

Indoor Localization Based on Deep Neural Network with Cameras

Liang-Yu Lin 、 Chow-Sing Lin 、 Yu-Hsiang Cheng
Department of Computer Science and Information Engineering
National University of Tainan
*E-mail : mikelin@mail.nutn.edu.tw

Abstract

With the advance of localization technology, wireless localization methods can position humans rapidly. However, the attenuation of signal transmission in the indoor environment caused by the Path loss and the noise leads to the instability of localization. Proposed by Google recently, Tensorflow Object Detection API, which is structured by a SSD neural network model, can detect people accurately. In this paper, we apply the Tensorflow Object Detection API to detect humans, and transform their images coordinates to real world positions with Inverse Perspective Mapping (IPM). We can track humans in camera sequences accurately and rapidly localize them at the cost of lowering the frame rate. Our experimental results show that our proposed method can correctly detect humans once a second, and the accuracy of positioning then can be improved by less than 100cm.

Keyword: deep learning 、 neuron network 、 machine learning 、 object detection 、 inverse perspective mapping

* Corresponding author:mikelin@mail.nut.edu.tw

DOI : 10.3966/2223448920190400901003

基於深度神經網路與攝影機之室內定位

林亮宇、林朝興*、鄭宇翔
國立臺南大學 資訊工程學系
*E-mail: mikelin@mail.nutn.edu.tw

摘要

隨著定位技術的進步，使用無線定位技術能夠快速地定位人類。然而，位於室內環境的無線訊號傳輸，與能量衰減以及干擾息息相關，導致定位不穩定。最近，Google 釋出使用 Single Shot MultiBox Detector(SSD)模型建構的 Tensorflow Object Detection API，可用於偵測人類。在本研究中，我們使用 Tensorflow Object Detection API 來進行偵測人物，並使用逆透視投影演算法(Inverse Perspective Mapping, IPM)精確地將其影像座標轉換為真實世界座標，透過降低影像序列幀率來準確與快速地進行偵測，並且定位人物。本研究提出的室內定位方法能夠以每秒一次準確地偵測人物，並將定位精準度提升到 100 厘米內。

關鍵字：深度學習、神經網路、機器學習、物體偵測、逆透視投影演算法

1. 緒論

在定位技術中，GPS 為常見的定位方式[2]，可使用於定位室外的人物所在位置，亦可用於追蹤車輛目前的位置，但因為現今高樓林立，GPS 訊號在室內會因為建築物的阻隔，導致訊號的衰減，而無法正確的定位[3]，即使能夠取得 GPS 訊號，也無法得知樓層的訊息。

隨著無線技術的發展，許多室內定位的技術也隨之誕生，例如：紅外線、藍芽、RFID[4]-[10]。各種不同的室內定位方式也有不同的特點，以紅外線為例，使用紅外線定位的精準度較高，但是因為紅外線的傳輸距離較短，若在環境複雜的室內空間中，布局需比其他定位方式更複雜，導致成本提高；而 RFID 的作用範圍較小，且為參與式之定位方法。若在障礙物較多的區域中，使用無線通訊技術進行室內定位，可能因為訊號的反射、折射或者被障礙物所阻擋，導致定位難度增加。為了維持定位精準度，在使用多種無線定位的方法下，可能導致成本增加。Li Qiyue 等人在[11]中提到，在較複雜的空間中，使用訊號強度波紋圖(Fingerprint)會因為無線訊號的反射、折射或是訊號的衰減，導致定位精準度降低，所以搭配 Wi-Fi 訊號，來進行複雜空間中的室內定位能夠有效的提升定位的精準度；但是若是使用 Wi-Fi，需要在特定區域中架設多台 AP，導致成本提升。

若使用事先架設之固定攝影機，從高處拍攝特定區域中的影像，在大部分情況下，俯視的攝影機拍攝之影像還是能夠涵蓋目標人物。另外，因為目前有許多圖像處理的函式庫，如：OpenCV、深度學習神經網路，能夠更方便、更快速的進行圖像處理的動作；並且因為在一張影像中，能夠表達的事物比用文字敘述的方式來得更多、更加清楚明瞭，所以本研究希望以攝影機影像透過影像處理的方式，快速及精確地偵測目標人物，取得目標人物的影像座標。因為現今很多場所可能基於某些目的會架設攝影機或是監視器來監控室內環境，可能是監視人們的行動亦或是監視物品儲藏情況，對於取得攝影機影像十分便利，若是應用在百貨公司，透過室內定位系統可以得知目標人物在室內空間的位置，可以進行推薦商品的動作，或是百貨公司可以統計人們在各個地方停留的時間。

