Up	If $ty < H/4$
Down	If $ty > H * 3/4$
None	otherwise

### 3-2 FPS 效能測試

視覺暫留 (Persistence of vision) 用來表示多個影像快速抽換，使人腦以為是連續影像，要達到視覺暫留的效果必須是兩張影像間隔 0.1-0.4 秒左右，也就是兩張 frame 之間隔時間為 100ms-400ms。一般電影的畫面更新頻率為每秒 24 張 frame，也就是 24 FPS (frame per second)，隨著科技的進步將會有更高的 FPS 產品推出。

在實際體驗中，60 FPS 相對於 30 FPS 有著更好的體驗。一般螢幕更新頻率通常是 60Hz，此時人眼會感受到非常流暢的影像體

驗。當 FPS 超過 85 之後，此時已經到達大腦處理影像的極限，人眼已經無法分辨更高更新率的差異。

自動運鏡系統輸出 FPS 的流程是：

1. 從攝影機接收影像：使用 OpenCV 函數從攝影機接收影像。
2. 處理辨識影像：使用 Viola's method 或是 OpenTLD 進行影像處理運算。
3. 輸出 frame：輸出影像處理到電腦螢幕上。

因此，要測量處理影像的速度可以由 FPS 為依據來測定。

處理圖形是非常消耗資源的工作，而如何降低資源的消耗並且提升準確率是一難題。一般影像串流為 30 FPS，每一秒鐘要處理 30 張 frame，在每一張 frame 尋找目標並且追蹤。FPS 的方式為記錄一張 frame 輸出時的系統時間，減掉下一張 frame 輸出的系統時間，假設第一張 frame 為  $frames_1$ ，下一張 frame 為  $frames_2$ ，公式為：

$$FPS = \frac{1sec}{frames_2 - frames_1}$$

由此公式可以知道 FPS 是一張 frame 輸出後，取得、處理影像後，輸出下一張 frame 的時間。以 FPS 採樣作為測試依據，採樣時間為 0 秒至 180 秒，追蹤一個特定臉型。採樣的方式為 180 秒，每

一秒鐘對輸出的 FPS 值進行記錄。記錄兩種方法所輸出的 FPS。將兩種方法所做出來的記錄畫成折線圖做比較。

### 3-3 對比度測試

追蹤影像的測試過程中，我們發現 Viola's method 容易被對比度較低的影像干擾而造成誤判，將程式寫成辨識到人臉就儲存影像。圖 6 為清晨 6 點 54 分，因為高強度光源造成影像局部區域對比度降低而導致誤判，測試地點為實驗室門口，該時段一小時內出現了 16 次誤判。因此推測 Viola's method 在低對比影像輸出的情況下容易產生誤判。



圖 6 清晨強光誤判圖例

圖 7 為凌晨 3 點 43 分正要天亮時的誤判圖，從 12 點 42 分到 3 點 57 分，共出現 521 張類似誤判的情況。

由上面的觀察發現影像局部區域內的相鄰像素值差異不高，使影像的特徵值不明顯。因此設計了類似的對比度環境來進行測試，比較 Viola's method 和 OpenTLD 方法的成功率。



圖 7 低對比度誤判圖

## 3-4 CPU 資源使用測試

建立一個自動運鏡系統，若實作在移動裝置上就必須考慮 CPU 的限制與電量消耗的問題。如果可以用更小的 CPU 消耗來達到同樣的目標，則 CPU 使用量愈低愈好。因此 CPU 測試是在同樣環境底下，記錄兩種方法消耗 CPU 資源的程度，圖 8 為範例。

