

國立交通大學

電控工程研究所

碩士論文

基於三維影像之坦克車即時姿態估測研究



Real-Time Tank Pose Estimation Based on 3D
Image Data

研究生：賴柏宏

指導教授：林昇甫 博士

中華民國一〇一年七月

基於三維影像之坦克車即時姿態估測研究

Real-Time Tank Pose Estimation Based on 3D Image Data

研 究 生：賴柏宏

Student: Po-Hung Lai

指 導 教 授：林昇甫 博士

Advisor: Dr. Sheng-Fuu Lin

國立交通大學

電控工程研究所



Submitted to Institute of Electrical Control Engineering

College of Electrical Engineering

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Electrical Control Engineering

July 2012

Hsinchu, Taiwan, R.O.C

中華民國一〇一年七月

基於三維影像之坦克車即時姿態估測研究

學生：賴柏宏

指導教授：林昇甫 博士

國立交通大學電控工程研究所

摘要

近年來，由於三維取像技術越來越成熟，以及取像工具的成本越來越低，三維影像逐漸成為重要的研究對象，在這幾年也有越來越多的三維辨識法則相繼提出，其中大多數的演算法開發，都沒有考慮到物體的形變，這些演算法如果遇到了物體外形變化，勢必會辨識失敗，所以開始有些演算法在開發時，會考慮到物體發生形變時的情況，而這類型的演算法，都必須對物件進行姿態估測才能夠繼續，而本論文的主要目標就是對坦克車達到即時且準確的姿態估測。

本論文選擇坦克車做為實驗對象，主要是考慮到，假設某一天在戰場上，發現了一台坦克車，比起辨識出該坦克車的型號，不如了解這台坦克是否具有威脅來的更有意義，所以本論文的演算法開發主要是考慮在不知道坦克車型號的前提下，進行姿態估測，並確認是否對我方具有威脅。

本論文的貢獻有二點：第一，本論文提出針對坦克車的切割法則，可以在沒有任何坦克車的尺寸資訊下，達成切割目標；第二，本論文利用提出以網格方式進行點雲篩選，可以使姿態估測結果較其他論文來的更好。

Real-Time Tank Pose Estimation Based on 3D Image Data

Student : Po-Hung Lai

Advisor: Dr. Sheng-Fuu Lin

Institute of Electrical Control Engineering
National Chiao Tung University

ABSTRACT

In recent years, the 3D camera technology has become more developed and cheaper. And the research of 3D images has become more popular. Among these years, more and more 3D recognition rules are proposed. Most of these algorithms did not consider about the articulate objects. Therefore, some articulate recognition rules are noticed. These kinds of algorithms can be used only after the pose estimation. So the main purpose of this paper is to design a real-time pose estimation system.

Imaging that, if one day, there is a tank appearing in the battlefield. Compared with to identify the model of the tank, it is more meaningful to understand whether the tank is threatening or not. Therefore, the algorithm of this paper is to estimate the pose of the tank to confirm the threat based on the situation without knowing the tank's model.

There are two contributions in this paper. First, we provide a segmentation rule of tank without the size information; second, we propose a method to filter the point cloud data based on grids, and that results in a better performance in pose estimation than other papers.

誌 謝

首先我要感謝我的指導教授林昇甫博士，感謝老師在我碩士生涯中，給了我許多的啟發與指導，讓我能夠在這時後順利的完成我的碩士論文，也必須要感謝我的家人們，在我努力完成碩論時，給與我許多正面的鼓勵與支持，也要感謝實驗室中的學長晉嘉、國育、士哲、啟耀、逸章、俊偉、裕筆，還有同學植諺、俊良、婷婷以及學弟妹們，有大家在實驗時幫我解決困難，讓我遇到問題時，不會有無助的感覺，也在我寫論文心煩時給予我最大的鼓勵，有大家的幫忙才有這篇論文的產生。

最後更要感謝從大學一直陪我到研究所的同學們，奕心、玫芳、鈺婷、逸棋，有他們一直在研究所時，讓我在一個陌生的環境時不會孤單，也在修課時相互扶持，有了大家，才讓我的碩士生涯可以話上一個完美的句點，由衷的感謝各位。



目錄

中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
圖目錄	vi
表目錄	ix
第一章 緒論	1
1.1 三維影像介紹	1
1.2 相關研究之探討	4
1.3 研究動機	7
1.4 論文架構與貢獻	7
第二章 相關知識及理論	8
2.1 三維影像取得	8
2.2 點雲資訊前處理	11
2.2.1 KD-tree	11
2.2.2 法向量估測	15
2.3 興趣目標搜尋	16
2.3.1 平滑切割	16
2.3.2 平面切割	18
2.3.3 歐西里德分類法	20
2.3.4 連結區塊分類法	21
2.4 姿態估測	22
第三章 系統流程	25

3.1	資料取得方式	25
3.2	系統流程圖	28
3.3	系統前處理	29
3.3.1	場景平面抽取及切割	29
3.3.2	物件分群	30
3.3.3	興趣目標搜尋	33
3.4	物件姿態估測	37
3.4.1	物件切割	37
3.4.2	角度估測	43
第四章	實驗結果及分析	51
4.1	實驗場景與設備	51
4.2	實驗結果	52
4.3	結果分析與比較	58
4.4	結果討論	63
第五章	結論與未來工作	66
參考文獻	68

圖目錄

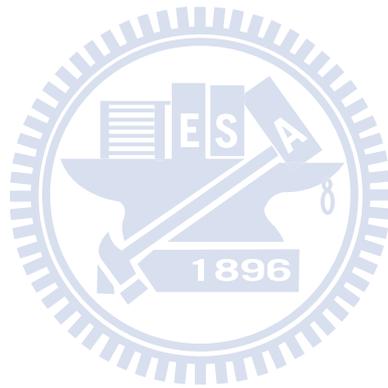
圖 1.1 灰階影像及距離影像比較圖。(a) 灰階影像；(b) 距離影像。.....	1
圖 1.2 點雲範例圖。.....	2
圖 1.3 二維影像物件辨識示意圖。.....	3
圖 1.4 文獻[6]切割範例圖，長方體包覆範圍為坦克車，'o'為炮管，'x'為砲塔。(a) 炮管、砲塔切割不完全；(b) 將車身部份誤判為炮管。.....	6
圖 2.1 微軟體感遊樂器以及座標示意圖。.....	8
圖 2.2 距離影像範例。.....	9
圖 2.3 kinect 座標點雲圖。.....	9
圖 2.4 實際座標點雲圖。.....	10
圖 2.5 KD-tree 示意圖。.....	11
圖 2.6 KD-tree 樹狀圖。.....	13
圖 2.7 平滑切割流程圖。.....	16
圖 2.8 鄰近點搜尋示意圖。.....	19
圖 2.9 歐西里德分類範例。(a) 分類前；(b) 分類後。.....	21
圖 2.10 連結區塊定義 (a) 4-鄰近點；(b) 8-鄰近點。.....	21
圖 2.11 連結區塊分類範例。(a) 分類前；(b) 分類後。.....	22
圖 2.12 隨機分佈之二維資料。.....	23
圖 2.13 最小平方法估測結果圖。.....	24
圖 3.1 實驗用坦克車。.....	25
圖 3.2 體感遊樂器(kinect)。.....	26
圖 3.3 實驗用鐵架。.....	26
圖 3.4 鐵架上拍攝的距離影像圖。.....	27
圖 3.5 從圖 3.4 計算得到的點雲圖。.....	27

圖 3.6 系統流程圖。	28
圖 3.7 場景影像。(a) 場景二維影像；(b) 對應之三為點雲圖。	29
圖 3.8 切割平面結果。	30
圖 3.9 去除平面後之場景。	30
圖 3.10 平面切割後點雲資訊。	31
圖 3.11 平面切割後對應之距離影像。	31
圖 3.12 連結區塊法分類結果。	32
圖 3.13 歐西里德距離分類法分類結果。	33
圖 3.14 不同距離拍攝之坦克車。(a) 15179 點；(b)10942 點。	34
圖 3.15 不同距離之坦克車圖。	34
圖 3.16 長寬閾值定示意圖。	35
圖 3.17 坦克車場景。	36
圖 3.18 物件分類結果。	36
圖 3.19 興趣目標搜尋結果。	37
圖 3.20 文獻[6]長方體包覆切割結果。'x'為砲塔，'o'為炮管，黑點為車身。	38
圖 3.21 利用高度資訊切割結果。	38
圖 3.22 未切割之坦克車圖。	39
圖 3.23 不同高度時點雲之寬度圖。	39
圖 3.24 寬度變化圖。	40
圖 3.25 根據 $z=0.1402$ 切割結果圖。	40
圖 3.26 待切割炮管。	41
圖 3.27 將圖 3.26 轉至 x 軸結果圖。	41
圖 3.28 兩個方向之寬度圖。	42
圖 3.29 炮管切割後之結果。紅色為炮管；藍色為炮塔。	42
圖 3.30 坦克切割結果圖，綠色為車身，紅色為炮管，藍色為炮塔。	43
圖 3.31 主成份分析主軸結果(從上往下看)。	44

圖 3.32 邊緣抽取範例圖。	44
圖 3.33 網格化示意圖。	45
圖 3.34 網格化結果。	45
圖 3.35 抽取最多點雲區塊結果。	46
圖 3.36 綠色點為對應之邊緣點雲。	46
圖 3.37 配合網格化邊緣抽取之主軸結果。	47
圖 3.38 未經切之之主軸分析。	47
圖 3.39 切割後之主軸分析。	48
圖 3.40 主成份分析法及最小平方法估測比較。	49
圖 3.41 旋轉後之炮塔。	49
圖 3.42 利用最小平方法計算之主軸。	50
圖 4.1 電子氣泡式角度尺。	51
圖 4.2 量角器。	52
圖 4.3 俯仰角與準確度關係圖	53
圖 4.4 車身角度誤差圖。	54
圖 4.5 炮塔旋轉角度誤差圖。	55
圖 4.6 炮管角度誤差圖。	55
圖 4.7 電腦模擬估測結果。	56
圖 4.8 連續場景估測結果一。	57
圖 4.9 連續場景估測結果二。	57
圖 4.10 車身角度誤差比較圖。	58
圖 4.11 炮塔角度誤差比較圖。	59
圖 4.12 炮管角度誤差比較圖。	60
圖 4.13 車身遮蔽比與精準度之比較圖。	61
圖 4.14 炮管遮蔽比與精準度之比較圖。	62
圖 4.15 長方體包覆誤差圖。	63

表目錄

表 4.1 估測所需時間表	61
表 4.2 與各論文結果比較表	64



第一章 緒論

本論文所要研究的目標是，在戰場中利用雷射測距儀(laser range finder)對坦克車進行姿態估測，來達到即時戰場防護預警系統。為達到即時防護的目標，在演算法的開發上，必須同時兼顧到速度以及準確性，這部分也將是本論文的開發重點。

本章將分為四個小節，1.1 節將介紹本論文所使用的三維影像，並簡介三維影像相較於二維影像的優缺點；1.2 節將介紹論文題目的研究動機；1.3 節將會說明過去該領域的研究成果；1.4 節則介紹本論文之架構與貢獻。

1.1 三維影像介紹

在過去，取像技術多是以二維影像為主，例如：灰階影像或彩色影像，因為取像方式非常容易且相當普及，所以過去所開發的演算法也都是因應二維影像所發展的。但隨著三維取像技術越來越成熟以及成本的降低，使得以三維影像為基礎的演算法也逐漸的開始發展。而本論文所使用的三維影像，則是利用三維取像工具：雷射測距儀所拍攝得來的，利用雷射測距儀所拍攝得到的影像，主要可以分為兩大類，第一類為距離影像(range image)，距離影像與灰階影像略為相似，如圖 1.1 所示：



(a)

(b)

圖 1.1 灰階影像及距離影像比較圖。(a) 灰階影像；(b) 距離影像。

雖然兩圖類似，但代表意義卻是完全不同，距離影像以及灰階影像每個像素(pixel)皆是以一個 0 至 255 的整數來做表示，但灰階影像中，每個像素的數字所代表的是灰階值(gray level)；而距離影像中，每個像素則是將雷射測距儀得到的距離依線性的方式轉換成 0 至 255 所得的結果。觀察圖 1.1，可以發現距離影像的某些位置中是完全黑的，與灰階影像無法呼應，會造成這些情況的原因有下列幾種：第一，可能是因為距離太遠，超出儀器設定的範圍，導致雷射測距儀無法讀取到反射回來的資訊，所以會造成某些位置資訊空白。第二種原因則是雷射光會穿透過玻璃這類的物質，造成雷射無法反射，而無法得到正確的距離資訊，所以在玻璃杯、電燈泡的部分就會是黑色的，無法測距。而雷射測距儀所能得到的第二類資訊則是點雲資訊(point cloud data)，點雲資訊是將得到的距離資訊以點的方式做呈現，如圖 1.2，點雲中的每個點，都是以三維空間中的座標的方式做為描述，其位置就是在真實空間中所對應的位置，單位則是公尺。

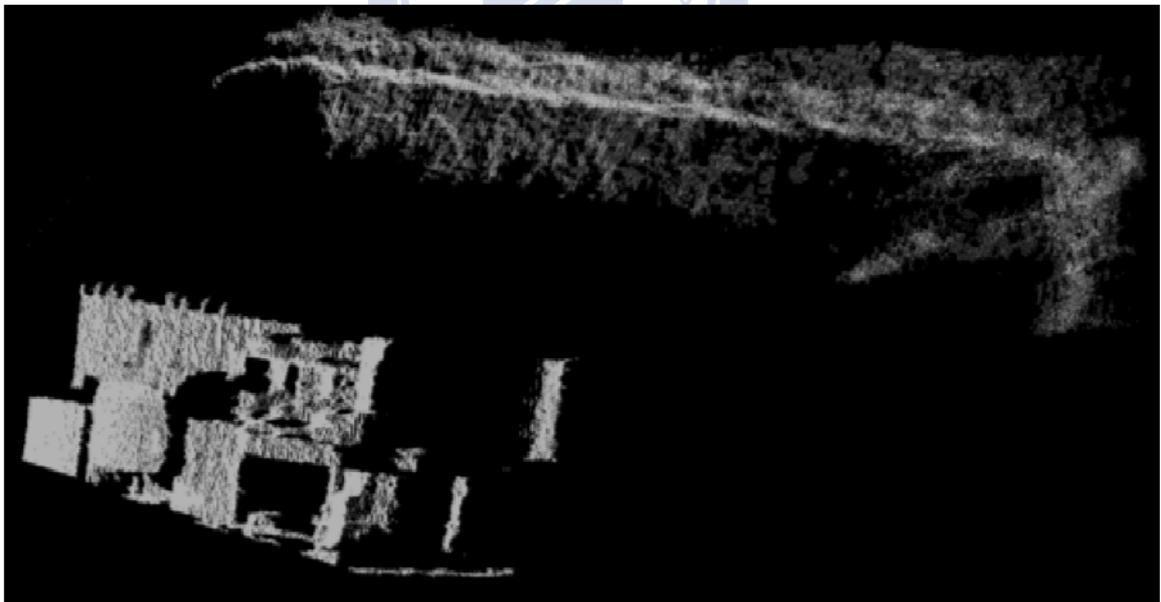


圖 1.2 點雲範例圖。

以上簡介了二維影像以及三維影像，兩者最大的差異在，三維含有了距離資訊，而且這個距離資訊所指的是實際的距離，並非是縮小比例的結果，這樣的特性讓三維

影像的演算法開發上不需要特別考慮到尺度的問題。例如圖 1.3，可以發現，左圖為實際的物件，而右圖只是一個該物件的圖片，如果使用一個二維的辨識系統，對物件抽取特徵時，都會相似，勢必會把右邊的物件圖片也辨識為搜尋的目標物；但是如果使用的是三維的影像時，因為有著對該物件的實際距離，所以右圖中，偵測到會是一個平面，可以很輕易的被刪除，由此可見，有著物件實際距離的三維影像，可以讓演算法開發上來的更有效率。



圖 1.3 二維影像物件辨識示意圖。

但是，雖然三維影像有著這樣豐富的資訊，但仍是存在著某些缺點，例如：遇到玻璃，會穿透過去，造成無法得到資訊，一旦欲辨識的物件屬於這類的材質，都無法使用三維影像進行處理；另外，雷射測距儀的價格與其精準的程度成正比，目前價格仍非常昂貴，設備的取得也是一個門檻；而最大的卻點，也是目前開發中較大的阻礙就是，三維影像在資訊上多了一個維度，但在計算上卻會變得非常的龐大，例如二維影像中，鄰近點搜尋非常容易，只要利用該像素的位置就可以找到其鄰近點，但在三維點雲中，搜尋鄰近點必須計算出所有點的距離，再找出最小值才能得到，這也讓三維影像演算法的實際使用上，有了較大的阻礙。

目前三維影像，最被廣泛的應用是在於距離量測、地形地貌監測、三維立體物件的量測等。利用雷射測距儀可以得到物體的輪廓、邊界、紋理等；這些影像資訊目前有被應用在機器人視覺，地形變化的量測，機械機密定位以及生物立體影像的重建等。

1.2 相關研究之探討

近年來，三維影像越來越被廣泛的使用，由其在物件辨識以及人體姿態估測上應用的更是越來越多，而本論文的目標是完成一即時防護系統，演算法重點是放在關節式鋼體的姿態估測，期中結合了目前常用的一些辨識演算法以及姿態估測的演算法，對其進行更新及改善，來完成這一系統，以下我將對目前較常見的三維姿態估測的方法作簡介。

文獻[1]中，疊代最近點法(iterative closest point, ICP)是三維點雲在辨識以及姿態估測都常使用的到方法，疊代最近點法是將兩組點雲資料以點座標位置為特徵進行疊合的一種演算法，這個方法的優點在於使用非常容易，只要設定好一物體的點雲資料庫，就可以將拍攝到的物體點雲資訊進行疊合，並計算出誤差值，根據誤差值就可得到辨識結果以及姿態的結果，但缺點是因為利用疊代的方式來進行疊合，是一種比較暴力的演算法，所需的計算時間很長，而且如果物體疊代的初始姿態選擇不好的話，可能會造成疊代失敗。也有人提出不同類型的方法，是利用抽取特徵的方式進行辨識，例如利用傅立葉描述子(Fourier descriptor)[2]，傅立葉描述子是使用距離影像，對物體抽取出輪廓特徵來進行辨識的，傅立葉描述子的優點是他可以針對物體的輪廓形狀進行描述，轉換成一個數值化的特徵向量，只要與資料庫的特徵向量做比對，即可以得到物件的類別及姿態但缺點則是必須對物體的多個角度抽取特徵向量放入資料庫等後比對，且這個方法僅針對了物體的輪廓做描述，並未對物體內容描述，也常造成辨識錯誤。相對於輪廓，也有針對內容的辨識演算法，如奈克矩描述子(Zernike moment descriptor) [3]，奈克矩描述子相反於傅立葉描述子，是一個考慮物體內容的特徵，但也因為只考慮到了內容，而沒有考慮到輪廓，所以在某些特定的物件也常常會有辨識錯誤的結果。為了同時考慮到兩者，則有了結合傅立葉描述子以及奈克矩描述子的方法[4]，效果則會較只使用單一個描述子來的好。但是在以上的方法中，有一個特點，都是針對無形變鋼體進行開發的演算法，所以即使這些方法可以順利的找出物件的類別以及當下的姿態，但是一但遇到鋼體形變，例如：車門打開了、機車掀起後車

廂、甚至是本論文考慮的坦克車的炮管升起、旋轉、…等，這些情況時，以上的演算法就無法使用了，或者必須對各種角度的變化都建立資料庫樣本等後比對，但這麼作是非常不實際的。

為了解決這類的問題，開始有研究者陸續的提出依些解決方法，在文獻[5]，作者提出了將資料庫先進行切割，再配合疊代最近點法來解決這類的形變問題，一開始利用手動的方式將一個物件在物體關節處進行切割，當每一筆測試資料輸入時，則將各個部位分別進行疊代最近點法，每當完成一個部位的疊代，就將符合的點刪除並繼續下個部位的疊代，最後再以剩下未被疊代的點的數量來決定目標是哪種物件。這個演算法優點是不論目標為何，都可以使用，缺點是需對每個可變部位都建立資料庫，且疊代時間長。而文獻[6]中，作者提出了一個針對坦克車的自動目標辨識法(automatic target recognition, ATR)，該論文的核心是利用最佳化的方式，求出對整個物件的最小長方體將該物體包覆，再利用資料庫中原本的模型來推估計算出來的長方體中，哪些點屬於炮管、砲塔、以及車身，並且分別求出各部位的姿態，最後再根據求出的角度，將資料庫中每台坦克根據該角度值進行旋轉，並且將旋轉過的資料庫點雲和測試資料點雲進行比對，利用最小平方法(least squares method)來達到辨識的結果，這篇論文的方法最大的優點在於他只需要對每台坦克車建立一個資料庫即可，但最大的缺點在於切割的部份做的並不理想，常會有切割不完全或將車身切成炮管的情況發生。所以在文獻[7][8]中，作者提出了基於貝氏定理(Bayes theorem)的關節式物件的追蹤與搜尋(tracking and searching)，在假設物件必存在目前場景的前提之下，於場景中對搜索出該物件，做法上與文獻[5][6]最大的差異是切割的部份，本篇論文是將搜尋到的興趣目標(region of interest, ROI)先利用地面法向量加上已知的高度資訊來對物體進行切割，切割的結果相較之前的方法都好上很多，分割過後，再利用預先設定好的機率分佈來計算該物件的姿態，由於是抽取特徵並非以疊代方式來進行比對，所以在速度上快上很多，但需要對不同部位建立機率分佈作為資料庫。根據上述幾篇論文可以發現，針對形變的鋼體的辨識以及姿態估測，多是利用切割的方式，再配合過去非變形鋼體的演算法對各部位進行辨識以及姿態估測，所以切割的好壞，不論是對姿態估測或辨

