

國立政治大學資訊科學系
Department of Computer Science
National Chengchi University

碩士論文

Master's Thesis

粒子群最佳化演算法於估測基礎矩陣之應用

Particle Swarm Optimization Algorithms for
Fundamental Matrix Estimation

研究生：劉恭良

指導教授：何瑁鎧

中華民國一百年七月

July 2011

粒子群最佳化演算法於估測基礎矩陣之應用

Particle Swarm Optimization Algorithms for
Fundamental Matrix Estimation

研究生： 劉恭良 Student： Kung-Liang Liu

指導教授： 何瑋鎧 Advisor： Maw-Kae Hor



A Thesis
submitted to Department of Computer Science
National Chengchi University
in partial fulfillment of the Requirements
for the degree of
Master
in
Computer Science

中華民國一百年七月

July 2011

粒子群最佳化演算法於估測基礎矩陣之應用

摘要

基礎矩陣在影像處理是非常重要的參數，舉凡不同影像間對應點之計算、座標系統轉換、乃至重建物體三維模型等問題，都有賴於基礎矩陣之精確與否。本論文中，我們提出一個機制，透過粒子群最佳化的觀念來求取基礎矩陣，我們的方法不但能提高基礎矩陣的精確度，同時能降低計算成本。

我們從多視角影像出發，以 SIFT 取得大量對應點資料後，從中選取 8 點進行粒子群最佳化。取樣時，我們透過分群與隨機挑選以避免選取共平面之點。然後利用最小平方中值表來估算初始評估值，並遵循粒子群最佳化演算法，以最小疊代次數為收斂準則，計算出最佳之基礎矩陣。

實作中我們以不同的物體模型為標的，以粒子群最佳化與最小平方中值法兩者結果比較。實驗結果顯示，疊代次數相同的實驗，粒子群最佳化演算法估測基礎矩陣所需的時間，約為最小平方中值法來估測所需時間的八分之一，同時粒子群最佳化演算法估測出來的基礎矩陣之平均誤差值也優於最小平方中值法所估測出來的結果。

關鍵字：影像處理、基礎矩陣、粒子群最佳化、最小平方中值法。

Particle Swarm Optimization Algorithms for Fundamental Matrix Estimation

Abstract

Fundamental matrix is a very important parameter in image processing. In corresponding point determination, coordinate system conversion, as well as three-dimensional model reconstruction, etc., fundamental matrix always plays an important role. Hence, obtaining an accurate fundamental matrix becomes one of the most important issues in image processing.

In this paper, we present a mechanism that uses the concept of Particle Swarm Optimization (PSO) to find fundamental matrix. Our approach not only can improve the accuracy of the fundamental matrix but also can reduce computation costs.

After using Scale-Invariant Feature Transform (SIFT) to get a large number of corresponding points from the multi-view images, we choose a set of eight corresponding points, based on the image resolutions, grouping principles, together with random sampling, as our initial starting points for PSO. Least Median of Squares (LMedS) is used in estimating the initial fitness value as well as the minimal number of iterations in PSO. The fundamental matrix can then be computed using the PSO algorithm.

We use different objects to illustrate our mechanism and compare the results obtained by using PSO and using LMedS. The experimental results show that, if we use the same number of iterations in the experiments, the fundamental matrix computed by the PSO method have better estimated average error than that computed by the LMedS method. Also, the PSO method takes about one-eighth of the time required for the LMedS method in these computations.

Keywords: Image processing, fundamental matrix, PSO, Least Median of Squares.



致謝

能順利完成碩士論文，要感謝何瑀鎧指導教授，從老師身上學習到如何做研究的態度與方法。老師曾勉勵我，做任何事情都要做好，我會謹記這句話，不論往後遇到任何困難與挑戰，都可以克服找到解決的方法。

感謝華梵大學唐政元教授，對我不辭辛勞的幫忙，給予我磨練的機會，學習到很多影像處理的相關知識和技術，並且提供豐富的資源讓我進行論文研究。

感謝口試委員唐政元教授和吳怡樂教授，他們給我很多寶貴的建議，讓論文內容可以更加完整。

在研究所學習階段中，感謝逸帆學長、振寰學長、元彰學長、坤信學長、淑怡學姐，以及立軒、瑞鴻、明龍、紹暉、姿旻、柏諺。特別要感謝博士班凱軒學長，給予我在課業上以及論文實驗上很多的協助，還有宏敏學長，對我的照顧和鼓勵。

感謝政治大學教務處課務組、教學發展中心、資訊中心、政大書院提供我工讀的機會，讓我獲得課業以外的寶貴經驗。

最後要感謝我的父母與姐姐，一直以來的照顧與支持，陪伴著我的成長過程，讓我有今天的成就，謝謝你們。

恭良 2011 年 11 月

目錄

第一章 緒論	1
1.1 前言.....	1
1.2 研究背景與動機.....	1
1.3 問題描述.....	3
1.4 論文貢獻.....	4
1.5 論文章節架構.....	4
第二章 文獻回顧	5
2.1 粒子群最佳化演算法.....	5
2.2 估測基礎矩陣.....	7
第三章 背景技術	10
3.1 粒子群最佳化演算法流程.....	10
3.2 隨機取樣.....	14
3.3 最小平方法.....	15
3.4 最小平方中值法.....	17
3.5 估測基礎矩陣流程.....	18
第四章 以粒子群最佳化演算法估測基礎矩陣	20
4.1 模擬基礎實驗.....	20
4.1.1 平面中粒子群最佳化.....	20
4.1.2 圓錐近似中粒子群最佳化.....	24
4.2 研究方法與流程.....	28
4.2.1 系統架構.....	28
4.2.2 評估基礎矩陣.....	31

第五章 實驗結果	33
5.1 平面中粒子群最佳化實驗結果.....	33
5.1.1 速度未經過正規化.....	34
5.1.2 速度經過正規化.....	46
5.1.3 討論與分析.....	55
5.2 圓錐近似中粒子群最佳化實驗結果.....	55
5.2.1 限制固定疊代次數.....	55
5.2.2 對 Gbest 收斂	57
5.2.3 限制 fitness 值和對 Gbest 收斂.....	64
5.2.4 討論與分析.....	64
5.3 以粒子群最佳化估測基礎矩陣實驗結果.....	65
5.3.1 第一組實驗(3DMAX)資料.....	65
5.3.2 第二組實驗(LIDAR)資料.....	69
5.3.3 討論與分析.....	73
第六章 結論	74
6.1 結論.....	74
6.2 未來工作.....	75
參考文獻.....	76

圖目錄

圖 1.1 : epipolar geometry 示意圖	3
圖 2.1 : 演算法經過三次演化後的路徑圖.....	6
圖 2.2 : 19 種估測方法時間比較.....	7
圖 2.3 : 圓錐體近似問題示意圖.....	8
圖 3.1 : 粒子群最佳化演算法流程圖.....	11
圖 3.2 : 正確資料(1-e)和錯誤資料(e)示意圖	14
圖 3.3 : 線性近似問題範例圖.....	15
圖 3.4 : 最小平方中值法示意圖.....	17
圖 4.1 : 二維平面問題示意圖.....	21
圖 4.2 : 圓錐近似中粒子群最佳化演算法流程圖.....	24
圖 4.3 : 以粒子群最佳化演算法估測基礎矩陣流程圖.....	29
圖 4.4 : 計算基礎矩陣誤差示意圖.....	32
圖 5.1 : 速度未正規化第一次實驗.....	38
圖 5.2 : 速度未正規化第二次實驗.....	42
圖 5.3 : 速度未正規化第三次實驗.....	46
圖 5.4 : 速度正規化第一次實驗.....	49
圖 5.5 : 速度正規化第二次實驗.....	51
圖 5.6 : 速度正規化第三次實驗.....	54
圖 5.7 : 次數為停止條件.....	57
圖 5.8 : convergence 值 <12.25 為停止條件	59
圖 5.9 : convergence 值 <9.8 為停止條件	60
圖 5.10 : convergence 值 <7.35 為停止條件	62
圖 5.11 : convergence 值 <6.12 為停止條件	63

圖 5.12 : convergence 值 <4.9 為停止條件	63
圖 5.12 : $G_{best_{fitness}} < 0.1$ 和 convergence 值 <12.25 為停止條件.....	64
圖 5.13 : 3DMAX 實驗相片	66
圖 5.14 : 3DMAX 拍攝場景	66
圖 5.15 : 3DMAX 實驗直方圖比較	68
圖 5.16 : 3DMAX 實驗折線圖比較	68
圖 5.17 : LIDAR 第一組實驗相片	69
圖 5.18 : LIDAR 第一組實驗直方圖比較	70
圖 5.19 : LIDAR 第一組實驗折線圖比較	71
圖 5.20 : LIDAR 第二組實驗相片	71
圖 5.21 : LIDAR 第二組實驗直方圖比較	72
圖 5.22 : LIDAR 第二組實驗折線圖比較	73



表目錄

表 1：速度向量正規化演算法.....	23
表 2：調整 X 位置向量演算法.....	27
表 3：速度未經過正規化第一次實驗.....	34
表 4：速度未經過正規化第二次實驗.....	39
表 5：速度未經過正規化第三次實驗.....	42
表 6：速度正規化第一次實驗.....	47
表 7：速度正規化第二次實驗.....	49
表 8：速度正規化第三次實驗.....	52
表 9：3DMAX 場景估測基礎矩陣結果.....	67
表 10：3DMAX 場景估測基礎矩陣標準差分佈.....	67
表 11：LIDAR 場景估測基礎矩陣第一組結果.....	70
表 12：LIDAR 場景估測基礎矩陣第一組結果標準差分佈.....	70
表 13：LIDAR 場景估測基礎矩陣第二組結果.....	72
表 14：LIDAR 場景估測基礎矩陣第二組結果標準差分佈.....	72

第一章

緒論

1.1 前言

在隨機取樣(random sampling)的問題中，如何在大量的搜尋空間中，以較少取樣次數尋找最佳結果，是過去研究人員追尋的目標。在人工智慧(artificial intelligence)[10]領域中，制訂出仿效大自然生態中的演化機制(evolution mechanism)，通稱為演化式演算法(evolution algorithms)，其「隨機性質」和「仿效自然」的做法有別於過去其他演算法，本研究選擇以其中一種演化式演算法：粒子群最佳化演算法(Particle Swarm Optimization Algorithms, PSO)[13]為基礎，經過模擬兩個基礎實驗後，接著以電腦視覺(computer vision)領域中估測基礎矩陣(fundamental matrix)[16]為實驗範例，透過使用粒子群最佳化演算法，達成減少取樣次數和計算的時間成本，維持過去估測基礎矩陣方法的準確率，以便於未來進行後續研究應用。

1.2 研究背景與動機

過去解決隨機取樣問題時，在沒有奢求時間成本之下，窮舉法(brute-force method)是我們可以直接採用的方法，其做法為將每種取樣組合計算後，直接比較結果即會找到最佳解。但是在搜尋空間龐大時，我們無法在有限的時間內完成

所有的取樣和計算時，要找到最佳解是不可能的事情，因此窮舉法並不適合使用在大量搜尋空間之下。

貪婪演算法(greedy algorithms)[17]常用於找尋最佳解，過去應用於解決很多知名問題，他是一種由上而下(top-down)的解題方式，在解決問題時會選定集合中任一個節點(node)出發，不斷的透過計算更新，在節點周圍空間繼續尋找較佳解，然後移至該較佳解，直到所有節點計算結束後，最後一個計算出的較佳解就是問題中搜尋到的最佳解，但是在搜尋空間龐大且需要計算複雜時，使用貪婪演算法會花費大量的計算時間。在需要兼具效能和準確率的目標下，是相當費時的解題方式。

參數估測(parameter estimation)是一種先將搜尋空間中取樣資料先分群的做法，利用計算誤差和設定門檻值的方式，將取樣的資料分為正確資料(inlier)和錯誤資料(outlier)，兩者的區分在於雜訊(noise)的大小，之後過濾掉錯誤資料以便於減少取樣次數，其中較常使用的方法為最小平方中值法(Least Median of Square, LMedS)，雖然可以估測出誤差較小的結果，但是計算的次數仍然太多，也得花費較多的時間成本。

為了能減少取樣次數並且可以維持過去學者研究方法的準確率，我們使用了粒子群最佳化演算法，並且以估測基礎矩陣為實驗範例，希望仿效自然界中個體、群體、群集互相影響之特性，保有演化式計算的原理之外，嘗試結合參數估測的工具，希望更能提升解題的效率和維持估測基礎矩陣的準確率。

1.3 問題描述

本研究使用粒子群最佳化演算法，估測高準確率的基礎矩陣。進而可以在未來進行影像處理後續研究。在電腦視覺領域尋找對應點的問題中，大多是利用兩張或是多張相片。以兩張相片為例，過去有尺度不變特徵轉換(Scale Invariant Feature Transform, SIFT)[12]和擴增(propagation)[11]等方法，可以找到具特徵性質的對應點，但是這些尋找對應點的方法，無法確保對應點的精確度，有時誤差相當大。因此有學者提出兩部相機 O_L 和 O_R 拍攝同一個物體 P 時，在兩張照片上成像為 p_L 和 p_R 。稱為極線幾何(epipolar geometry)(圖 1.1)的關係，由於我們無法知道兩部相機在座標系上的絕對位置，因此我們透過估測基礎矩陣，可以計算出兩部相機的相對位置，且任一張照片上的特徵點可以在另外一張相片繪出極線(epipolar line)(如圖 1.1 中 E_L 和 E_R)，我們可以在極線附近尋找對應點可能存在的位置，因此，估測基礎矩陣在電腦視覺領域中一直是熱門的研究重點。

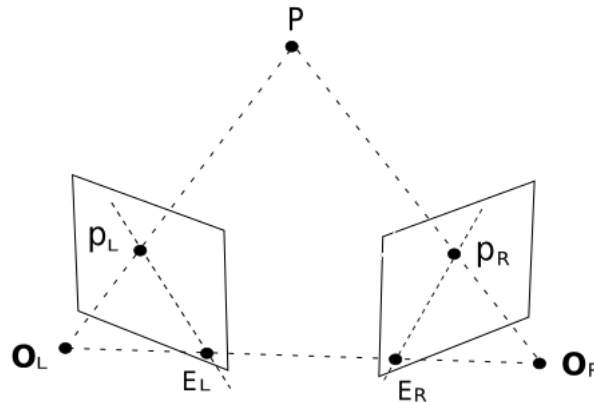


圖 1.1：epipolar geometry 示意圖

由於估測一組基礎矩陣，使用八點演算法，僅需要八組對應點便可以計算出結果，因此若 n 組對應點的情況下所有搜尋空間為 C_8^n ，時間複雜度為 $O(n!)$ ，如何有效的減少取樣次數並且維持過去演算法的準確率便是我們的目標。

1.4 論文貢獻

本論文共有以下三點貢獻：

1. 降低計算時間成本：我們使用粒子群最佳化演算法，和傳統演算法相比，可以降低取樣次數。和過去方法相比，演算法計算複雜度較低，根據實驗結果顯示：與強健式估測方法(最小平方中值法)比較，在接近的取樣次數下，只需要 1/8 左右的計算時間。
2. 維持實驗準確率：跟最小平方中值法比較平均誤差值，我們提出的方法仍可以將誤差再縮小，根據實驗數據，粒子群最佳化演算法的標準差小於平均值的比例較高。
3. 結合其他方法及應用：除了使用粒子群最佳化演算法之外，結合過去幾種參數估測工具，如應用最小平方中值法內建立最小平方中值表格，加入演算法步驟中，協助設計演算法並執行。

1.5 論文章節架構

本論文共分六章。本章為緒論；第二章為文獻回顧，介紹過去我們使用的演算法和過去學者如何估測基礎矩陣；第三章為相關的背景技術介紹；第四章為研究方法，描述我們如何推演套用演算法和應用；第五章為實驗結果，對於不同的實驗條件，我們做比較與分析；第六章為結論和未來工作。

第二章

文獻回顧

本章節分為兩個重點，第一我們介紹粒子群最佳化演算法，並列出過去一些使用粒子群最佳化演算法的相關文獻，第二我們介紹過去估測基礎矩陣使用的幾種方法，並且會選擇一種過去的估測方法和我們的實驗結果做比較與分析。

2.1 粒子群最佳化演算法

粒子群最佳化演算法是由 J.Kennedy 和 R.Eberhart 於 1995 年提出一套演化式計算的演算法，此方法是一種基於個體、群體、群集並且結合人工智慧以解決搜尋最佳解問題，演算法起源於自然界中鳥群覓食的行為，作者將其推演至社會學行為的問題。在演算法中，個體的行為會因為個體本身過去的經驗和認知影響，並且也會受到整個群體過去的經驗和認知影響，是一個群集的活動，模擬簡化的社會模型。演算法利用隨機分佈一定數量的粒子個體(particles)組成群體，在社群空間中不斷演化計算，演化至下一世代之前，會根據粒子個體本身過去經驗和群體過去經驗進行計算，並且持續不斷的在搜尋空間中找尋最佳解，直到達到我們給予演算法的停止條件。

每個粒子個體在搜尋空間中會有自己的位置向量、速度向量，其中向量的維度會隨著欲解的問題不同而調整，在第四章時，我們會討論在進行主要實驗之前，先行推演的兩個實驗，向量維度就不同。除了上述兩個向量外，粒子個體還會有

區域性最佳解(Local best solution，或稱 Personal best，簡稱 Pbest)，全部粒子群體會有全域性最佳解(Global best solution，或稱 Global best，簡稱 Gbest)，上述這些數值會不斷經過數學式子計算，不斷尋找到粒子個體中區域最佳解、群體中全域性最佳解，且根據這兩個過去經驗以求得最佳解，我們將在第三章時介紹整個演算法流程，並說明上述幾個參數是如何推演。

過去國內外有許多學者研究粒子群最佳化演算法，並且有各種不同的應用，尹邦嚴等人[1]提出粒子族群最佳化的視覺化及開發工具，利用程式語言將演算法可以視覺化的展示，讓使用者可以清楚了解粒子群最佳化的演化過程，並且可以顯示在圖片上，如圖 2.1 所示。

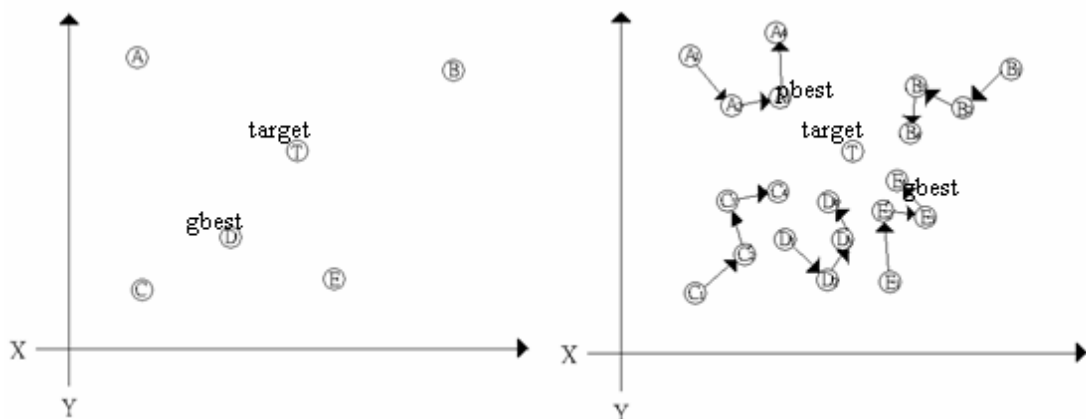


圖 2.1：演算法經過三次演化後的路徑圖，圖像取自[1]

蔡元介等人[5]提出建立一個以 PSO 求解多點最佳路徑的行動地理資訊系統，藉由網路服務(web services)技術，透過呼叫遠端伺服器的演算法來進行複雜的地圖路徑計算，以求得最佳路徑；蔡賢量[6]等人提出基於粒子族群最佳化之不完全資料處理，應用在知識發掘(knowledge discovery)的研究，透過演算法處理不完全資料的問題，並且延伸至後續應用在中文文字處理[2]。

2.2 估測基礎矩陣

Xavier 和 Joaquim[18]於 2003 年曾提出並且整理過去一些估測基礎矩陣的方法，在整理的 19 種方法中，又可以分為三大類，分別是線性方法(linear method)(圖 2.2 中 1~4)、疊代法(iterative methods)(圖 2.2 中 5~11)和強健式方法(robust methods)(圖 2.2 中 12~19)，作者將 19 種方法實作後，用現實拍攝的影像進行實驗，結果顯示，線性方法在對應點準確率高的情況下，實驗結果是誤差小且正確率高。疊代法可以處理一些有高斯雜訊(gaussian noise)的情況，但是對於錯誤資料太多的狀況下處理是相當沒有效率的。強健式方法可以同時處理高斯雜訊和存在錯誤對應點的情況，因此在本研究會選一種強健式估測基礎矩陣的方法當作比較範例，在第五章實驗結果時會有討論與分析。