CPU 資源使用測試方法為記錄從 0 秒到 180 秒，每秒鐘記錄一次 CPU 使用率。將兩次的記錄做成折線圖來做比較。

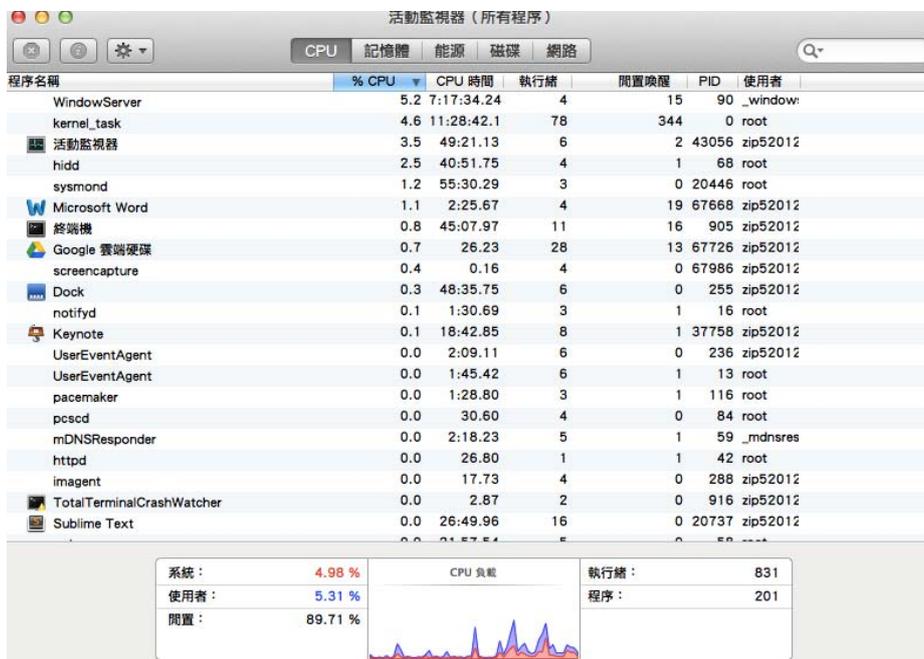


圖 8 CPU 負載範例圖

## 第四章 研究結果

### 4-1 FPS 效能測試

以 FPS 來測試兩種方法的穩定度與效率，使用測試的機器為 2010 年末 MacBook Air，CPU 處理器為 1.86GHz Intel Core 2 Duo。

根據實驗結果，OpenTLD 的平均 FPS 是 11.6。Viola's method 的平均 FPS 是 6.1 較慢。

圖 9 為 FPS 測試結果，從 0 到 180 秒每一秒記錄一次。追蹤單一不固定位置臉型，臉型會任意不規律移動。兩種方法追蹤臉型之後輸出的 FPS 並畫出折線圖，縱軸為 FPS 值，橫軸為時間。