本研究提出的室內定位系統流程圖如圖 1，大致上可將研究的目標分成偵測以及計算人物座標兩個部分。首先，將事前架設好的攝影機拍攝的影像序列輸入到系統中，接著對影像序列中的幀進行人物偵測，取得目標人物在畫面中的影像座標，最後藉著影像座標來還原三維的真實世界座標。

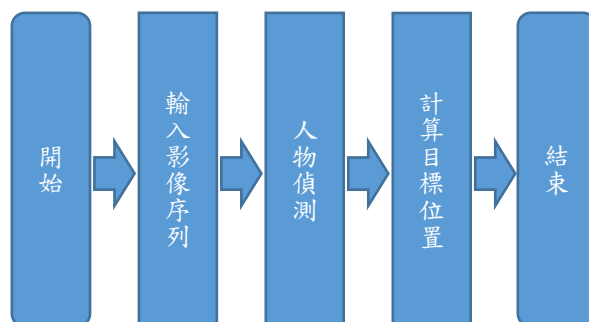


圖 1 室內定位系統流程圖

首先，為了定位目標人物的真實世界座標，必須先取得目標人物在畫面中的影像座標，這個部分須仰賴當今許多函式庫中的人物偵測的方法；偵測的部分目標在於希望將

畫面中的所有人物標示出來，即使人物在畫面中的上半身或是下半身被遮蔽，導致在畫面中無法看見完整的人身。在第 5 節中，將比較 OpenCV 與 Tensorflow API 的偵測效果。

攝影機所拍攝的影像能夠將三維世界座標中的畫面以二維的影像保存下來，但三維轉換為二維將損失空間中的深度資訊，若透過深度相機所拍攝的影像雖然能夠記錄深度資訊，但是考慮到成本以及普及性，在本研究中使用的是 CCD 相機所拍攝的影像，所以需要藉由逆透視投影演算法(Inverse Perspective Mapping, IPM)，將二維的影像座標轉換為真實世界中的三維座標。

郭偉凱在[1]的實驗結果中提到，利用架設在車內的 CCD 相機所拍攝的影像，使用逆透視投影轉換演算法來計算與前車的距離，若單純使用逆透視投影演算法來計算距離時，在實際距離為 10 公尺時，誤差只有 1.28%，對於本研究希望在特定區域中欲計算出人物的座標已經十分準確，所以計算目標人物座標的部分，我們決定使用逆透視投影演算法將被偵測之人物的影像座標轉換為真實世界座標來達到室內定位的目標。

本研究在第 2 節將解釋人物偵測的方法，第 3 節解釋計算真實世界座標的方法，第 4 節討論特殊情況下如何轉換目標人物的座標，第 5 節實驗的結果以及此研究的效能，最後第 6 節是結論以及參考文獻。

2. 人物偵測

人物偵測的目標在於將畫面中的人物標示出來，即使人物的半身被遮蔽，只有部分部位在畫面中。偵測的方法有許多種，此處以 OpenCV 與 Tensorflow object detection API 做比較。

2.1. OpenCV 偵測方法

OpenCV 使用 Histogram of oriented gradients(HOG)特徵[12]進行人物偵測，優點：偵測速度快；缺點：偵測錯誤率高。

HOG 特徵是一種在電腦視覺和圖像處理中用來進行物體檢測的特徵描述子，利用計算和統計圖像局部區域的梯度方向直方圖來構成特徵，將特徵搭配 SVM 分類器，已被廣泛用於圖像辨識技術中。