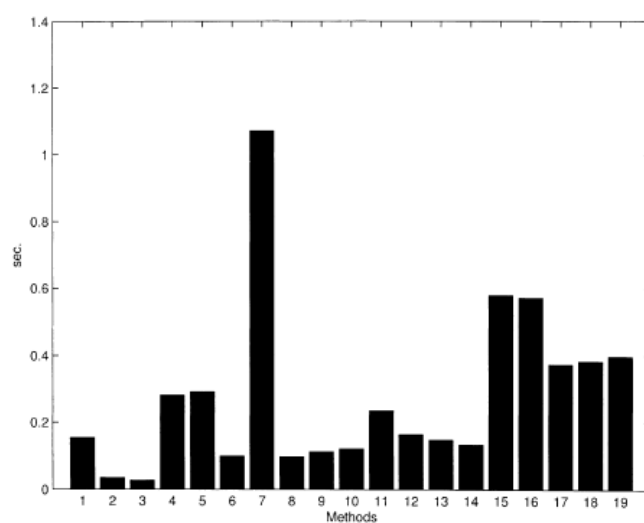


圖 2.2：19 種估測方法時間比較，圖像取自[18]

過去有許多學者持續以強健式方法為基礎進行研究，Zhang[19]於 1997 提出一篇強健式參數估測用於圓錐體近似，由於大部份電腦視覺的問題常常會遇到雜訊資料干擾影響估測結果，所以 Zhang 模擬圓錐體近似實驗，並且分佈資料點可

能是正確資料或是錯誤資料交錯之下(如圖 2.3)，因此使用線性解法並不適當，所以 Zhang 使用了參數估測中的最小平方方法(least-squares)和最小平方中值法希望可以過濾掉錯誤資料，我們將在第三章描述其相關技術，並且在第四章以粒子群最佳化演算法加入解題，因為圓錐體可以在二維平面上繪圖出來，所以我們先觀察演算法用在解類似問題下是否適合，進而使用在估測基礎矩陣實驗。

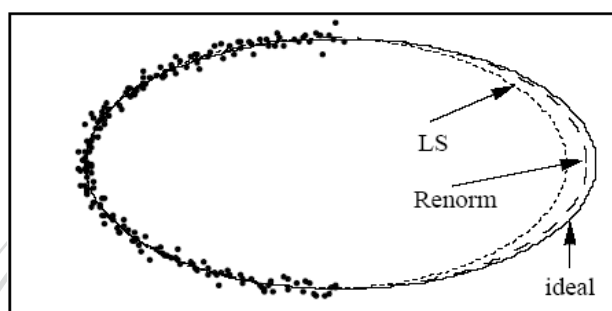


圖 2.3：圓錐體近似問題示意圖

鄒慎財[3]提出強健式估測基礎矩陣之研究，利用強健式估測方法，如最小平方中值法和隨機抽樣(Random Sample Consensus, RANSAC)演算法進行估測，其中要使用最小平方中值法時，要注意錯誤資料需要佔全部可取樣點數的 50% 之內，這部分我們會在第三章介紹此方法，而使用 RANSAC 演算法雖然可以在錯誤資料超過 50% 情況下完成較準確的估測，但只能對特殊情況設定門檻值，並且以票選的方式，當超過門檻值的數量大於制訂的標準後，便可以信任估測的結果，最後提出一套判斷隨機取樣是否正確的判斷式，有 70% 以上的機率，能判斷取樣錯誤，藉此提升估測的效能。

廖怡儂[4]提出應用在電腦視覺強健式估測之研究，使用多種強健式估測方法，並且以線性近似(Line Fitting)和估測基礎矩陣為實驗範例，分析錯誤資料佔全部資料在不同的百分比之下，使用何種強健式方法可以得到最好的解，最後結合不同的強健式方法，再改善實驗結果，提供未來繼續研究的空間。

賴岳宏[7][8]提出利用演化式計算做最佳化之研究，引用田口方法(taguchi method)[14]和演化式演算法中的基因演算法(genetic algorithms)[9]估測基礎矩陣，利用模擬大自然的基因交配，留下最適合適應環境的方式，在隨機取樣的問題中，可以透過複製、交配和突變來求取最佳解，估測出好的基礎矩陣，在實驗中，雖然可以找到較佳的基礎矩陣，但是演化速度相當費時，較不符和計算成本。

本研究除了以粒子群最佳化演算法估測基礎矩陣外，也結合強健式方法中的最小平方中值法內幾個步驟，另外獨立實作最小平方中值法作為比較，我們將在第五章列出實驗結果，並且討論與分析。



第三章

背景技術

本章節我們將描述研究中應用到的各種相關技術，首先我們介紹粒子群最佳化演算法的流程，以及我們如何進而套用在估測基礎矩陣。接著說明隨機取樣，在含有正確資料和錯誤資料情況下，隨機取樣成功的機率是可以透過公式計算出取樣次數，再來是兩種強健式方法，最小平方法和最小平方中值法，最後介紹估測基礎矩陣，以極線幾何關係述說兩張影像的關係，並列出相關技術。

3.1 粒子群最佳化演算法流程

粒子群最佳化演算法是基於粒子個體和群體不斷演化計算的方法，目的是將群體中每個粒子個體透過不斷移動搜尋，直到達到停止條件為止。我們一開始會在搜尋空間中放置數個隨機取樣的粒子個體，接著隨著要解的問題不同，我們會制訂評估值計算方式，有了評估值之後，便可以求出 P_{best} 和 G_{best} 兩個值，接著運用這兩個值可以計算出粒子個體新的速度向量，並且和原本位置向量相加後，便可以得到新的位置，接著回到前面計算評估值，透過不斷的疊代來找到較佳解。傳統的粒子群最佳化演算法流程可以由圖 3.1 顯示，我們將對每個步驟做詳細的介紹。並且說明如何應用在估測基礎矩陣。

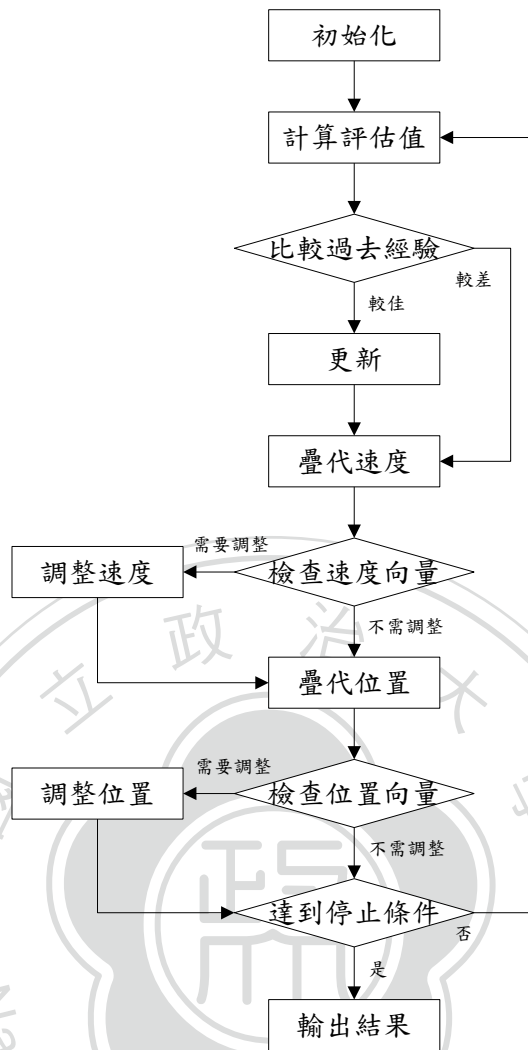


圖 3.1：粒子群最佳化演算法流程圖

初始化

隨機取樣 i 組粒子個體，其中每個粒子個體包含位置向量 X_{id} 和速度向量 V_{id} ， d 表示向量維度，在不同的問題下，維度大小並不一定相同。由於我們的搜尋空間並非無限大，因此 X_{id} 會受限於最小和最大範圍， V_{id} 是粒子個體的移動速度，通常我們會限定 V_{id} 的最小與最大值(負號表示方向不同)，避免粒子個體在一次疊代步驟後改變太大的距離，使得我們錯過沒搜尋到的目標。

計算評估值

計算評估值(fitness)，演算法中最重要的一步驟，是將粒子個體和目標之間透過公式計算出一個數值，若能制訂良好，就能幫助演算法提升解題效率，相反的，若制訂不佳，很有可能會產生退化問題甚至會影響最後實驗數據，這邊退化問題是指計算成本不減反增，失去原先使用此演算法的意義。我們會根據不同的問題找尋評估好壞的依據，進而制訂計算評估值的公式。

比較過去經驗

粒子個體過去最佳的經驗，我們稱為 P_{best} ，表示過去曾經到過最好位置；群體過去最佳的經驗，我們稱為 G_{best} ，表示群體中某一粒子個體曾經到達過最好的位置，這邊表示的最好位置，便是由前一步驟計算出來的 fitness 值當作評估的好壞。之後計算出的結果較前面好時，我們會取而代之前者，倘若計算結果沒有比較好，便直接進行疊代速度。

更新

第一次計算時，每組粒子個體都會更新本身的 P_{best} ，再來會從全部的粒子個體中挑選出 G_{best} ，當再次更新 P_{best} 時，我們會紀錄較好的評估值和位置向量，並在群體中檢查 G_{best} 是否也有較好的評估值，當有較好時我們也會紀錄 G_{Best} 的評估值和位置向量。

疊代速度、檢查速度向量、疊代位置、檢查位置向量

由初始化產生的 X_{id} 位置向量和 V_{id} 速度向量，會經由數學式子(式 3.1、3.2)的計算，得到粒子群最佳化演算法第二回合或是往後幾個回合開始時的 X_{id} 向量和 V_{id} 向量。

$$V_{id} \leftarrow w \times V_{id} + C_1 \times \text{rand}(0,1) \times (PbestX_{id} - X_{id}) + C_2 \times \text{rand}(0,1) \times (GbestX_d - X_{id}) \quad (3.1)$$

$$X_{id} \leftarrow X_{id} + V_{id} \times t \quad (3.2)$$

權重值 w 是參考上一回合速度向量的一個加權值， $\text{rand}(0,1)$ 是隨機產生 0 到 1 之間的亂數，目的是要讓新的速度向量有機會產生更多種組合， $PbestX_{id}$ 為過去第 i 個粒子個體的經驗中最佳位置向量， $GbestX_d$ 為過去群體經驗最佳位置向量， C_1 和 C_2 是兩個加速度的權重值， C_1 表示參考 $Pbest$ 的權重， C_2 則是參考 $Gbest$ 的權重，一般會設定為常數，但也有學者研究會隨著疊代次數改變 C_1 和 C_2 [15]。

當實驗中有限制速度向量的最大值和最小值時，我們要判斷是否有超過的情況，同樣地，若有限制位置範圍時，在位置疊代的步驟也要檢查。

達到停止條件

若達到，將輸出一組最佳評估值和每個粒子個體最後結果，若尚未達到，則往下一步驟繼續進行實驗，其中停止條件的設定也會根據解的問題不同而有不同的制訂方式，如何制訂停止條件在粒子群最佳化也是相當重要。

由於使用八點演算法下，估測基礎矩陣僅需八組對應點，因此我們會將維度設定為八維空間，而我們進行的實驗，便是初始分佈數個粒子個體後在八維空間中進行搜尋，我們將在第四章詳細討論。

3.2 隨機取樣

隨機取樣在機率學中是可以數學表示成功機率(S)的，如圖 3.2，假設在樣本空間中存在正確資料(1-e)%和錯誤資料(e)%時，若我們計算一組答案需要取樣 p 個正確資料時，機率便為(1-e)^p，結合前面幾個參數，我們可以將隨機取樣表示為式 3.3。

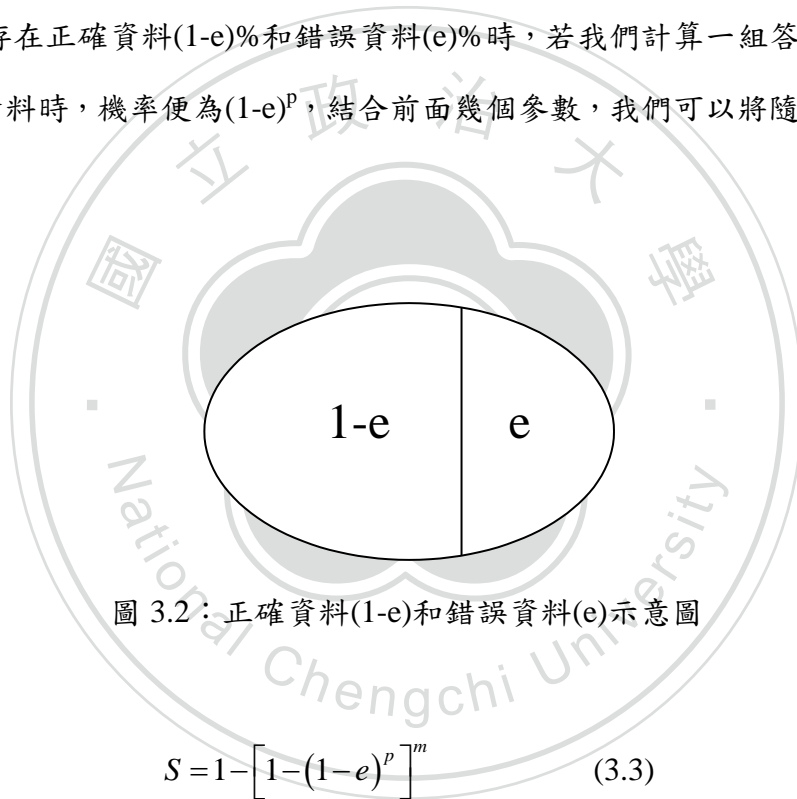


圖 3.2：正確資料(1-e)和錯誤資料(e)示意圖

$$S = 1 - [1 - (1 - e)^p]^m \quad (3.3)$$

其中 m 表示為需要取樣的次數，即當我們取樣 m 次樣本時，可以在錯誤資料佔有 e% 的情況下，達到 S 的機率會有一次一組取樣內容皆是正確資料。

我們在實作最小平方中值法時，會結合隨機取樣，並且設定機率 S 值，如此會計算出不同的取樣次數，利用這些不同的次數會有不同的實驗結果，我們將在第四章說明詳細流程。

3.3 最小平方法

在有兩點或是多點的二維座標系中，我們要找到一組(a,b)集合解，在線性近似問題中(如圖 3.3)，求出一條最逼近、誤差值最小的直線方程式(如式 3.4)，最小平方法是一個好的解題方法。

$$y = ax + b \quad (3.4)$$

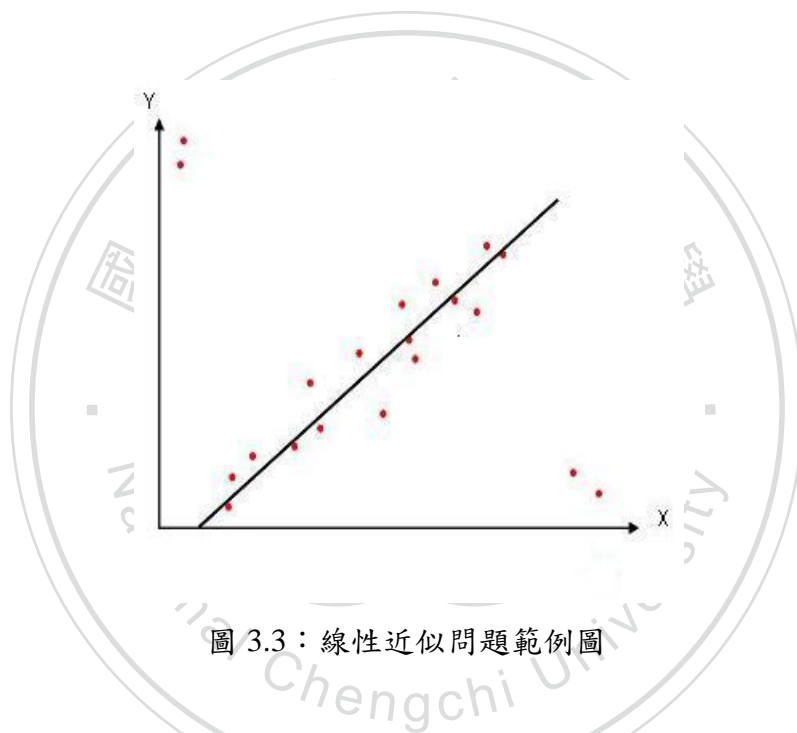


圖 3.3：線性近似問題範例圖

由於每個座標點並不在我們近似的直線上，因此每個點皆可以代入式 3.4，計算出來一組殘餘誤差值 r (如式 3.5)。為了避免統計誤差時正負值會抵消，所以採用誤差平方和為最小($\min \sum r_i^2$)。

$$r_i = |ax_i + b - y_i| \quad (3.5)$$

當有 n 個點的時候，我們可以寫成 n 組直線方程式(如式 3.6)。

$$\begin{aligned} ax_i + b &= y_i \\ \{x_i, y_i \mid i=1, 2, 3, \dots, n\} \end{aligned} \quad (3.6)$$

接著經過微積分的計算後，我們可以寫出式(3.7、3.8)計算出 a 、 b ，並且將 (a, b) 代回式 3.4 後，即完成解線性近似問題。

$$b = \frac{\sum x_i y_i - \frac{\sum x_i \sum y_i}{N}}{\sum x_i^2 - \frac{(\sum x_i)^2}{N}} \quad (3.7)$$

$$a = \frac{\sum y_i}{N} - b \frac{\sum x_i}{N} \quad (3.8)$$

使用最小平方法的前提是分佈的點雜訊要盡可能的很小，以圖 3.3 的範例看來，在接近 X 軸和 Y 軸的地方皆有錯誤資料，很有可能會因為少許的錯誤資料，雖然計算出來的誤差是最小，但卻無法是幾個正確資料計算誤差最小，也就是說，最小平方法會受到錯誤資料的影響，而且無法排除，因此我們將在下一小節介紹最小平方中值法，希望可以改善存有錯誤資料的問題。

3.4 最小平方中值法

是一種強健式估測方法，由於錯誤資料是無法避免並且常常會佔有一定的比例，因此當我們使用最小平方中值法時，希望可以有效率的過濾掉錯誤資料。在計算的過程中，會產生最小平方中值表格，我們將所有的誤差值經過排序後，捨棄誤差值較大的 50% 資料，保留前 50% 的值，希望讓平均值以上的資料可以繼續被留下來應用，進行後續的實驗，如圖 3.4 所示。

	H ₁	H ₂	H ₃	H ₄	...	H _m
	r_{11}^2	r_{21}^2	r_{31}^2	r_{41}^2		r_{m1}^2
	r_{12}^2	r_{22}^2	r_{32}^2	r_{42}^2		r_{m2}^2
	⋮	⋮	⋮	⋮		
Min	Median					
	⋮	⋮	⋮	⋮		
	r_{1n}^2	r_{2n}^2	r_{3n}^2	r_{4n}^2		r_{mn}^2
	(residual)					

圖 3.4：最小平方中值法示意圖

我們採用最小平方中值法，並且應用在計算一組粒子個體的評估值，當我們估測出一組結果時，可以將原先所有資料代入計算，這時每組資料都會有一組殘餘誤差值(如 3.3 小節中式 3.5)，接著我們會將誤差值由小到大排序，並且取中位數，當作我們的評估值，這表示我們相信前面 50% 的資料，後續實驗會以這樣的作法來取評估值。

3.5 估測基礎矩陣流程

Hartley 的書中曾提過八點演算法(式 3.9)，即表示有八組對應點座標時，即可以計算出一組基礎矩陣，在第一章時我們曾介紹極限幾何關係(圖 1.1)，接著我們將對一些參數做說明。

$$p_R^T F p_L = 0 \quad (3.9)$$

其中 p_L 為第一張相片上的特徵點，座標為 $(u, v, 1)^T$ ， F 表示基礎矩陣， p_R 表示 p_L 對應在第二張相片上的對應點，座標為 $(u', v', 1)^T$ 。對左右兩張相片上的每組對應點 p_L 、 p_R ，上述算式皆會成立，我們可以將式 3.9 改寫成式 3.10，看出基礎矩陣中包含的幾個參數。

$$[u', v', 1] \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = 0 \quad (3.10)$$

若我們有八組對應點，並且給予編號 1~8 時，代入式 3.10 後，可以得到以下式子如 3.11。

$$\begin{bmatrix} u_1 u'_1 & u_1 v'_1 & u_1 & v_1 u'_1 & v_1 v'_1 & v_1 & u'_1 & v'_1 \\ u_2 u'_2 & u_2 v'_2 & u_2 & v_2 u'_2 & v_2 v'_2 & v_2 & u'_2 & v'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_8 u'_8 & u_8 v'_8 & u_8 & v_8 u'_8 & v_8 v'_8 & v_8 & u'_8 & v'_8 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \end{bmatrix} = - \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (3.11)$$

要估測出基礎矩陣，需要求解 3.11 聯立方程式，只要有八組對應點，就可以有一組答案，但我們主要問題在於估測出來的基礎矩陣要是好的。倘若基礎矩陣解出來不好，在後續找密集對應點的過程中會產生錯誤，我們該如何使用粒子群最佳化演算法幫助初始隨機取樣後的結果呢，將在第四章時會說明。