識都有一定的影響，但過去在對坦克車切割的方式中，其中，文獻[6]是少數利用長方體包覆的方式進行切割的，但根據文獻內可以發現效果並不佳，如圖 1.4：

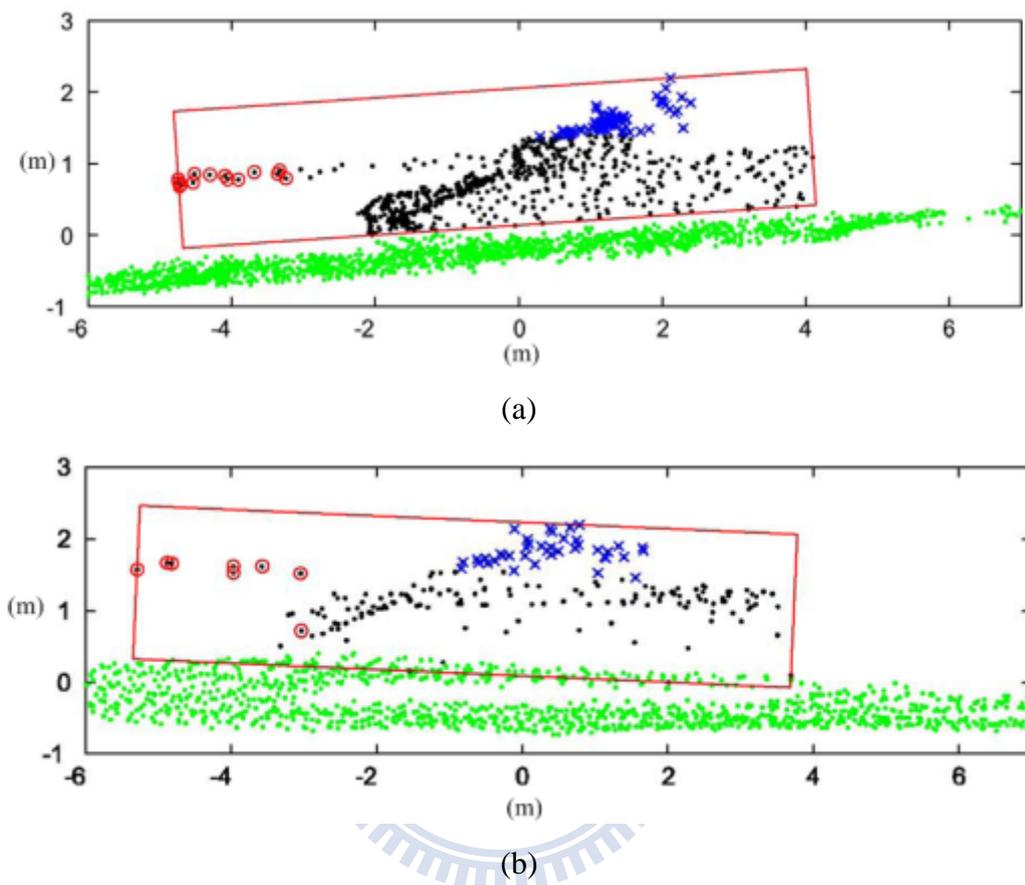


圖 1.4 文獻[6]切割範例圖，長方體包覆範圍為坦克車，'o'為炮管，'x'為砲塔。(a) 炮管、砲塔切割不完全；(b) 將車身部份誤判為炮管。

上圖 1.4 可以發現切割的效果並不好，如圖 1.4(a)，炮管、砲塔的部分並沒有完整的切開，圖 1.4(b) 則是將不屬於炮管的部份也切成炮管，這說明了該方法還是有改進空間的；而在文獻[7][8][9]則是利用平面法向量以及坦克車的高度進行切割，切割效果較[6]好很多，但是這樣的切割法有一個假設前提，必須先知道坦克車的高度，才能夠進行切割，但在未辨識前，要怎麼得到坦克車型號呢？所以本論文只能將實驗預設在執行搜尋的前提，才能成功，否則沒有高度資訊的情況，將無法執行後續的演算法。

1.3 研究動機

在相關研究領域中發現，過去的研究裡，姿態估測常常只是一個前處理，但在真正的戰場上，知道坦克車的攻擊方向、目標，相信比會知道坦克車的種類還來的重要，所以本論文的目標在完成一個即時防護預警系統，試著考慮以下這種情況：當戰場上當某武裝小隊被賦與一個攻擊任務時，該小隊勢必會專心的注意目標，並前去攻擊，但這時候有可能發生某些狀況，例如：全心的前往該目標的途中，卻被某個非該小隊目標的單位攻擊。考慮到這種情況，過往的處理方法是在坦克車或戰機上裝設有感測器，當被敵方的雷射測距儀掃描時，才會發現自己可能已經成為敵方的目標，但雷射測距儀因為角度大，即使被掃到，也不代表已經被鎖定，就算真是被鎖定了，卻也為時已晚。所以本論文希望能開發出一個主動式即時防護預警系統，這樣的系統可以準確的判斷出是否真的在敵方的攻擊範圍內，不再是被動的反擊，可以更優先的將我方隊伍為目標的坦克車清除，相信這樣的一個系統將會比過往的設備來的更有效。

在系統中，為了兼顧速度及準確性，同時利用距離影像以及點雲資料兩種資訊來進行，利用點雲資訊可以方便對場景進行切割以及角度估測；而距離影像則是可以更快的搜尋場景的可能物件，同時使用兩種影像，可以讓本系統兼顧速度以及準確性。

1.4 論文架構與貢獻

本論文在第一章的部份，簡述了本論文的研​​究方向、過去相關領域研究以及研究動機；第二章則會介紹本論文使用的相關知識及理論；第三章將介紹本論文的系統架構以及本論文的核​​心技術；第四章則會模擬戰場中坦克車的姿態改變，並實驗本論文所提出的姿態估測法則是否能達到即時且準確的結果；最後將在第五章的部份敘述論文貢獻以及未來工作的發展。本論文的貢獻有二點：第一，本論文提出針對坦克車的切割法則，可以在沒有任何坦克車的尺寸資訊下，達成切割目標；第二，本論文利用提出以網格方式進行點雲篩選，可以使姿態估測結果較其他論文來的更好。

第二章 相關知識及理論

針對場景中坦克車的姿態估測，本論文將利用三維影像資訊來完成，在本章將會介紹過去的相關知識及理論，並對這些演算法進行優缺點的比較。本章分為四個小節，2.1 節首先介紹論文中使用的三維影像取得方式；2.2 節將介紹如何如何將一群點雲進行組織；2.3 節介紹如何抽取出興趣目標；2.4 節則介紹姿態估測的相關方法。

2.1 三維影像取得

在第一章的部分，有提到目前三維影像是靠雷射測距儀來取得的，過去由於雷射測距儀成本高，所以在實驗常常是利用電腦模擬的方式來進行，但模擬的方式因為沒有雜訊干擾，且精準度相當的高，導致這樣模擬的方式由於獲取資訊太過完美，不免讓人懷疑當資訊不在如此完美時，演算法是否仍然作用呢？所以本論文為了讓演算法能夠真正的應用於實際，所以選擇使用微軟公司開發的體感遊樂器(kinect)來進行實驗，如圖 2.1

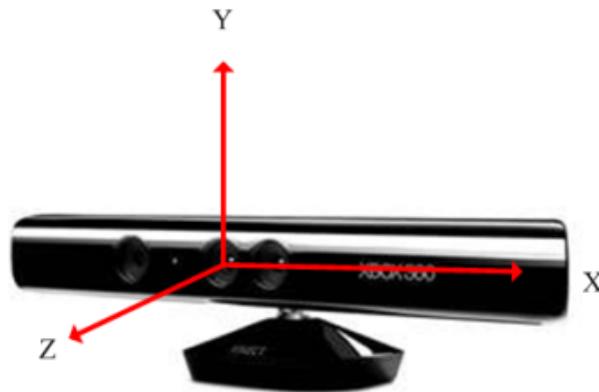


圖 2.1 微軟體感遊樂器以及座標示意圖。

體感遊樂器可以同時提供二維彩色圖片以及三維距離影像，考慮到一般雷射測距儀並不會提供二維影像的前提下，本論文在演算法開發上，不使用二維影像資訊，僅使用三

維影像資訊。體感遊樂器本身是利用紅外線偵測距離的，根據紅外線飛行時間，就可以達到測距的效果，下圖 2.2 為利用體感遊樂器得到的距離影像：



圖 2.2 距離影像範例。

而本論文開發的演算法同時使用了距離影像及點雲資訊，所以首先必須將距離影像轉換成點雲資訊，當輸入一張距離影像時，該影像本身的座標是以體感遊樂器所建立的如圖 2.1 所示，以該座標系統建立結果如下圖 2.3：

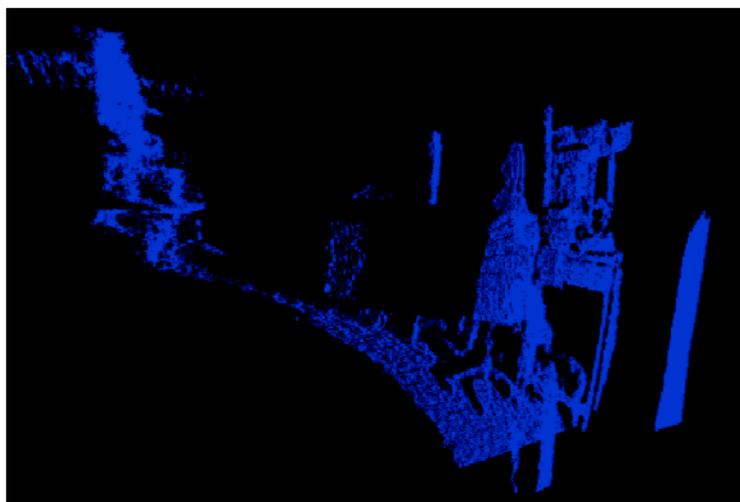


圖 2.3 kinect 座標點雲圖。

圖 2.3 中，整個點雲資訊是根據儀器本身所定義出來的，可以看到這樣的座標資訊與肉眼實際看到的物體是不一樣的，所以必須有一個轉換方式，使得儀器的座標轉換成實際的座標，參考[10]所提供的公式：

$$\begin{bmatrix} x_p \\ y_p \\ z_p \end{bmatrix} = range(x_r, y_r) \begin{bmatrix} (x_r - 640) \times \frac{1}{f_{x_z}} \\ (y_r - 480) \times \frac{1}{f_{y_z}} \\ 1 \end{bmatrix} \quad (2.1)$$

其中， x_p 、 y_p 、 z_p 表示實際點雲座標， x_r 、 y_r 代表距離影像中的相素位置， $range(x_r, y_r)$ 則是代表距離影像中 (x_r, y_r) 的深度值， f_{x_z} 以及 f_{y_z} 則是代表焦距，分別是 594 以及 591 [10]。將圖 2.2 的所有數值代入上式(2.1)，則可得到下圖 2.4：

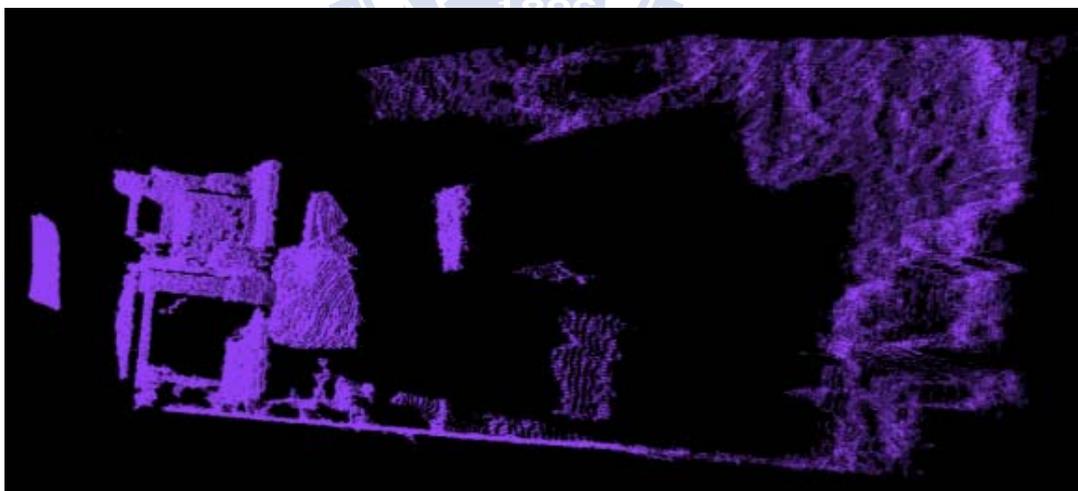


圖 2.4 實際座標點雲圖。

以上介紹了體感遊樂器的三維影像以及如何將距離影像轉乘實際的點雲座標資訊，本論文將同時使用距離影像以及點雲資訊來完成。

2.2 點雲資訊前處理

當我們利用體感遊樂器獲取了場景的點雲資料時，必須在一個場景之內找到目標物，畢竟點雲資訊本身是一個沒有任何相對關係的點集合，為了找出興趣目標，勢必必須將點雲的相對關係先找出來，再進階利用這些關係找出那些點資訊是可能屬於同一物件的。本節將分為兩小節，2.2.1 節將介紹快速搜尋鄰近點的方法；2.2.2 節介紹如何找出各點法向量。

2.2.1 KD-tree

在點雲資料中，由於所得到的資料只是一群離散的點集合，所以搜尋鄰近點是一個必要動作。經過對鄰近點的搜尋，才得以將離散的點雲資料建構起連結關係，透過這樣的連結關係，才能夠得到法向量甚至是曲率等相關資訊，所以本小節將介紹如何搜尋 k 個鄰近點。獲取 k 個鄰近點，最直覺的方法，就是算出每個點到該點的距離後，再找出距離最近的 k 個點，但是如果利用這樣的方法來做 k 個鄰近點搜尋時，速度非常慢，必須算出每個點到該點的距離後，再搜尋出最近的 k 個點，這樣的計算量是相當龐大且沒效率的。為了解決該問題，我們利用 KD-tree[11]來試圖解決速度上的問題，KD 一詞即為 K 維的意思，是可用於高維度資料的樹狀資料結構。KD-tree 示意圖如下圖 2.5：

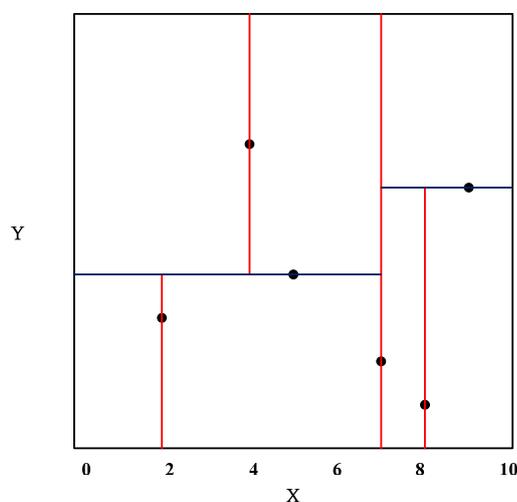


圖 2.5 KD-tree 示意圖。

KD-tree 做法是以第一個維度的中位資料為界線，將資料以第一維度一分為二，再以第二維度的中位資料將資料分為兩大類，依序執行到 K 維後，再回到第一維，重複以上動作，直到每個小區塊都只有一筆資料為止。其中，中位資料是指將數據按大小順序排列起來，形成一個數列，居於數列中間位置的那個數據，例如：有一筆資料為 $[4,2,5,9,8]$ ，經排列後得 $[2,4,5,8,9]$ ，則中位資料等於 5。我們可以將圖 2.5 中的所有資料建立如圖 2.6 的樹狀結構，詳細過程如下：

所有資料內容： $x_1 = (2,3)$ ， $x_2 = (4,7)$ ， $x_3 = (5,4)$ ， $x_4 = (7,2)$ ， $x_5 = (8,1)$ ， $x_6 = (9,6)$

步驟一：根據第一維度找出中位資料

第一維度資料有 $[2,4,5,7,8,9]$ ，經由排列可以得到 $[2,4,5,7,8,9]$

在資料為偶數筆時，選擇較大資料的來進行二分，所以選擇 $x_4 = (7,2)$ 來二分，

因此可得第一層為 $(7,2)$ 。

步驟二：以 x_4 為做為分界，將資料分為第一維度較 7 小的一群 $A : (x_1, x_2, x_3)$ ，以及較 7 大的一群 $B : (x_5, x_6)$ 。

A 群中，利用第二維度資料排序得 $[3,4,7]$ ，中位資料 4 屬於 x_3 ，選擇 x 來二分，因此樹狀圖左半邊第二層得到 $x_3 = (5,4)$ 。

B 群中，利用第二維度資料排序得 $[1,6]$ ，資料為偶數筆，選擇中位資料 6 屬於 x_6 ，選擇 x_6 來二分，因此樹狀圖右半邊第二層得到 $x_6 = (9,6)$ 。

步驟三：以樹狀圖左半邊來看，將步驟二 A 群中剩餘的資料中，找出第一維度較 5 小的一群 $A_a : (x_1)$ ，較 5 大的一群： $A_b : (x_3)$ ，兩邊都只剩一筆資料，於是將較小的 x_1 加入樹狀圖中 $x_4(5,4)$ 的左半邊， x_2 則放到右半邊。

以樹狀圖右半邊來看，將步驟二 B 群中剩餘的資料中僅剩一筆資料，則將該筆資料放到樹狀圖中 x_6 的下面。

最後得到的結果如圖 2.6。

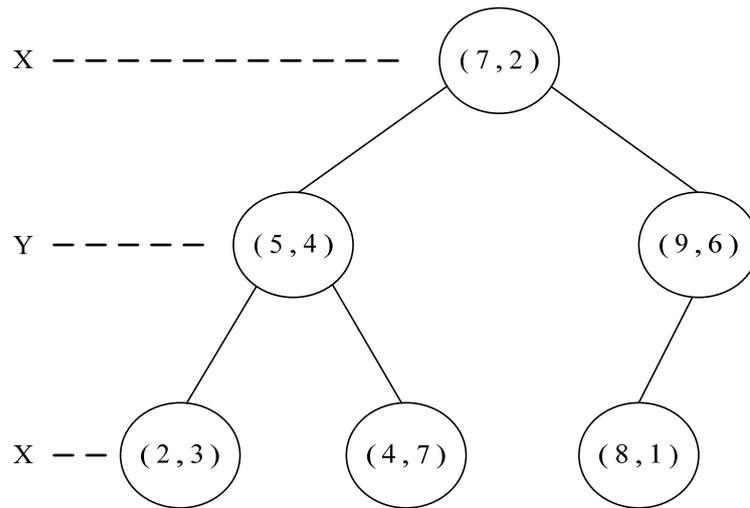


圖 2.6 KD-tree 樹狀圖。

圖 2.6 是 KD-tree 建立後的樹狀圖。但是，在執行 KD-tree 的建立時，中位資料的搜尋，其實是相當耗時的一件工作，為了讓整體演算法更有效率，對中位資料的搜尋做一點補強，我們將中位資料這個步驟改為隨機挑選(本部分將在下一段詳細解說)，當然，隨機取點所做出來的 KD-tree 與原始的 KD-tree 是有些許差異的，但是在速度上卻是可以快上非常多。

在中位資料搜尋的部分最簡單的方法即是將所有資料的第 n 維度排列後，再取中位值做一分為二動作，但每完成一個維度後，要在重新排列是非常沒有效率的，計算量為 $O(\frac{n^2+1}{2})$ ，所以將利用以下演算法來改善該狀況。快速搜尋演算法的核心想法是：「假如要搜尋第 k 小的資料，那代表該陣列中仍有 $k-1$ 個比該值小的資料」[12]。實際的做法是，先隨機取其中一筆資料，判斷該維資料比該資料小的，則丟到第一個矩陣，大於該值的則丟到第二矩陣，再根據第一個矩陣的值數量推測中間值應該存在哪一邊，重複以上步驟，直到找出需要的資料。詳細流程為：

假設有一組一維的資料 x_1, x_2, \dots, x_n ，希望找到中位資料 x_k ，其中 $k = \frac{n}{2}$ 。

步驟一：隨機取一筆資料 x_i ，並對除了自己以外的所有資料進行比較，內容 $\leq x_i$ 的丟入

A 矩陣，資料內容 $> x_i$ 則丟入 B 矩陣。

步驟二：假設 A 矩陣內有 s 筆資料，

若 $s = k - 1$ 則表示 x_i 是希望的结果，則完成中位資料搜尋。

若 $s \geq k$ 則 $k_{new} = k_{old}$ ，並將資料更新為 A 矩陣回到步驟一，

若 $s < k$ 則 $k_{new} = k_{old} - s - 1$ 並將資料更新為 B 矩陣回到步驟一。

重複以上步驟直到找出中位資料即可結束。

以下舉一個一維資料範例來做說明：

假設原始資料為 $Data = [2, 7, 6, 5, 4]$ ，而我們所需要的中位資料為 $Data[4] = 5$

因為總共有五筆資料，所以我們需要的將會是第三小的資料，令 $k = 3$

步驟一：隨機挑選一筆資料，假設挑到了第三筆 $Data[3] = 6$

步驟二：找出與該筆資料比較大小過後的矩陣 A 、 B

得到 $A = [2, 5, 4]$ ， $B = [7]$ ，

$length(A) = 3 = k$ ，所以可以知道中位資料必定出現在 A 矩陣，其中

其中， $length(X)$ 表示 X 矩陣擁有的資料個數。

$k_{new} = k_{old} = 3$ ，更新資料為 A 矩陣，回到步驟一。

步驟一：在矩陣 A 裡隨機挑一筆資料，假設為 $A[1] = 2$

步驟二：找出與該筆資料比較大小過後的矩陣 A 、 B

得到 $A = []$ ， $B = [5, 4]$ ，

$length(A) = 0 < k$ ，所以中位資料會出現在 B 矩陣

$k_{new} = k_{old} - length(A) - 1 = 2$ ，更新資料為 A 矩陣，回到步驟一。

步驟一：在矩陣 B 裡隨機挑一筆資料，假設為 $B[1] = 5$

步驟二：找出與該筆資料比較大小過後的矩陣 A 、 B

得到 $A = [4]$ ， $B = []$ ，

$length(A) = 1 = k - 1$ ，所以中位資料就是 5。

這樣的演算法因為有隨機取值的步驟，無法精確算出計算量為多少，但本演算法若考慮最糟糕的情況(每次都取到最大值或最小值)，計算量為 $O(\frac{n^2+1}{2})$ ，與原本的演算法相同，但本方法不僅可以用於取中位值，也可以用於取任意第 k 小或第 k 大的值，與原本的方法比較，在速度上有了較大的優勢，並且功能沒有任何減少。