提取 HOG 特徵的步驟如下：第一步將影像做 Gamma 校正，對輸入影像進行顏色空間的標準化(歸一化)，目的是調節影像的對比度，降低影像局部的陰影和光照變化所造成的影響；第二步計算影像中每個像素點的梯度大小以及方向；第三步將圖像分為若干個 cell，每個 cell 由數個像素構成，例如 3×3 個像素為一個 cell，統計 cell 中所有像素點的梯度，即可得到梯度直方圖；第四步與第三步類似，將若干個 cell 組成一個 block，例如 2×2 個 cell 組成一個 block，將 block 內的各個 cell 的特徵向量串連起來即可得到此 block 的特徵向量；第五步收集所有重疊的 block 的特徵，最終形成完整影像的 HOG 特徵，最後再將此特徵，透過 SVM 分類器，可達成人物辨識的目標。

2.2. Tensorflow Object Detection API 偵測方法

Huang J 等人在[13]中，使用各種深度學習神經網路的模型來進行人物的偵測，在本研究中使用 Single Shot MultiBox Detector(SSD)模型[14]來進行人物偵測，所以在此將說明 SSD 之運作模式，SSD 模型之架構圖如下圖 2。

Wei Liu 等人在[14]中提到，SSD 是基於一個向前傳播的卷積神經網路，產生一系列固定大小的 bounding boxes，以及每一個 box 中包含物體的可能性，也就是 score。最後，進行非極大值抑制(Non-maximum suppression)得到最後的 predictions。

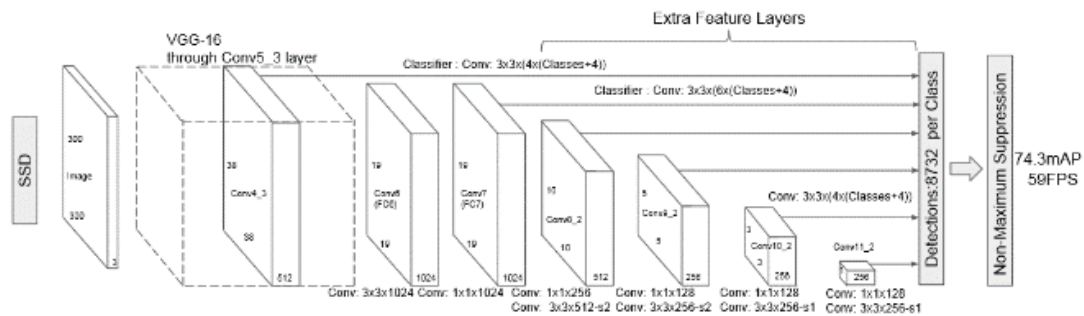


圖 2 SSD 模型架構圖[14]

2.3. 人物被遮蔽之偵測效果比較

在實驗中發現，使用 OpenCV 的偵測方法，當人物距離攝影機較近的時候，只會有上半身出現在畫面中，將無法偵測到此人物，這可能是因為 OpenCV 偵測方法的訓練資料是以人的全身進行訓練，所以當人物只有半身出現在畫面中的時候，OpenCV 的偵測方法無法準確偵測到目標人物，但使用 Google 釋出的 Tensorflow Object Detection API[15]來進行人物偵測，即使人物沒有全身都在畫面中，也能夠精準的偵測出此人物，並且將其標示在圖像上，實驗結果如圖 3 所示，圖 3 (a)使用 OpenCV 在人物只有上半身在畫面中時無法準確地偵測出人物，甚至會偵測到一些背景的其他物體，圖 3 (b)使用 Tensorflow API 即使人物並沒有全身都在畫面當中，亦可偵測出人物，所以我們決定在偵測人物的部分使用 Tensorflow object detection API 來進行人物偵測。