第四章

以粒子群最佳化演算法估測基礎矩陣

本章第一小節介紹兩個基礎實驗，分別是粒子群最佳化演算法應用於二維平面和圓錐近似兩個實驗，透過模擬的實驗，我們可以更清楚演算法的流程和運作，第二小節是估測基礎矩陣的研究方法與流程，會詳細說明每個步驟和設計的想法，實驗結果將在第五章。

4.1 模擬基礎實驗

4.1.1 平面中粒子群最佳化

本實驗將在平面中模擬粒子群最佳化演算法，由於三維以上的空間我們很難想像，因此我們用一個最簡單的例子，如圖 4.1，是一個二維平面找尋最佳解的問題，我們將最佳解的目標假定在平面裡其中一個座標點位置，並且在限制位置大小的範圍隨機分佈粒子個體，最終我們希望初始分散的所有粒子個體，可以逼近至我們假定的最佳解目標，以及達到停止條件制訂的門檻值，實驗流程可以參考圖 3.1。

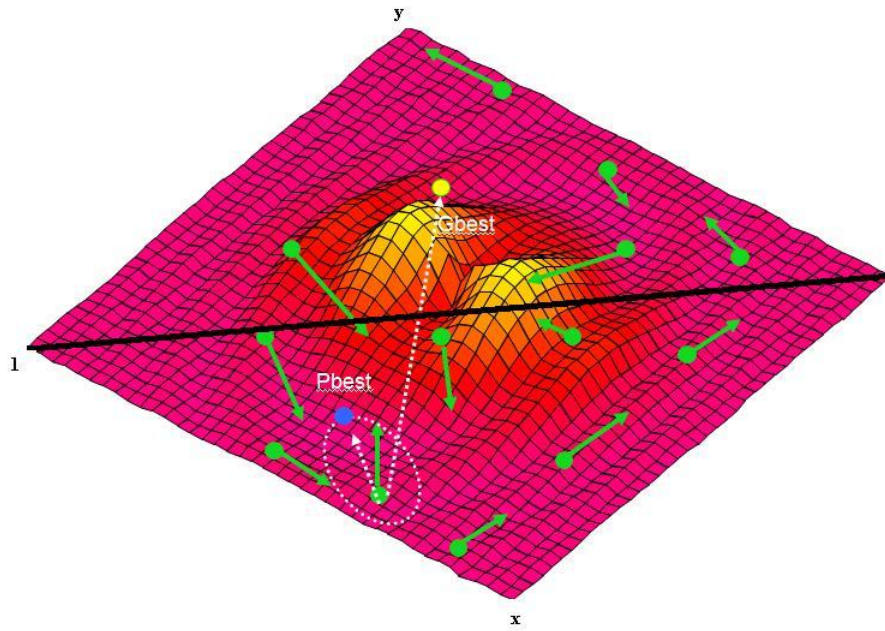


圖 4.1：二維平面問題示意圖

初始化

同 3.1 小節，在本實驗我們將維度設為二維，因為這是在平面中模擬的實驗，位置向量的範圍即是二維的範圍，速度向量範圍會依據可移動的位置向量範圍做調整。

計算評估值

同 3.1 小節，在平面實驗中，我們制訂了一個公式用來計算評估值 f ，如式 4.1。在這個模擬實驗我們假定最佳解是知道的，目的是為了觀察粒子個體的移動。

$$f = \sqrt{(X_i - Goal_x)^2} + \sqrt{(Y_i - Goal_y)^2} \quad (4.1)$$

比較過去經驗、更新

和 3.1 小節皆相同。

疊代速度、檢查速度向量、疊代位置、檢查位置向量

因為在平面中搜尋範圍是有限制的，我們必須控制粒子個體的移動範圍、速度向量，舉例來說，假設是 1000×1000 的搜尋空間，粒子個體在疊代步驟計算後，位置向量很有可能超過我們預設的搜尋空間，因此我們這邊多了一個判斷的步驟來確保粒子個體會存在範圍內。

其中速度向量在二維實驗中若有調整時，可再細分為兩種，速度向量正規化和速度向量未正規化，表 1 是速度向量正規化演算法，第五章列出實驗結果實我們會比較兩種做法的實驗結果，以及適合使用的時機。

表 1：速度向量正規化演算法

輸入：疊代速度後需要調整的速度向量(V_{xi}, V_{yi})

輸出：正規化計算後的速度向量(V_{xi}, V_{yi})

1. if ($V_{xi} > V_{max}$) then $t_x = V_{max} / V_{xi}$
else if ($V_{xi} < V_{min}$) then $t_x = V_{min} / V_{xi}$
else $t_x = 1$
2. if ($V_{yi} > V_{max}$) then $t_y = V_{max} / V_{yi}$
else if ($V_{yi} < V_{min}$) then $t_y = V_{min} / V_{yi}$
else $t_y = 1$
3. $t_{min} = \min(t_x, t_y)$
 $V_{xi} = V_{xi} * t_{min}$
 $V_{yi} = V_{yi} * t_{min}$

其中 t_x 、 t_y 和 t_{min} 為實驗暫存數值，跟實驗結果沒有直接關係。

達到停止條件

在二維實驗中，實驗的停止條件是對所有粒子個體的 fitness 值皆小於 0.1，因為起先制訂計算 fitness 值的公式時(式 4.1)，可以發現我們是用粒子個體座標和目標有直接的計算關係，因此當值小於到一定的程度後，便可以在圖上顯示所有粒子個體皆已經逼近至預設的目標。

透過此實驗，我們可以更清楚了解粒子群最佳化演算法是一種在搜尋空間中不斷跑動搜尋，逼近最佳解的演算法。

4.1.2 圓錐近似中粒子群最佳化

本實驗是將圓錐近似當作實驗範例，加入粒子群最佳化演算法的模擬實驗，我們假定最佳解的目標在平面中一個二元二次方程式的橢圓(式 4.2)，粒子個體初始分佈是採用一定比例的正确資料和錯誤資料，仿造估測基礎矩陣的幾個實驗條件，同樣要進行隨機取樣，希望可以有效逼近假定的最佳解，並且藉由幾個實驗條件的設定，得到一些結論與分析，流程圖稍做修改如圖 4.1。

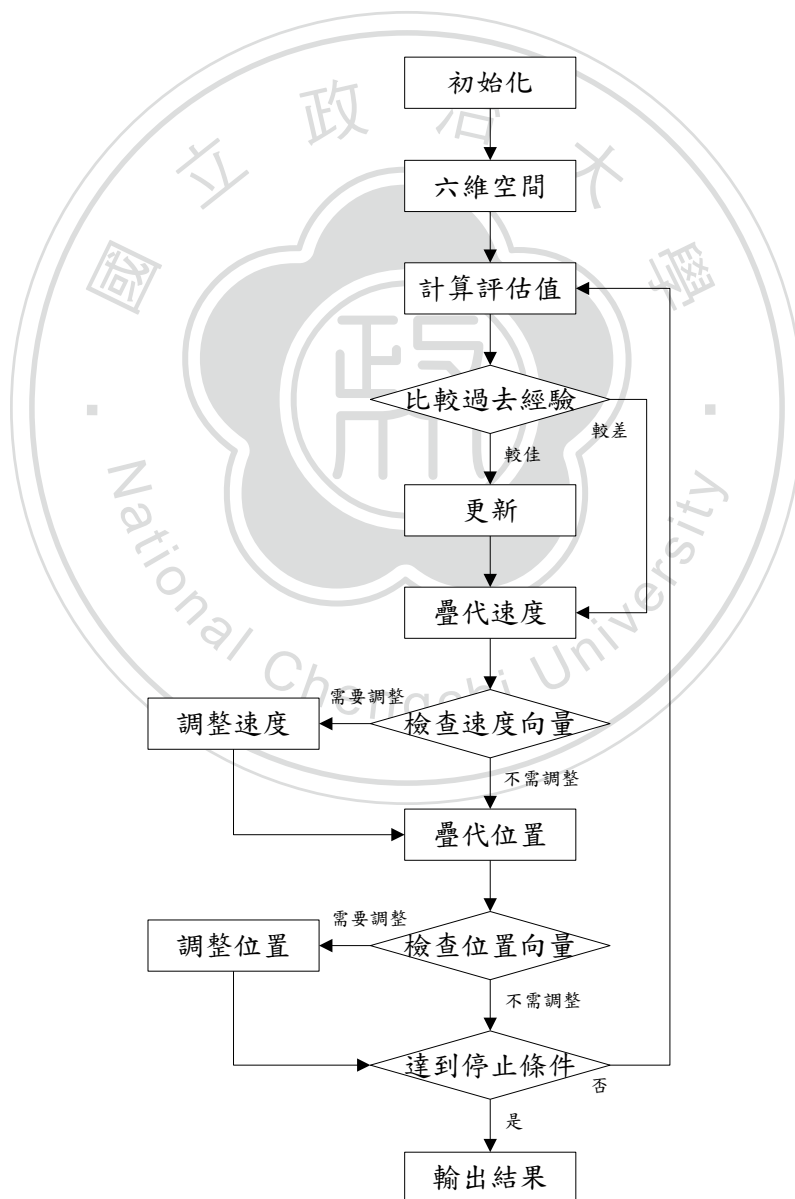


圖 4.2：圓錐近似中粒子群最佳化演算法流程圖

$$Q(x_i, y_i) = Ax_i^2 + 2Bx_iy_i + Cy_i^2 + 2Dx_i + 2Ey_i + F = 0 \quad (4.2)$$

本研究中會將式 4.2 做正規化，將 $A+C=1 \Rightarrow C=1-A$ ，則式 4.2 可以改寫成為式 4.3。

$$Q(x_i, y_i) \Rightarrow A(x_i^2 - y_i^2) + 2Bx_iy_i + 2Dx_i + 2Ey_i + F = -y_i^2 \quad (4.3)$$

經由線性代數轉換成以下式 4.4。

$$\begin{bmatrix} x_i^2 - y_i^2 & 2x_iy_i & 2x_i & 2y_i & 1 \end{bmatrix}_{i \times 5} \begin{bmatrix} A \\ B \\ D \\ E \\ F \end{bmatrix}_{5 \times 1} = \begin{bmatrix} -y_i^2 \end{bmatrix}_{i \times 1} \quad (4.4)$$

$\{x_i, y_i \mid i = 1, 2, 3, \dots, N\}$

再將式子簡化，可以寫成 $\Rightarrow Ap = b \Rightarrow Ap - b = [0]_{i \times 1}$ ，其中 A 集合表示式 4.4 中 $i \times 5$ 矩陣，p 集合表示式 4.4 中 5×1 矩陣，b 集合表示式 4.4 中 $i \times 1$ 矩陣，殘餘誤差值 r 可以寫成 $r = Ap - b = [0]_{i \times 1}$ 。

在推導過後，可以由式 4.5 求得 p 集合解。

$$p = (A^T A)^{-1} A^T b \quad (4.5)$$

求出 p 集合後，我們代回式 4.3 得到一組二元二次方程式，並且在平面上繪出圖形，這是以一組取樣六個座標點可以計算出一組圖形的方法，接著我們對圖 4.1 的流程圖介紹步驟。

初始化

在本實驗中初始化要先決定所有點的分佈情形，例如正確資料和錯誤資料的比例分配，並且對每個點給予編號，接著是分配一定數量的粒子個體，每個粒子個體內會隨機取樣六個點，再來給予初始的速度向量，本實驗將有六個維度。

六維空間

和平面中的實驗條件不同，由於這邊六個點我們可以看成獨立事件，彼此之間並不會互相影響，因此我們將粒子個體內的六個點編號進行由小到大排序，並且分割為六個維度，如此一來，每個粒子個體在後面程序時，便可以依照不同維度內進行粒子群最佳化演算法的步驟。

計算評估值

本實驗的評估值計算和先前不同，我們會用到最小平方中值法中的其中一個步驟，建立最小平方中值表格，即一組粒子個體可以計算出一組橢圓方程式，接著將初始每個點都代入方程式後，可以計算出殘餘誤差值，將誤差值由小到大排序過後，我們取中位數當作我們的 fitness 值，這樣採用評估值的好處是，我們相信前 50% 誤差較小的點，是可以當作參考的依據。

比較過去經驗、更新

和 3.1 小節相同，

疊代速度、檢查速度向量

和 3.1 小節相同，但需要特別注意，每個維度之間都要遵守最大的移動速度，若有超過，皆需要調整回最大。

疊代位置、檢查位置向量

本實驗中搜尋空間是以初始分佈的點數量決定，舉例來說，當我們分佈 n 個點時，一個維度的搜尋空間就是 $1\sim n$ ，在這邊實驗內，需要透過六個不同的點計算出一組方程式，所以不能挑選重複的點，因此在疊代位置時是要注意的部份，當有重覆的情況產生時，便要透過我們設計的調整位置向量演算法，如表 2。

表 2：調整 X 位置向量演算法

輸入：疊代位置完後的粒子個體

輸出：粒子中六個點編號皆不重複

1. 對粒子個體六個點編號檢查，從第 1 個開始，判斷是否有重覆。
2. 用氣泡旋轉方式檢查，若出現最大值，則前者-1，若出現最小值時，則後者+1，一般正常狀況是後者直接+1。
3. 重複步驟 2-3，直到所有粒子檢查結束。
4. 輸出各組粒子的六個不重複點編號。

達到停止條件

在圓錐近似這個問題中，其實較難去設置停止條件，因此我們採用了幾種不同的停止條件並觀察其結果，分別是：限制疊代次數，對 Gbest 收斂，其中收斂又可以調整收斂的範圍，第三是限制 fitness 值和對 Gbest 收斂，我們將在第五章時呈現實驗結果。

圓錐近似的問題中，我們嘗試用粒子群最佳化演算法加入解題，並且仿效估測基礎矩陣問題中假設一些實驗條件，有助於我們繼續進行估測基礎矩陣的研究，在下一小節，將著重在本研究中的問題，設計研究方法與實驗流程。

4.2 研究方法與流程

4.2.1 系統架構

在前面章節中，已經介紹估測基礎矩陣的原理和流程，因此在本節我們將直接加入粒子群最佳化演算法，和前面基礎實驗中不同的是，實驗資料是兩張相片中，先取得大量且可靠的對應點，因為 SIFT 可以針對平移和旋轉的照片尋找對應點，所以在本研究中採用 SIFT 方法，之後將找到的對應點給予編號，接著會根據相片解析度，切割為數個固定大小的籃子(bucket)，因為從幾何關係的角度上，倘若隨機取樣的對應點有共平面發生時，會影響我們估測的結果，所以我們先將尋找到的對應點，依照座標位置落在各個籃子中，在各個籃子的對應點也給予編號，我們只要先選取籃子，再選取籃子內的對應點，就可以盡量避免共平面問題。

當這些資料準備就緒後，即可以開始進行粒子群最佳化演算法估測基礎矩陣，
流程圖我們將以圖 4.2 表示。

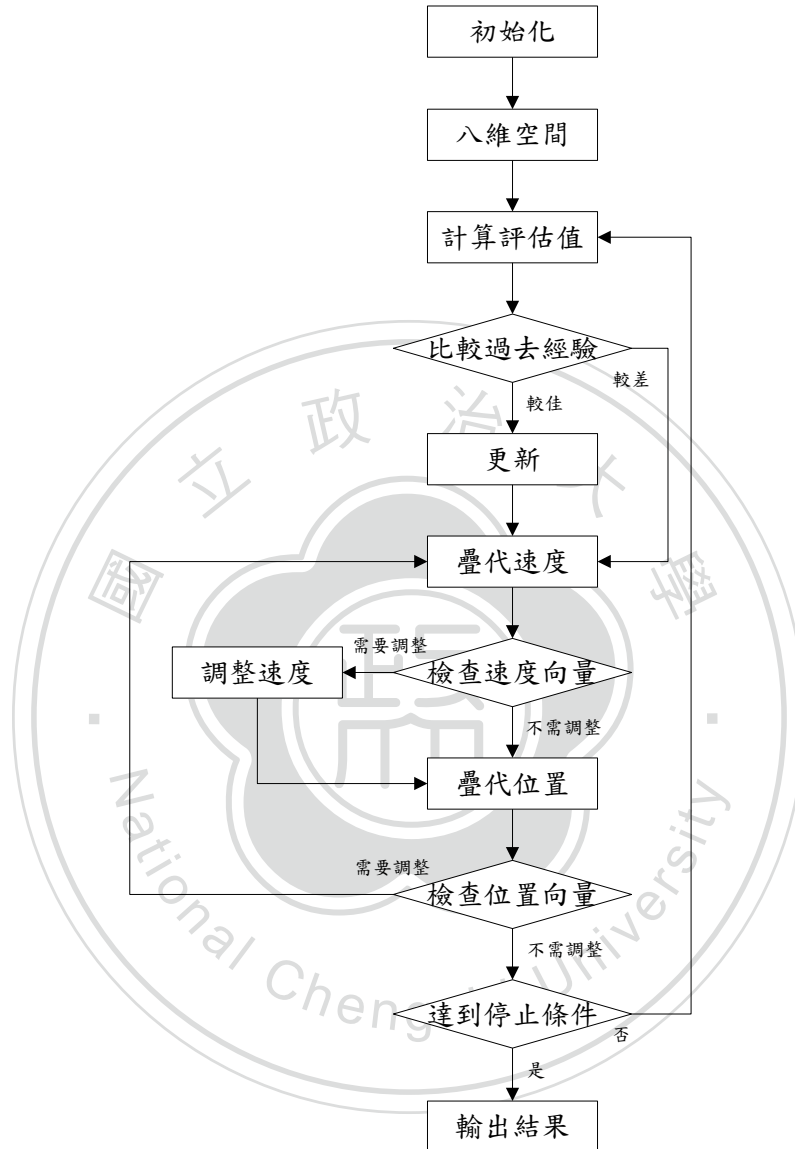


圖 4.3：以粒子群最佳化演算法估測基礎矩陣流程圖

初始化

初始化在 SIFT 尋找對應點和分配至籃子工作完成後，便決定粒子個體的數目，這邊為了避免共平面，我們是採用先隨機取有對應點的籃子，決定八個籃子後，再從挑選出的籃子內隨機取一組對應點，接著藉由籃子和籃子內的編號，可以推回去原本的對應點編號，接著在分配速度向量，本實驗會有八個維度。

八維空間

取樣的八個點編號會進行由小到大排序，並且分割為八個維度，接著便可以設定 i 為我們取樣的粒子個體數量， d 為維度，代入 3.1 節中公式 3.1、3.2 進行實驗。

計算評估值

挑選完八組對應點後，我們可以計算出一組基礎矩陣，接著將 SIFT 找到的對應點逐次代入這組基礎矩陣後，可以得到每組對應點計算出的殘餘誤差值，接著建立最小平方中值表格後，取中位數當作 fitness 值，因為 SIFT 在尋找對應點上可以相信有 50% 以上的準確率，因此我們這樣子決定 fitness 值的制訂方法是合理的。

比較過去經驗、更新

和 3.1 小節相同。

疊代速度、檢查速度向量

和 4.1.2 小節相同。

疊代位置、檢查位置向量

本實驗搜尋空間是以 SIFT 找到的對應點數量決定，舉例來說，當我們找到 n 個點時，一個維度的搜尋空間就是 $1\sim n$ ，在估測基礎矩陣實驗中，需要八組不同的對應點計算出一組基矩陣，所以不能挑選重複的點，在初始化時我們曾利用籃子幫忙挑選對應點，因此在疊代位置時，雖是用對應點編號進行疊代，但確會用籃子編號進行檢查，當有籃子編號重複時，我們便會回到疊代速度步驟，計算新的速度向量，直到算出來的位置向量不需調整，便可以進行下一步驟。

達到停止條件

本實驗用疊代次數當作停止條件，因為每一個粒子個體經過一次疊代後，都會有新的一組結果，為了和過去幾種估測方法比較，我們選擇用接近的取樣次數，來比較實驗結果，至於該怎麼衡量估測基礎矩陣的好壞，便需要地面實況(ground truth)資料輔助我們計算結果好壞。

4.2.2 評估基礎矩陣

和先前實驗較不同的地方在於，平面和圓錐近似兩個實驗可以繪製在圖形上顯示出結果，直接在圖上判斷結果好壞，但是若我們要評估基礎矩陣的好壞，並沒有辦法畫在圖上，只能由兩張相片的極線幾何關係評估，將第一張影像上任一組特徵點，透過估測出來的基礎矩陣在第二張相片上畫出極線，接著計算第一張