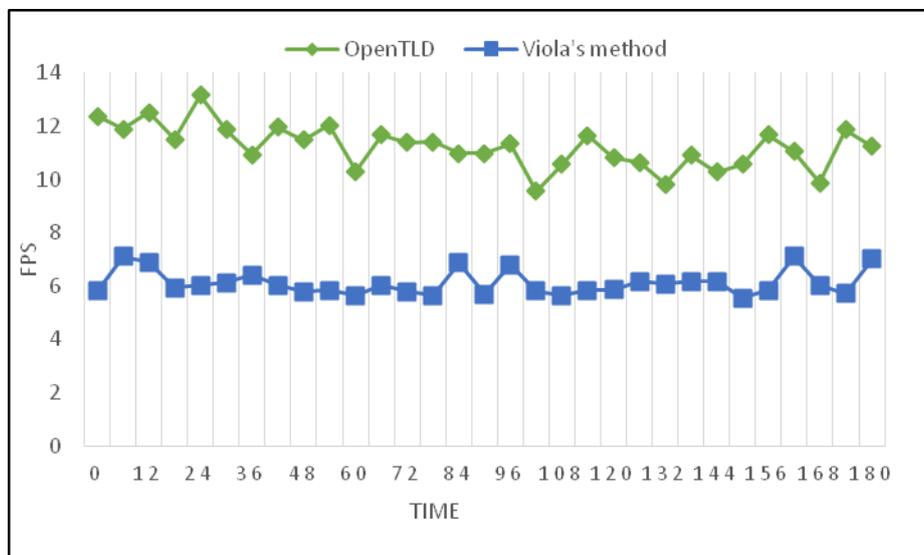


圖 9 兩種方法 0 到 180 秒 FPS 取樣

## 4-2 對比度測試結果

在影像辨識中，普遍操作的方法是對每個 pixel 的灰階值做處理。將圖 6 與圖 7 每個 pixel 灰階值計算平均值與標準差。

圖 6 的灰階平均值為 58.57 灰階值，標準差為 78.80 灰階值。

圖 7 的灰階平均值為 76.14 灰階值，標準差為 36.80 灰階值。

一般情況下影像灰階標準差為 60 灰階值左右。

圖 6 有過曝現象導致亮部區域全亮、暗部區域全暗失去細節。亮部與暗部兩個區域加總計算後的灰階像素標準差比一般狀況下高。

圖 7 的情況則是曝光不足導致整個影像的細節不明顯、對比度降低，使整個影像的灰階像素標準差降低，使特徵值不明顯。推論像素灰階標準差越低，越容易使 Viola's method 產生誤判。

為了要製造低對比度的影像，利用燈光、手電筒控制環境光源來重製近似於圖 7 的低對比度影像。辨識單一臉型，臉型會在螢幕上任意不規則移動。從 0 到 180 秒每一秒鐘進行一次辨識紀錄，成功率為 R，成功辨識次數總數為 A，所有計次辨識總數為 G，公式如下：

$$R = \frac{A}{G}$$

進行實驗後發現在對比度較低的情況下進行測試 0 到 180 秒每秒辨識 1 次。Viola's method 成功辨識次數為 94 次，成功辨識率是 52%。

OpenTLD 成功辨識次數為 155 次。由於 OpenTLD 會自動即時更新訓練集並過濾低對比的背景，因此不易受低對比影像影響，其成功辨識率為 86%。因此如果要使用 Viola's method 追蹤的影像要注意曝光程度，過曝或是曝光不足都容易使影像失去細節與降低對比度造成誤判，而 OpenTLD 則不易受到低對比度影響。

### 4-3 CPU 資源消耗測試結果

在同一個環境底下，用兩種方法追蹤同一個臉型。臉型在畫面中不規則移動，測試時間為 0 到 180 秒，每秒鐘記錄一次 CPU 使用負載。

圖 10 為 CPU 資源消耗統計圖，從 0 到 180 秒每秒取一次 CPU 使用率。當我們執行 Viola's method 時因為運算量較大，CPU 進入超頻模式，平均使用率有 170%。OpenTLD 則是維持在 98%使用率。