2.2.2 法向量估測

要估測法向量，最簡單的方法就是利用主成份分析法(principal component analysis, PCA)[13]，可以用來實現法向量的估測的。主成份分析法是一個向量空間轉換法，用來找出一組多維度向量的主要成份，也廣泛的使用在高維度資料的降維或壓縮上。主成份分析能找出資料的主要分佈方向，透過計算共變異矩陣(covariance matrix)的特徵向量，可視為一組特徵空間的正交基底，可以提供整體向量的結構，使其他向量能快速投影到此特徵空間中。其中最小特徵值所對應的特徵向量即為法向量。

為了實現法向量的估測，必須利用周圍的 k 個鄰近點來對法向量估測。搜尋出 k 個鄰近點後，再對這 k 筆資料進行主成份分析法的動作，主成份分析法演算法如下：

步驟一：找出所有點的重心

$$\bar{x} = \frac{1}{k} \sum_{n=1}^k x_n \quad (2.2)$$

步驟二：將每筆點雲資料減去重心：

$$\hat{\mathbf{x}}_n = \mathbf{x}_n - \bar{\mathbf{x}}, \quad n = 1, \dots, k \quad (2.3)$$

步驟三：產生資料矩陣 \hat{X} ：

$$\hat{X} = [\hat{\mathbf{x}}_1 \quad \hat{\mathbf{x}}_2 \quad \dots \quad \hat{\mathbf{x}}_k] \quad (2.4)$$

步驟四：求得共變異矩陣 C ：

$$C = \hat{X}\hat{X}^T \quad (2.5)$$

步驟五：求解特徵向量：

$$\lambda_j \mathbf{e}_j = C\mathbf{e}_j \quad (2.6)$$

其中，最小特徵值對應的特徵向量，就是法向量。

2.3 興趣目標搜尋

興趣目標搜尋的部分，主要是介紹如何在一個隨機場景中找出坦克車，本章節將介紹過去常用的興趣目標抽取方式，2.3.1 節先介紹全自動的切割方法：平滑切割法；2.3.2 節則介紹抽取場景平面的演算法；2.3.3 節介紹如何利用點雲將點雲分類；2.3.4 則介紹如何利用距離影像進行物件分類。

2.3.1 平滑切割

平滑切割(smoothness constraint)[14]，是以點雲之間的距離以及法向量夾角為基礎的物件切割方法，流程圖如圖 2.7：

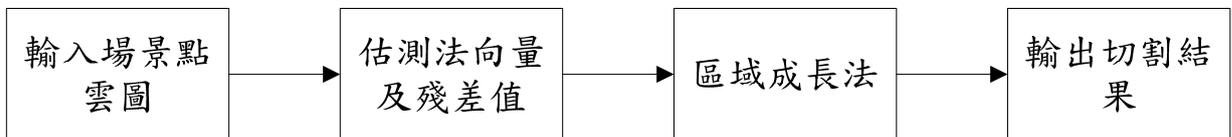


圖 2.7 平滑切割流程圖。

主要是利用點雲中每個點的法向量以及殘差值做為分類的指標，平滑切割的演算法主要分為兩大部分：

第一部分：前置處理

步驟一：輸入場景點雲 \mathbf{P} ，並建立 KD-tree。

步驟二：利用主成份分析法求出每個點的法向量集合 \mathbf{N} ，以及殘差值(residuals)集合

\mathbf{r} ，其中殘差值的定義為主成份分析法中共變異矩陣的最小特徵值。

步驟三：設定角度閾值 θ_{th} 以及殘差閾值 r_{th} 。

第二部份：區域成長法

步驟一：初始分類矩陣 \mathbf{R} 為空集合。

步驟二：對所有的點進行以下步驟：

1. 設定當前分類集合 \mathbf{R}_c 為空集合，當前種子點 \mathbf{S}_c 為空集合。

2. 找出未被分類點中，殘差值最小的點 \mathbf{P}_{min} 做為起始點。

3. 將 \mathbf{P}_{min} 加入 \mathbf{S}_c 以及 \mathbf{R}_c ，並將 \mathbf{P}_{min} 設定為以分類。

4. 對 \mathbf{S}_c 中的每個點 \mathbf{P}_s 進行以下步驟：

i) 找出 \mathbf{P}_s 的鄰近點集合 \mathbf{B}_c

ii) 對 \mathbf{B}_c 中的每個點 \mathbf{P}_b 進行下列步驟：

a. 若 \mathbf{P}_b 尚未被分類，且 $\cos^{-1}(\left| \langle \mathbf{N}\{\mathbf{S}_c\{\mathbf{P}_s\}\}, \mathbf{N}\{\mathbf{P}_b\} \rangle \right|) < \theta_{th}$

則 將 \mathbf{P}_b 加入 \mathbf{R}_c ，並設定 \mathbf{P}_b 為已分類，並檢查 b.

否則 則進入下一個鄰近點

b. 若 $\mathbf{r}\{\mathbf{P}_b\} < r_{th}$

則將 \mathbf{P}_b 加入 \mathbf{S}_c 。

iii) 直到對所有 \mathbf{B}_c 中的每個點處理後，進入 5.。

5. 將分類結果 \mathbf{R}_c 加入分類矩陣 \mathbf{R} ，進入步驟三。

步驟三：直到所有點都被分類則進入步驟四。

步驟四：將分類矩陣 \mathbf{R} 中每一類別根據點數量由多至少進行排列。

步驟五：輸出分割結果 \mathbf{R} 。

其中， $\langle \mathbf{a}, \mathbf{b} \rangle$ 定義為兩向量之內積，假設， $\mathbf{a} = [a_1 \ a_2 \ a_3]$ ， $\mathbf{b} = [b_1 \ b_2 \ b_3]$ 則， $\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^n a_i b_i$ 。以上為平滑切割的演算法，平滑切割是依靠角度閾值以及殘差閾值來決定切割範圍的，角度閾值決定了各點延伸的程度，角度越大，則可以得到較大的分類平面，角度小則反之；而殘差閾值則是決定了何處要停止平面的延伸，殘差閾值越小則會切割出較多的平面，反之，則會切割出較少的平面，為了讓平滑切割演算法的可用度更高，作者也提出了閾值的建議，建議角度閾值 15 到 20 度以及殘差閾值在所有點的前 2% 到 5% 之間做為閾值會得到較好的結果。實際測試了平滑切割法，發現若物件本身屬於較平滑的物件，例如汽車，這樣流線形的物件，則平滑切割確實可以順利的找出物件；但是如果物體本身並不是平滑的，或者屬於較為方形的物件，例如本論文所使用的坦克車，則無法順利的進行切割。參考了許多論文，大多數自動切割的方法，都是依靠曲率以及法向量夾角等資訊來找出平滑面，而進行分類的，所本論文的目標物，坦克車勢必無法使用這樣的演算法進行自動切割的。

2.3.2 平面切割

在平面切割的部分，本論文參考[15]來實現的，在前一小節介紹了區域成長法配合曲率以及法向量夾角等資訊來進行物件的切割，雖然無法使用在坦克車的抽取，但是卻能夠實現找出平面的功能，但是體感遊樂器在每張圖中提供了約三十萬個點，要對其進行 KD-tree 的建立以及法向量的計算都非常的耗時，所以[15]中則是利用了不一樣的方法來進行法向量及取率的估測，這個方法主要是對於有組織的點雲圖來進行的，靠的就是利用原始的距離影像配合點雲一起來執行法向量的估測，在 2.1 的部分有提到法向量的估測做法是對 k 鄰近點進行主成份分析法，最小特徵直對應的特徵向量就會是該點的法向量，而本方法最大的改變就是鄰近點搜尋方法的不同，做法是，找出該點的在距離影像中的四周的點，做為鄰近點，如圖 2.8，大部分的時候，周圍的點就會是鄰近點，但如果剛好抽取到邊緣點的時候，不只是非鄰近點，甚至可能是距離非常遙遠的點，所以為了避免有太遙遠的點造成法向量估測準確度過差，所以在執行時會利用權重主成份分

析法來進行，權重主成份分析法與主成份分析法類似，共變異矩陣定義為式(2.7)，

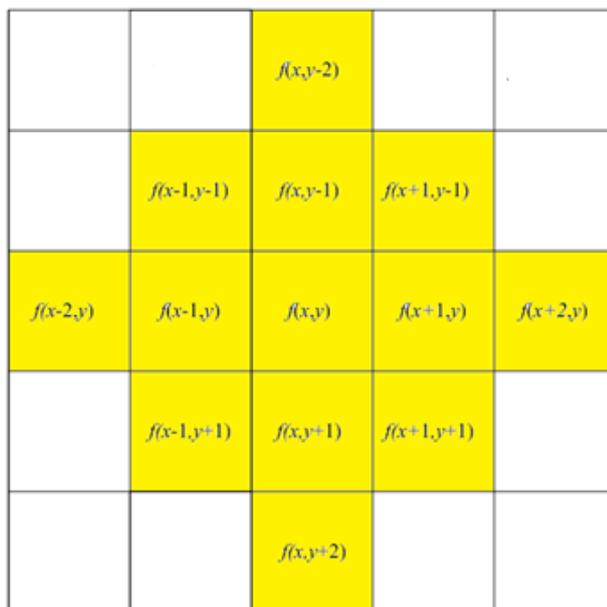


圖 2.8 鄰近點搜尋示意圖。

$$\mathbf{C} = \frac{1}{k} \sum_{i=1}^k \xi_i \cdot (\mathbf{p}_i - \bar{\mathbf{p}}) \cdot (\mathbf{p}_i - \bar{\mathbf{p}})^T \quad (2.7)$$

其中， k 為鄰近點的個數， ξ_i 為第 i 個點的權重值， $\bar{\mathbf{p}}$ 為重心，定義如下：

$$\bar{\mathbf{p}} = \frac{1}{k} \sum_{i=1}^n \xi_i \cdot \mathbf{p}_i \quad (2.8)$$

而權重值的定義是若與點 $\mathbf{p}_{(x,y)}$ 距離小於閾值， $\xi_i=1$ ；否則 $\xi_i=0$ ，其中距離的計算是利用其對應的點雲所算出的實際距離。利用這樣的方法就可以將距離過遠的點刪除，計算出共變異矩陣後再對其求特徵向量，最小特徵值對應的特徵向量就會是法向量，而最小特徵直就是曲率。

利用上述的方法求出法向量與取率特徵後，就可以代入前一章節的演算法中，利用區域成長法的方式找出平面，並找到該平面對應的平面方程式。

2.3.3 歐西里德分類法

歐西里德分類法(Euclidean clustering)，是一個以歐西里德距離為基礎的分類方法，主要目的是將距離足夠近的兩點分為同一類，時常用在高維度的特徵向量上，例如：特徵分類以及兩特徵相似度比較，都常常會使用歐西里德分類法，而本論文是將歐西里德分類法運用在物件的分類，由於一般的場景點雲圖都是連結在一起的，所以本方法必須與前面所提到的平面切割一起搭配使用才能得到較好的效果，而為了加速鄰近點的搜尋，歐西里德分類法也會搭配前面所提到的 KD-tree 一併使用。歐西里德分類法的演算法如下：

步驟一：輸入場景點雲 P ，並建立 KD-tree。

步驟二：設定一個類別空集合 C ，以及一個等待處理的佇列(queue) Q 。

步驟三：對每個點 $p_i \in P$ ，進行下列步驟：

1. 將點 p_i 加入佇列 Q 。

2. 對每個點 $p_i \in Q$ ，進行下列步驟：

i) 搜尋 p_i 半徑 r 內的鄰近點 P_i^k 。

ii) 對每個鄰近點 $p_i^k \in P_i^k$ ，檢查是否已經被分類，若還沒則加入佇列 Q 。

3. 當佇列 Q 中所有的點都被處理過後，將佇列 Q 加入類別集合 C ，並清空佇列 Q 。

步驟四：當所有點都被分類則結束。

以下利用圖 2.9 隨機點雲圖做為範例，圖 2.9 (a) 是一個隨機產生的點雲圖，圖 2.9 (b) 是利用歐西里德分類法分類後的結果。結果圖中可以發現到，歐西里德分類法可以順利的將四個區塊分成四種類別，但是，若根據不同的閾值設定，也可能會造成不同的結果，所以閾值的定義是非常重要的，而閾值的定義方法將會在後續實驗部分進行說明。

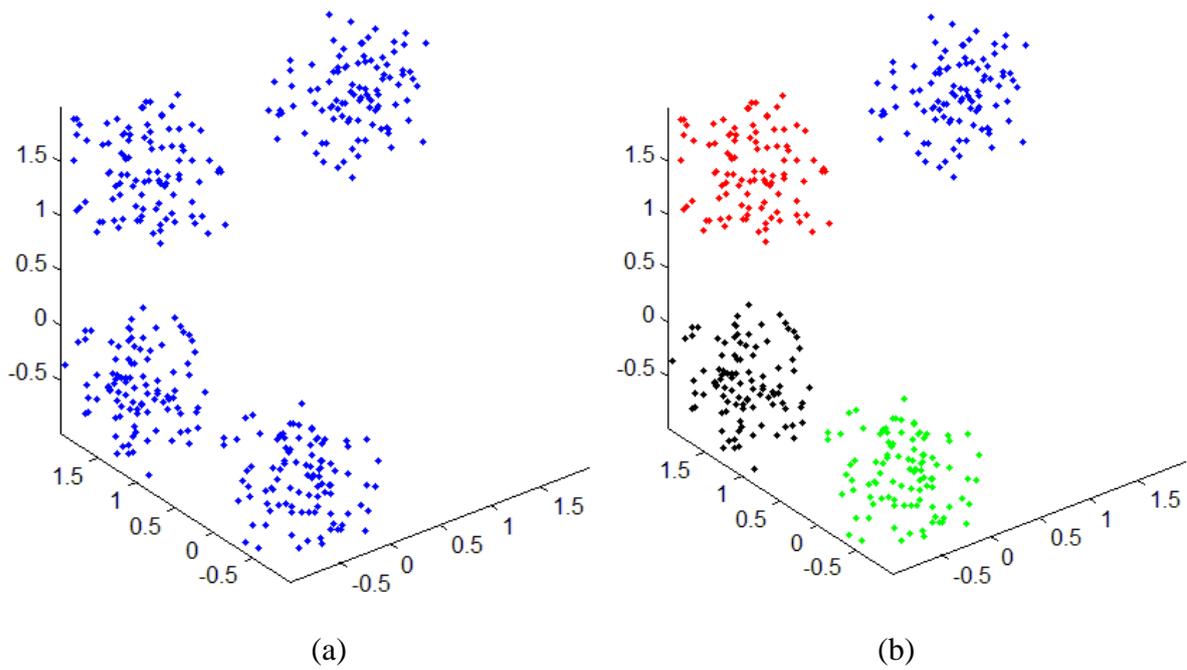


圖 2.9 歐西里德分類範例。(a) 分類前；(b) 分類後。

2.3.4 連結區塊分類法

連結區塊分類法(connected component labeling)[16]，是在過去二維影像中常常使用的分類法，主要目的是將連結再一起的區塊標記為同一個值，一般都會與二值化演算法一起使用，而其連結的定義通常有兩種，如圖 2.10：

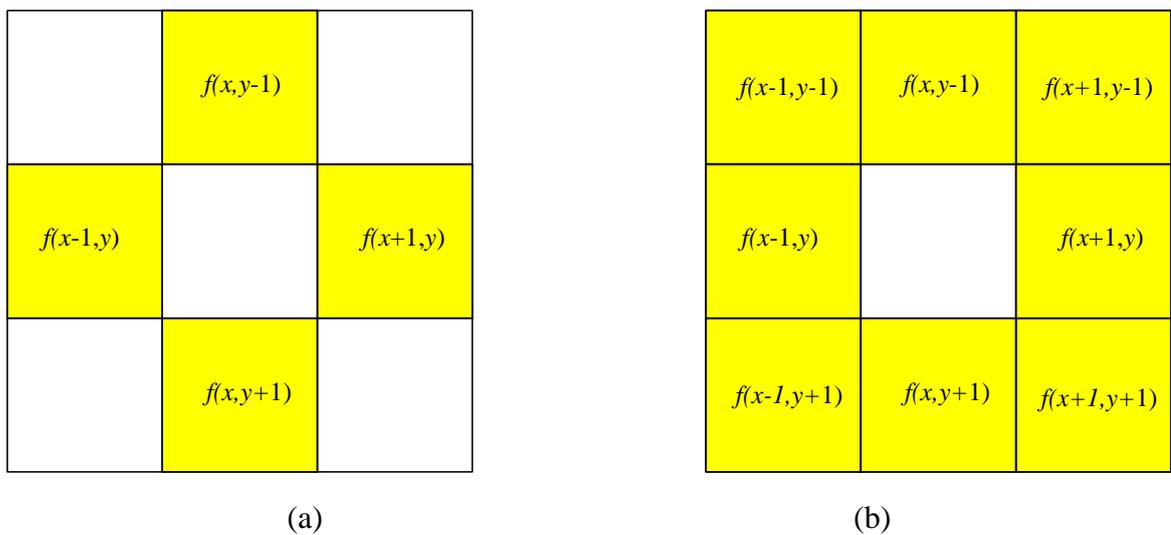


圖 2.10 連結區塊定義 (a) 4-鄰近點；(b) 8-鄰近點。

圖 2.10 (a)中是 4-鄰近點(4-adjacent)，圖 2.10 (b)中是 8-鄰近點(8-adjacent)。而連結區塊分類法主要分為兩個步驟：

步驟一：由左至右，由上至下的掃描影像中的所有像素，並給予標記值，標記值是以鄰近點中，標記值最小的為主，若鄰近點都未被標記，則給予一個新的標記值。

步驟二：再次掃描所有像素，若相鄰點標記不同，則重心標記為鄰近點中的最小值。

以下圖 2.11 做為範例：

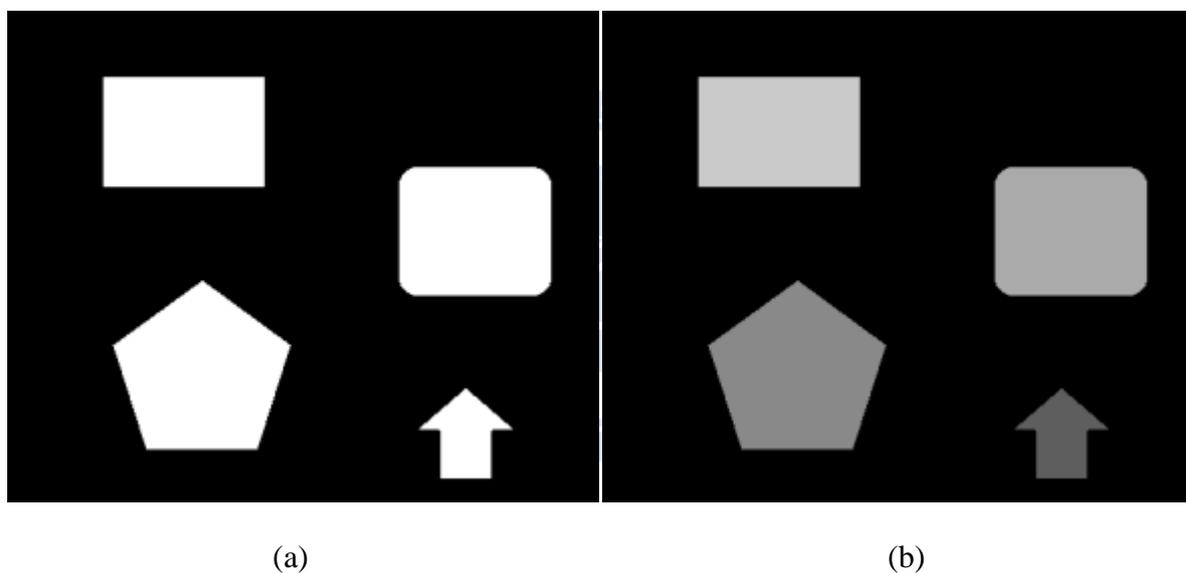


圖 2.11 連結區塊分類範例。(a) 分類前；(b) 分類後。

圖 2.11 (a)中是一張二值化的影像，可以看到裡面有大小形狀不同的四個物件，通過連結區塊分類的方法後，可以得到如圖 2.11 (b)的結果，四個物件分別被標記為不同數值，並達成分類的結果。

2.4 姿態估測

姿態估測是本論文實驗的重點，目前姿態估測最常被使用也最簡單的方法就是利用

前面章節介紹的主成份分析法來進行，利用主成份分析法，找出共變異矩陣中最大特徵值對應的特徵向量，就是物體的姿態，但是由於主成份分析法對於一個不完整物體的主軸估測會有很大的影響，為了達到更好的姿態估測結果，所以本論文選用最小平方法 (least square method) 來進行姿態估測的動作，最小平方法的做法是：