影像上特徵點在第二張影像上的對應點，由這個對應點，和極線計算距離誤差，同理，將第二張影像上的特徵點，經由同樣的方法，也可以在第一張相片上畫出極線，由對應在第一張影像上的點進行計算距離誤差，若是同一組對應點時，可以取平均值當作最後的一組誤差，當我們統計的點數越多時，越可以清楚知道我們估測基礎矩陣的好壞。圖 4.3 是計算誤差的示意圖，計算公式如式 4.6。

$$d(m, L) = \frac{|ax + by + c|}{\sqrt{a^2 + b^2}} \quad (4.6)$$

m 為對應點的座標， L 為 m 在另外一張相片畫出的極線方程式 $ax + by + c = 0$ ，公式帶入後求出的 d 值即為誤差值。

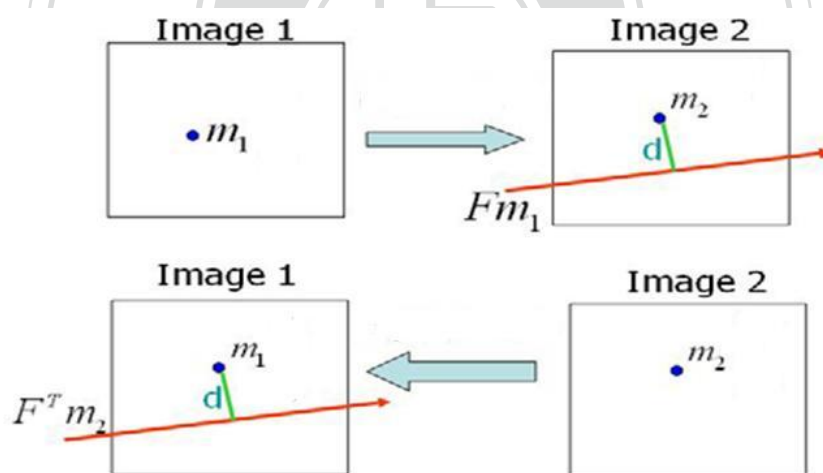


圖 4.4：計算基礎矩陣誤差示意圖

第五章

實驗結果

在本章，我們會展示第四章兩個基礎實驗和本研究估測基礎矩陣實驗的結果，並且說明與討論，首先我們先列出平面中粒子群最佳化實驗，除了驗證粒子群最佳化演算法的執行效率和適合解最佳化問題之外，我們假設並配置幾個實驗條件，解決於平面中找尋一個最佳座標點的問題，再來以圓錐近似中粒子群最佳化為第二個實驗，除了和先前的演算法略有變動之外，我們首次採用以調整點編號決定取樣的方式，希望能用粒子群最佳化演算法有效的達到圓錐近似的目標，最後，我們列出本研究中主要問題，以粒子群最佳化估測基礎矩陣實驗結果。

5.1 平面中粒子群最佳化實驗結果

在 4.1.1 小節時我們曾提過速度正規化的觀念，為了驗證結果，我們在平面中粒子群最佳化進行兩次實驗，分別為速度未經正規化和速度經過正規化，雖然都可以達到實驗目標，但兩者花費的疊代次數是有明顯差別的。

在平面中我們假設搜尋空間為 $1000*1000$ ，近似目標為座標點(500,500)，fitness 值的制訂如式子 4.1，每次實驗我們會列出初始分佈和結束時的數據表格，並列出每經過 100 次疊代後粒子移動後的圖形。

5.1.1 速度未經過正規化

在速度未經過正規化的實驗中，我們列出其中三次實驗結果，實驗數據放在表 3、4、5，實驗流程在 4.1.1 小節，初始化採取隨機分佈 20 組粒子，以表 3 為例，粒子 1 初始 fitness 值經過計算為 103.793，經過 526 次疊代計算後，停止的 fitness 值為 0.657727。其餘粒子我們也將記錄初始 fitness 值，實驗停止條件為所有粒子的 fitness 值皆要小於 0.1，實驗即停止，同時我們也記錄了初始 Gbest 和實驗停止後的 Gbest，受限於篇幅，我們無法列出所有座標點的詳細(x,y)座標，但可由圖 5.1~圖 5.3 看出實驗進行的過程。

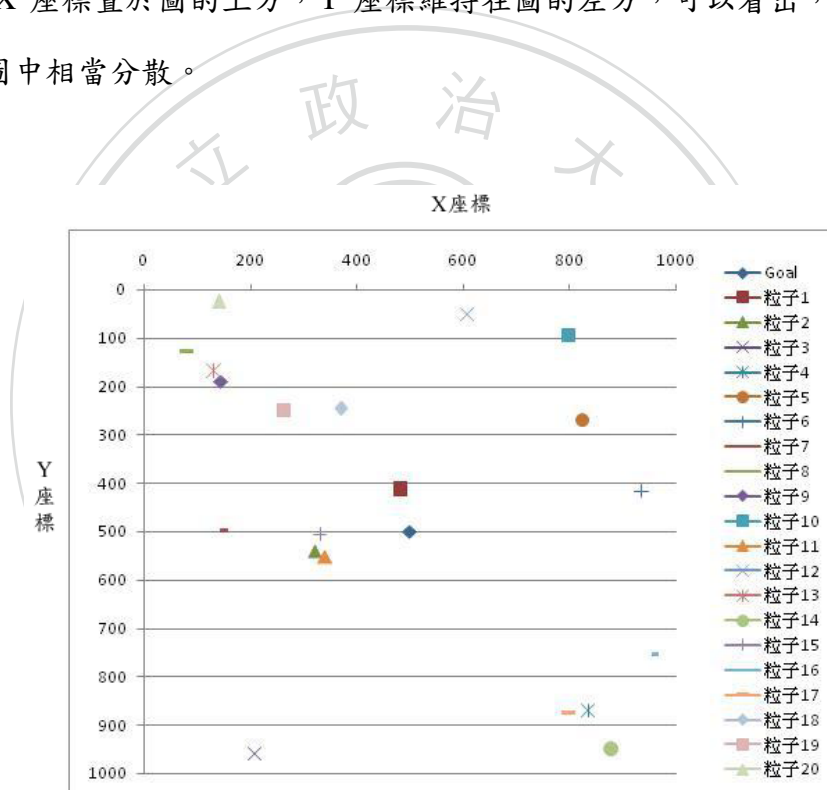
速度未經過正規化第一次實驗

表 3：速度未經過正規化第一次實驗

粒子編號	初始 fitness	停止 fitness
粒子 1	103.793	0.657727
粒子 2	229.591	0.439575
粒子 3	760.094	0.67022
粒子 4	717.948	0.436645
粒子 5	566.149	0.33785
粒子 6	525.407	0.531189
粒子 7	362.133	0.464233
粒子 8	802.881	0.299561
粒子 9	674.612	0.717736
粒子 10	713.37	0.772005
粒子 11	220.923	0.801949
粒子 12	567.248	0.852063
粒子 13	709.616	0.540056
粒子 14	838.282	0.739844
粒子 15	183.477	0.847061
粒子 16	717.795	0.68312
粒子 17	684.133	0.830808

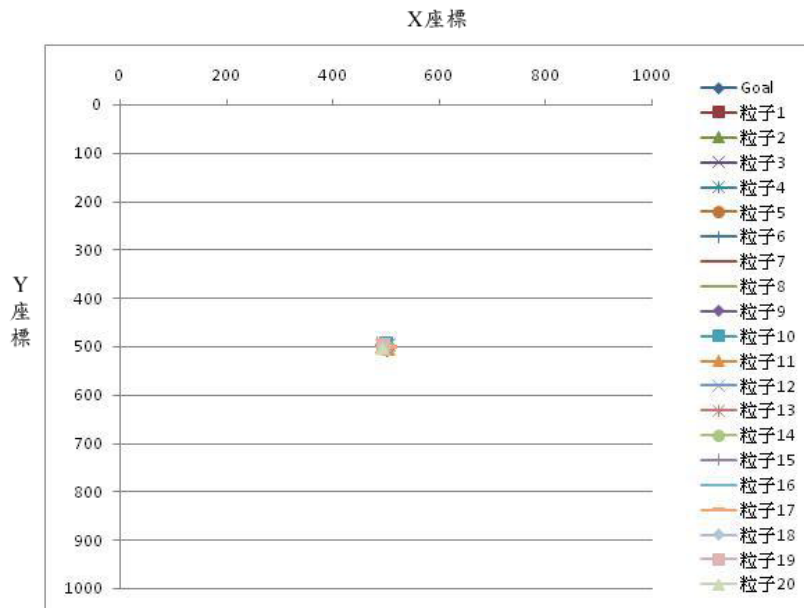
粒子 18	392.376	0.93267
粒子 19	495.773	0.172525
粒子 20	844.203	0.894932
Gbest	103.793	0.172525
平均值	555.49	0.631088
標準差	226.697	0.211664
疊代次數	526	

在圖 5.1(a)中，是實驗初始 20 個粒子的分佈圖，我們採取了影像處理的座標模式，將 X 座標置於圖的上方，Y 座標維持在圖的左方，可以看出，初始的隨機分佈在圖中相當分散。

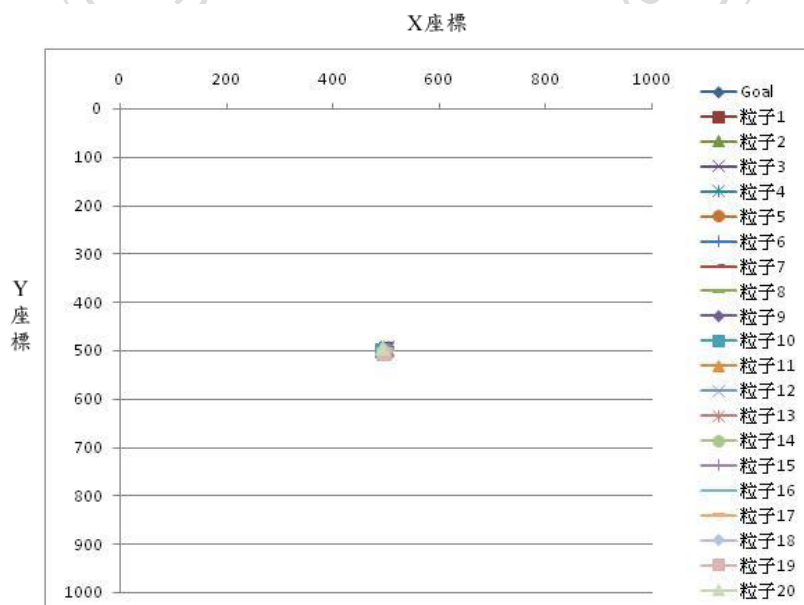


(a)

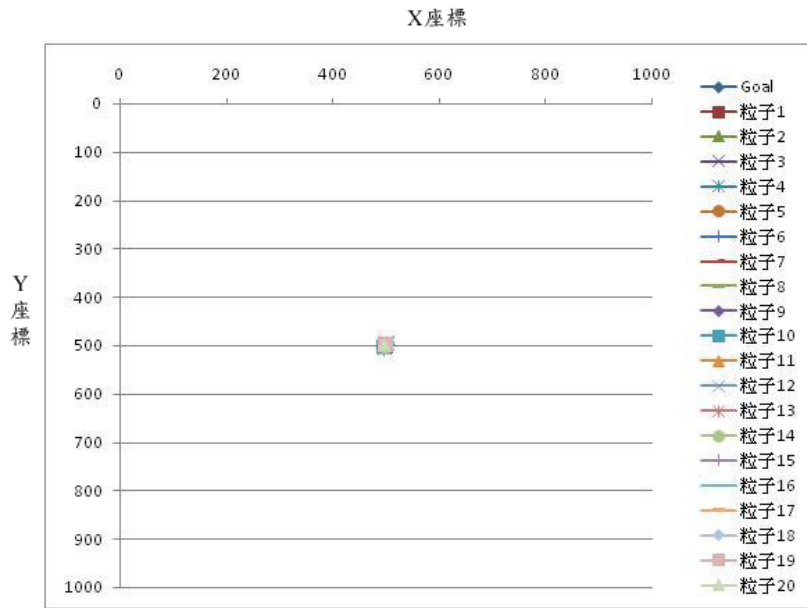
經過 100 次疊代計算後，可以看出 20 個粒子已經漸漸收斂至我們假定的最佳座標點，但是因為尚未達到實驗停止條件，因此實驗繼續進行，在 200 次疊代計算後，其實粒子群變動的幅度已經不明顯，都收斂在最佳座標點附近，直到最後實驗停止條件達成後，我們將實驗停止，並且列出最後結果。



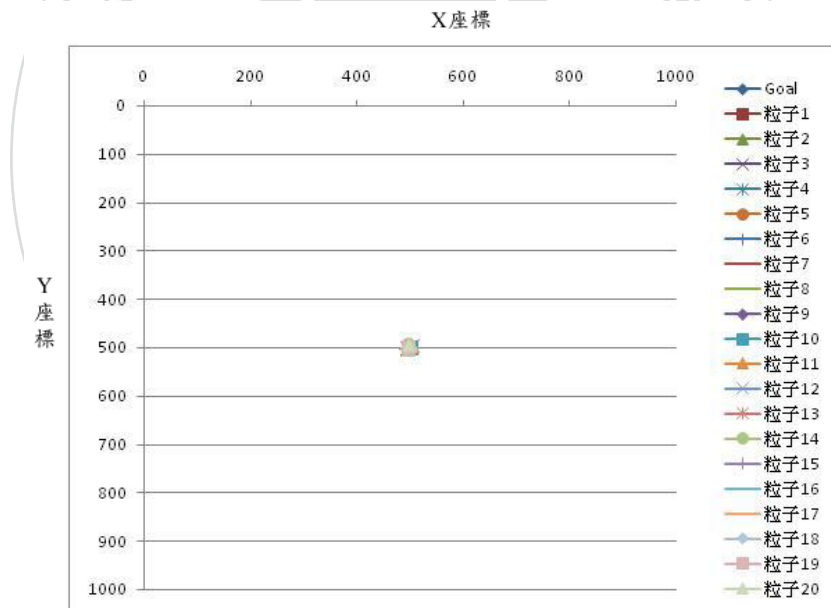
(b)



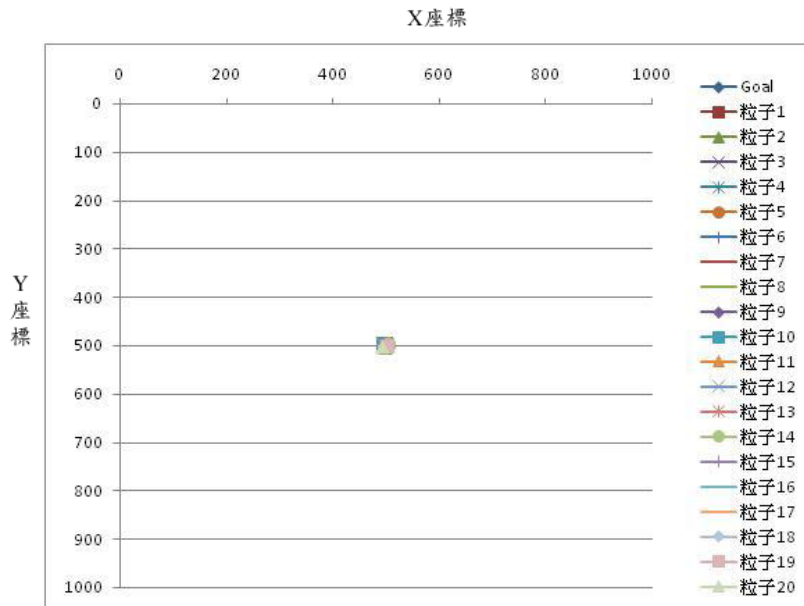
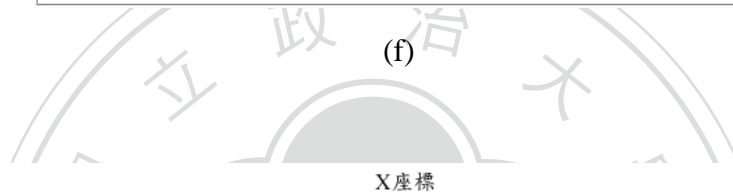
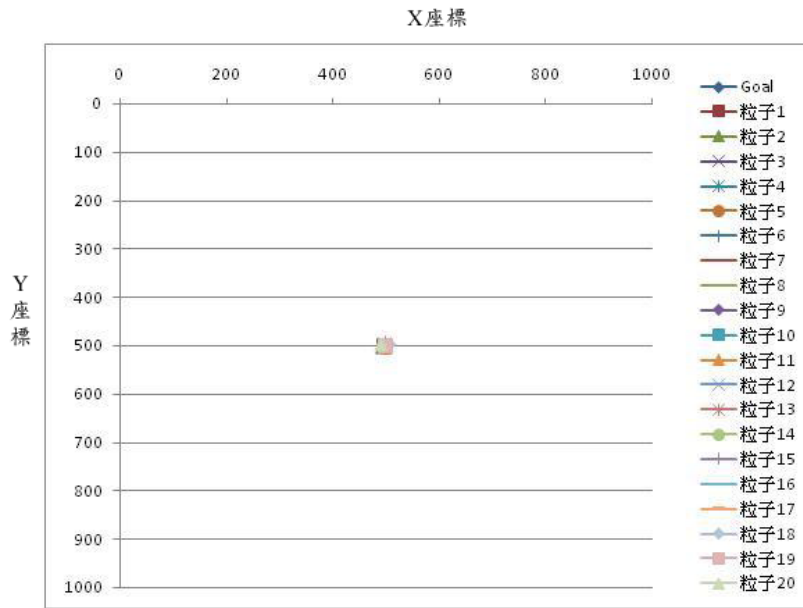
(c)



(d)



(e)



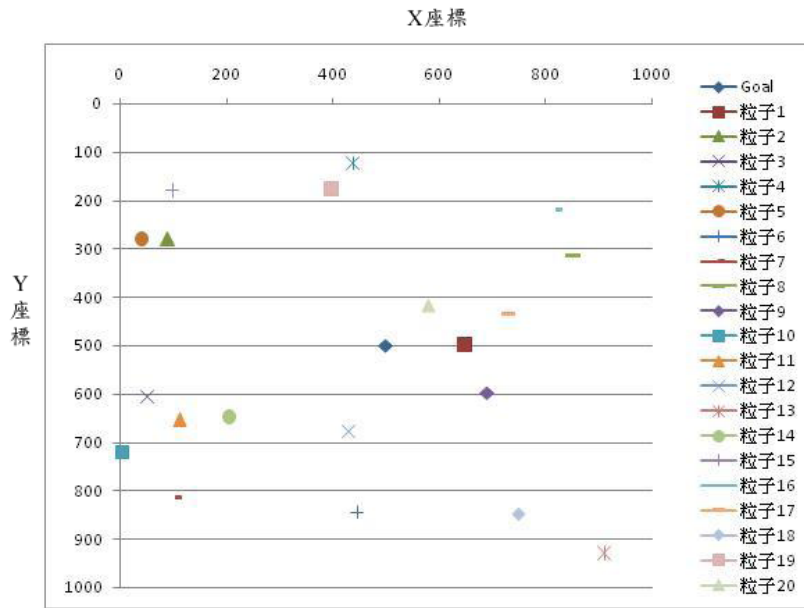
(g)

圖 5.1：速度未正規化第一次實驗 (a)初始分佈 (b)疊代 100 次後 (c)疊代 200 次後 (d)疊代 300 次後 (e)疊代 400 次後 (f)疊代 500 次後 (g) 實驗停止