(a)使用 OpenCV 進行偵測



(b)使用 Tensorflow API 進行偵測

圖 3 兩種偵測方法在人物只有上半身在畫面中之比較

3. 人物定位

由第 2 節的方法，偵測出畫面中的人物之後，可以得知目標人物的影像座標，為了定位人物，需要將影像座標轉換為真實世界之座標，我們決定使用逆透視投影演算法來

將二維的影像座標，轉換為三維的真實世界座標。

3.1. 逆透視投影演算法

攝影機能夠將原本三維空間中的畫面以二維的影像保存下來，但是這將會失去物體在空間中的深度資訊，若想要將二維影像座標還原成三維的真實世界座標可利用逆透視投影演算法。

逆透視投影演算法是利用相機之世界座標、相機之視角，以及拍攝影像時相機的旋轉角以及俯角作為參數，將像素之二維影像座標轉換為三維真實世界座標。

3.1.1. 取得相機之視角

相機之水平以及垂直視角，是由相機鏡頭的焦距以及感光元件之尺寸而定。計算公式如式(1)。

$$\alpha = 2 \tan^{-1} \frac{d}{2f} \quad (1)$$

其中 d 為感光元件水平或是垂直方向的尺寸，而 f 為相機鏡頭的焦距，如下圖 4、圖 5。

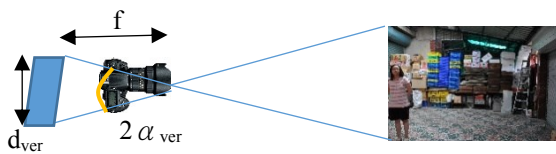


圖 4 垂直視角示意圖

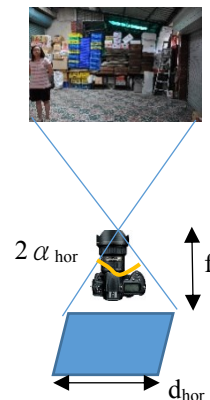


圖 5 水平視角示意圖

3.1.2. 取得旋轉角以及俯角

如圖 6(a)，將視軸 σ 投影到 $Z=0$ 的平面上形成 η 向量與 X 軸方向所形成的角度稱為旋轉角；如圖 6(b)將視軸 σ 與 X 軸方向所形成的角度稱為俯角。

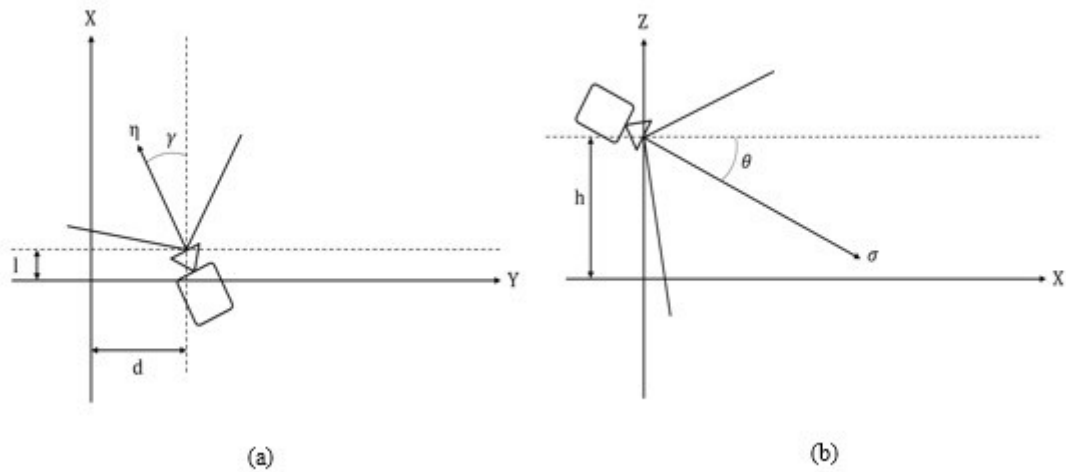


圖 6 相機各項參數示意圖

3.1.3. 計算公式

取得各項參數之後，接著利用式(2)、式(3)的轉換公式，將影像座標轉換為真實世界座標。其中 m, n 為影像解析度， u, v 為像素之二維影像座標，以及 l, d, h 為相機的世界座標。

$$x(u, v) = h \times \cot \left[(\theta - \alpha_{ver}) + u \frac{2\alpha_{ver}}{n-1} \right] \times \cos \left[(\gamma - \alpha_{hor}) + v \frac{2\alpha_{hor}}{m-1} \right] + l \quad (2)$$

$$y(u, v) = h \times \cot \left[(\theta - \alpha_{ver}) + u \frac{2\alpha_{ver}}{n-1} \right] \times \sin \left[(\gamma - \alpha_{hor}) + v \frac{2\alpha_{hor}}{m-1} \right] + d \quad (3)$$