假設有 n 筆隨機資料： $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ，希望找到一個函數： $f(x) = ax+b$ 來逼近這組資料，但用一般的方式，兩個變數加上 n 個式子，這將會是無解的，所以用最小平方法來求解這樣的函數。最小平方法定義誤差：

$$e = \sum_{i=1}^n \|y_i - f(x_i)\|^2 \quad (2.9)$$

最小化誤差 e 來找到逼近樣本資料的曲線。考慮圖 2.12：

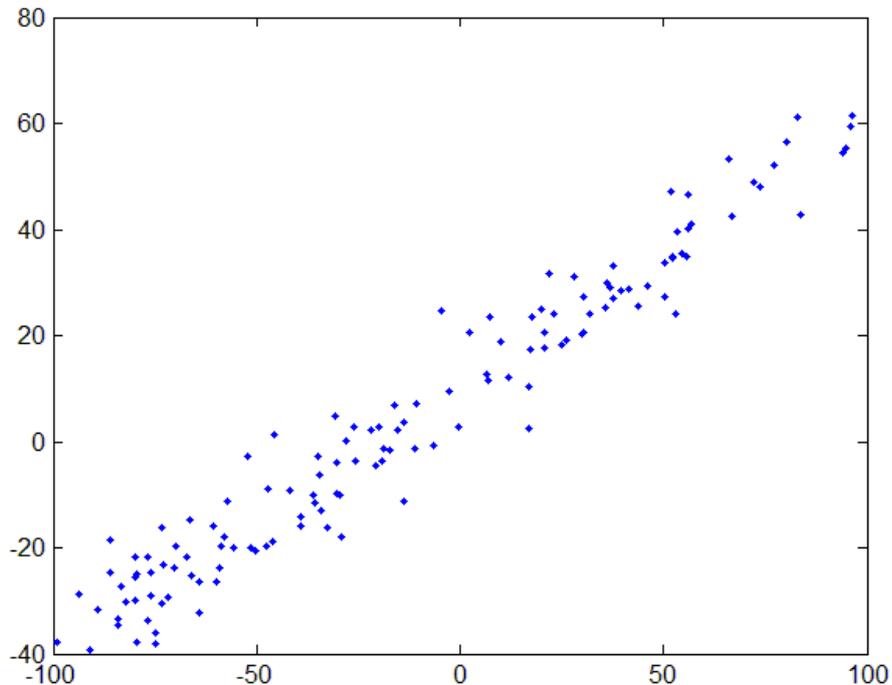


圖 2.12 隨機分佈之二維資料。

對圖 2.12 進行最小平方方法的估測可以得到結果如圖 2.13：

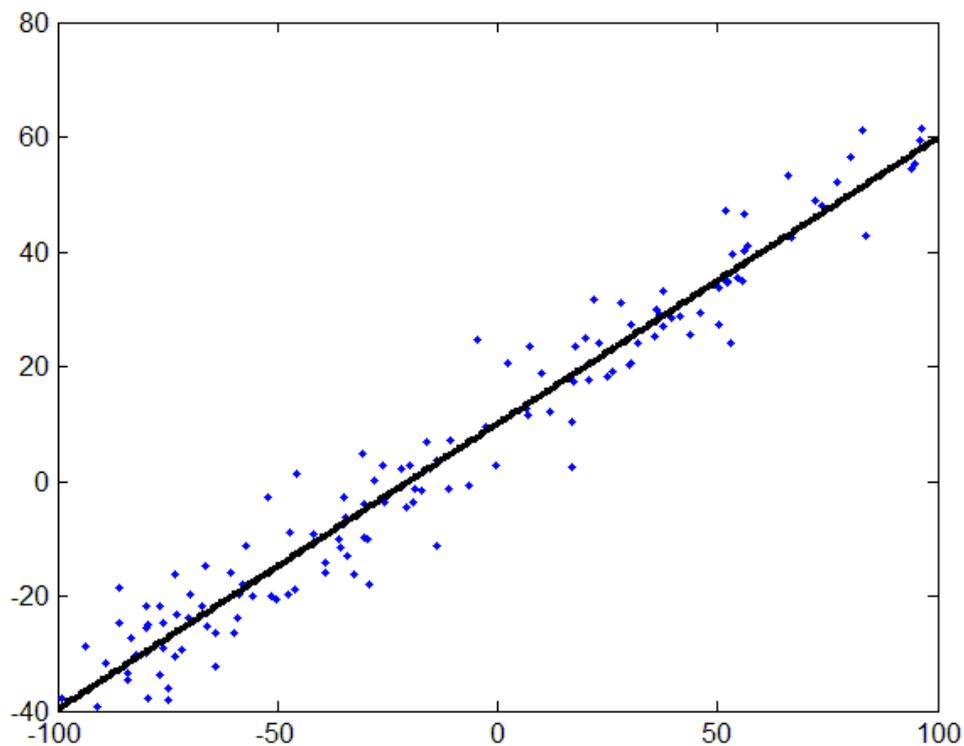


圖 2.13 最小平方方法估測結果圖。

從圖 2.13 可以看到利用最小平方方法，可以找到這筆隨機點資訊的逼近函數，而論文也將以這樣的方式來對炮管以及炮塔的角度進行姿態估測，下一章將開始將對本論文的系統架構進行說明，包含找尋坦克的前處理，以及如何對坦克進行切割，最後再提出本論文姿態估測的相關演算法。

第三章 系統流程

在第三章的部分，將會對本論文提出的系統架構進行詳細的說明，首先說明資料的取的方式，接著再說明當取得場景資料後，如何找出興趣目標，以及如何進行姿態估測。本章將分為 4 個小節，3.1 節先介紹本論文的坦克車資料取得來源；3.2 節會介紹系統的完整流程圖；3.3 節會介紹如何找出場景中的坦克車；3.4 節則會介紹如何對坦克車進行姿態估測。

3.1 資料取得方式

本論文為了讓整個理論有更高的實用性，考慮實際雷射測距儀會含有雜訊的情況，所以並未使用電腦模擬的方式來進行取像，而是使用了等比例縮小的坦克車模型，如圖 3.1：



圖 3.1 實驗用坦克車。

圖 3.1 本論文所使用的坦克車模型，為德國虎式三代，一比十六的模型，而本論使用的取像工具是由微軟公司所開發的體感遊樂器做為取像的來源，如圖 3.2：



圖 3.2 體感遊樂器(kinect)。

圖 3.2 是本論文使用的取像工具，可以用來提供三維距離影像，而本論文為了使計算出的姿態能夠有好的比較標準，所以利用鐵架來幫助世界座標的定位，如圖 3.3：

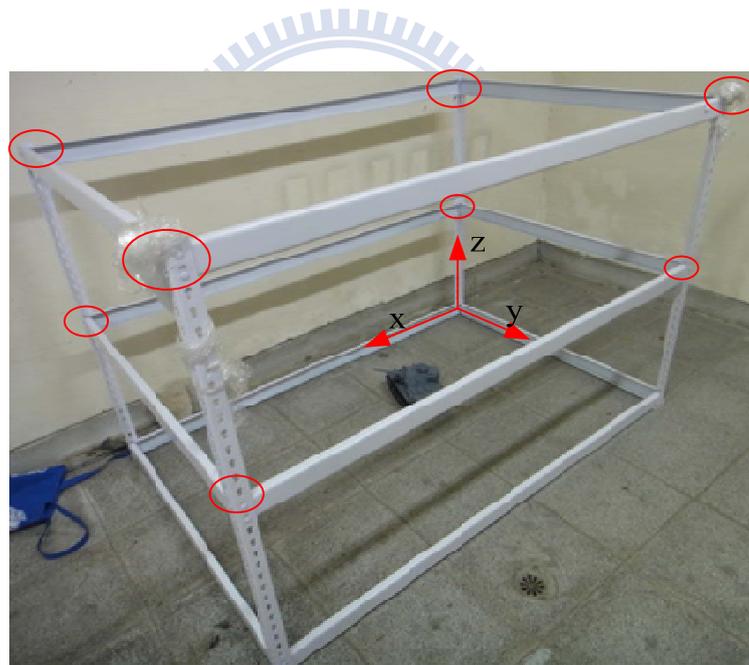


圖 3.3 實驗用鐵架。

圖 3.3 中的紅圈是本論文用來架設體感遊樂器的位置，大約相對戰場會是周圍高山、堡壘頂端，或者直升機飛過的高度位置。最裡面點上的 x 、 y 、 z 則是本論文定義的視界座標，利用鐵架的好處是，只要根據攝影機架設的位置，就可以很容易的講點雲轉成世界座標。而圖 3.4 以及圖 3.5 是利用體感遊樂器對場景拍攝的範例圖以及點雲圖：

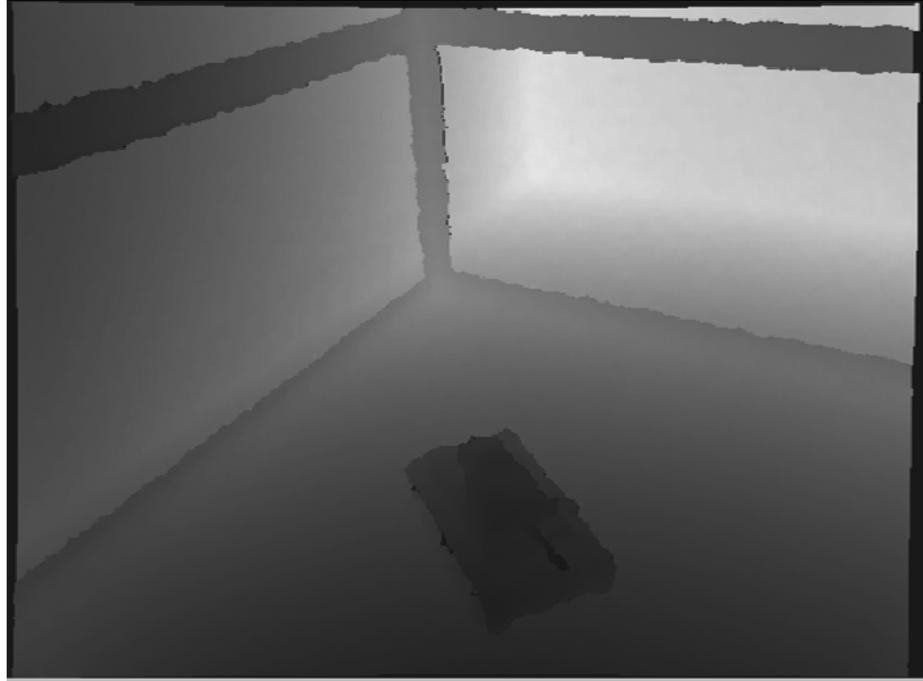


圖 3.4 鐵架上拍攝的距離影像圖。

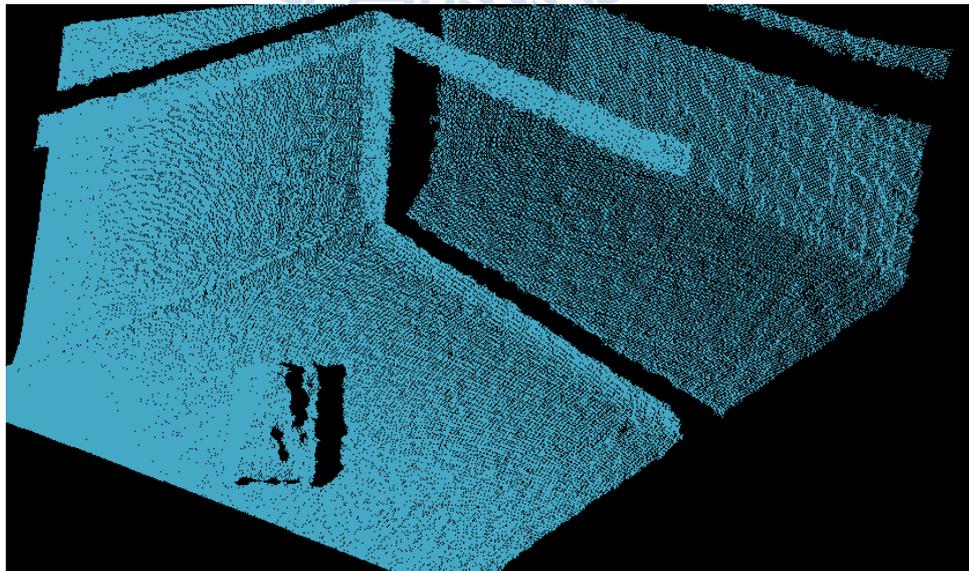


圖 3.5 從圖 3.4 計算得到的點雲圖。

圖 3.4 是在鐵架上拍攝到的距離影像，利用式(2.1)就可以將距離影像轉換成點雲資訊，可得到如圖 3.5 的結果，本論文也將同時利用兩種資訊來進行實驗，來達到更高的效率。

3.2 系統流程圖

本論文的流程中，主要分為兩個部分，第一部份：前處理，目的是找出整個場景中可能是坦克車的物件；第二部分：姿態估測，目標是將找到的坦克車進行切割，並對其進行姿態估測，整體流程圖如圖 3.6：

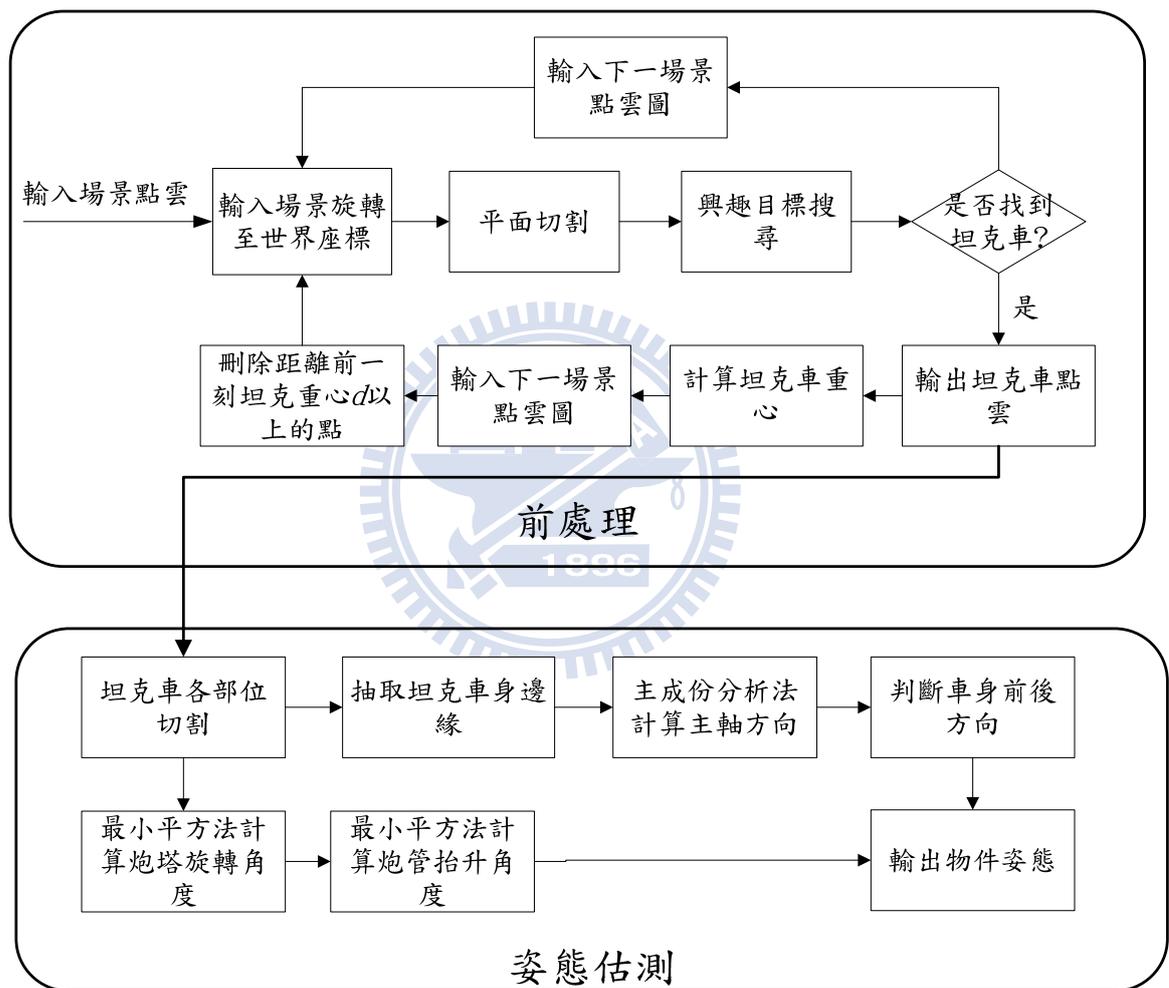


圖 3.6 系統流程圖。

圖 3.6 中，前處理的部分，由於解析度為 640×480 ，每次會有三十萬點需要被處理，會造成速度非常慢，為了不要每次都對如此大量的點雲處理，所以當前處理判斷已經找到可能的目標物件時，會對找到的興趣目標計算其重心位置 C ，下一場景開始，會先對整

個場景中，距離重心 C 距離超過閾值的點雲進行刪除的動作，這樣的一個處理，可以讓整體需要被處理的點雲減少到五至十萬點，不僅可以減少找錯興趣目標的機率，也可以加速整個前處理的流程。

3.3 系統前處理

系統前處理主要分為三個步驟，將分別利用一個小節進行介紹，3.3.1 節首先介紹利用點雲資訊找出場景中的平面的結果；3.3.2 節則會利用距離影像對切割過後的結果進行物件分群；3.3.3 節會將搜尋分群後的物件中，找出有可能是坦克車的物件。

3.3.1 場景平面抽取及切割

在場景平面抽取及切割的部分，是利用前一章所提出的方法來執行的，考慮下列場景，如圖 3.7：



(a)

(b)

圖 3.7 場景影像。(a) 場景二維影像；(b) 對應之三為點雲圖。

利用平面切割演算法，可得到平面點雲圖，如圖 3.8，觀察可以發現，順利的找出場景中的平面。找出平面後，將場景中的平面刪除，可以得到圖 3.9，圖中的平面已經被去除，剩下許多點雲區塊，接下來，為了找出場景中的坦克車，首先必須先將各個區塊的物件進行分類，下一小節將對物件分類進行討論。



圖 3.8 切割平面結果。

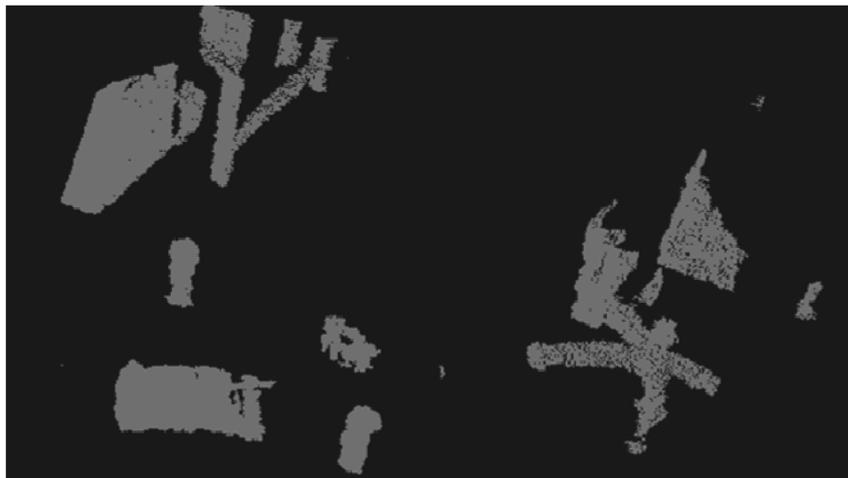


圖 3.9 去除平面後之場景。

3.3.2 物件分群

在前面的部分有提到點雲資訊是由距離影像經過一定的運算得來的，也就是說，距離影像上的每個像素，都可以對應到點雲資訊上的某一個點，反過來說，每當利用平面切割的方法刪去點雲中任一個點的時候，也等於將距離影像中的某一個像素值歸為零，假設有一場景經過平面切割後，如圖 3.10：

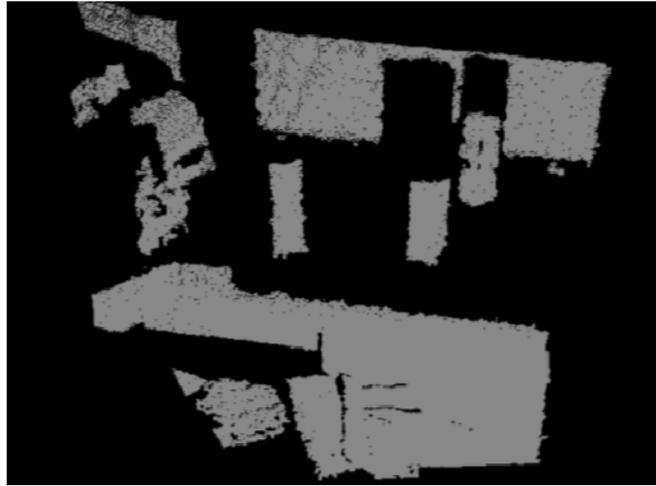


圖 3.10 平面切割後點雲資訊。

可以將圖 3.10 這筆點雲資訊轉換成為距離影像如圖 3.11：

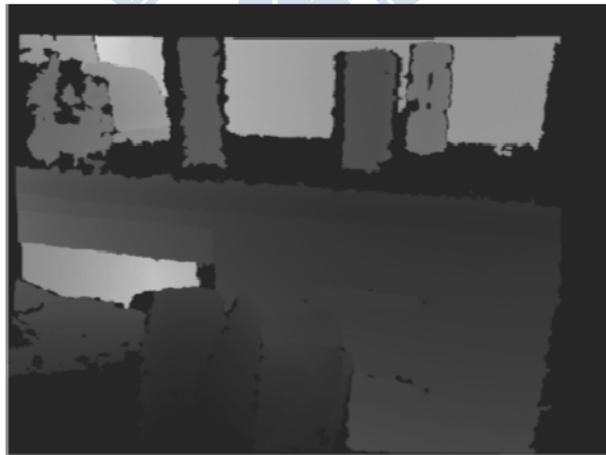


圖 3.11 平面切割後對應之距離影像。

圖 3.11 為將平面切割後點雲圖對應的距離影像，這邊利用連結區塊分類法，可以將物件進行分群，而這邊不同於過去連結區塊法的地方是，過去的連結區塊法會配合二值化一起使用，但本論文提出了不同的方法，不利用二值化，而是使用距離影像擁有深度的這個特色來進行切割，做法與傳統連結區塊分類法有些微的不一樣，傳統連結區塊的演算法是：

步驟一：由左至右，由上至下的掃描影像中的所有像素，並給予標記值，標記值是以鄰近點中，標記值最小的為主，若鄰近點都未被標記，則給予一個新的標記值。

步驟二：再次掃描所有像素，若相鄰點標記不同，則重心標記為鄰近點中的最小值。

而本論文的做法不同，給予標記值之前，會先計算兩個點的深度差，如果小於閾值才給予為同一個標記值，如果大於閾值則給予新的標記值，若同時有多個不同位置符合結果，則是以標記時較小的為主。利用這樣的方法可以得到結果，如圖 3.12



圖 3.12 連結區塊法分類結果。

在物件分群中，會設定每個類別的最大值以及最小值以避免掉過多不必要的分類，根據圖 3.12 可以發現，在圖的左上角可以看到有四塊鄰近的區塊，如果以過去傳統連結區塊法會連結成同一區塊，但是利用深度差的方式來執行的話，可以將四塊不同的物件都切割開來。為了證明本方法的正確性，最後也將這樣分類的結果與過去常使用的歐西里德距離法來進行分類的結果比較，如圖 3.13：