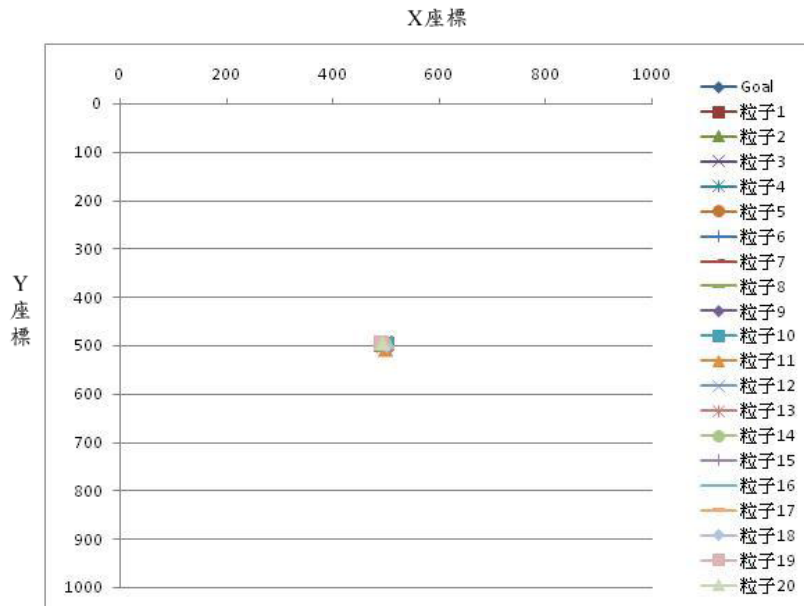
速度未經正規化第二次實驗

表 4：速度未經過正規化第二次實驗

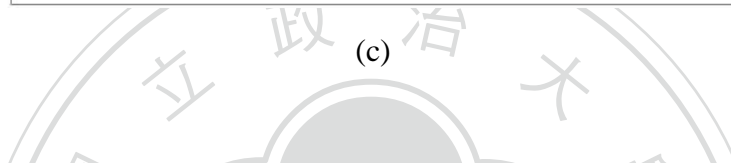
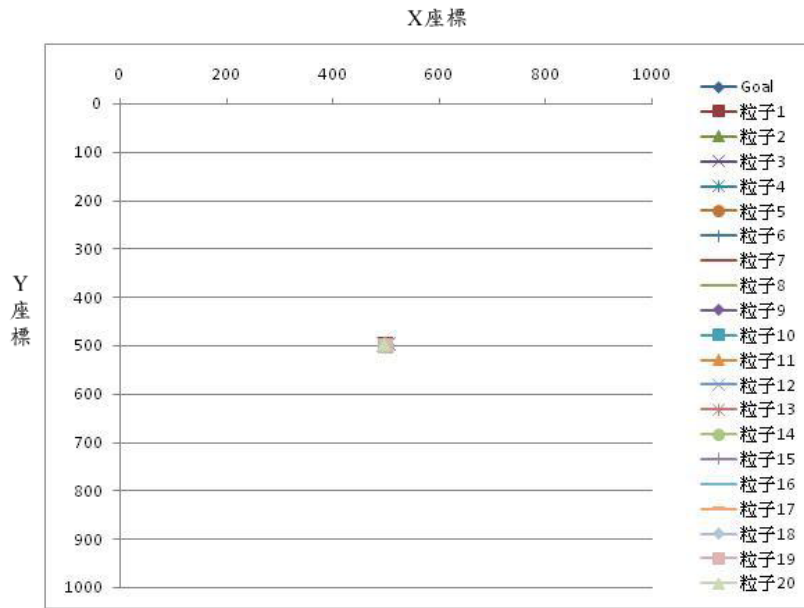
粒子編號	初始 fitness	停止 fitness
粒子 1	149.632	0.578291
粒子 2	638.63	0.427444
粒子 3	564.287	0.993407
粒子 4	447.188	0.776515
粒子 5	687.002	0.162064
粒子 6	408.185	0.106686
粒子 7	717.215	0.903953
粒子 8	549.638	0.794998
粒子 9	298.593	0.694554
粒子 10	726.188	0.361354
粒子 11	551.012	0.952971
粒子 12	257.057	0.706375
粒子 13	851.436	0.842508
粒子 14	452.254	0.681268
粒子 15	728.446	0.910115
粒子 16	610.096	0.462617
粒子 17	306.009	0.334668
粒子 18	608.081	0.81664
粒子 19	434.919	0.204055
粒子 20	162.511	0.45435
Gbest	149.632	0.106686
平均值	507.419	0.608242
標準差	193.873	0.269995
疊代次數	345	



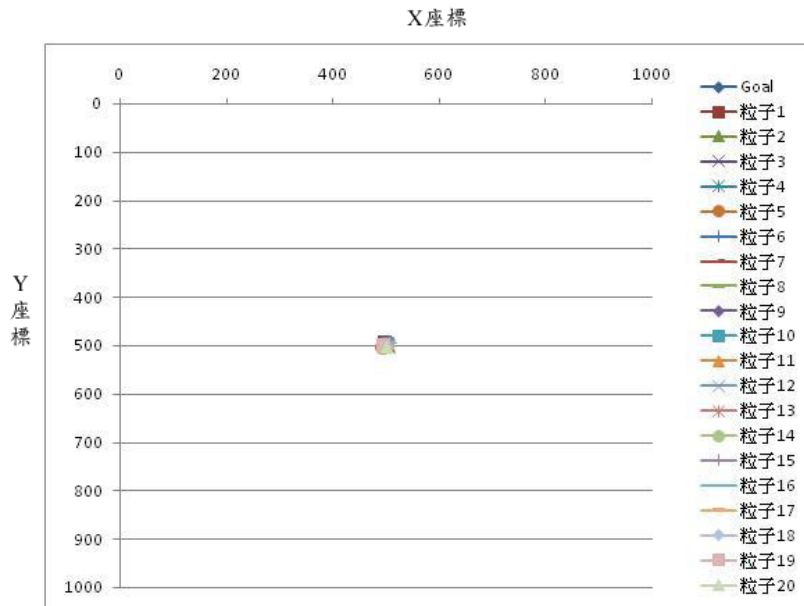
(a)



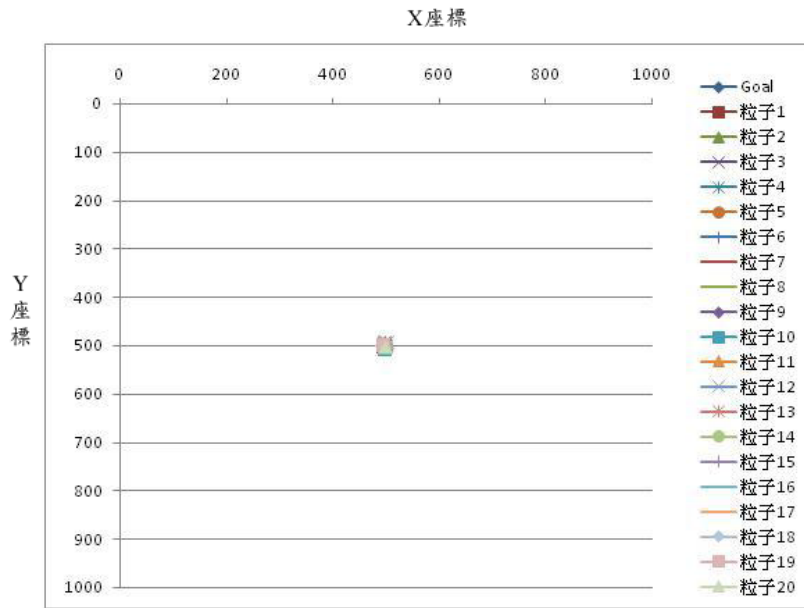
(b)



(c)



(d)



(e)

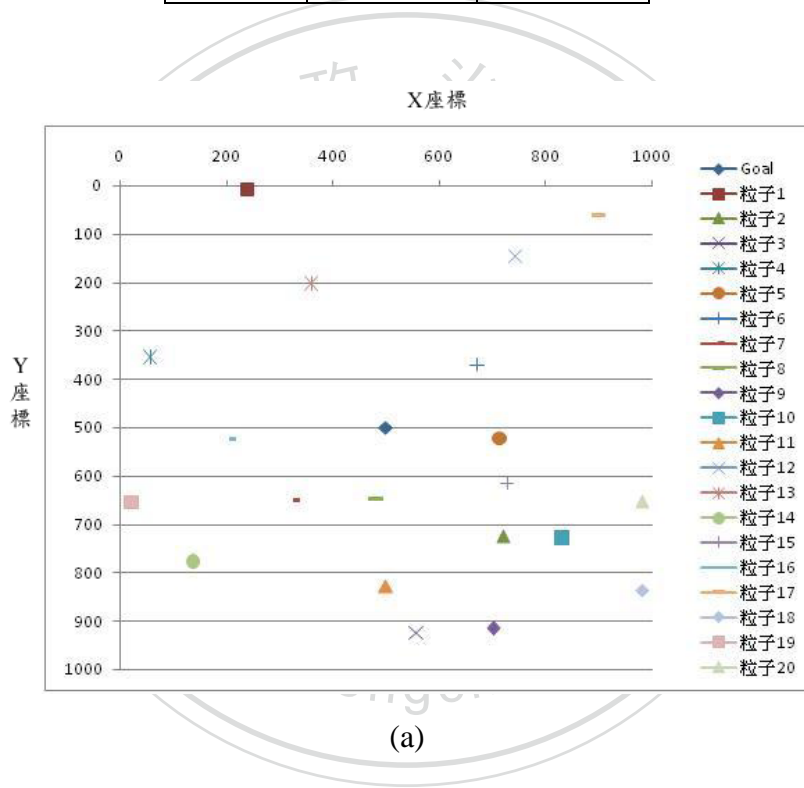
圖 5.2：速度未正規化第二次實驗 (a)初始分佈 (b)疊代 100 次後 (c)疊代 200 次後 (d)疊代 300 次後 (e)實驗停止

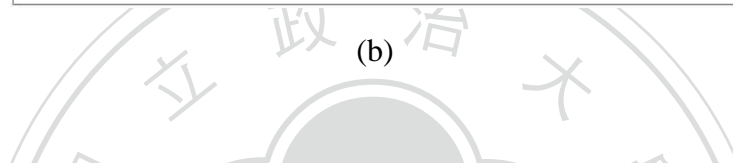
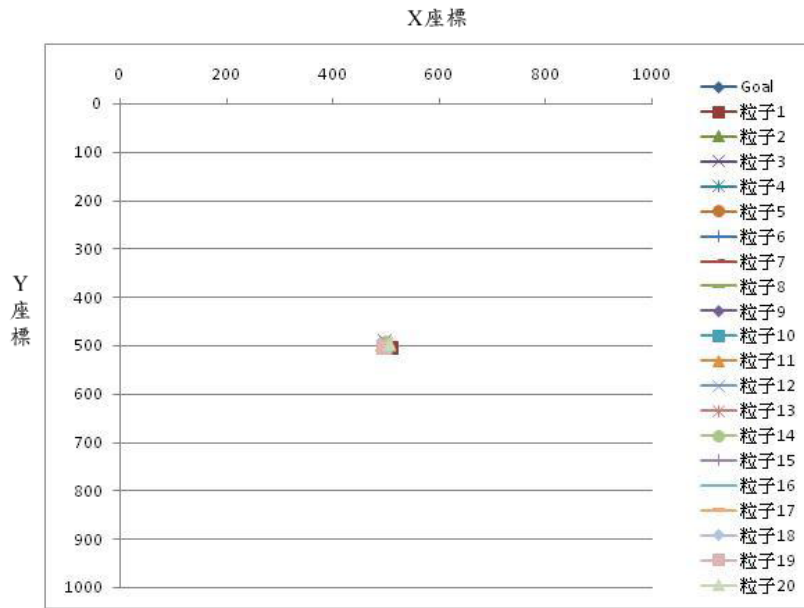
速度未經正規化第三次實驗

表 5：速度未經過正規化第三次實驗

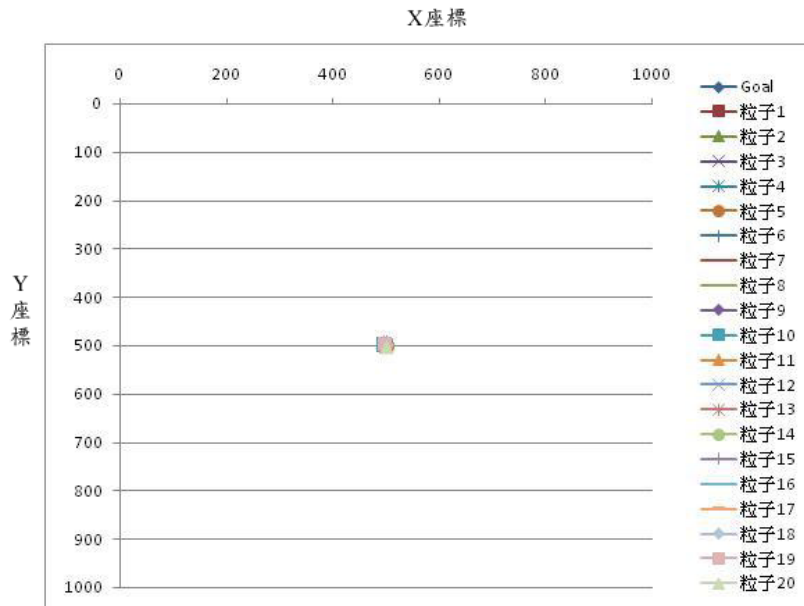
粒子編號	初始 fitness	停止 fitness
粒子 1	761.193	0.035626
粒子 2	454.543	0.924146
粒子 3	493.851	0.915178
粒子 4	597.644	0.902284
粒子 5	237.22	0.300982
粒子 6	312.296	0.880012
粒子 7	331.675	0.535077
粒子 8	164.403	0.893676
粒子 9	627.888	0.767696
粒子 10	569.72	0.64459
粒子 11	338.389	0.846648
粒子 12	608.325	0.856534
粒子 13	445.57	0.94182

粒子 14	648.885	0.276665
粒子 15	342.906	0.575668
粒子 16	317.728	0.928936
粒子 17	849.757	0.600338
粒子 18	828.211	0.687454
粒子 19	642.811	0.854468
粒子 20	644.612	0.321167
Gbest	164.403	0.035626
平均值	510.881	0.684448
標準差	191.859	0.261441
疊代次數	430	

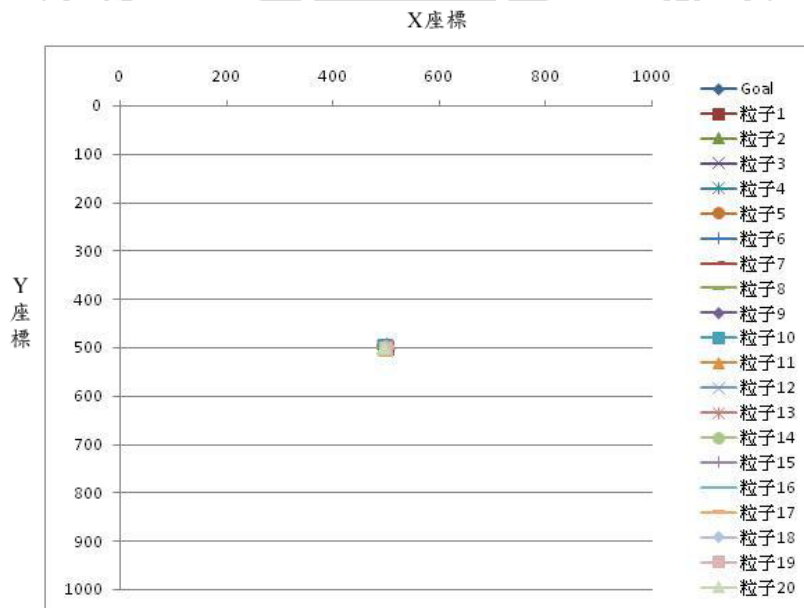
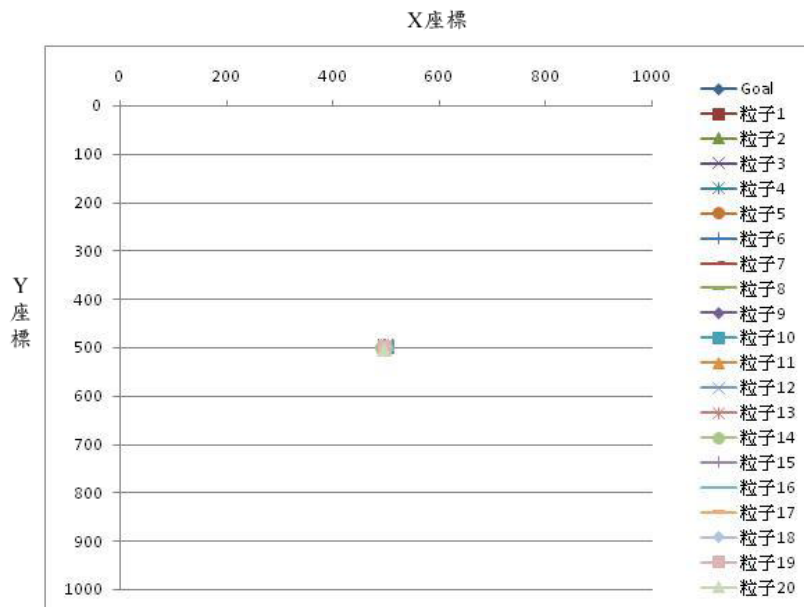




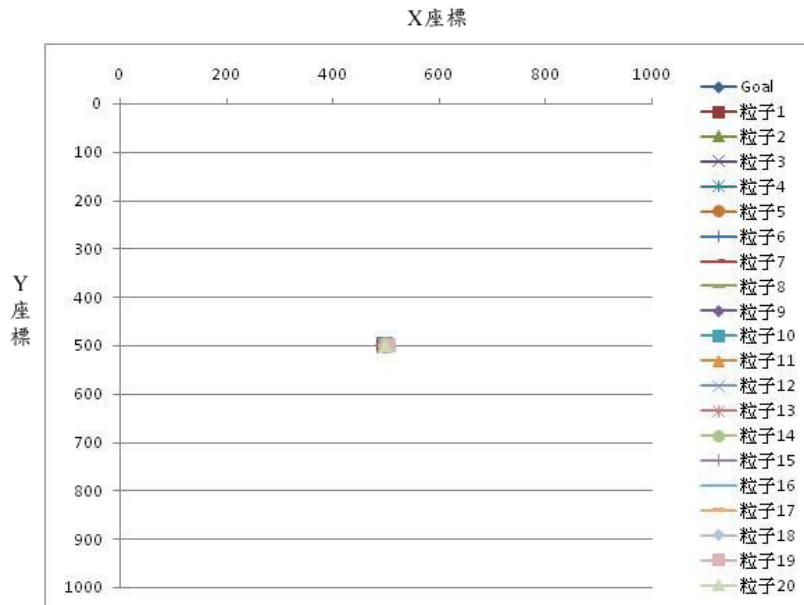
(b)



(c)



(e)



(f)

圖 5.3：速度未正規化第三次實驗 (a)初始分佈 (b)疊代 100 次後 (c)疊代 200 次後 (d)疊代 300 次後 (e)疊代 400 次後 (f)實驗停止

平均疊代次數：433.67

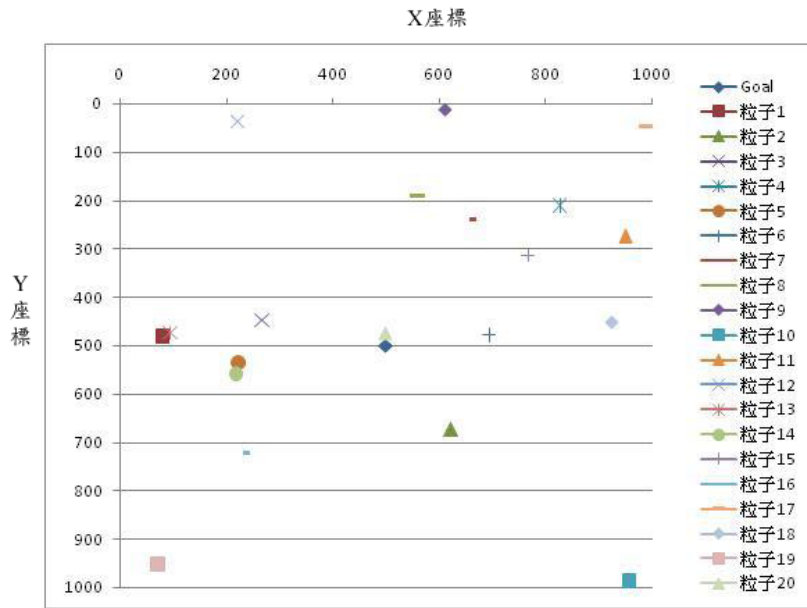
5.1.2 速度經過正規化

在速度經過正規化的實驗中，我們同樣列出三次實驗結果，實驗數據放在表 6、7、8，初始化採取隨機分佈 20 組粒子，以表 6 為例，粒子 1 初始 fitness 值經過計算為 103.793，經過 526 次疊代計算後，停止的 fitness 值為 442.915。其餘粒子我們也將記錄初始 fitness 值，實驗停止條件為所有粒子的 fitness 值皆要小於 0.1，實驗即停止，同時我們也記錄了初始 Gbest 和實驗停止後的 Gbest，受限於篇幅，我們無法列出所有座標點的詳細(x,y)座標，但可由圖 5.4~圖 5.6 看出實驗進行的過程，可以看出，平均花費的疊代次數減少。