3.1.4. 效果評估

使用第 2 節所提到之偵測方法，可取得標示目標人物矩形之底邊中點的二維影像座標，並將其代入式(2)、式(3)，即可將影像座標轉換為真實世界中，目標在地表平面上的座標。本方法轉換之效果將在第 5.3 節進行誤差之評估。

4. 特殊情形之座標轉換

我們以偵測結果底邊中點的影像座標來代表目標人物，而人們在室內空間中可能會隨意地走動，無法保證在所有情況下，攝影機都能完整拍攝到人物全身的影像，所以本研究提出幾種人物位在攝影畫面中可能的情況，並說明如何計算腳底之影像座標。

室內定位系統會每秒偵測一次畫面中的人物，並且計算人物的長寬比，透過持續的計算並且記錄人物長寬比，可以得知人物是否已經全身進入畫面，使用適當的影像座標進行轉換。我們將此特殊情形分成三種情況，並且在下方詳細說明。

4.1. 目標人物進入或離開畫面下方

當人物的長寬比與前一秒的長寬比變化較大，並且目標人物接近畫面底部，我們的系統會將此目標人物視為正在進入或是離開畫面下方，此時，人物的下半身通常會被遮

蔽，無法從畫面中看到人物的腳底，如圖 7 所示，若是用底邊中點則無法準確的代表目標人物的位置。

在此情況中，系統會使用先前紀錄目標人物的最大長寬比，以及偵測結果的寬，反推出人物腳底的座標，示意圖如圖 8，透過式(4)、式(5)推算出目標人物的腳底的影像座標，最後再透過逆透視投影轉換演算法，轉換目標人物的影像座標。

$$(u, v) = \left(\frac{left + right}{2}, top + rate * w \right) \quad (4)$$

$$w = right - left \quad (5)$$

其中，(u, v)為代表此人物的影像座標點，left、right、top 為偵測結果的邊界座標值，rate 則是在定位過程中所記錄的目標人物的最大長寬比例。

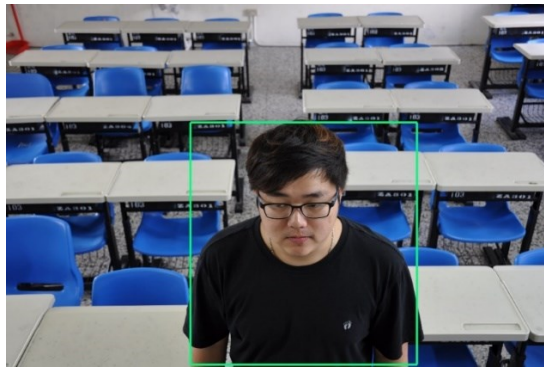


圖 7 人物下半身被遮蔽

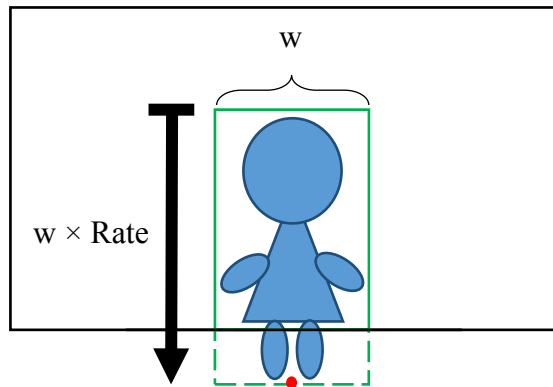


圖 8 計算腳底座標示意圖

4.2. 目標人物全身在畫面中

當目標人物長寬比與上一秒的長寬比相近時，室內定位系統會將此目標人物視為全身在畫面中，此時目標人物的腳底可在畫面中被看見，如下圖 9，所以在這個情況下，直接以偵測結果的底邊中點使用逆透視投影轉換演算法，即可將目標人物的影像座標轉換為真實世界座標。