圖 3.13 歐西里德距離分類法分類結果。

可以發現分類結果是一樣的，證明利用距離影像進行分類，因為不必建立 KD-tree，並且利用 4-鄰近點的方式也減少了點雲中搜尋鄰近點的步驟，所以在時間上有很大的優勢，並且結果也是一樣的，從圖 3.13 中可以看到在深度不同，多物件連接的位置，利用連結區塊法也是可以順利分割，而決定是否可以分割的依據是靠閾值來決定的，閾值若設定過小會造成同一物件被過度分割，相反的閾值若設定太大會造成兩個不同物件被分為同一類，所以如何設定閾值是非常重要的工作，在這邊本論文也提供閾值設定的方法，閾值 r_{th} 本身代表的意義是兩個物件若距離超過 r_{th} 則會被分成兩類，所以在設定時，則是以考量物件會距離多遠來進行設定，但是要考慮到點雲的密集問題，也不能將閾值設定太小，以本論文為例，設定閾值為 2 公分。

3.3.3 興趣目標搜尋

興趣目標搜尋的部分主要是依靠點雲來執行的，由於三維影像是以距離作為表示，所以不論雷射測距儀距離物件多遠，量到物件的尺寸只會有點數的差異，而不會有大小的不同，如圖 3.14：

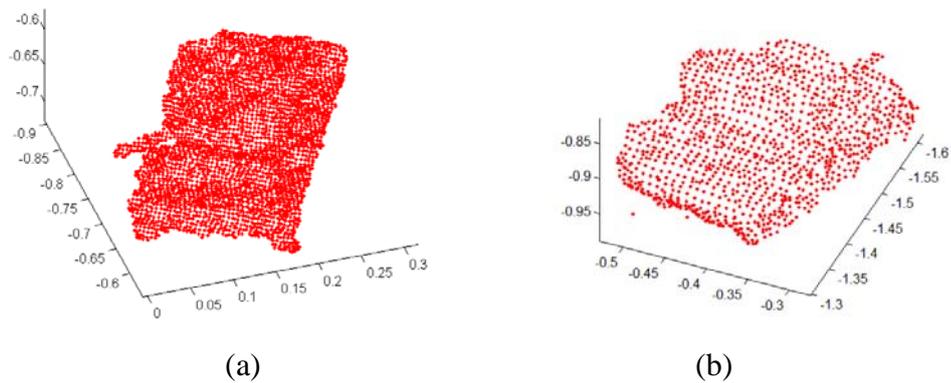


圖 3.14 不同距離拍攝之坦克車。(a) 15179 點；(b)10942 點。

圖 3.14 中，(a) 圖與(b) 圖是同一台坦克在不同位置及距離下拍攝的結果，可以發現不同距離之下，點數相差甚多，(a)有 15179 點，(b)則只有 10942 點，但若將兩圖放在同一座標軸，如圖 3.15

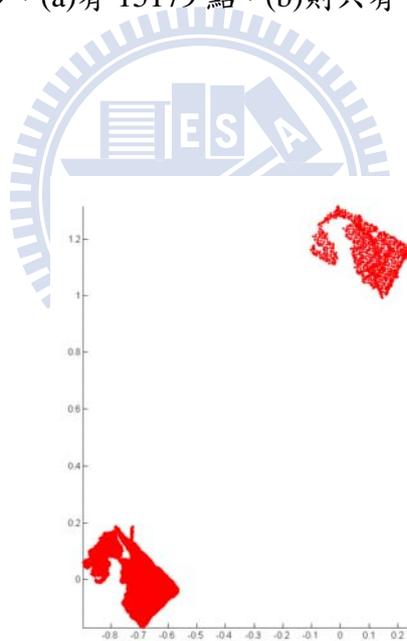


圖 3.15 不同距離之坦克車圖。

可以發現，兩圖雖然點數不同，但是整體大小並不會因為距離而改變，而本論文就是依靠這樣的特性來確認哪些物件為興趣目標。

考慮到物體會形變，找出來的尺寸在不同的形態下會有不一樣的結果，本論文的作法是：

步驟一：將物件根據與平面之距離 r 分為上下兩個部份。

步驟二：對下部份的物件進行主成份分析，求出長度 l 以及寬度 w 。

步驟三：根據長度以及寬度，檢查是否大於長寬閾值 l_{th} 、 w_{th} ，

若大於閾值則非坦克車；

若未大於閾值進入步驟四。

步驟四：找出上部分的高度是否大於高度閾值 h_{th} ，

若大於閾值則非坦克車；

若未大於閾值則為坦克車。

其中 r 的設定主要目的只是要找出下部分皆為坦克車身，並非要很精準的將物件分為車身及砲塔，所以 r 的設定只要不要過高，切到形變部分即可，以本論文為例，設定 $r=5$ 公分(換算實際距離為 $5*16=80$ 公分)； l_{th} 的設定則是以物件的對角線會為最大值， w_{th} 的設定則是當長軸為對角線時所對應的寬度，如圖 3.16 所示：

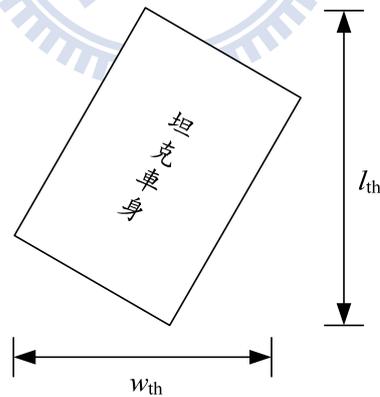


圖 3.16 長寬閾值定示意圖。

由於本論文目標是對所有坦克車進行姿態估測，所以長寬閾值並不能只為單一台坦克車做設計，但是大部分的坦克車在長度以及寬度都相似，所以設定閾值時則是以大部份坦克車的大小來做設定，以本論文為例，閾值設定為 $l_{th}=50$ 公分(換算實際距離為 $50*16=800$ 公尺)， $w_{th}=20$ 公分(換算實際距離為 $20*16=320$ 公尺)，最後高度閾值 h_{th} 的部分的設定目

標是將類似樹、圍牆以及房子等這些可能出現在戰場中，設定上也是依照大部份坦克車炮管高度來調整，以本論文為例 $r = 20$ 公分(換算實際距離為 $20 * 16 = 3.2$ 公尺)，利用以上的參數搭配本論文提出的演算法就可以順利的找到場景中的興趣目標，以下圖 3.17 為例：



圖 3.17 坦克車場景。

圖 3.17 中為含有坦克車之場景，經過平面切割以及分類後可以得到下圖 3.18：

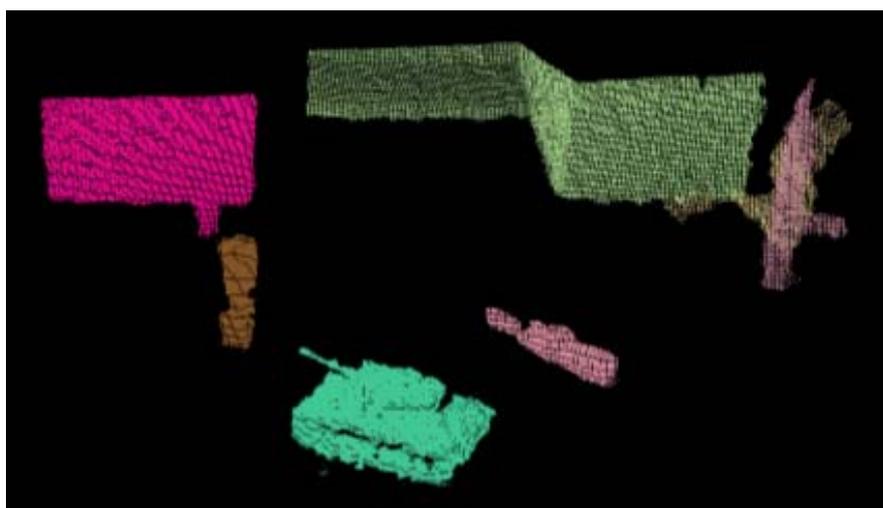


圖 3.18 物件分類結果。

圖 3.18 中是利用連結區塊分類得到結果，再利用興趣目標切割法可以得到如圖 3.19：

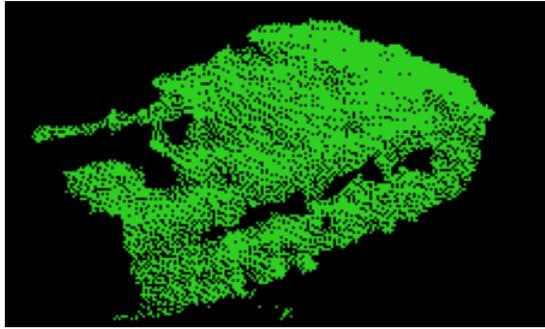


圖 3.19 興趣目標搜尋結果。

圖 3.19 中可以發現，利用本論文的演算法可以順利獲得坦克車的點雲資訊，下一章節將開始介紹本論文姿態估測的相關演算法。

3.4 物件姿態估測

物件姿態估測的部分在過去時常使用疊代最近點的方式進行，或者是利用主成份分析法，但是疊代最近點的缺點有以下幾點，第一是速度很慢；第二是若初始位置不佳，可能會使疊代失敗；最後是需要對物件建立資料庫才得以使用。而主成份分析法相較於前者，速度快非常多，但是對於局部物件的估測效果很差，常常會偏離期望的主軸方向。

綜觀以上幾點，本論文的目標是希望速度快，且準確度要高，疊代最近點的方法勢必是無法使用，主成份分析法速度快，但是對局部點雲資訊敏感度很高，所以本論文首要目標是提出一個良好的物件切割，利用一前處理的方式來解決主成份分析法主軸偏離的問題，本章將分為兩個小節，3.4.1 節首先介紹如何進行物件切割；3.4.2 節再介紹本論文如何進行物件的姿態估測。

3.4.1 物件切割

人造的機械中，絕大多數都是由許多小關節所組成的，而本論文的實驗對象坦克車也是一樣，要對這種可形變的剛體進行影像處理，切割是不能少的，所以如何將物體在

關節處切割開來變成為了重要的問題了，參考過去的論文[5][6][7][8][9]中，除了[5]是利用疊代最近點的方式估測，其餘皆是利用切割的方式來做，這幾篇論文中提出了幾種切割方法，i) 多長方體包覆切割[6];ii)利用物體高度資訊切割出車身以及砲塔[7][8][9]。

首先考慮第一種多長方體包覆切割，這個方法是先將整個形變剛體力用一個長方體進行包覆，然後計算其密度，若密度不到閾值，則再多用一個長方體包覆一直到每個長方體包覆的密度都達到目標，如圖 3.20：

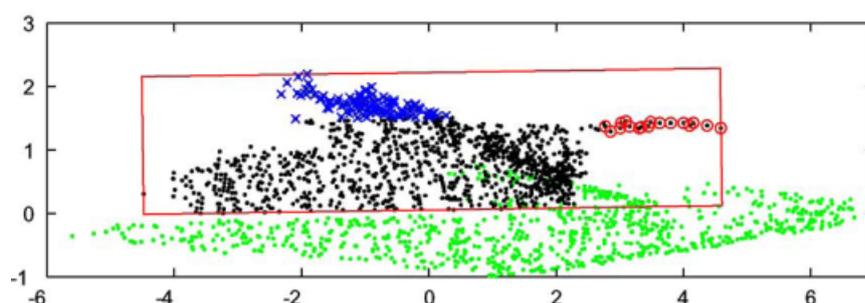


圖 3.20 文獻[6]長方體包覆切割結果。'x'為砲塔，'o'為炮管，黑點為車身。

圖 3.20 可以看到最後的切割結果，分別找到了砲塔、炮管以及車身，但是並沒有很完整的將三個部分切割開；而另外一種常用的切割方法是利用高度的方式切割坦克車，這樣的方法切割效果不錯，但只能用於車身與砲台及大炮的切割，並無法將炮管以及砲塔切開，如圖 3.21：

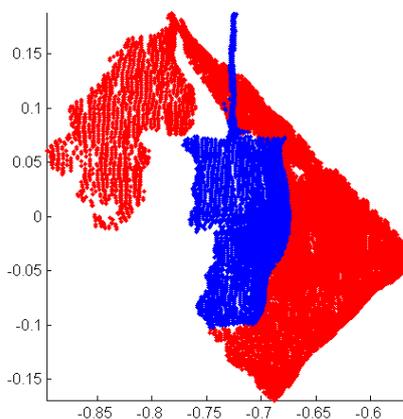


圖 3.21 利用高度資訊切割結果。

圖 3.21 可以發現很簡單的就找出了車身以及砲塔部分，但論文中並未提及如何切割炮管及砲塔，並且這樣的演算法是必須知道坦克車的高度才有辦法執行的，但本論文的目标是大多數的坦克車都能夠適用，所以這樣的切割方法除非能夠自動找出高度，否則無法使用的。根據上述幾點，本論文的切割目标有以下幾個，i)自動找出坦克車的高度，以利車身及砲塔的切割，ii)自動的切割炮管以及砲塔。

自動找出高度的部分，考慮圖 3.22 這樣未切割的坦克車圖，本論文提出下列演算法來達成：

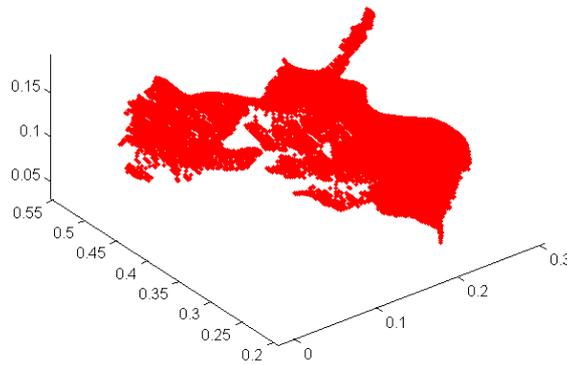


圖 3.22 未切割之坦克車圖。

步驟一：將點雲資料根據 z 軸的高度 h_x 由上至下，每次增加 h_x/n 的大小，將物件分為上下兩部份，並對上部分以主成份分析法分析其寬度(y 軸差)。如圖 3.21：

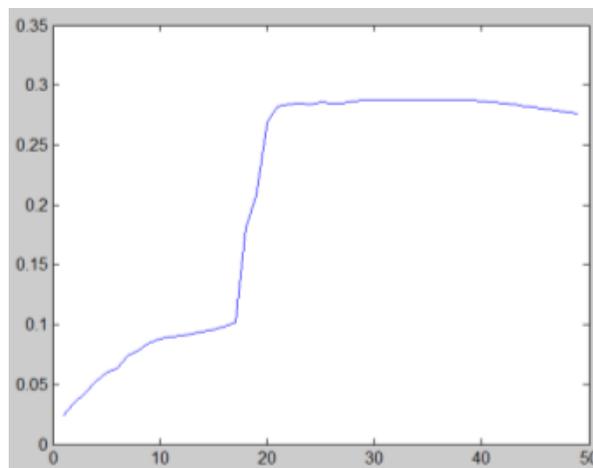


圖 3.23 不同高度時點雲之寬度圖。

步驟二：找出寬度圖中分別變化量最大的位置，如圖 3.23 不同高度時點雲之寬度圖，得到當 $step=17$ 時，變化量為 $\Delta_z=0.07933$ ，如圖 3.24：

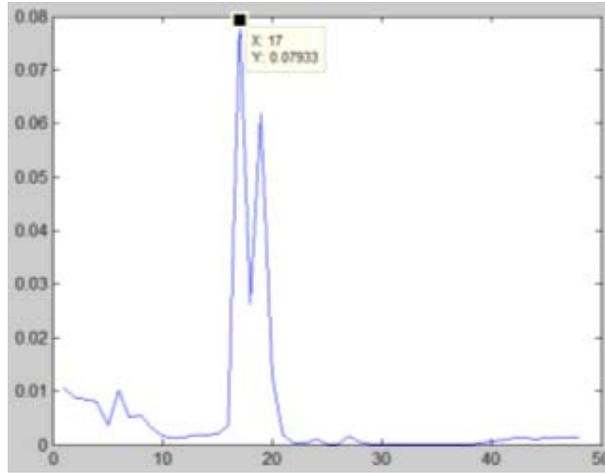


圖 3.24 寬度變化圖。

步驟三：找出 $step=17$ 對應的高度 $z=0.1402$ ，根據該高度直進行切割，上部即為炮管、炮塔，下部則為車身，即完成切割。結果如圖 3.25：

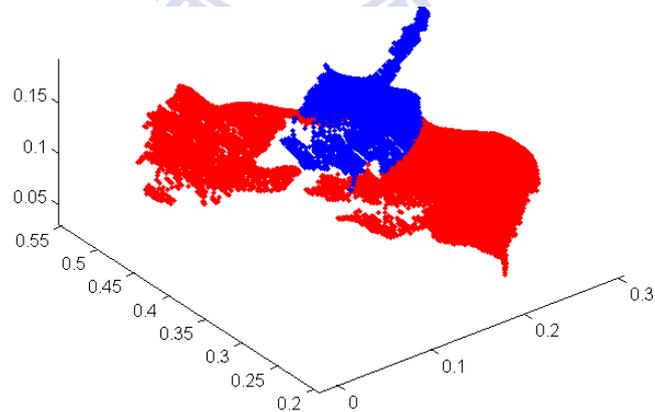


圖 3.25 根據 $z=0.1402$ 切割結果圖。

根據上述幾個步驟就可以完成高度的自動搜尋，接下來將繼續對炮管以及炮塔的部分進行切割。對炮管切割的部份，本論文提出一個演算法，是利用炮管跟炮塔的寬度不同這個特性來完成的，考慮下圖 3.26：

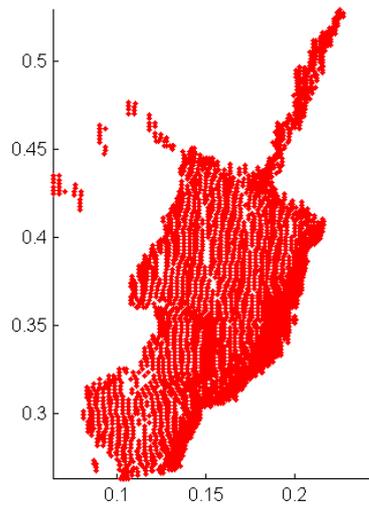


圖 3.26 待切割炮管。

圖 3.26 中可以發現炮管以及炮塔的寬度不同，所以利用以下演算法來進行切割：

步驟一：對上部點雲進行主成份分析，並主軸轉至 x 軸。

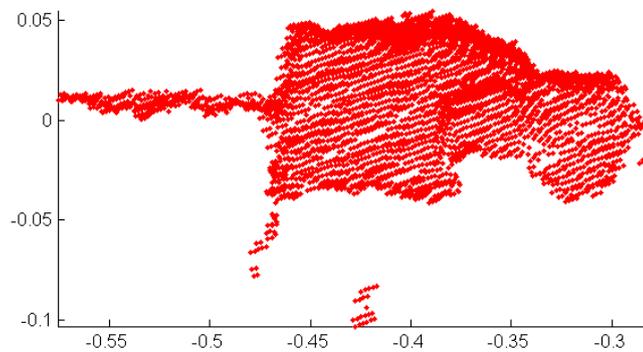


圖 3.27 將圖 3.26 轉至 x 軸結果圖。

步驟二：將點雲資料根據 x 軸的長度 l_x 由左至右，以及由右至左以每次增加 l_x/n 的大小，並計算其寬度(y 軸差)。如圖 3.28，

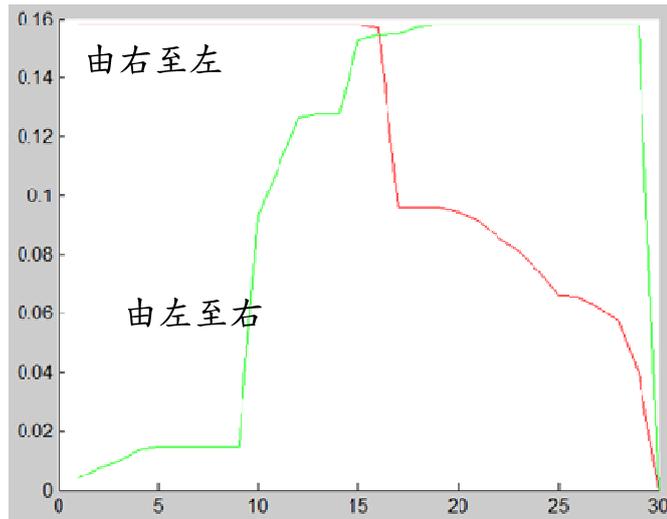


圖 3.28 兩個方向之寬度圖。

步驟三：找出寬度圖中分別變化量最大的位置，如圖 3.28，由右至左的寬度圖變化最大為 $step=16$ ，變化量為 $\Delta_r=0.0605$ ；由左至右的寬度圖變化最大為 $step=9$ ，變化量為 $\Delta_l=0.0789$ ；