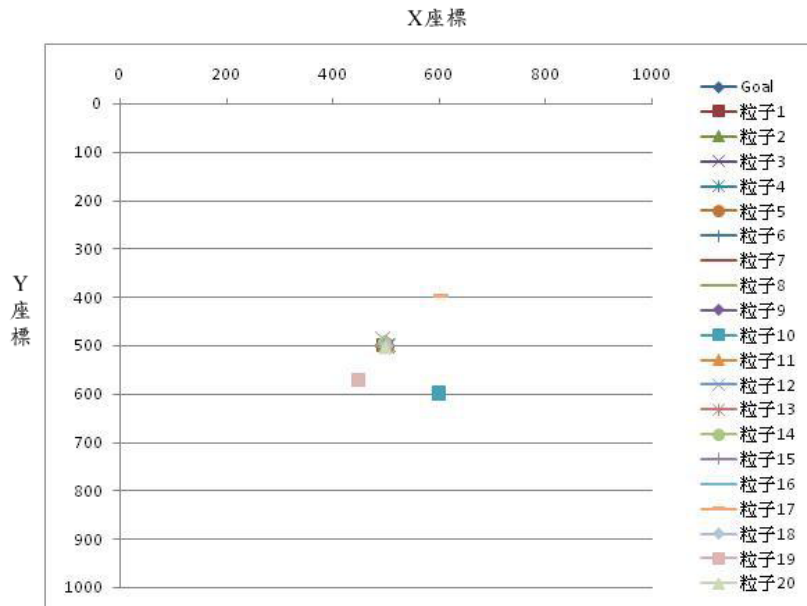
速度經過正規化第一次實驗

表 6：速度正規化第一次實驗

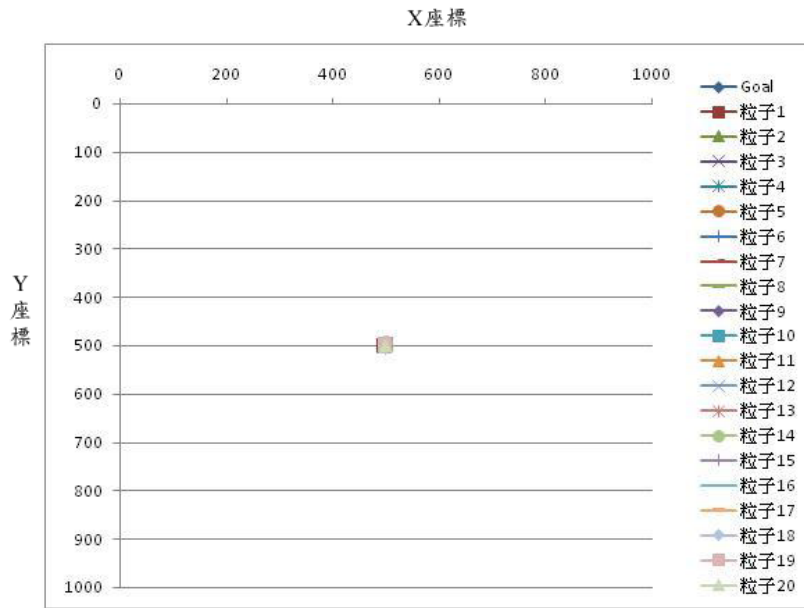
粒子編號	初始 fitness	停止 fitness
粒子 1	442.915	0.529885
粒子 2	303.232	0.946604
粒子 3	293.222	0.667126
粒子 4	622.15	0.399617
粒子 5	317.972	0.877642
粒子 6	220.954	0.76358
粒子 7	425.336	0.464154
粒子 8	379.04	0.96982
粒子 9	605.182	0.266683
粒子 10	953.246	0.568593
粒子 11	682.272	0.815373
粒子 12	748.466	0.427144
粒子 13	433.882	0.623231
粒子 14	347.484	0.976441
粒子 15	460.372	0.405031
粒子 16	497.879	0.269902
粒子 17	953.001	0.72099
粒子 18	477.157	0.921068
粒子 19	886.227	0.569809
粒子 20	27.3446	0.752216
Gbest	27.3446	0.266683
平均值	503.867	0.646745
標準差	239.743	0.224104
疊代次數	172	



(a)



(b)



(c)

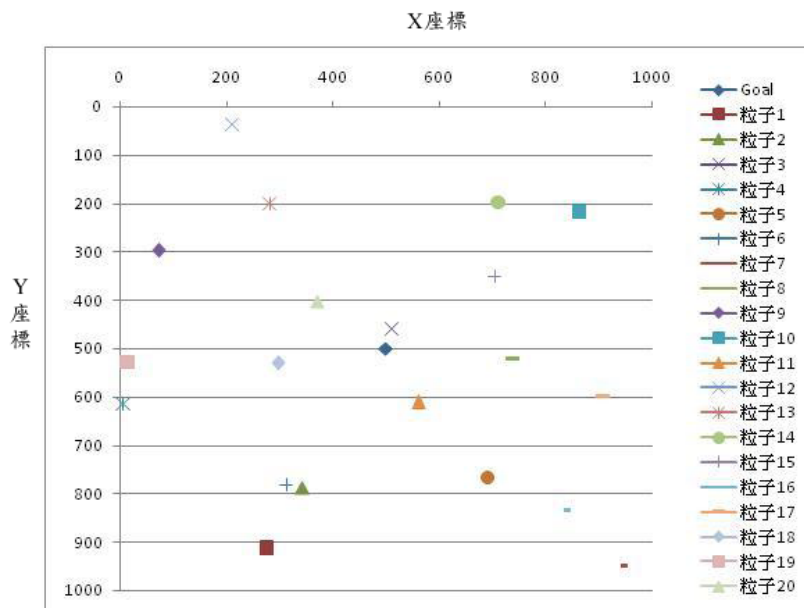
圖 5.4：速度正規化第一次實驗 (a)初始分佈 (b)疊代 100 次後 (c)實驗停止

速度經過正規化第二次實驗

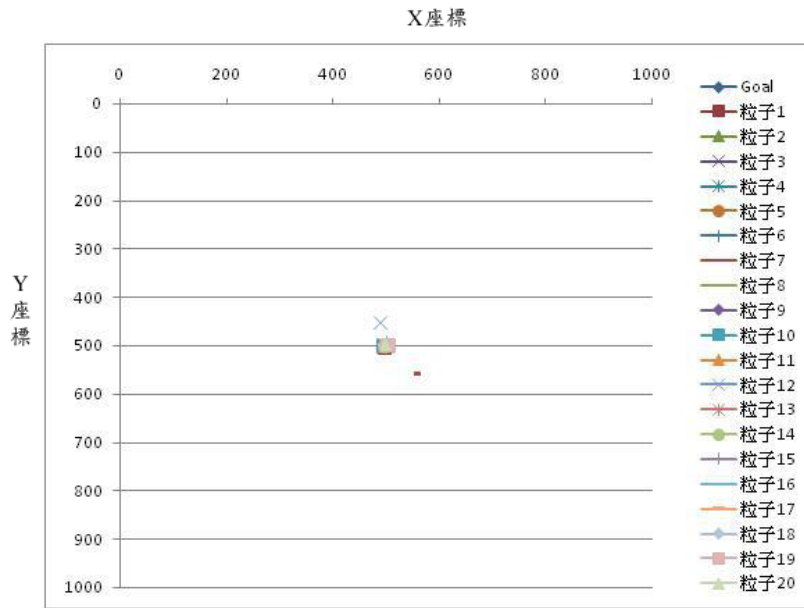
表 7：速度正規化第二次實驗

粒子編號	初始 fitness	停止 fitness
粒子 1	643.208	0.733114
粒子 2	450.85	0.673877
粒子 3	58.9923	0.767924
粒子 4	615.589	0.2485
粒子 5	465.285	0.77931
粒子 6	475.661	0.694566
粒子 7	897.519	0.137358
粒子 8	268.075	0.240469
粒子 9	636.28	0.505828
粒子 10	652.821	0.652019
粒子 11	177.831	0.25602
粒子 12	755.394	0.260135
粒子 13	525.956	0.82313
粒子 14	520.188	0.850336
粒子 15	362.682	0.579484

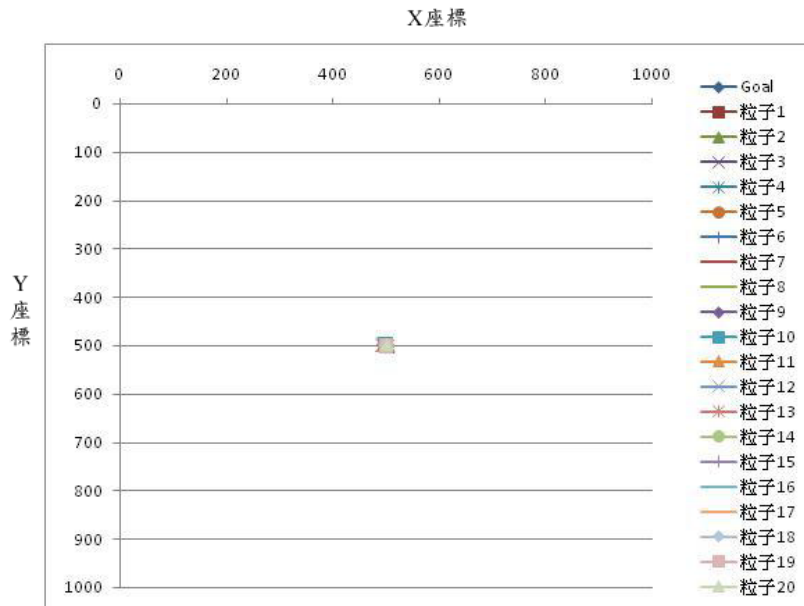
粒子 16	678.823	0.097698
粒子 17	512.375	0.811489
粒子 18	241.005	0.927598
粒子 19	519.15	0.4073
粒子 20	233.833	0.907129
Gbest	58.9923	0.097698
平均值	484.576	0.567664
標準差	205.052	0.267629
疊代次數	180	



(a)



(b)



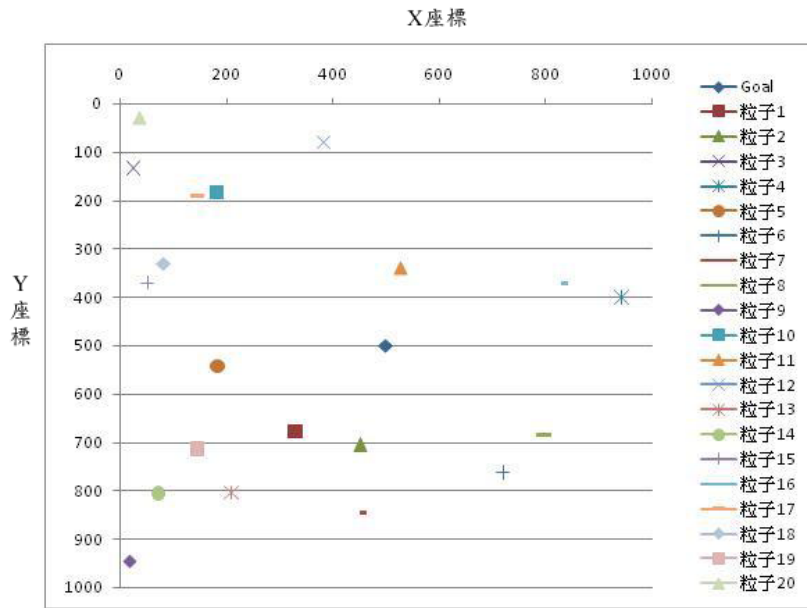
(c)

圖 5.5：速度正規化第二次實驗 (a)初始分佈 (b)疊代 100 次後 (c)實驗停止

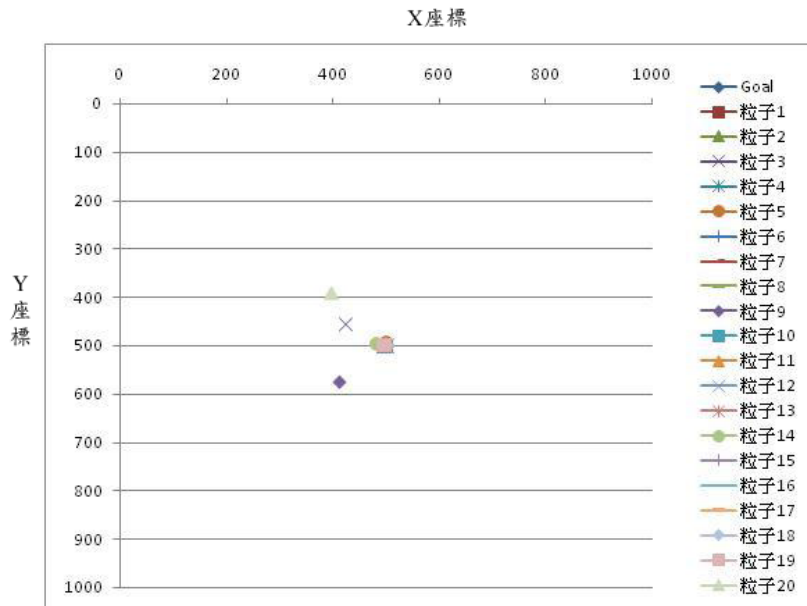
速度經過正規化第三次實驗

表 8：速度正規化第三次實驗

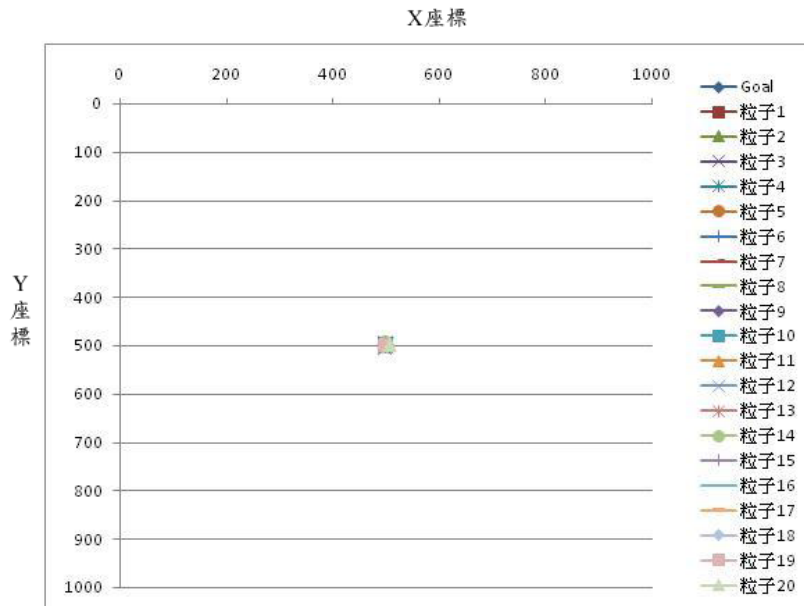
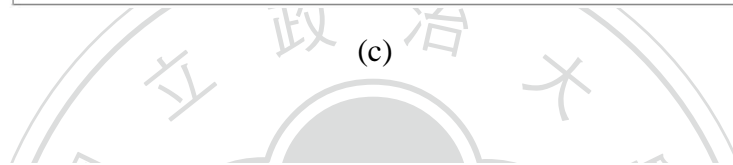
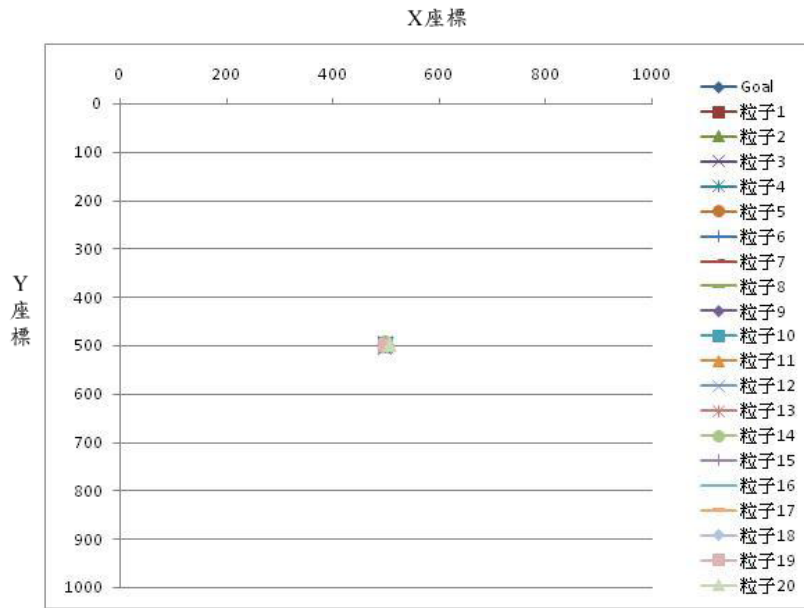
粒子編號	初始 fitness	停止 fitness
粒子 1	355.052	0.824258
粒子 2	258.278	0.627105
粒子 3	846.919	0.987563
粒子 4	546.953	0.968214
粒子 5	365.001	0.729819
粒子 6	492.569	0.615142
粒子 7	397.198	0.529327
粒子 8	487.533	0.572341
粒子 9	934.751	0.290347
粒子 10	639.637	0.558223
粒子 11	192.907	0.740161
粒子 12	544.694	0.544433
粒子 13	602.618	0.394315
粒子 14	742.149	0.461306
粒子 15	580.34	0.441606
粒子 16	461.531	0.775901
粒子 17	667.653	0.740729
粒子 18	591.082	0.203654
粒子 19	573.84	0.909929
粒子 20	939,177	0.76918
Gbest	192.907	0.203654
平均值	560.994	0.634178
標準差	196.626	0.209261
疊代次數	247	



(a)



(b)



(d)

圖 5.6: 速度正規化第三次實驗 (a) 初始分佈 (b) 疊代 100 次後 (c) 疊代 200 次後 (d)

實驗停止

平均疊代次數：199.67

5.1.3 討論與分析

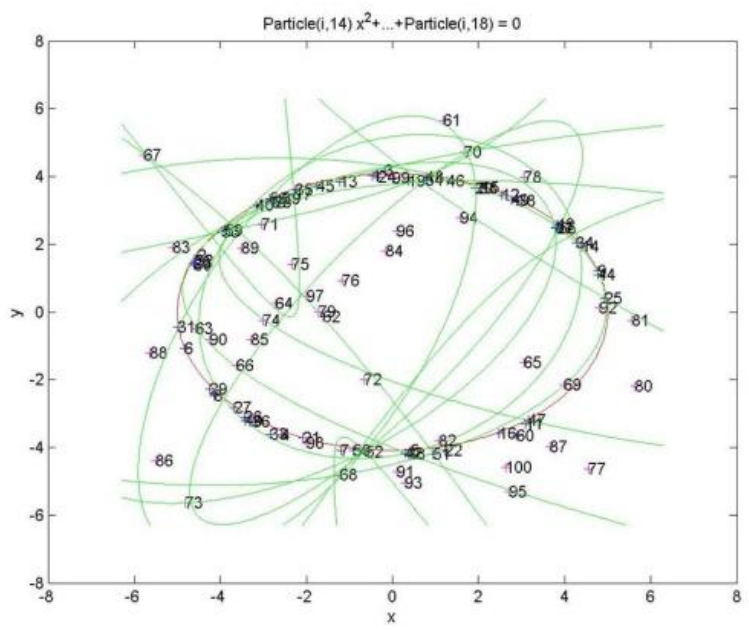
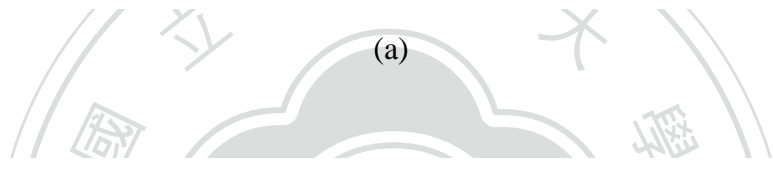
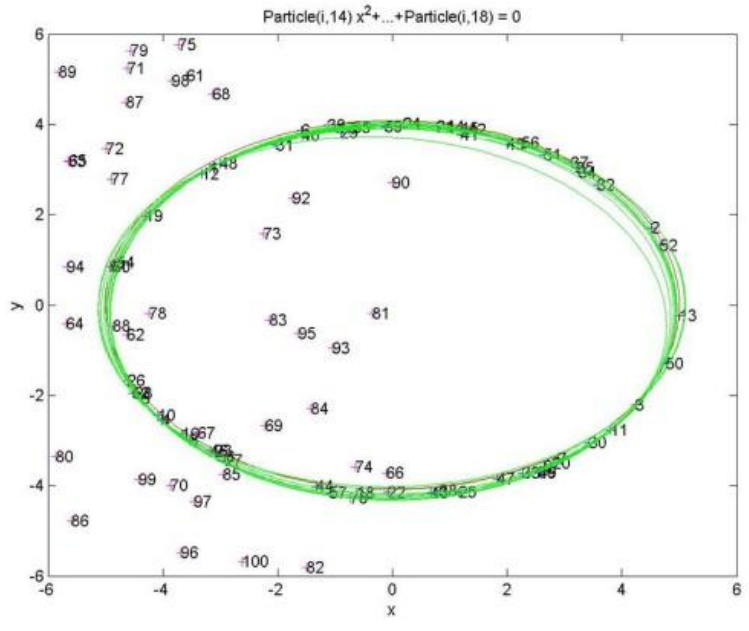
在 5.1.1 和 5.1.2 兩節中我們將兩種實驗結果用表格記錄和繪製到平面圖上，從數據中我們可以得到，有做速度正規化的疊代次數會比速度未正規化來的少，也驗證了我們在 4.1.1 節時曾提到當兩個維度具有相依性時，是需要做速度正規化的，另外可以從表 3~8 和圖 5.1~5.6 中可以發現，一開始粒子的分佈是很分散的，所以初始 fitness 值計算出來都相當大，直到經過一定的疊代次數後，發現各組粒子會越來越逼近我們假定的最佳點座標，直到達到停止條件後，我們可以看到所有粒子在表格中停止 fitness 值皆會小於 0.1，這是我們經過多次實驗後制訂出的門檻值，在此條件下，所有粒子的(x,y)座標會和最佳點座標誤差會在 0.05pixel 之內。