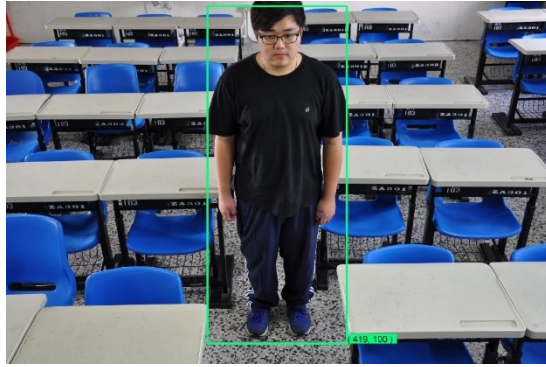


圖 9 人物全身進入畫面中

4.3. 目標人物進入或離開畫面上方

當目標人物長寬比與上一秒的長寬比變化較大，並且目標人物接近畫面上方，室內定位系統會將此目標人物視為正在進入或是離開畫面上方，此時目標人物的腳底可在畫面中被看見，如下圖 10 所示，所以在這個情況下與第 4.2 節的情況相同，直接使用偵測結果底邊中點的二維影像座標進行逆透視投影轉換演算法，將目標人物的影像座標轉換為真實世界座標。



圖 10 人物上半身被遮蔽

4.4. 追蹤畫面中目標人物

系統會持續地計算目標人物的長寬比，透過長寬比的變化來判斷目標人物是否全身在畫面中，並且依據不同的情況，使用不同的方式取得代表目標人物位置的影像座標；但在目標人物全身進入畫面之前，系統還是會持續的計算目標人物的位置，所以當人物全身進入畫面後，系統會使用目標人物的最大長寬比來將先前所推算的真實世界座標修正。

5. 實驗結果

在本研究中，將實驗分為三個部分，首先是比較 OpenCV 函式庫中的偵測方法以及 Tensorflow API 偵測方法的效果評估，評估的指標是以 Intersection over Union(IoU)來評估偵測效果。IoU 是一種評估偵測物體的方法的指標，其計算方式是以偵測結果的矩形與 Ground truth 的矩形之交集面積除以兩矩形聯集之面積；IoU 的數值越大代表越接近 Ground truth，代表此偵測方法越能精準的偵測到目標。第二部分是比較兩種方法偵測影像需花費的時間。第三部分則是計算使用逆透視投影演算法轉換目標人物之世界座標與

實際世界座標的誤差。

實驗使用的攝影機為 Nikon D90，其感光元件之大小為 $23.6\text{ mm} \times 15.8\text{ mm}$ ，並且安裝焦距為 18 mm 之鏡頭，可透過式(1)取得所使用相機之水平以及垂直視角，分別為 66.5 及 47.4 度。將攝影機正下方之的平面座標假設為真實世界座標之原點，並且腳架高度經過量測為 155 公分，所以相機之世界座標 (l, d, h) 為 $(0, 0, 155)$ 。拍攝影像時，相機之旋轉角為 5 度，俯角為 8 度。最後是拍攝之影像解析度為 4288×2848 。

實驗的測試資料集為一行人繞著長為 297 公分，寬為 198 公分之矩形，且行人步長為 33 公分，所形成之 30 張影像序列，實驗測試資料示意圖如下圖 11。

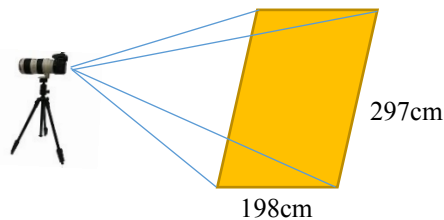


圖 11 測試資料示意圖

5.1. 偵測效果比較

在 OpenCV 函示庫中，使用 HOG 特徵來偵測人物，實驗中使用相同的 30 張測試集影像，分別以兩種方法進行偵測，接著以 IoU 指標來評估兩種方法的偵測效果，評估結果如下。

從下圖 12 的評估結果中可看見，使用 OpenCV 方法時， 30 張影像的 IoU 數值皆不超過 0.5 ，而 Tensorflow API 在大部分情況下，所偵測到的人物非常接近目標人物，可見若使用 OpenCV 的方法並無法達到預期的偵測效果。