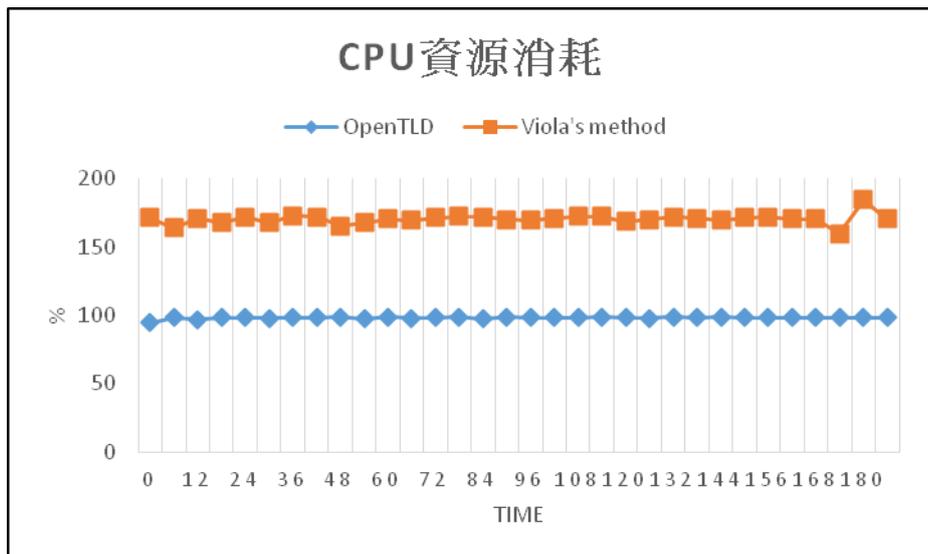


圖 10 CPU 資源統計圖

圖 11 為縮小影像為原本的三分之一後進行的測試，避免超頻現象。

Viola's method 的平均 CPU 使用率為 96.11%。

OpenTLD 的平均 CPU 使用率為 54.2%。

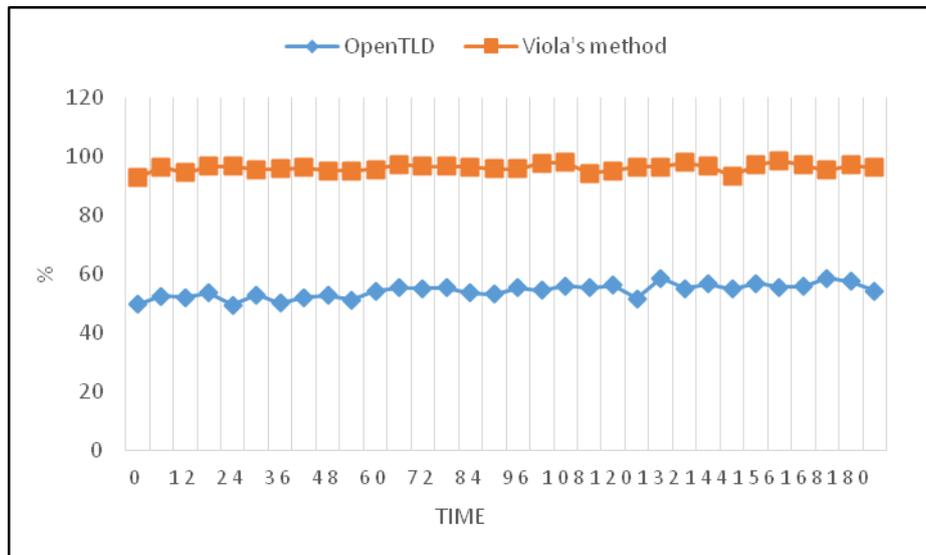


圖 11 縮小影像後 CPU 資源統計圖

## 4-4 綜合比較結果

表 3 為 Viola's method 與 OpenTLD 的各項比較表，Viola's method 的學習方法需要使用數個已標記的訓練樣本來執行，OpenTLD 只需要使用一張 frame 作為樣本就可以執行。

為了避免 Viola's method 超頻的現象，將影像縮小為原本的三分之一降低運算量。表 3 第五行為縮小影像後 CPU 使用率。

Viola's method 反映的 FPS 較低、使用的 CPU 資源較高。

OpenTLD 會動態更新訓練集，FPS 輸出會隨著時間減少，而使用 CPU 資源穩定不變。

表 3 Viola's method 與 OpenTLD 比較表

	Viola's method	OpenTLD
Learning Method	Adaboost	Semi-supervised Learning
低對比度辨識率	52.22%	86.11%
FPS (Average)	6.105±0.471	11.23±0.818
平均 CPU 使用率	170.329%	98.1438%
Resize image CPU	96.11%	54.2%

## 第五章 結論

自動運鏡應用在監視系統上可以減少攝影機的死角，使用在拍攝紀錄上也可以減少人力，我們嘗試做出一個有辦法自動追蹤目標的攝影機。

首先遇到的挑戰是該使用哪種方法可以快速的反應且穩定的追蹤目標。Viola's method 被證實是快速且穩定的追蹤目標，至今仍然在 OpenCV 上實作，直到 OpenTLD 被提出有更快的方法進行物件辨識追蹤，因此我們在這裡將兩種方法進行比較。

經過實驗，Viola's method 的缺點是需要事先提供數個訓練樣本來進行訓練。在執行階段對整個影像進行滑動掃描辨識，所以會消耗較多的 CPU 資源，且使用特徵值方法進行辨識會在局部低對比的情況下容易誤判。但其優點則是在執行期間 FPS 穩定不變。