5.2 圓錐近似中粒子群最佳化實驗結果

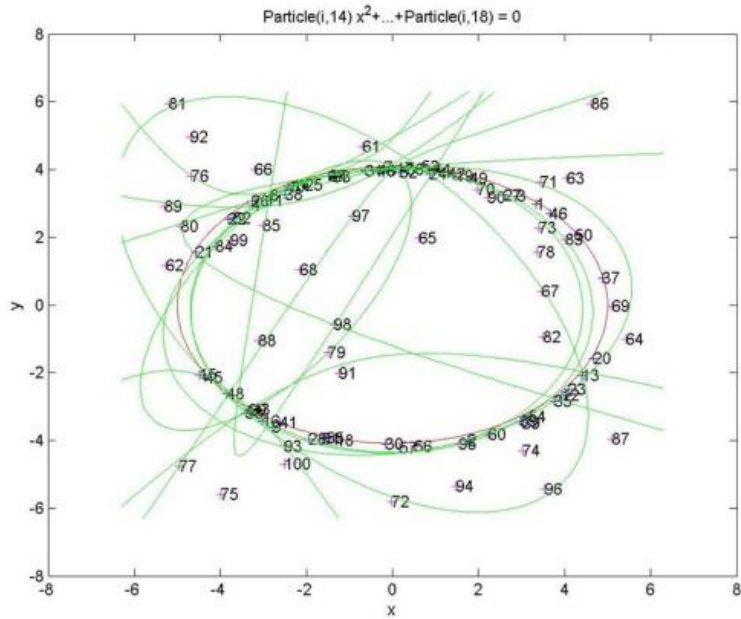
在本節中，我們將列出圓錐近似中的幾次實驗結果，我們嘗試了三種不同的停止條件，後面章節的圖中，x、y 軸接表示座標，點旁邊的數字表示點編號。

5.2.1 限制固定疊代次數

本節實驗我們是採取疊代次數當作停止條件，但因為隨機分佈的狀況不同，利用固定的實驗次數是相當不保險的做法，圖 5.7 是我們進行 5000 次疊代後的實驗結果。



(b)



(c)

圖 5.7：疊代次數為停止條件(a)第一次(b)第二次(c)第三次 實驗結果

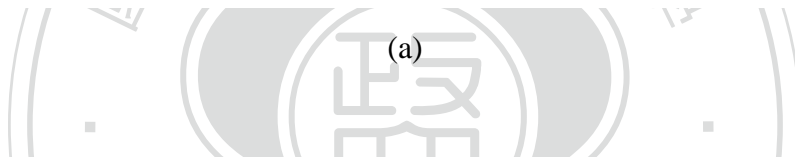
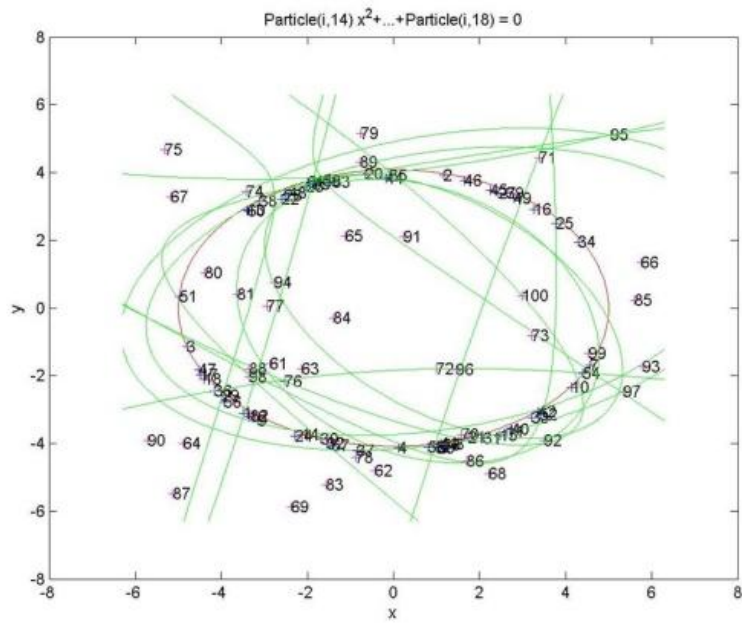
5.2.2 對 Gbest 收斂

利用收斂當作停止條件其實有一部分是參考平面中的 fitness 值制訂，我們計算收斂值公式是利用各組粒子中的六個點編號和 Gbest 的六個點編號計算出來，式 5.1 是我們計算的公式。

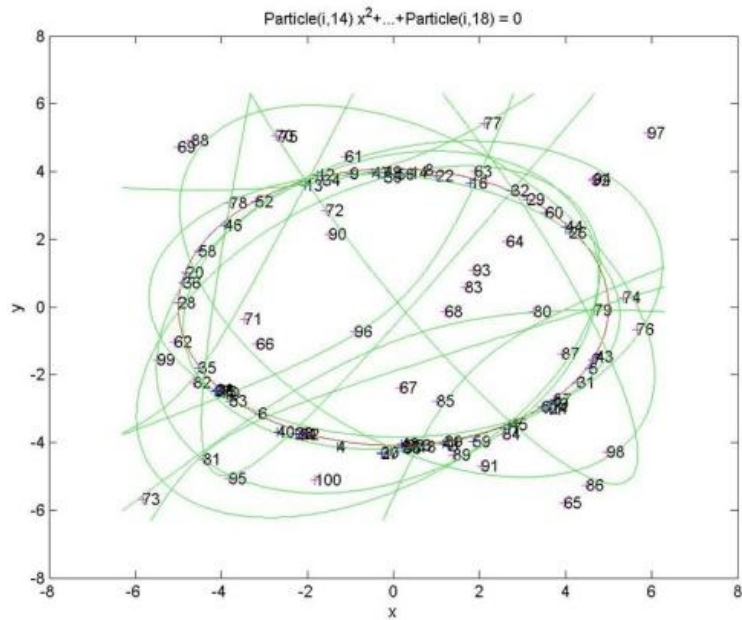
$$convergence = \sum_{N=1}^6 \sqrt{(Particle_{i_N} - Gbest_{i_N})^2} \quad (5.1)$$

其中 $Particle_i$ 表示第幾個粒子個體， N 表示第幾個點編號， c 表示我們計算出的收斂值，接著我們將收斂值 $convergence$ 值調整後，有以下的幾次實驗。

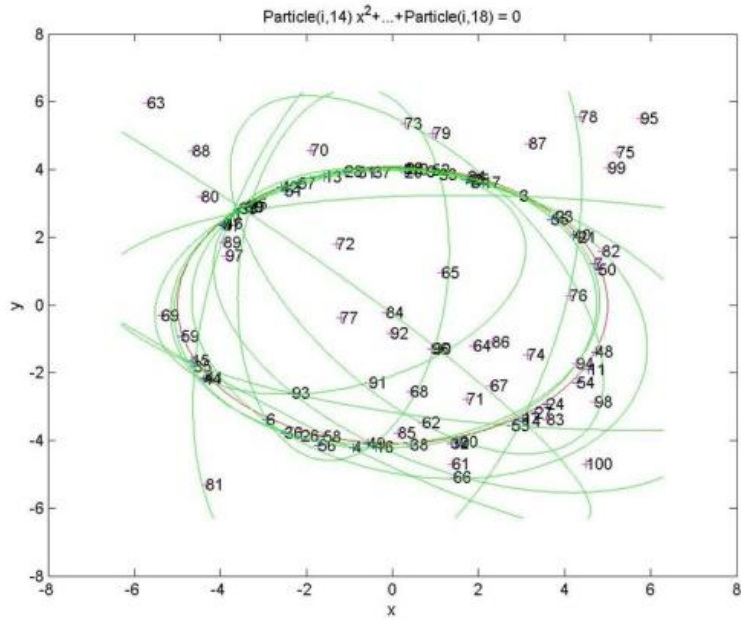
convergence 值<12.25



(a)



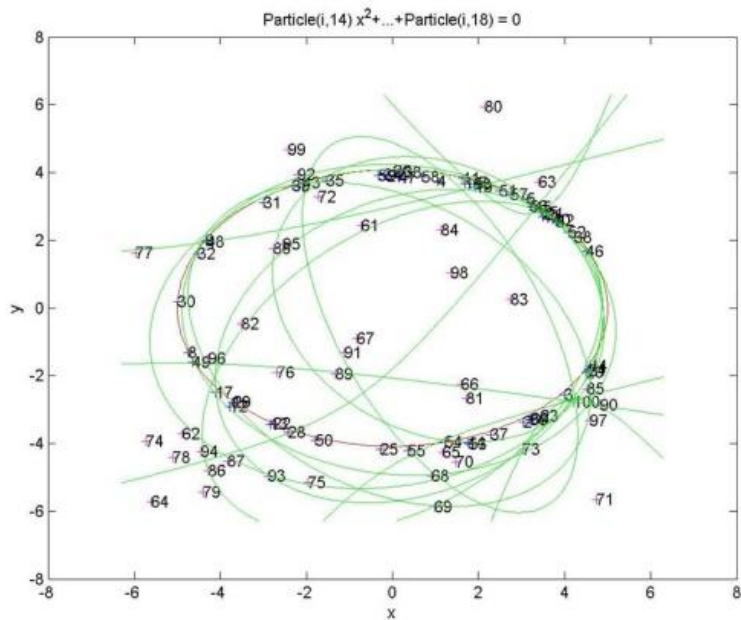
(b)



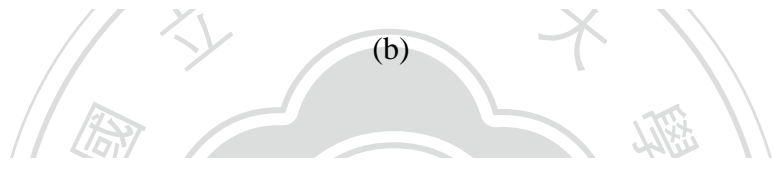
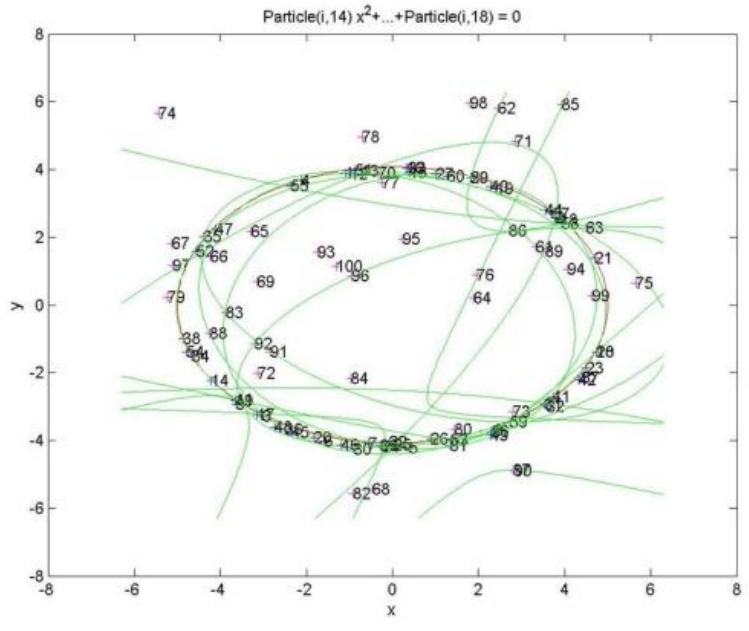
(c)

圖 5.8 : convergence 值 < 12.25 為停止條件 (a) 第一次 (b) 第二次 (c) 第三次 實驗結果

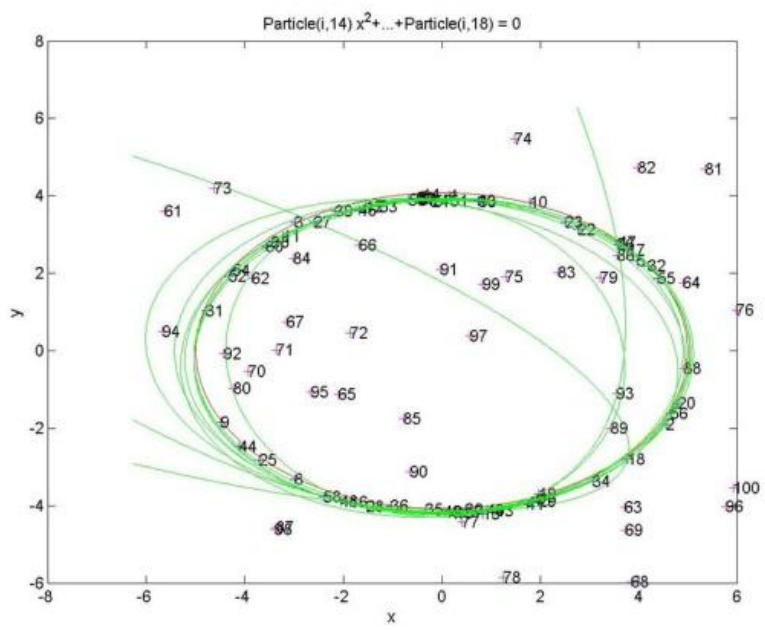
convergence 值 < 9.8



(a)



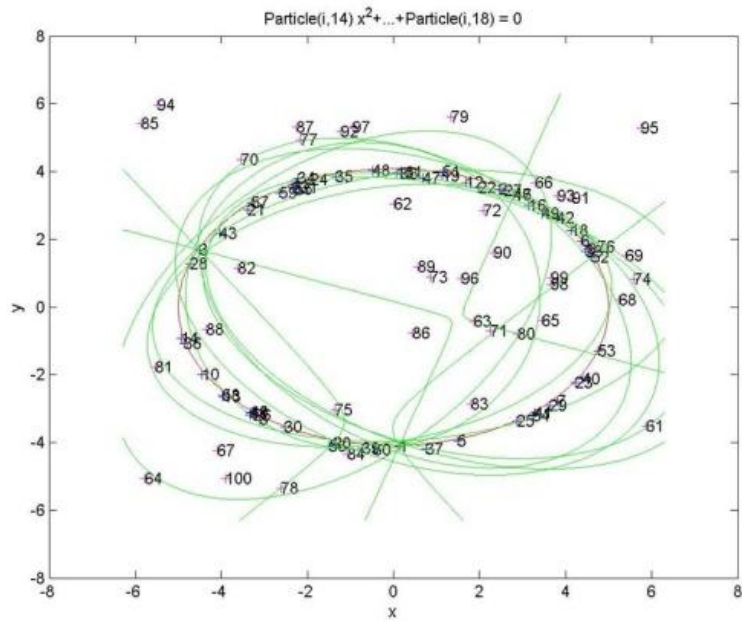
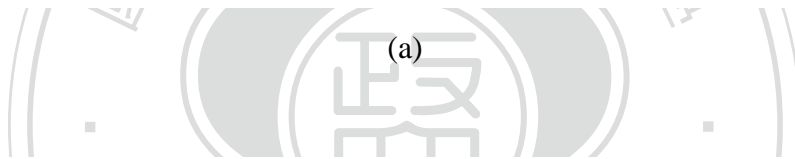
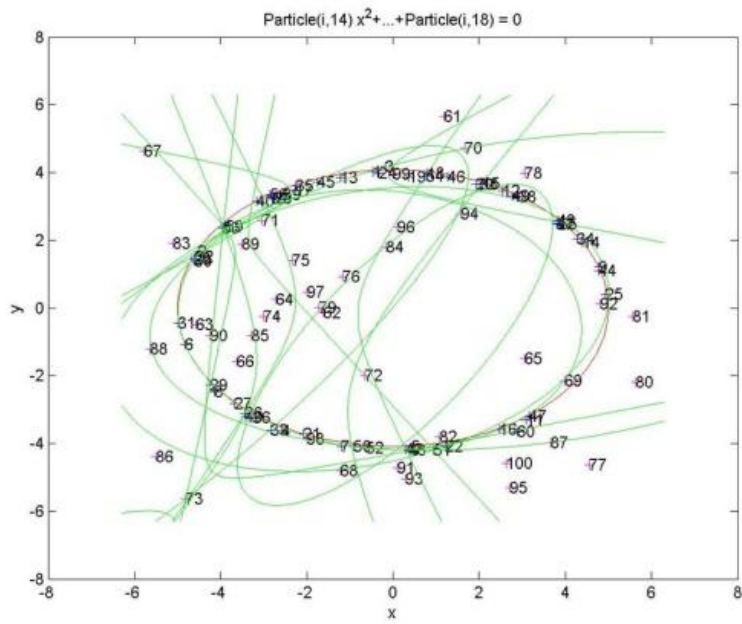
(b)



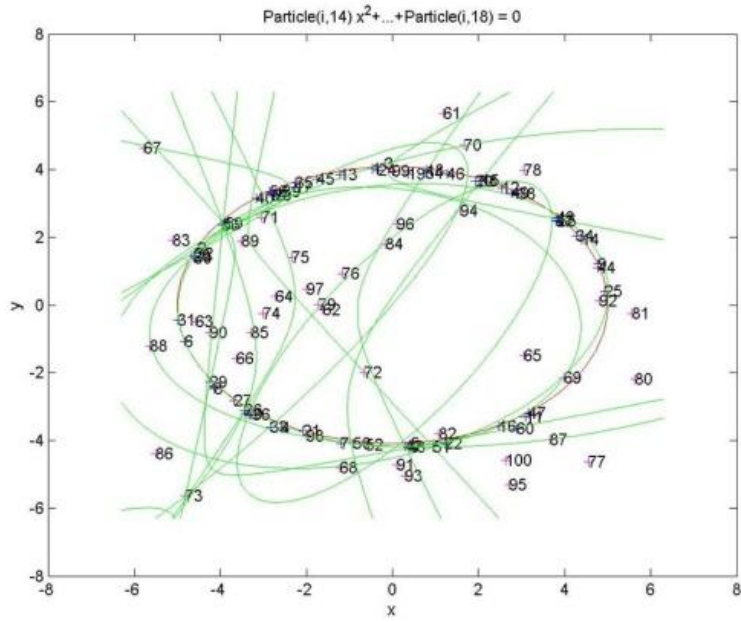
(c)

圖 5.9 : convergence 值 < 9.8 為停止條件 (a) 第一次 (b) 第二次 (c) 第三次 實驗結果

convergence 值 <7.35



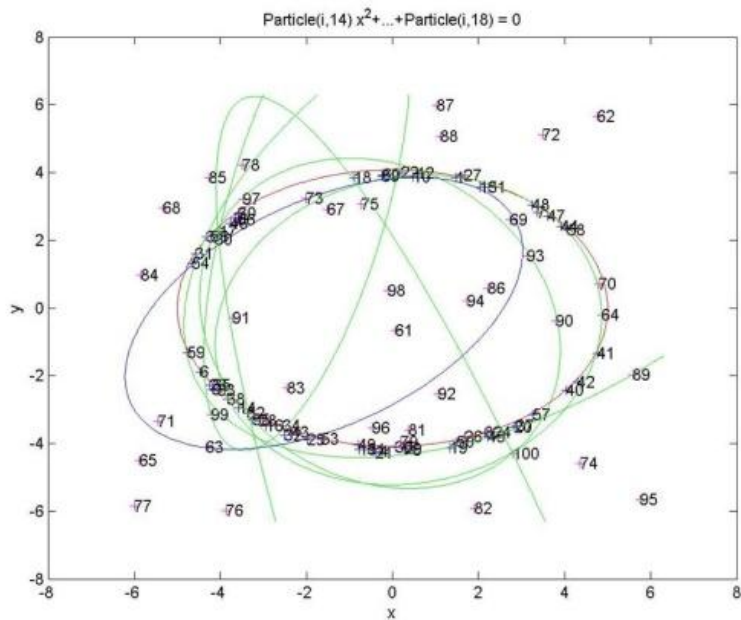
(b)



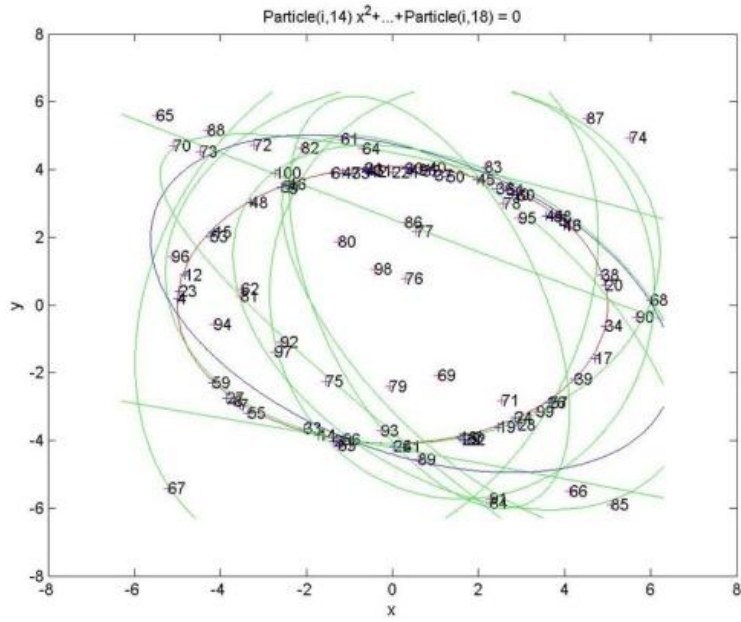
(c)

圖 5.10 : convergence 值 < 7.35 為停止條件 (a) 第一次 (b) 第二次 (c) 第三次 實驗結果

convergence 值 < 6.12



(a)



(b)