步驟四：比較 $\Delta_l > \Delta_r$ ，找到切割點為 $step=9$ 對應的切割位置 $x = -0.4956$ 。

步驟五：找出炮管後再次對炮管部份執行主成份分析法，依該角度將資料轉至 x 軸，並重複執行以上步驟二至步驟四，直到炮管角度變化小於閾值。

步驟六：得到切割結果，如圖 3.29：

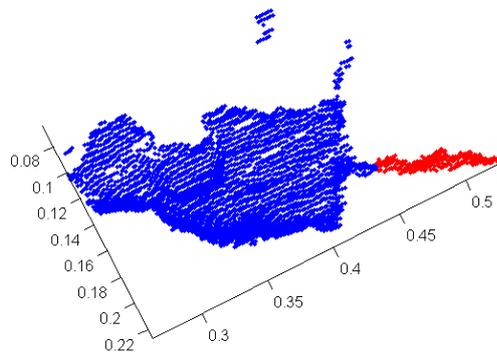


圖 3.29 炮管切割後之結果。紅色為炮管；藍色為炮塔。

根據以上的步驟，就可以得到如圖 3.30 的坦克切割圖。

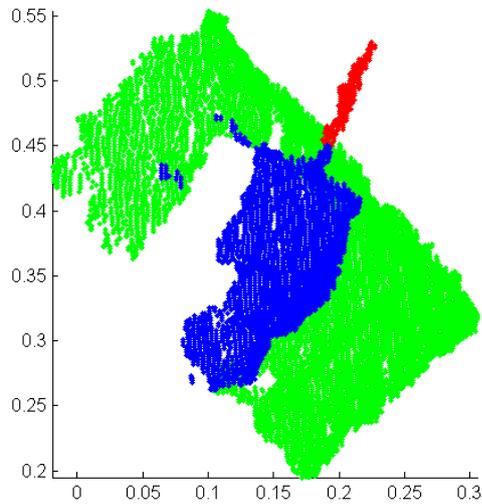


圖 3.30 坦克切割結果圖，綠色為車身，紅色為炮管，藍色為炮塔。

成功的將坦克車做切割後，即可開始對坦克車進行角度估測。

3.4.2 角度估測

在角度估測的部分，分為兩個部份，第一是車身角度；第二是炮塔及炮管角度。前小節已經提出了，利用自動搜尋高度的方式就可以切割出坦克車車身的部分，也提出過要對一比資訊快速計算角度的方式就是利用主成份分析法，若直接將切割過後的車身部分進行主成份分析法，會得到圖 3.31，從圖中的綠色主軸線可以發現，由於是側拍得到的局部點雲資訊，造成主軸偏差，很明顯的並非我們所期待的車身方向。為了得到更正確的車身方向，本論文的做法，考慮車身的方向等同於車身邊緣(edge)的方向，如果能夠抽取出車身的邊緣，那車身方向也就唾手可得，所以本論文第一步是想辦法找出車身的邊緣。

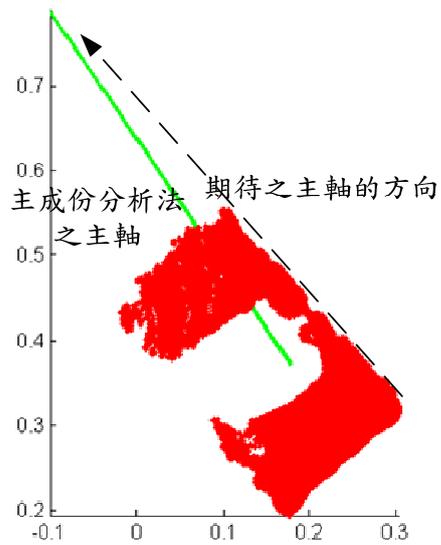


圖 3.31 主成份分析主軸結果(從上往下看)。

過去邊緣抽取的演算法，都是針對二維影像所開發的，針對三維點雲資訊，本論文的做法是，首先將三維點雲投影到 xy -平面，再將資料網格化，考慮下圖 3.32，由於物件的邊緣有高低的落差，框框內的邊緣會拍攝到較多的點數，所以本論文利用這樣的特點來對點雲進行網格化的邊緣抽取。

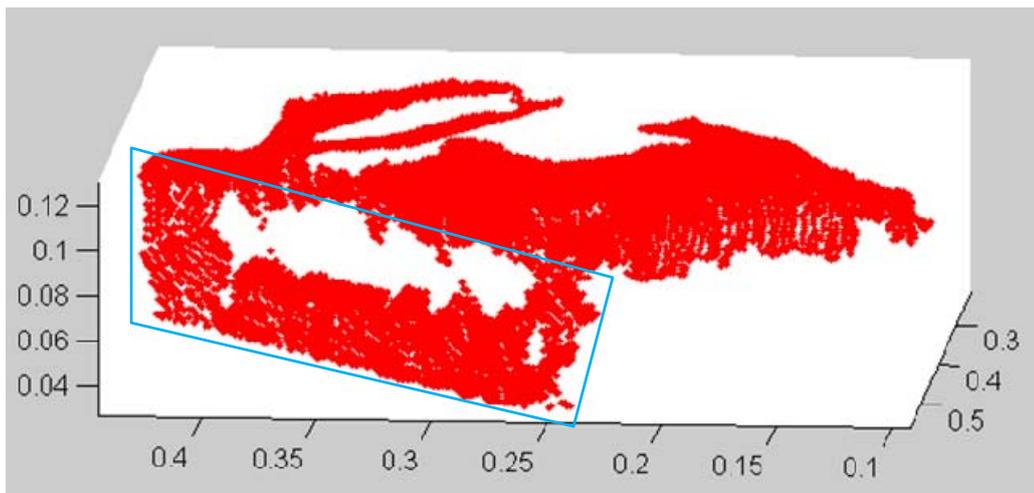


圖 3.32 邊緣抽取範例圖。

首先將點雲資料投影到一個 xy -平面，如圖 3.33，再分別在 x 軸以及 y 軸以一定的距離 d 將點雲資訊給與一個索引值， d 的設定建議是與儀器的精準度相同。

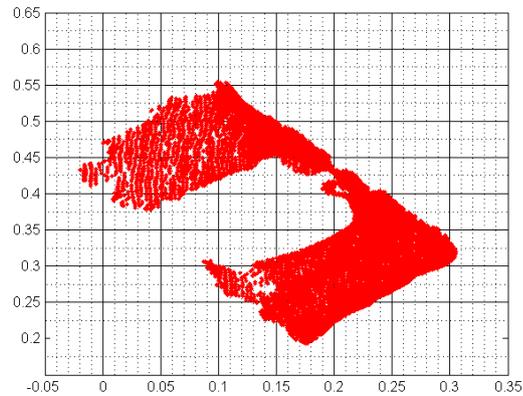


圖 3.33 網格化示意圖。

以本論文為例，是根據體感遊樂器的精度設定 $d=1$ 公分。網格化後，計算每一格中，含有多少個點雲資訊，以二維影像的方法表示如圖 3.34，圖 3.34 中可以看到邊緣部分亮度相對中間來的高，這是因為邊緣的部分是從高到低都有點雲分部，邊緣的像素質會因為含有較多點雲資訊所以來的較高，對圖 3.34 找出像素質高的前 n 筆可以得到如圖 3.35 的結果。



圖 3.34 網格化結果。

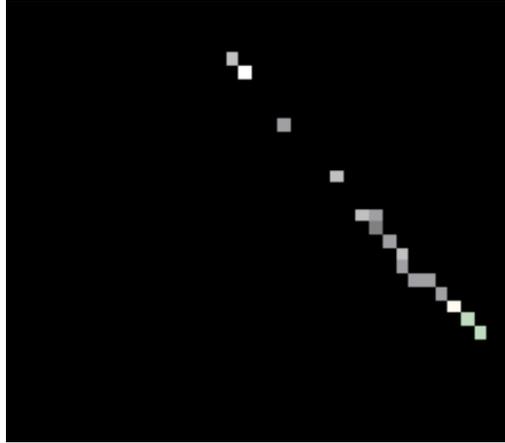


圖 3.35 抽取最多點雲區塊結果。

並利用圖 3.35 的結果，找出這些像素對應的點雲資訊，如圖 3.36，抽出邊緣點雲後，因為資料已經不再是局部資訊所以直接進行主成份分析法，即可獲得車身角度，如圖 3.37。觀察圖 3.37 可以發現估測出的主軸較圖 3.31 所估測出的主軸，與期待的车身方向更為符合，但估測車身方向，仍然有一個問題，就是到底哪邊是車頭？哪邊是車尾？兩者相差一百八十度，僅靠主成份分析抽取出的主軸，是無法得知的，所以觀察了坦克車本身，發現由於一般坦克車中，座艙一般皆會位於坦克車的後方，為了要讓裡面能夠運載更多士兵，通常空間會較大、較高，而車頭的部分因為是引擎的位置，所以並不需要太高，這樣的原因，讓大多數的坦克車，有著前面低，後面高的特性，所以本論文也依照這樣的特性，比較主軸前端以及後端點雲的高度，較低者為車頭；較高者為車尾，若主軸估測相反，則將主軸向量乘上-1 就可以得到正確的主軸向量。

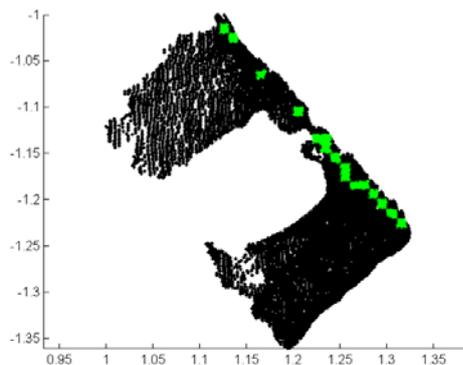


圖 3.36 綠色點為對應之邊緣點雲。

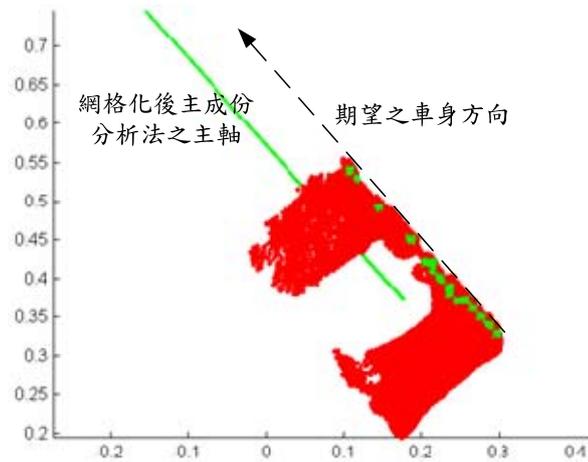


圖 3.37 配合網格化邊緣抽取之主軸結果。

有了主軸向量 $v=(v_x, v_y, v_z)$ 後，將向量帶入下式(3.1)，就可以得到車身角度(yaw)，以上是車身方向的估測方法，接下來將介紹炮塔以及炮管的估測方式。

$$\theta = \cos^{-1}\left(\frac{v_x}{\sqrt{v_x^2 + v_y^2}}\right) \quad (3.1)$$

炮管及炮塔的角度估測與車身估測的方式類似，都是搭配主成份分析法來完成的，在過去，大多數的論文並未對炮管進行切割，對炮塔的估測都是將炮塔加上炮管是為一個物體來進行角度的估測，但這樣效果並不好如圖 3.38：

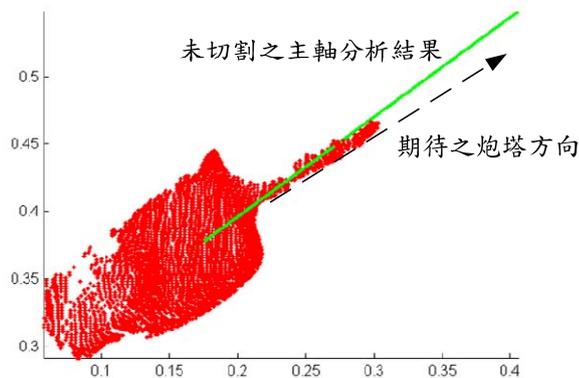


圖 3.38 未經切割之主軸分析。

觀察圖 3.38 中可以發現未經切割的主軸分析結果，與期待的炮塔方向是有落差的，主要原因也是因為單側拍攝時，資料不全所導致的結果，但如果經由前一章節的方式進行切割，同樣的對炮管部分進行網格化並找出點數多的區塊，如圖 3.39 的綠色點，再對這些點雲資訊進行主軸分析，如圖 3.39：

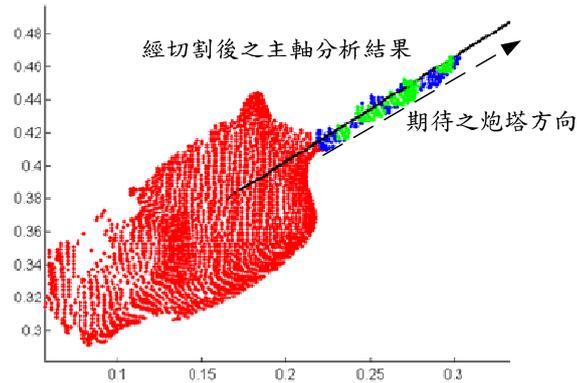


圖 3.39 切割後之主軸分析。

圖 3.39 中，可以觀察切割、網格化後再加上主成份分析法的結果，較圖 3.38 未切割的結果來的更好，但是結果與期待的方向仍有些微的差異，這是因為炮管本身小，相對的雜訊影響更為嚴重，所以會導致主軸偏差嚴重，本論文為了能找到更好的角度結果，利用以下方法來進行修正，首先將炮管投影至 xy -平面，再對點雲利用最小平方法來進行角度估測，結果如圖 3.40。從圖 3.40 中，可以發現利用最小平方法估測出的角度，相較於主成份分析法所估測出的較度來的更為準確，而估測出了主軸向量後，仍需考慮方向是否有完全相反，本論問評估的方法是計算出炮塔重心至炮管重心的向量 \vec{v} ，只要估測出的向量與 \vec{v} 的夾角小於 90° 則是正確方向，反之則代表方向相反，只要乘上 -1 就會是正確的炮塔方向最後再把炮管方向向量代入式(3.1)就可以得到旋轉角度。

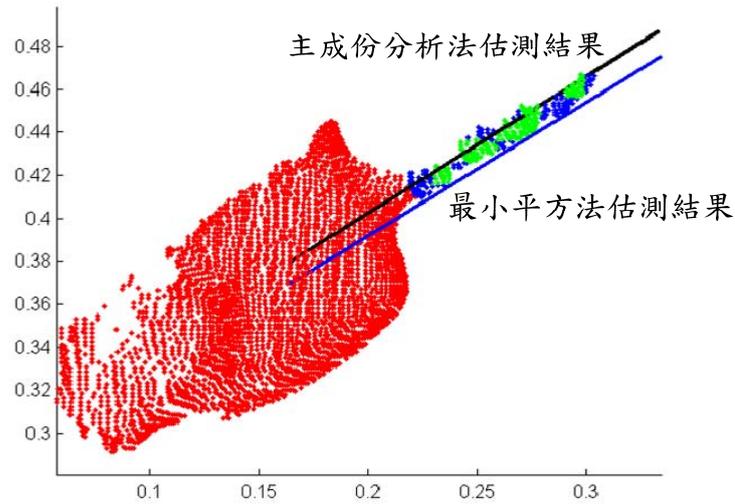


圖 3.40 主成份分析法及最小平方法估測比較。

估測出炮塔角度後，最後的目標是對炮管抬升角進行估測。炮管估測的部分，仍然利用了前面提到的方法進行網格化並找出點數多的區塊，對留下的部分進行處理，根據前面實驗得知，使用主成份分析法分析的結果不佳，所以在炮管部份，直接使用最小平方法來進行角度估測，進行之前，先用先前求出的炮塔旋轉角，代入三維旋轉公式，式 (3.2)：

$$R_{n \times 3} = P_{n \times 3} \times \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

其中 R 為旋轉後的點雲， P 為旋轉前的點雲， n 為點雲數量。將炮塔旋轉至 x 軸，如圖 3.40：

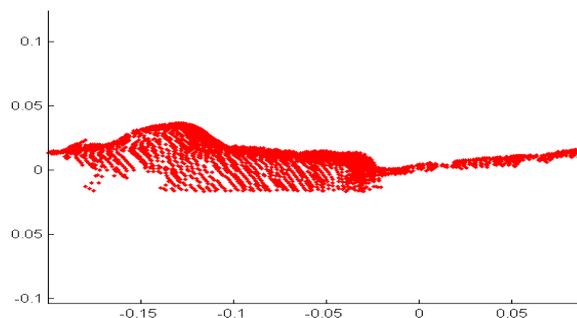


圖 3.41 旋轉後之炮塔。

考慮圖 3.40 中的炮管，利用最小平方法的估測結果如圖 3.42：

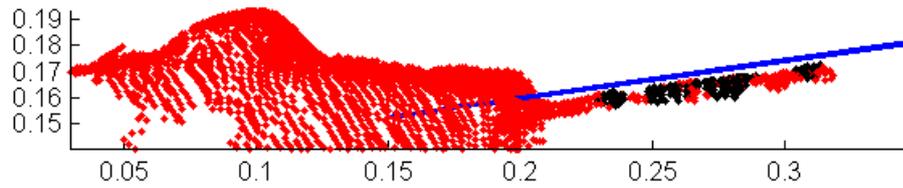
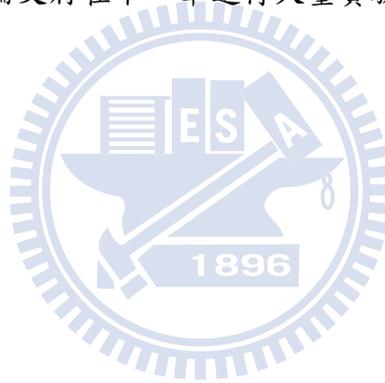


圖 3.42 利用最小平方法計算之主軸。

將主軸方向代入式(3.1)，就可以得到對應的角度，即完成了抬升角的估測。以上就是本論文估測坦克車各部位的方法，利用抽取邊緣，減少主成份分析法受局部點雲的影響，來找出車身角度；也利用網格化的方法將炮管中較重要的部分利用最小平方法估測來得到炮管及炮塔的角度，本論文將在下一章進行大量實驗來驗證本論文演算法的正確性。



第四章 實驗結果及分析

為驗證本論文所提出的演算法，是否能夠順利達到需求，本章將進行大量實驗來測試本系統。本章將分為四個小節，4.1 節的部分將先說明實驗平台，並介紹本論文的量測工具以及實驗環境；4.2 節則是對本論文演算法進行實驗，並提供相關數據；4.3 節的部分則會比較本論文與其他論文的姿態估測結果；4.4 節則對本論文的結果進行討論，並分析與其他論文比較的優劣之處。

4.1 實驗場景與設備

本節將介紹本論文的實驗平台，本論文的實驗硬體規格是使用 Intel CPU i5 M450 2.4GHz，記憶體為 8GB，64 位元的 Windows 7 作業系統的筆記型電腦，以 Visual Studio 2010 的環境中完成整個演算法，並搭配 Matlab R2010a 來畫數據圖，另外，也有使用到 PCL(<http://pointclouds.org/>)的程式資料庫。在角度量測的部分，使用了氣泡式角度尺以及量角器，如圖 4.1 以及圖 4.2：



圖 4.1 電子氣泡式角度尺。



圖 4.2 量角器。

圖 4.1 是用來測量抬升角度的，測量時，平放於地面利用白色按鈕歸零，再放到坦克車炮管上即可量測到抬升角度，精準度可達 0.1° ；圖 4.2 是用來量測炮塔旋轉角以及車身方向的，其精準度為 1° 。坦克車模型是使用德國虎式三代 1:16 之模型，如圖 3.1，其炮塔旋轉角度約為 330° ，炮管抬升約為 30° 。而取向工具是使用微軟開發的體感遊樂器，如圖 3.2，其距離偵測精準度為 1 公分。而為了使世界座標更容易定義，本論文使用鐵架來固定攝影機的架設位置，如圖 3.3，尺寸為 $1.8*1.2*1.2$ ，單位為公尺。利用鐵架定好世界座標的三軸，同時可以對各位置設定好轉換矩陣，這樣才能對量測出來的數值與計算出來的數值進行比較。

4.2 實驗結果

本章節實驗結果的部分，將分別對坦克車車身角度估測、炮塔旋轉角估測以及炮管抬升角進行實驗及分析。本論文實驗預設情況是以攝影機不移動，車身不移動，僅炮塔以及炮管會轉動的前提之下去進行的，但由於坦克車移動時要分別量測各部位的角度，並不太容易，所以本論文在實驗結果的部份將分適用範圍、靜態實驗以及動態實驗三份進行實驗，適用範圍實驗部分，主要是要探討俯仰角與本論文演算法的關係；靜態實驗的部分則是實驗本論文在角度估測的準確度；而動態實驗的部分將會以連續曲線圖來探討本論文對一連續序列場景的估測結果。

首先再使用範圍的部分，這部分主要是針對車身估測所進行的實驗，由於本論文是

利用邊緣垂直部分的點數較多的特點來進行姿態估測的，所以必須考慮到的是，怎麼樣的情況有可能會造成抽取失敗，而水平角的不同以及距離得遠近，只會造成整體物件點數下降，並不會造成邊緣抽取的失敗，若邊緣抽取失敗，最主要的原因是與俯仰角有關，例如：在俯仰角 90° 時，會因為拍攝角度的關係，造成只拍攝到物件的上表面，而邊緣部分因為點數比例與其他位置一樣多，所以會完全無法抽取邊緣。為了驗證本論文在各角度的準確度，本論文利用電腦模擬的方式，以隨機水平角，以及 $0^\circ \sim 90^\circ$ 的俯仰角來測試各俯仰角本論文演算法的準確度，本論文以俯仰角 10° 為間隔，每次拍攝 50 張水平角的模擬圖來測試其準確度，結果如圖 4.3，