至於 OpenTLD 則不需要數個訓練樣本做訓練，只需要一張 frame 作為樣本就可以開始追蹤。過濾低對比度的背景可以降低 CPU 使用的資源，因此 FPS 輸出也較快速。缺點是動態新增訓練樣本雖然可以增加追蹤物件的準確率，但是增加訓練樣本之後 FPS

會逐漸降低。動態增加訓練樣本有機會造成追蹤目標與原先設定的目標有少許差異。

在監控的實際應用上，若追蹤的目標有足夠的訓練樣本。可以使用 Viola's method 穩定的追蹤目標，但是要注意曝光程度，過曝或是曝光不足容易造成誤判。而在沒有足夠訓練樣本的情況下，OpenTLD 只需要一張 frame 就可以進行追蹤目標，但是選定追蹤的目標必須準確，否則會造成後面追蹤的目標與原本選定的目標少許不同。

## 参考文献

[1] F. Smach, M. Atri, J. Mitéran, and M. Abid, "Design of a Neural Networks Classifier for Face Detection," in *World Academy of Science, Engineering and Technology*, 2005, pp. 274-277.

[2] K. Nam-Chul, J. Ick-Hoon, K. Jong-Dae, and K. Jae-Kyoon, "Real-Time Automatic Target Tracking Based on Spatio-Temporal Gradient Method with Generalized Least Square Estimation," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, 1988, pp. 1088-1091 vol.2.

[3] Y. Hori, K. Shimizu, Y. Nakamura, and T. Kuroda, "A Real-Time Multi Face Detection Technique Using Positive-Negative Lines-of-Face Template," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, 2004, pp. 765-768 Vol.1.

[4] P. A. Viola and M. J. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," in *Computer Vision and Pattern Recognition*, 2001, pp. 511-518.

[5] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-Learning-Detection," in *IEEE Trans Pattern Anal Mach Intell*, 2011.

[6] A. Khan, S. S. A. Ali, M. Omer, and K. Raza, "Real-Time Automatic Target Detection and Tracking Using Visual Feedback," in *Intelligent and Advanced Systems (ICIAS), 2014 5th International Conference on*, 2014, pp. 1-5.

- [7] "[Http://Opencv.Org/](http://OpenCV.org/)."
- [8] C. P. Papageorgiou, M. Oren, and T. Poggio, "A General Framework for Object Detection," in *Proceedings of the Sixth International Conference on Computer Vision*, 1998, p. 555.
- [9] G. Swaminathan, V. Venkoparao, and S. Bedros, "Multiple Appearance Models for Face Tracking in Surveillance Videos," in *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*, 2007, pp. 383-387.
- [10] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting," in *Proceedings of the Second European Conference on Computational Learning Theory*, 1995, pp. 23-37.
- [11] A. K. Rainer Lienhart, Vadim Pisarevsky, "Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection," in *Lecture Notes in Computer Science*, 2002.
- [12] Y. Zhiguo, Y. Fang, W. Jian, S. Yongjie, L. Chao, and S. Mingxia, "Face Orientation Detection in Video Stream Based on Harr-Like Feature and Lqv Classifier for Civil Video Surveillance," in *Smart and Sustainable City 2013 (ICSSC 2013), IET International Conference on*, 2013, pp. 161-165.
- [13] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *Int. J. Comput. Vision*, 2004, pp. 91-110.
- [14] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, "Introduction to Semi-Supervised Learning," in *Morgan and Claypool Publishers*, 2009, p. 130.

[15] J. Bilmes, "A Gentle Tutorial of the Em Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," in *International Computer Science Institute*, 1998.

[16] "[Http://Www.Arduino.Cc/.](http://www.Arduino.Cc/)"

[17] P. Gorur and B. Amrutur, "Skip Decision and Reference Frame Selection for Low-Complexity H.264/Avc Surveillance Video Coding," in *Circuits and Systems for Video Technology, IEEE Transactions on*, 2014, pp. 1156-1169.