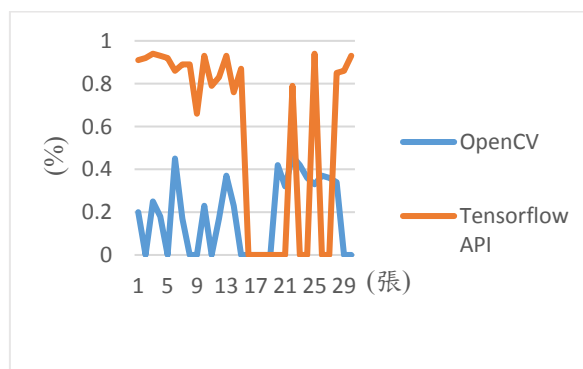


圖 12 偵測效果評估

在上圖 12 的偵測結果比較中可以發現：在影像序列的前幾張影像，Tensorflow API 之偵測效果明顯比 OpenCV 好許多，這是因為目標人物在前幾張影像中較接近相機，導致目標人物的腳有小部分並無在畫面中，導致在序列編號較前面的影像，使用 OpenCV 偵測的效果比 Tensorflow API 低許多，甚至許多時間無法偵測到目標人物，實際偵測情形如下圖 13。



(a) OpenCV 偵測結果



(b) Tensorflow API 偵測結果

圖 13 人物較接近相機之偵測效果比較

而在影像編號 19 附近時，OpenCV 方法能夠偵測到人物，而 Tensorflow API 無法偵測到目標人物，偵測結果如圖 14，造成此現象的原因推測為目標人物所在位置的背景太過複雜，導致 Tensorflow API 在此情況下無法正確地偵測出目標人物。



(a) OpenCV 偵測結果



(b) Tensorflow API 偵測結果

圖 14 影像編號 19 之偵測結果

5.2. 偵測時間比較

實驗設備之顯示卡規格為 GTX 1050，由下圖 15 可見，使用 OpenCV 幾乎是瞬間就偵測完成，而 Tensorflow API 在偵測第一張照片時花費較多時間，其餘之照片平均處理一張影像需要一秒。雖然使用 Tensorflow API 在每張影像上需要花費更多的時間，但綜合第 5.1 節的結果，Tensorflow API 在偵測的效果上比 OpenCV 好，況且一個人在一秒內能夠移動的距離不會太遙遠，所以處理每張影像需要花費一秒的時間，在可容忍之範圍內。

另外，Tensorflow API 在前幾張的偵測時間較長，在後面幾張影像時才維持在大約一秒，關於這個實驗結果，可能的原因在偵測前幾張影像時，須載入神經網路的權重，導致在偵測前幾張影像時花費的時間較久。

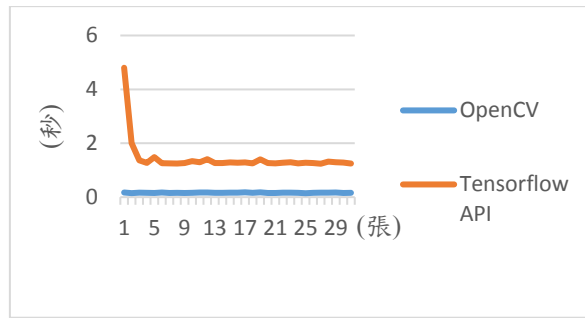


圖 15 兩種方法的時間比較

5.3. 使用逆透視投影演算法計算座標之誤差

將上述參數帶入逆透視投影演算法，並以測試資料集作為輸入使用 Tensorflow API 進行偵測，並且將目標人物之影像座標轉換為真實世界座標，最後計算實驗值以及實際值之誤差。

由下圖 16 觀察發現，編號 10 至 20 之影像為人物較接近畫面中間之影像，使用逆透視投影演算法轉換目標人物之座標獲得較低的誤差值，但是若目標人物不在影像的中間部分，有些點的實驗值與實際值之差距非常大。造成上述結果可能的原因有以下兩種：實驗使用的相機鏡頭焦距過短，視野過於廣角，導致畫面有邊緣扭曲的現象，而在我們中並沒有先進行修正畫面扭曲的動作。第二種可能的原因取得相機之參數有誤，導致將參數代入轉換公式並無法正確地將影像座標轉換為真實世界座標。