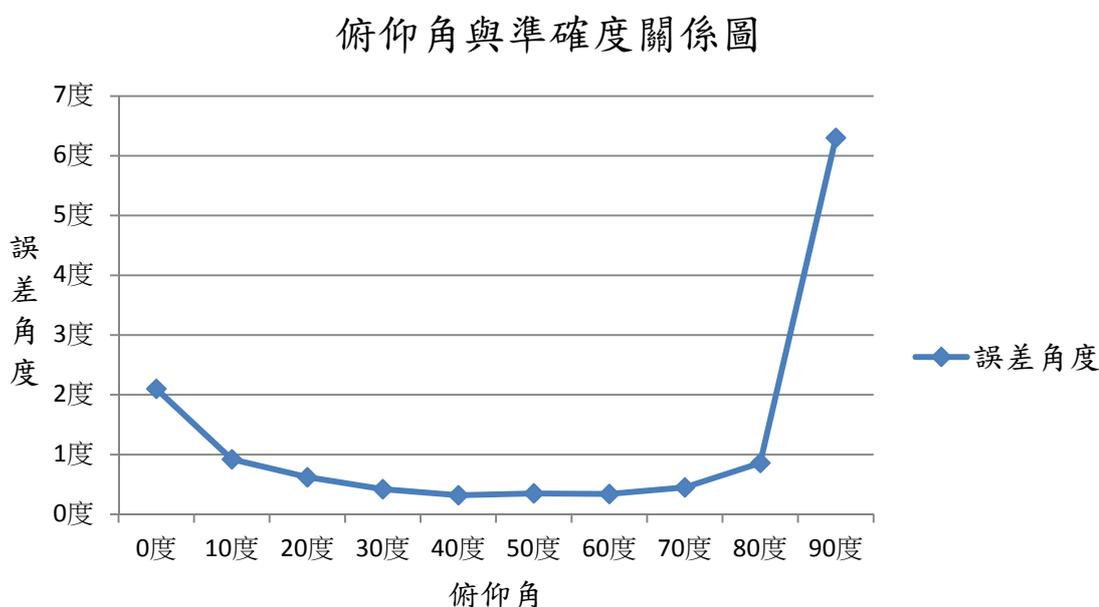


圖 4.3 俯仰角與準確度關係圖

根據圖 4.3 可以發現在絕大部分俯仰角位置中，因為邊緣較為明顯，所以整體估測的誤差都在 $\pm 1^\circ$ 內，除了俯仰角 0° 以及 90° 時，會有較大的誤差，首先討論俯仰角 0° 時，因未完全只拍攝到邊緣，而沒有其他位置的點雲料，所以邊緣的點雲以經不再是相對多的點位置，而是整體資訊投影後，每個點幾乎都一樣多，所以無法完全正確的抽取出邊緣，但也由於只有邊緣的點資訊，所以即使抽取不到完全正確邊緣，所估測出來的角度仍不

會誤差太大；俯仰角 90° 時則會出現比較嚴重的誤差，主要就是因為俯仰角 90° 時，完全拍攝不到邊緣，所以在執行網格抽取邊緣時，會完全無法抽取到邊緣，所以造成抽取出來的結果非常不穩定，幾乎是沒有規則的隨機亂跳。經過以上實驗，本論文所提出的演算法，僅適用在俯仰角度 $10^\circ \sim 80^\circ$ ，其餘的角度，則無法達到較佳的結果。

在靜態實驗的部分，本論文在鐵架上的各角度，各拍了許多張靜態的坦克車影像，這樣的實驗方式，除了要考驗本論文提出的演算法在精確度上是否達到要求外，也可以確定本論文的演算法在不同的距離以及角度是否都能得到相同的結果。在靜態實驗的部分，總共拍攝了 208 張三維影像來進行實驗，首先討論車身角度估測的部分，利用本論文的演算法，得到的結果如圖 4.4：

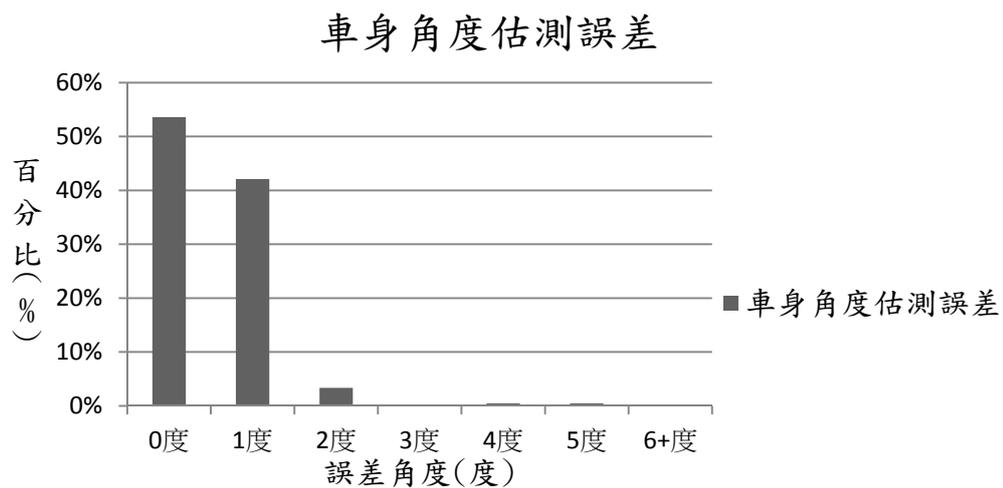


圖 4.4 車身角度誤差圖。

圖 4.4 中角度誤差的計算方式是對估測角度與實際角度的差取四捨五入後的結果，可以發現由於經過邊緣抽取，再加上車身因為大小較大，通常點數會比較多，所以估測到的角度準確度相當良好，誤差 1° 內的有 95%。而在炮管及炮塔的部分，本論文的切割演算法配合最小平方方法的姿態估測方法，經過實驗後，結果如圖 4.5：

炮塔旋轉角誤差

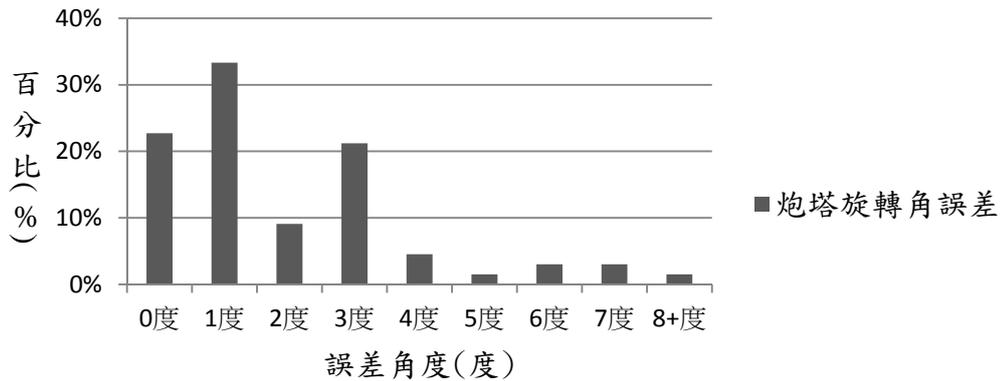


圖 4.5 炮塔旋轉角度誤差圖。

觀察圖 4.5 可以發現，炮管旋轉估測的準確度 1° 以內的大約為 55%，結果並沒有像車身估測來的高，但仍約有 90% 的角度估測誤差都可以在 5° 以內，效果上還是可以接受的。而在炮管的角度估測實驗部分，經過本論文演算法的實驗過後，得到結果如圖 4.6：

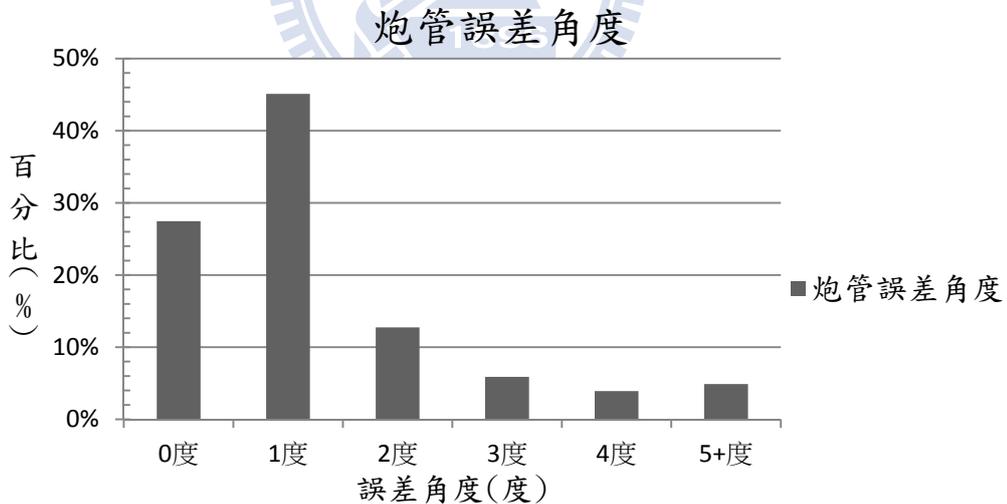


圖 4.6 炮管角度誤差圖。

觀察圖 4.6 可以發現結果上，炮管角度誤差 1° 以內約占 70%，以上三張直方圖是本論文實驗精準度的結果。

以上是本論文利用體感遊樂器所拍攝到的實際資料結果，但是考慮到實際拍攝時，

並無法拍攝到足夠的多的角度來驗證本論文的結論，所以本論文也另外利用電腦模擬的方式來以隨機水平角，俯仰角 $10^{\circ} \sim 80^{\circ}$ 之間的多筆資料進行姿態估測，結果如圖 4.7，

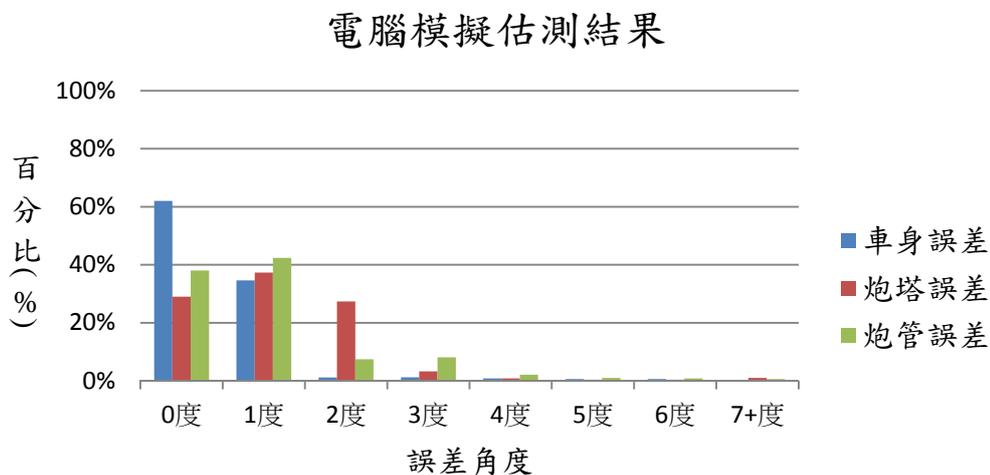


圖 4.7 電腦模擬估測結果。

從圖 4.7 中可以發現，某些角度與前面實際做出的結果來的不完全一樣，但大多數結果都來的比較好，主要的原因是在電腦模擬的拍攝圖並沒有雜訊干擾，而且其精準度非常的好，所以估測結果勢必會來的比實際拍攝的好，而本論文也特別將模擬位置相對於本論文所拍攝的八個位置的點雲資訊額外取出來進行結果比對，發現其估測結果與本論文實際拍攝的結果，相差並不大，都在 $\pm 2\%$ 以內，這也證明了本論文的演算法，不論是在模擬或者實際拍攝，結果都是非常不錯的。

在介紹完靜態實驗後，由於本論文主要是希望能提供一個預警系統，所以連續時間的角度估測是必要的，所以本論文做了以下的實驗：首先將坦克車放置在架中，並架設好攝影機，利用遙控的方式，讓炮塔從左旋轉到右，或從右旋轉到左，而炮管不動，這樣的拍攝方式。其中，為了讓搜尋興趣目標來的更快速，本論文在第一次找到興趣目標時，會計算其重心，讓下一張圖，距離重心太遠的點先刪除掉，免去一些過多的計算，這樣能更有效的加快整個系統的執行。而以上的實驗環境，經過姿態估測得到結果如圖 4.8 以及圖 4.9：

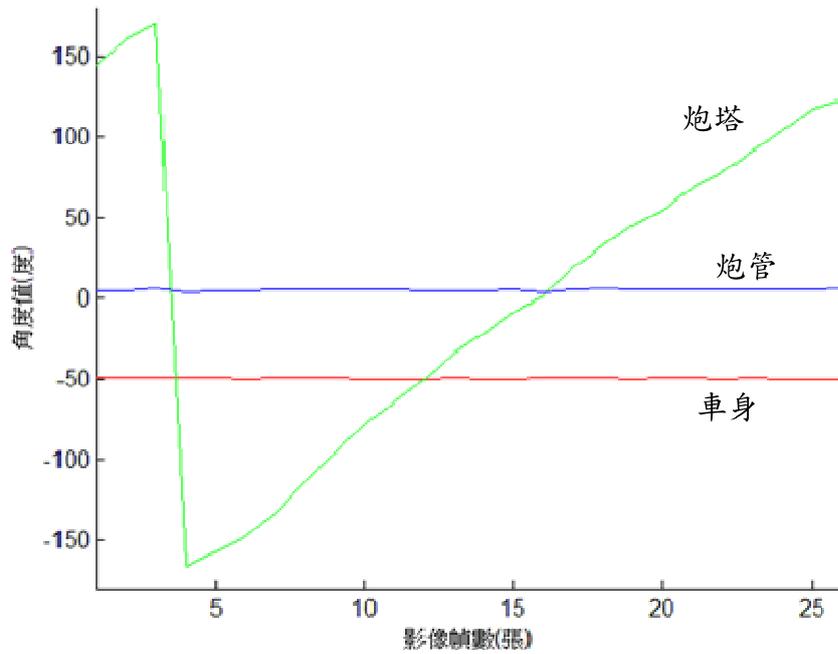


圖 4.8 連續場景估測結果一。

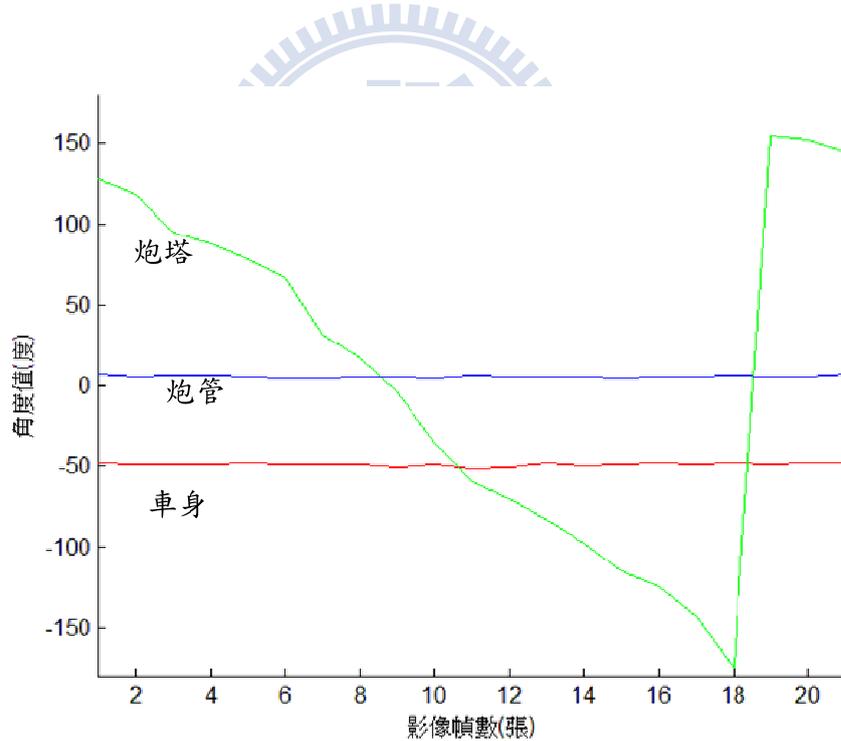


圖 4.9 連續場景估測結果二。

圖 4.8 以及圖 4.9 是攝影機從不同位置，對同一位置的坦克車進行姿態估測的結果，不同的是，前者是順時針旋轉炮塔，後者是逆時針旋轉炮塔，觀察圖 4.8 以及圖 4.9，可以發現雖然攝影機在不同位置，但是因為有做世界座標的轉換，所以估測道的角度皆為

-49°左右，而在炮管的抬升估測部分，兩張圖的結果也都一致，而炮塔旋轉的部分，兩張圖中都可以很明顯的看出整個坦克車在場景中移動的軌跡，在估測時間的部分，在前處理搜尋興趣目標的部分，平均每張約耗時 0.22 秒，而姿態估測的部分，平均每張耗時 0.1 秒，平均每秒大約可執行 3 至 4 張影像，由於坦克車這類機械，在旋轉炮塔時，並不會移動非常快速，所以這樣的速度已經足夠達到即時姿態估測。看完本論文在動態以及靜態實驗後，接下來的部分將對結果進行分析，並且與他人論文進行比較，來表現本論文所提出的演算法在姿態估測上的能力。

4.3 結果分析與比較

4.2 節中，對本論文的演算法進行了角度估測的實驗，但是單看這些數據是很難分辨優劣的，所以本論文在這邊與其他論文進行比較，來證明本論文的演算法是強健的。首先比較本論文方法、利用主成份分析法以及文獻[6]，利用長方體包覆的方式來進行姿態估測的方法，其中，文獻[6]的數據來源是實做該方法，並套用到本論文的點雲資料，所得到的結果。另外，也比較文獻[18]，利用機率方式的估測方法，而文獻[18]的數據來源是直接由論文上取得。實驗結果如圖 4.10：

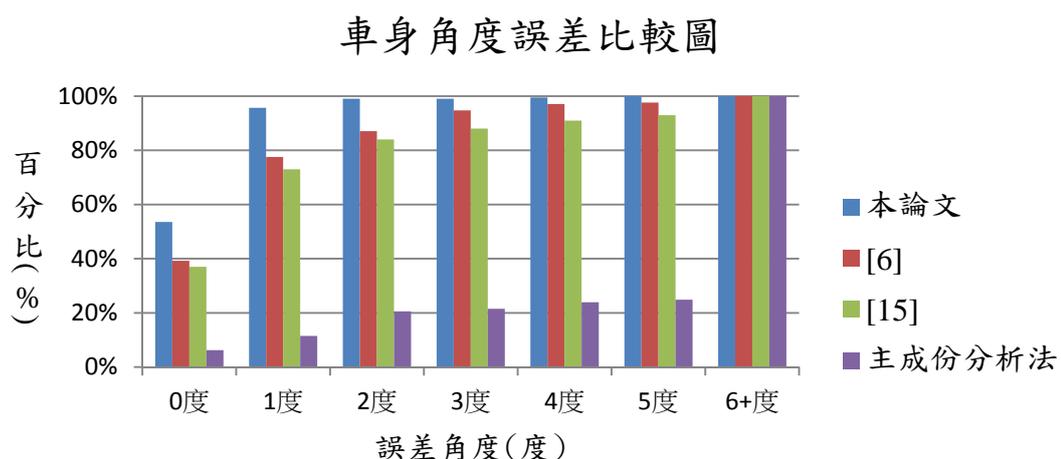


圖 4.10 車身角度誤差比較圖。

從圖 4.10 中可以觀察到，本論文使用的方法，配合邊緣抽取加上主成份分析法，有 95%

以上的誤差小於 1 度，而單純使用主成份分析法來進行角度估測，誤差小於 1 度以內的只有 20%，而文獻[6]所使用的長方體包覆法，效果也不差，誤差小於 1 度的有 85%，但是在速度上就慢了許多，本論文平均單張執行時間約為 0.08 秒，但是長方體包覆因為是利用疊代的方式來進行，所以時間上來的比較慢，每張平均約 1.68 秒。而文獻[18]中利用機率的方式來進行估測，最大的問題在於需要事先得知一些相關資訊才能進行，而本論文的法則，並不需要任何相關資訊，就可以進行姿態估測，而且在準確率上，本論文也是較有優勢的。再來討論炮塔旋轉的部分，從圖 4.5 中，可以發現整體角度誤差較車身角度估測來的大，炮塔旋轉的估測主要是依靠炮管來進行的，一般論文也都是如此，主要原因是估測較為簡單，且如果利用炮塔進行估測，在形狀描述非常不易，加上炮塔的位置是可能有人在上面，種種因素都會造成估測上的困難，而我們利用炮管進行旋轉角度估測，相對會來的簡單許多，但是由於炮管比較小，所以拍攝到的點雲數量也相對來的少很多，所以在估測上自然是無法像車身來的精準，本論文一樣比較了上述幾篇論文，實驗得到結果如圖 4.11：



炮塔角度誤差比較圖

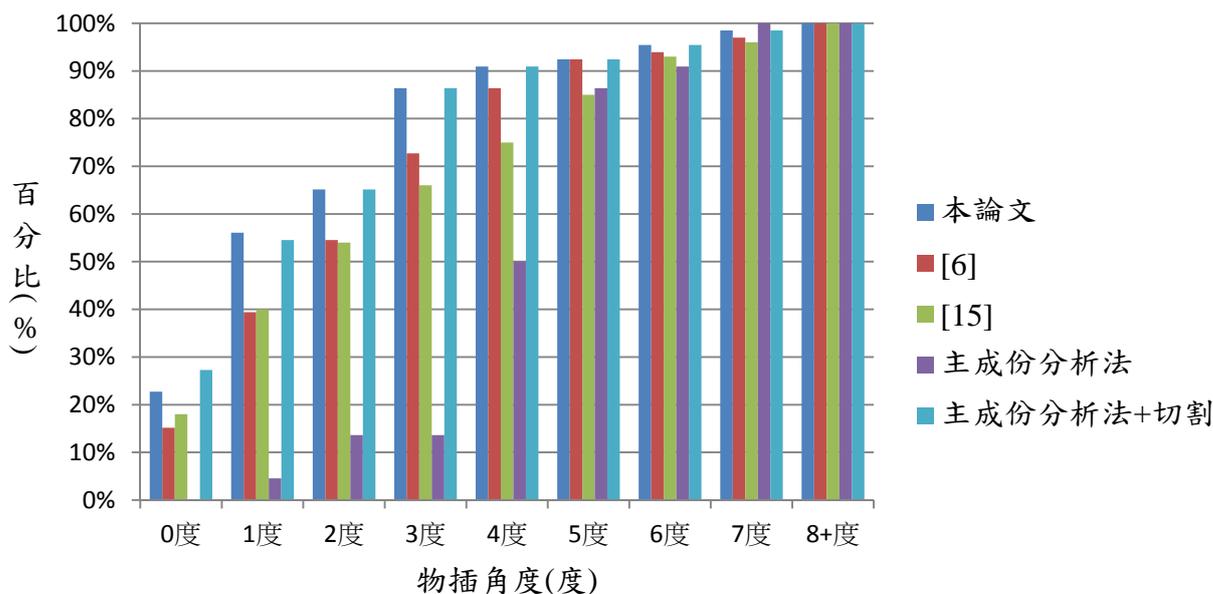


圖 4.11 炮塔角度誤差比較圖。

觀察圖 4.11 可以發現，本論文的演算法，由於有對炮管部分進行加強，所以相對得到的資訊更為完整，所以在角度估測時，可以得到較其他論文更好的結果，前面討論過主成份分析法對局部資訊以及雜訊的抵抗效果並不好，所以幾乎偏差都會到四至五度，而本論文的演算法則可以達到 80% 以上精準度都達 3° 以內；而文獻[6]中，長方體包覆的方式，因為並未再切割炮管部份加強，所以得到的結果並沒有比本論文方法來的好，而且由於必須不斷計算最佳的長方體包覆，所以時間上也來的較慢，平均單張需要 0.46 秒，而本論文的方法除了準確度更高外，單張時間也只需要 0.17 秒，速度來的更外；最後文獻[18]的部分，利用機率方式計算，精準度較本論文差一點，而且需要大量資訊才能夠開始進行角度估測，相較於本論文不需要任何物體資訊，是要來的差一點。最後，炮管估測比較的部分，由於[18]並未提及相關估測方法，所以將不與其做比較，實驗比較結果如圖 4.12：