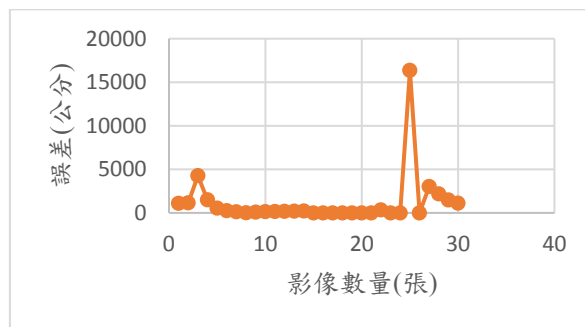


圖 16 利用逆透視投影演算法計算目標人物座標與實際世界座標之誤差

6. 結論

在本研究中所描述的使用攝影機影像快速計算目標人物位置的方法，當目標人物在較靠近特定區域中央時，才能夠較精確的將目標人物的影像座標轉換為真實世界座標，在未來的研究中，希望能夠提升偵測每張影像的速度，藉此提高輸入影像序列之幀率，並且使目標人物不僅在畫面中央能夠精準定位，即使在畫面較側邊之部分，也能夠將二維的影像座標轉換為三維的真實世界座標。在未來的研究中，將嘗試先將攝影機所拍攝到之畫面進行扭曲修正的動作，並且在代入參數之前，先進行相機校正，確保取得到之相機參數無誤，再開始進行轉換座標，希望藉由這兩種方法能夠使得轉換之誤差降低。

致謝

本研究成果部分由科技部研究計畫經費贊助，計畫編號 MOST 106-2221-E-024-003 和 MOST 106-2813-C-024-024-E。

參考文獻

- [1] 郭偉凱。類神經網路修正逆透視轉換後之前車距離偵測系統。明志科技大學電機工程研究所。2011。
- [2] H. S. Maghdid, I. A. Lami, K. Z. Ghafoor, and J. Lloret, “Seamless Outdoors-Indoors Localization Solutions on Smartphones,” *ACM Comput. Surv.*, vol. 48, no. 4, pp. 1–34, 2016.
- [3] Y. Gu, A. Lo, I. Niemegeers, “A survey of indoor positioning systems for wireless personal networks,” *IEEE Communications Surveys and Tutorials*, vol. 11, no. 1, pp. 13-32, 2009.
- [4] M. E. Rida, F. Liu, Y. Jadi, A. A. A. Algawhari and A. Askourih, “Indoor Location Position Based on Bluetooth Signal Strength,” 2015 2nd International Conference on Information Science and Control Engineering, Shanghai, 2015, pp. 769-773.
- [5] Dai Hepeng and Su Donglin, “Indoor location system using RFID and ultrasonic sensors,” 2008 8th International Symposium on Antennas, Propagation and EM Theory, Kunming, 2008, pp. 1179-1181.
- [6] Y. Liu, Z. Yang, “Location Localization and Localizability” in *Location awareness Technology for Wireless Networks*, Springer, 2010.
- [7] Y. Wang, S. Ma, and C. L. P. Chen, “TOA-Based passive localization in quasi-synchronous networks,” *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 592–595, 2014.
- [8] J. K. Y. Ng, K. Y. Lam, Q. J. Cheng, and K. C. Y. Shum, “An effective signal strength-based wireless location estimation system for tracking indoor mobile users,” *J. Comput. Syst. Sci.*, vol. 79, no. 7, pp. 1005–1016, 2013.
- [9] Z. Yang, C. Wu, and Y. Liu, “Locating in fingerprint space,” *Proc. 18th Annu. Int. Conf. Mob. Comput. Netw. - Mobicom '12*, p. 269, 2012.
- [10] H. Wang, A. Elgohary, and R. R. Choudhury, “No Need to War-Drive : Unsupervised Indoor Localization,” *Proc. 10th Int. Conf. Mob. Syst. Appl. Serv. (MobiSys '12)*, pp. 197–210, 2012.
- [11] Q. Li, W. Li, W. Sun, J. Li, and Z. Liu, “Fingerprint and assistant nodes based Wi-Fi localization in complex indoor environment,” *IEEE Access*, vol. 4, pp. 2993–3004, 2016.
- [12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 886-893 vol. 1.
- [13] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors,” *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

- [14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, et al., “SSD: Single-shot Multibox Detector,” in European Conference on Computer Vision (ECCV), 2016.
- [15] Jonathan Huang. “Supercharge your Computer Vision models with the TensorFlow Object Detection API.” Retrieved July 17, 2017, from <https://opensource.googleblog.com/2017/06/supercharge-your-computer-vision-models.html>