炮管角度誤差比較圖

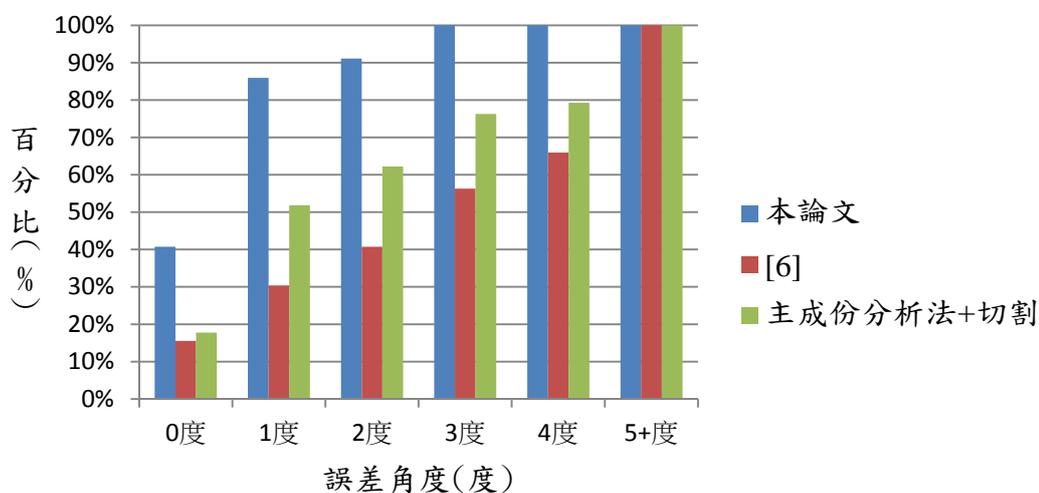


圖 4.12 炮管角度誤差比較圖。

在炮管抬升角的估測部分，本論文利用最小平方法估測的結果較其他結果來的都好，主要原因也是因為本論文在切割時盡可能的炮管切的完整，所以才能得到這樣的結果，相對其他論文，速度及準確度都相對來的好。其中與各論文速度比較的結果如表 4.1，

表 4.1 估測所需時間表

方法 \ 部位	車身	炮塔	炮管
本論文	0.1	0.04	0.03
文獻[6]	0.32	0.08	0.06
主成份分析法	0.04	0.04	X
主成份分析法+切割	0.04	0.06	0.03

單位：秒(s)

最後，將討論遮蔽對本論文所提演算法的影響，由於本論文是將各部為切割後才進行姿態估測，所以在遮蔽影響的部分，本論文將三個部位分別進行討論，首先是車身的部分，本論文將誤差 3° 以內視為正確，其中遮蔽的實驗方法是分別從水平以及垂直方向來進行的，經過實驗後，發現垂直方向的遮蔽對整體估測影響非常的大，因為本論文是對車體邊緣進行抽取，而垂直方向的遮蔽會完全的將邊緣遮蔽掉，導致抽取邊緣失敗，所以是垂直方向的遮蔽，幾乎是無法抵抗，而水平方向的遮蔽實驗結果如圖 4.13：

車身遮蔽估測圖

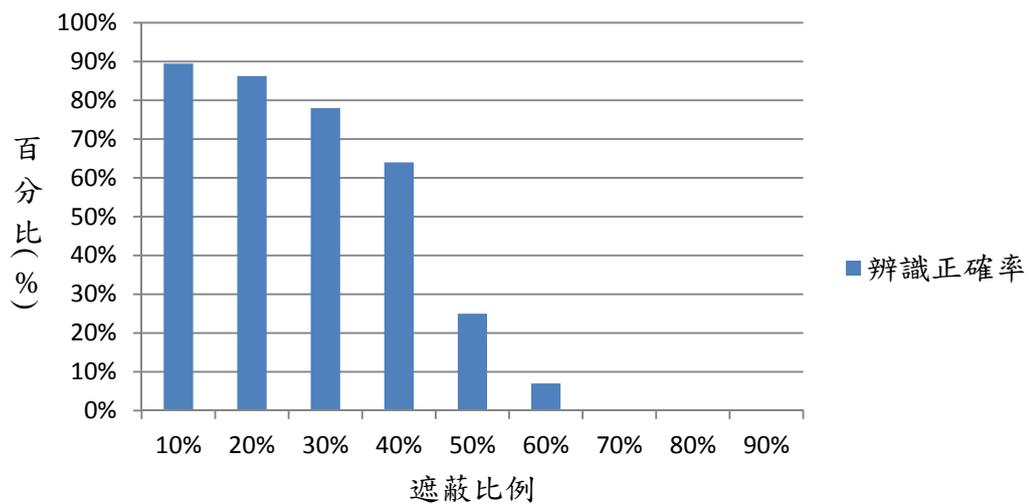


圖 4.13 車身遮蔽比與精準度之比較圖。

圖 4.13 中，可以看到在遮蔽比達到 40% 以上時，整個曲線開始劇烈的向下掉，估測效果會變得非常差，主要是因為當遮蔽達 40% 以上，在邊緣抽取部分會變得容易抽取到非邊緣部分，加上若車頭或車尾有其中一邊被遮蔽，則會無法判斷前後，所以造成了估測角度的困難；而在炮塔的部分，由於本論文的演算法中，並未利用炮塔部分進行姿態估測，僅利用來對炮管切割，所以在這部分，將不討論角度的誤差，而是討論遮蔽對切割的影響。經過實驗發現，炮塔遮蔽的影響主要是與遮蔽方向有關，與遮蔽比關係較小，由於本論文的切割方法是依據寬度的變化，如果遮蔽方向是從水平方向來，那對遮蔽的抵抗率會相當不錯，因為寬度差的資訊都還存在；但若遮蔽是水平方向來的話，那對遮蔽抵抗相對會低非常多，這是因為本身在拍攝時就已經是局部資訊，以經會造成一定的寬度資訊消失，但若是在加上水平方向的遮蔽，那在切割上會非常困難，若切割不成功，那後續炮管估測的結果也會變得相當差；最後討論炮管遮蔽的部份，炮管遮蔽實驗結果如圖 4.14：



炮管遮蔽估測圖

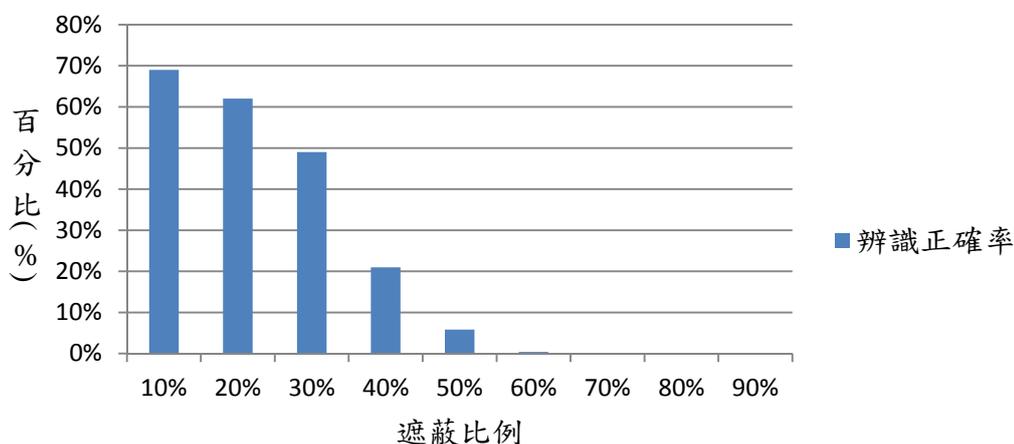


圖 4.14 炮管遮蔽比與精準度之比較圖。

從圖中可以發現，炮管抗遮蔽相對車身來的較差，大約在 30% 的遮蔽後，就開始嚴重的下降，最大原因在於炮管原本點數就少，所以抗遮蔽性自然較車身來的差，而且點數少的情況也會造成主軸的抽取更為困難，所以在這畢情況才會掉的如此快速。

4.4 結果討論

本小節中主要探討本論文實驗與其他論文實驗的不同之處，並整理本論文與其他論文的優劣關係。本論文的實驗是希望能夠達到及時坦克姿態估測，欲設用途是在戰場中的預警系統，所以再開發的目標，主要是希望能夠對絕大多數的坦克車都能夠達成效果，所以演算法的開發上，並不會預設針對某種坦克車進行估測，這點非常不同於以往的論文，以[18]為例，該論文實驗的目的是搜尋特定戰場目標，這篇論文的做法是先在資料庫中建立好 n 個物件的特徵，在系統執行時，先假設希望找到場景中的 A 坦克，系統就會以 A 坦克的相關的高度資訊、機率分佈資訊等等，來對物件進行姿態估測以及搜尋，這樣的作法只能針對某些特定物件來執行。而文獻[6]的作法中，從 4.3 節中的結果中可以看到，這樣的方法表現其實並不差，估測的結果也較[18]來的好，該論文是以長方體包覆的方式來進行，而用密度比的方式來對坦克車進行切割的方法，是不需要資料庫的，但是切割結果從前面的圖 3.20 中可以發現切割結果並不好，因為炮管與車身過於接近，常常會造成只切到炮管尾端的結果，而炮塔部分也常常只切割到上部分，這樣切割不完全的情況是造成準確度不高的其中之一；另外的原因是，長方體的包覆定義是找出最小體積的長方體，但是最小體積在對局部資訊的包覆時常常遇到如圖 4.15 這種情況：

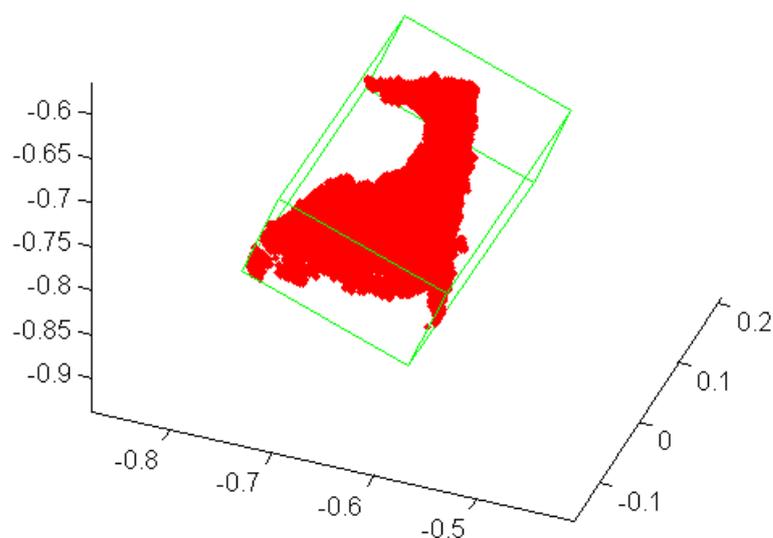


圖 4.15 長方體包覆誤差圖。

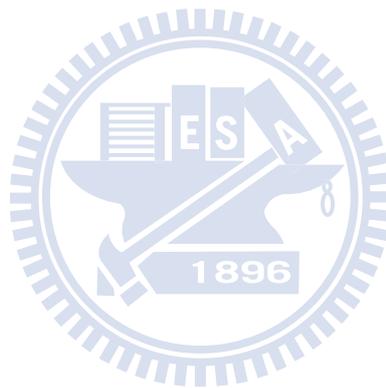
圖 4.15 中，這種幾乎只拍到一半的坦克車，會造成長方體包覆與預期不符，導致整個主軸偏差很大，而本論文因為有利用邊緣抽取再進行估測，所以可以先把車身部分切割出來，然後再進行估測，所以準確度來的更好，因為這些原因，所以長方體包覆的方式才會在估測準確度上不及本論文。而主成份分析法的部分是誤差最大的一個方法，主要原因就是因為遮蔽導致主軸偏移，在以往的論文中，常常使用主成份分析法來完成姿態估測，更常常看到有利用主成份分析法配合疊代最近點的做法，不過這樣的方法必須對每個物件的每個物件都建立資料庫，與本論文的主軸不符，所以並未在更深入的探討兩者搭配的效果。本論文將用表 4.2 來整理本論文與其他論文的比較結果：

表 4.2 與各論文結果比較表

方法 \ 特性	是否需要建立資料庫	車身 準確度	炮塔 準確度	炮管 準確度
本論文	利用寬度差找出切割點，無資料庫	0.5283	1.7242	1.422
文獻[6]	利用長方體以及密度比切割，無資料庫	1.066	2.4848	2.467
文獻[18]	需要機率分佈及車輛高度資訊為資料庫	1.28	2.73	X
主成份分析法	無資料庫	4.91	4.409	X
主成份分析 加切割法	無資料庫	4.91	1.8939	2.148

表 4.2 中，準確度的定義是指誤差 1° 內的百分比，從這個表格裡面，可以看到本論文為了達到對所有坦克車都能進行姿態估測的目標，所以是完全不使用資料庫的，而在精準度方面，車身因為抽取邊緣的結果，所以結果要其他方法都來的好；炮塔的部分效果也是裡面最佳的，其中主成份分析法分為切割與否兩個來分別進行討論，有許多論文因為沒有炮管切割法則，所以都是直接的連帶炮塔及炮管一起進行主成份分析法來估測角度，但如果將主成份分析法搭配本論文提出的切割方法來一同使用的話，可以有效的將估測

準確度由 4.5% 提升到 54%；炮管準確度的部分由於文獻[18]並未提及相關演算法，所以無資料，而原始主成份分析法也因為沒有對炮管切割的相關做法，所以再炮塔角度的部分，也是沒有資料的，再炮管估測準確度的部分，由於本論文的切割方法能夠得到較為完整的炮管資訊，所以結果較文獻[6]來的更好，而搭配本切割方法使用主成份分析法則是因為拍攝到的場景，不可能拍到完整的炮管，一定會缺一面，所以導致了估測角度時結果沒有來的這麼好。本論文靠著良好的坦克車切割演算法以及邊緣抽取等演算法，使得姿態估測來個更為精準，透過本章的實驗以及相關論文比較結果，也證明了本論文提出的演算法來的更為可靠。



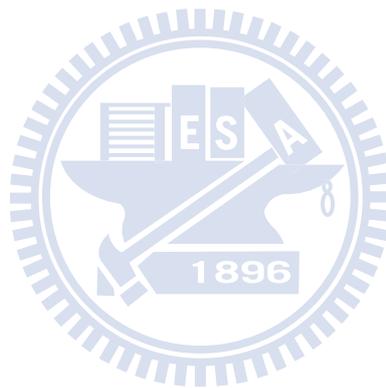
第五章 結論與未來工作

本論文主要提出了坦克車姿態即時估測系統。再興趣目標抽取前處理的部分，主要是利用點雲先對平面進行切割後，再利用改良型的連結區塊分類法，來進行興趣目標的搜尋，最後利用物體的長、寬、高等資訊，來剔除掉過大或過高的物件。抽取到坦克車後，為了節省往後每一張影像的處理速度，本論文用重心追蹤的方式刪除距離重心過遠的點，可以減少掉約 75% 的點雲，能夠大幅加快處理速度。而姿態估測的部分，本論文為能夠達到對所有坦克車通用的目的，所以也提出了利用不同高度下，寬度變化量的特徵來進行上下部分的切割，也利用在不同長度的位置下，寬度變化進行炮管與炮塔的分割，再分別對其進行姿態估測。車身估測中，本論文提出三維影像邊緣抽取方法，利用編原來進行姿態估測，不同於以往的對整台車進行估測，從圖 4.10 中看到，有 95% 以上的坦克車姿態估測誤差都在以內，較其他論文結果都來的好；而在炮管及炮塔姿態估測的部分，從圖 4.11 以及圖 4.12 可以看到未執行切割的姿態估測與切割後再進行姿態估測的結果相差許多，這也更加證明了局部資料對大部份的估測方法都有很大的影響，當然對本論文的演算法也不例外，所以本論文利用網格方法先對炮管進行重點部分抽取，再進行估測，可以發現整個結果可以改善非常多。

從第四章的實驗數據看來，本論文基於寬度的切割方式，可以在完全沒有坦克車尺寸資訊的前提下，成功的將各個部位進行切割，而網格法的重點部位抽取方法也讓整體姿態估測能夠得到更好的準確度。

本論文的貢獻主要有二點：第一，本論文的演算法則，可以在沒有任何坦克車尺寸的前提下完成各部位的切割，結果如圖 3.30；第二，本論文基於網格化的點雲篩選方式，可以有效的抽取出整個物件中對主軸方向較為重要的點資訊，也在圖 4.11 以及圖 4.12 中分別對網格化及位網格化的結果進行比較，發現本論文的網格法可以有效的修正誤差角度。

而本論文在實驗例上，考慮的情況是在攝影機不移動，車身不移動，僅移動炮塔及炮管的情況下來執行的，主要設定場景是將攝影機架設在戰場附近的高地，或者基地高處來進行的實驗，而未來或許可以朝著攝影機移動時仍達成姿態估測的目標前進，這樣的系統如果能夠搭載在飛機上，由於飛機速度快，飛行距離廣，勢必能夠在戰場中獲得更多的資訊，而在攝影機會移動的情況中，如何在移動場景中，即時的將各時間點的影像進行結合，將會是演算法開發重點，或許可以參考過去同步定位及建立地圖 (simultaneous localization and mapping, SLAM) 的技術，對其進行改良，讓演算法的速度加快，以達到即時的效果，過去這類的技術都是利用機器人來配合實驗，但執行速度是否能配合飛機這樣高速飛行的物體仍是個考驗，也是未來可以努力的方向。



參考文獻

- [1] D. Shaoyi and Z. Nanning, "Affine Registration of Point Sets Using ICP and ICA," *IEEE Transactions on Signal Processing Letters*, vol. 15, no. 2, pp. 689-692, 2008.
- [2] D. Zhang and G. Lu, "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures," in *Proceedings of the International Conference on Multimedia and Distance Education*, Fargo, North Dakota, USA, pp. 1-9, 2001.
- [3] W. Y. Kim and Y. S. Kim, "A Region-Based Shape Descriptor Using Zernike Moments," *Signal Processing: Image Communication*, vol. 16, no. 1-2, pp. 95-102., 2000.
- [4] D. S. Zhang and G. Lu, "An Integrated Approach to Shape Based Image Retrieval," in *Proceedings of 5th Asian Conference on Computer Vision*, Melbourne, Australia, pp. 652-657, 2002.
- [5] S. Erik, D. Joel, and E. Gil, "3D LADAR ATR Based on Recognition by Parts," in *Proceedings of the SPIE*, vol. 5094, pp. 29-40, 2003.
- [6] C. Gronwall, F. Gustafsson, and M. Millnert, "Ground Target Recognition Using Rectangle Estimation," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3400-3408, 2006.
- [7] W. Carmen, J. Klaus, H. Marcus, and A. Walter, "A Novel Segmentation Method for Object Tracking and Recognition," in *Proceedings of the SPIE*, vol. 6569, pp. 65690D, 2009.
- [8] A. Walter, "Model-Based Object Recognition in Range Imagery," in *Proceedings of the SPIE*, vol. 7481, pp. 748102, 2009

- [9] A. N. Vasile and R. M. Marino, "Pose-Independent Automatic Target Detection and Recognition Using 3D Laser Radar Imagery," *Lincoln Laboratory Journal*, vol. 15, no. 1, pp. 61-78, 2005.
- [10] N. Burrus, "Kinect Calibration," Retrieved : Apr. 27, 2012, from <http://nicolas.burrus.name/index.php/Research/KinectCalibration>, 2011.
- [11] Robert F. and Sproull, "Refinements to Nearest-Neighbor Searching in k-Dimensional Trees," *Algorithmica*, vol. 6, no. 1-6, pp. 579-589, 1991.
- [12] 林信良, "快速排序法", 搜索日期: 2012年5月18日, "<http://caterpillar.onlyfun.net/Gossip/AlgorithmGossip/QuickSort1.htm>", 2003.
- [13] A. M. and A. A., "On normals and projection operators for surfaces defined by point sets," in *Proceedings of Symposium on Point-Based Graphics 04*, pp. 149-155, 2004.
- [14] A. T. Rabbani and C. G. Vosselman, "Segmentation of Point Clouds Using Smoothness Constraint," in *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Dresden, Germany, vol. 36, part 5, pp. 248-253, 2006.
- [15] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-Time Plane Segmentation Using RGB-D Cameras," in *Proceedings of 15th RoboCup International Symposium*, Istanbul, Turkey, no. D, pp. 230-239, 2011.
- [16] K. Oleksandr, R. Abha, K. Stefan, and S. Ralf, "Connected Component Labeling on a 2D Grid Using CUDA," *Journal of Parallel and Distributed Computing*, vol. 71, no. 4, pp. 615-620, 2011.
- [17] G. Barequet, "Efficiently Approximating the Minimum-Volume Bounding Box of a Point Set in Three Dimensions," *Journal of Algorithms*, vol. 38, no. 1, pp. 91-109, 2001.
- [18] N. Joerg and A. Walter, "Segmentation, Classification, and Pose Estimation of Military Vehicles in Low Resolution Laser Radar Images," in *Proceedings of the SPIE*, vol. 5791, pp. 218-225, 2005.

- [19] R. B. Rusu, "Downsampling a PointCloud using a VoxelGrid filter," Retrieved Mar. 18, 2012, from http://pointclouds.org/documentation/tutorials/voxel_grid.php#voxelgrid, 2011.