圖 5.11 : convergence 值 < 6.12 為停止條件 (a) 第一次 (b) 第二次 實驗結果

convergence 值 < 4.9

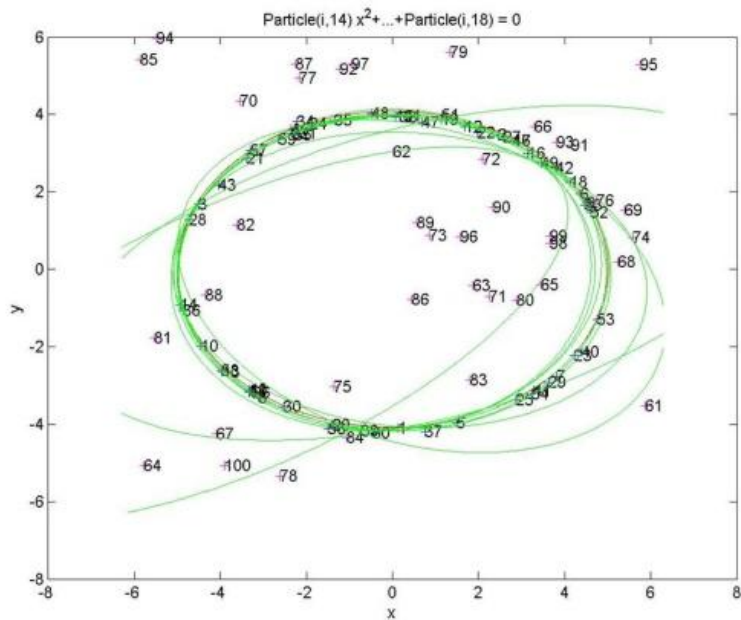


圖 5.12 : convergence 值 < 4.9 為停止條件 第一次 實驗結果

5.2.3 限制 fitness 值和對 Gbest 收斂

在本節實驗停止條件是限制 Gbest 的 fitness 值小於一定的門檻值，再利用收斂當作我們的停止條件，因為在 5.2.2 節中只用收斂到 Gbest 這組粒子個體當作停止條件，會發現很有可能還沒有近似到一定的準確率時就已達停止條件，因此我們再修改了停止條件，目前先進行了幾次實驗如下。

Gbest_{fitness} < 0.1 和 convergence 值 < 12.25

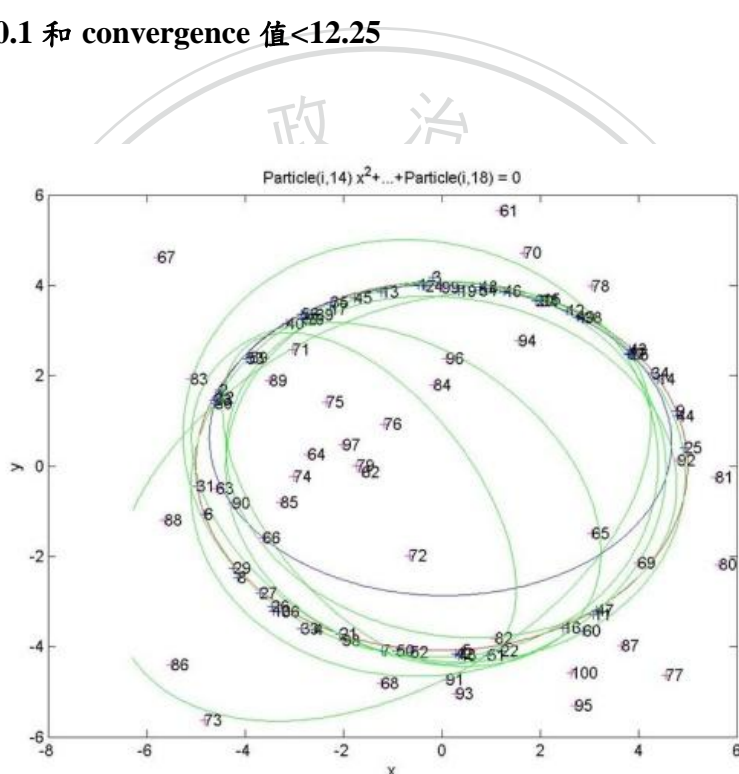


圖 5.12：Gbest_{fitness} < 0.1 和 convergence 值 < 12.25 為停止條件第一次實驗結果

5.2.4 討論與分析

在 5.2.1、5.2.2 和 5.2.3 節中我們將三個不同實驗停止條件放在圓錐近似的實驗中，得到的結果也不盡然相同，但目前仍有再改善的空間，目前幾個可能的問題如下，在 5.2.1 節中，只單純的限制疊代次數是風險較大的做法，因為我們沒

有過去經驗告訴我們說粒子群最佳化演算法可以在一定的計算次數中找尋到最佳解，5.2.2 節內收斂算是合理的做法，但是還有一些細節的問題是我們待釐清的，因為透過收斂的步驟，本意上是往 Gbest 收斂了，離他的距離會隨著 c 值越小而越靠近，但是這邊的靠近指的是點編號的靠近，由於我們在圓錐近似中粒子的組成是採取六個點編號的方式，因此光是編號的靠近，是否真的可以當作是一個評估的標準，還是有待驗證的地方，5.2.3 小節內結合限制 fitness 值和收斂的做法，限制 fitness 值是一個不錯的方式，因為先前章節有提到，當 fitness 值制訂的好，fitness 值便是很好的一個參考依據，所以我們未來實驗依然可以從限制 fitness 值出發，搭配收斂也是不錯的方法，雖說光是編號靠近尚無法驗證其好壞，但是當 fitness 值收斂到一定的值，可能會變得很有意義，並且可以提升實驗結果，這是未來可以研究的部分。

5.3 以粒子群最佳化估測基礎矩陣實驗結果

在本節我們將列出估測基礎矩陣的兩組實驗結果，第一組資料是由 3DMAX 模擬場景，由兩台相機拍攝多視角影像，由於場景是我們模擬配置，因此可以取得 ground truth 資料，第二組資料是由 Christoph[20]提供多視角影像和光達點雲資料，也可以由投影矩陣計算出多視角相片中的 ground truth 資料幫助計算誤差，我們將透過三種方法比較實驗結果，分別是 PSO、LMedS_PSO、LMedS，最後做討論與分析。

5.3.1 第一組實驗(3DMAX)資料

我們用 3DMAX 建模並且假設兩台相機拍攝兩張相片(圖 5.13)，圖 5.14 顯示我們的實驗場景，相片解析度為 800x600，籃子大小為 80x60，共有 100 個，初

始找尋對應點時採用 SIFT 演算法，共可以找到 2494 組對應點，而用來驗證的 ground truth 資料，則有 1306 組對應點。



圖 5.13：3DMAX 實驗相片

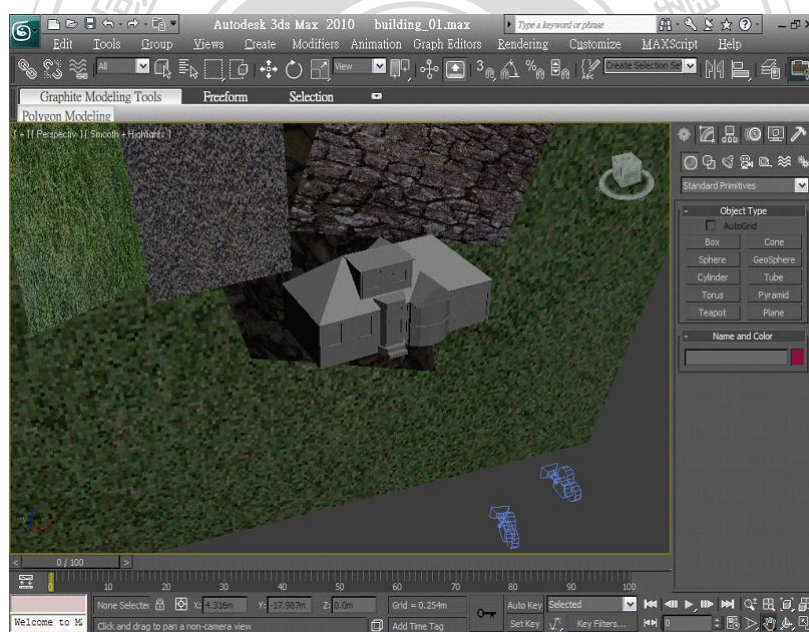


圖 5.14：3DMAX 拍攝場景

我們採用較接近的疊代次數來比較實驗結果，PSO 實驗採用 10 組隨機取樣的粒子個體經過疊代計算 117 次，共進行 1170 次疊代，而 LMedS_PSO 方法先計算成功取樣機率為 80%，正確資料為 50%的一組結果，套用隨機取樣公式需

要 412 次，接著隨機取樣 9 組粒子個體後，結合 LMedS 的一組結果，之後 10 組再進行疊代 76 次，共需要 1172 疊代次數，而 LMedS 的方法直接將成功取樣機率設為 99%，正確資料為 50%，需要疊代 1177 次。

完成上述的步驟後，我們稱為進行一組實驗，在本研究中，將使用 1000 組資料的平均誤差當作實驗結果，如表 9、10。

表 9：3DMAX 場景估測基礎矩陣結果

方法	1000 組平均誤差	1000 組標準差	1000 組需要時間
PSO	0.077 像素	0.0547 像素	5 小時
LMedS_PSO	0.086 像素	0.0548 像素	18 小時
LMedS	0.097 像素	0.0548 像素	41 小時

表 10：3DMAX 場景估測基礎矩陣標準差分佈

方法	小於平均標準差	小於 0~1 倍	小於 1~2 倍
PSO	678 個	667 個	11 個
LMedS_PSO	632 個	629 個	3 個
LMedS	581 個	577 個	4 個

從表 9、10 中我們可以發現，PSO 的平均誤差和花費的時間皆是最少的，而小於平均標準差的數量也是最多，接著我們統計三種方法的一千組結果，計算總平均和標準差，繪製出直方圖和累積的折線圖，如圖 5.15、5.16。

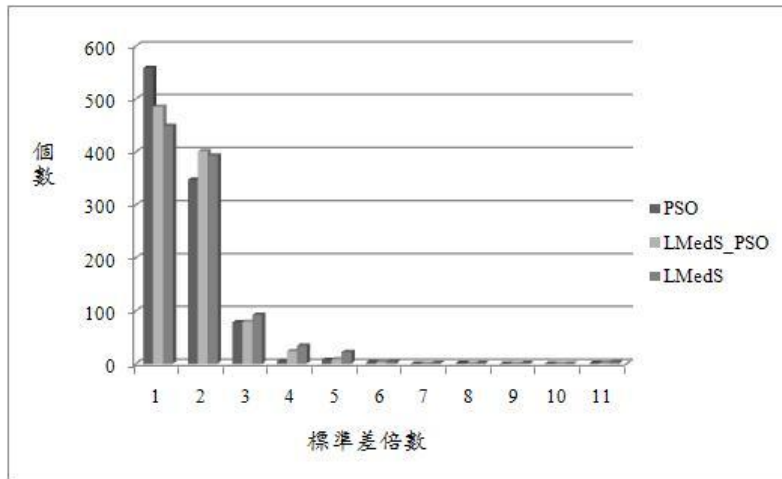


圖 5.15：3D MAX 實驗直方圖比較

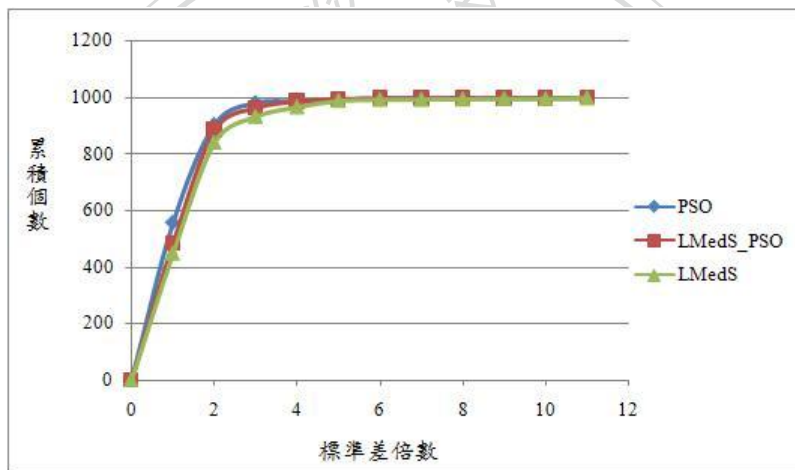


圖 5.16：3D MAX 實驗折線圖比較

從圖表中我們看出，PSO 的方法在接近的疊代次數中，時間成本和平均誤差值皆是表現最好的，接著下一節我們將列出第二組實驗結果。

5.3.2 第二組實驗(LIDAR)資料

第一組 LIDAR 資料

光達資料是一種雷射成像系統，可以將三維模型中的座標記錄下來，相機對物體拍攝相片後，只要透過投影矩陣，即可計算出三維物體在相片上的座標，我們可以透過這個步驟獲得 ground truth 資料，圖 5.17 是實驗用的兩張相片。



圖 5.17：LIDAR 第一組實驗相片

相片解析度為 3072×2048 ，籃子大小為 256×256 ，共有 96 個，初始找尋對應點時採用 SIFT 演算法，由於電腦記憶體不足以處理較大的相片解析度，因此我們在找尋對應點時，會先將相片解析度縮小一半為 1536×1024 ，找到後的對應點座標在乘上兩倍，便可以獲得對應點資訊，共可以找到 1336 組對應點，而用來驗證的 ground truth 資料，則有 2121 組對應點。

採用 5.3.1 小節的三種方法和接近的疊代次數，我們可以有以下的實驗結果，如表 11、12。

表 11：LIDAR 場景估測基礎矩陣第一組結果

方法	1000 組平均誤差	1000 組標準差	1000 組需要時間
PSO	0.237 像素	0.1056 像素	8 小時
LMedS_PSO	0.247 像素	0.1056 像素	29 小時
LMedS	0.278 像素	0.1057 像素	70 小時

表 12：LIDAR 場景估測基礎矩陣第一組結果標準差分佈

方法	小於平均標準差	小於 0~1 倍	小於 1~2 倍
PSO	673 個	633 個	40 個
LMedS_PSO	642 個	599 個	43 個
LMedS	564 個	541 個	23 個

從表 11、12 可以發現，當我們資料量越多，花費的時間成本也相對增多，不過 PSO 的方法仍是誤差最小，且時間花費最少的，接著我們統計三種方法的一千組結果，計算總平均、標準差，繪製出直方圖和累積的折線圖，如圖 5.18、5.19。

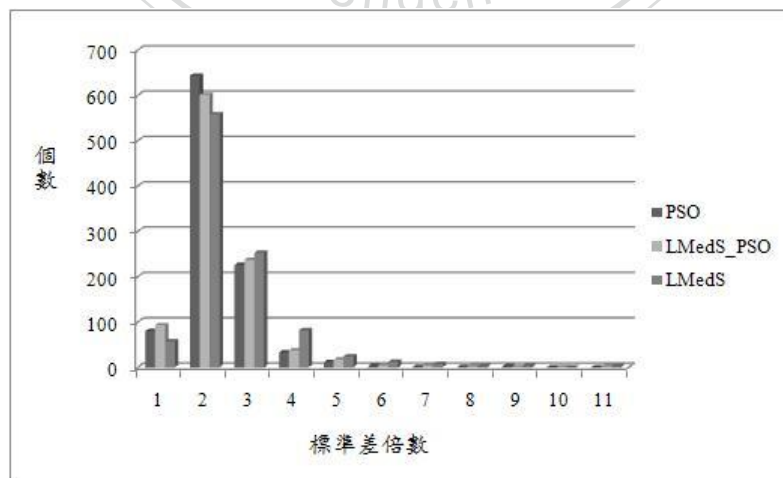


圖 5.18：LIDAR 第一組實驗直方圖比較

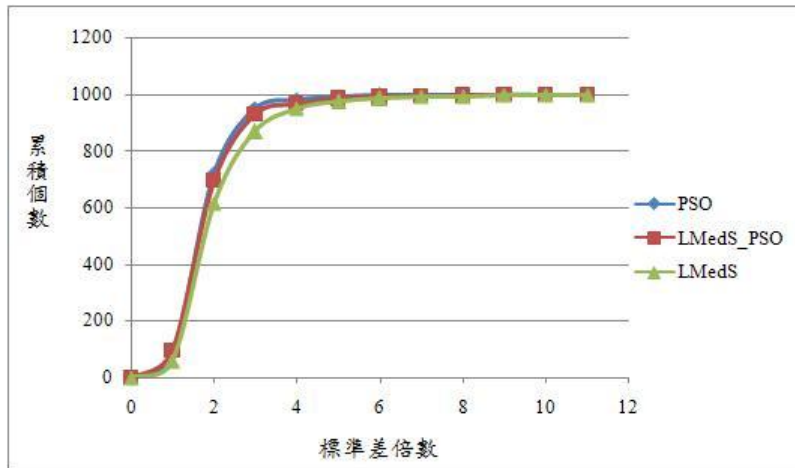


圖 5.19：LIDAR 第一組實驗折線圖比較

第二組 LIDAR 資料

第二組資料的兩張照片如圖 5.20。



圖 5.20：LIDAR 第二組實驗相片

相片解析度為 3072x2048，籃子大小為 256x256，共有 96 個，共可以找到 1157 組對應點，而用來驗證的 ground truth 資料，則有 1733 組對應點，實驗結果如表 13、14。

表 13：LIDAR 場景估測基礎矩陣第二組結果

方法	1000 組平均誤差	1000 組標準差	1000 組需要時間
PSO	0.385 像素	0.1683 像素	6 小時
LMedS_PSO	0.386 像素	0.1684 像素	23 小時
LMedS	0.431 像素	0.1685 像素	56 小時

表 14：LIDAR 場景估測基礎矩陣第二組結果標準差分佈

方法	小於平均標準差	小於 0~1 倍	小於 1~2 倍
PSO	638 個	520 個	118 個
LMedS_PSO	627 個	492 個	135 個
LMedS	515 個	420 個	95 個

PSO 的方法仍是誤差最小，且時間花費最少的，接著我們統計三種方法的一千組結果，計算總平均、標準差，繪製出直方圖和累積的折線圖，如圖 5.21、5.22。

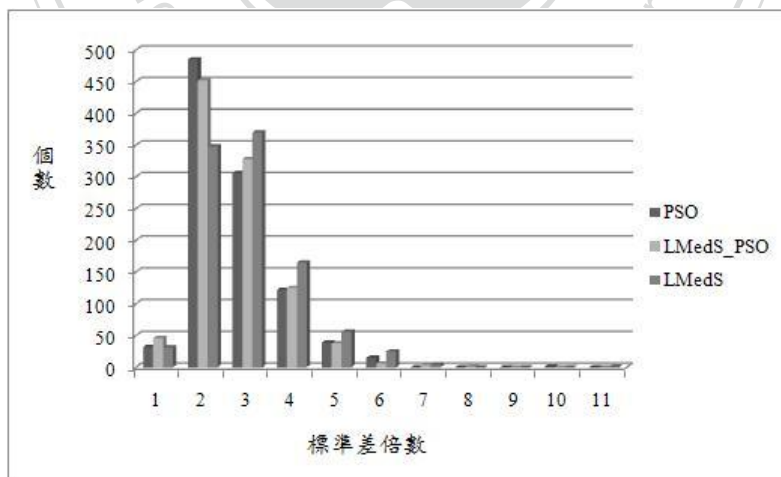


圖 5.21：LIDAR 第二組實驗直方圖比較

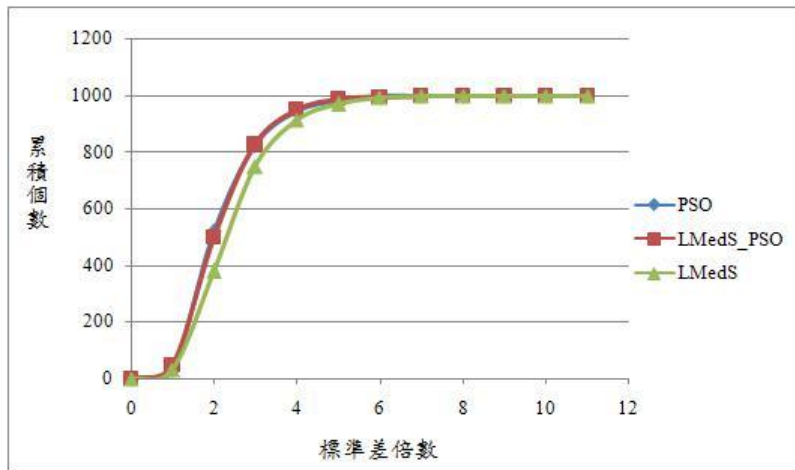


圖 5.22：LIDAR 第二組實驗折線圖比較

5.3.3 討論與分析

從 5.3.1 和 5.3.2 共三組實驗結果看出來，三種方法用接近的疊代計算次數比較，使用 PSO 的方法比過去使用強健式方法 LMedS 快上不少，平均誤差值也較小，而我們另外結合兩種方法的 LMedS_PSO 則是在時間成本、平均誤差值皆介於中間，這是可以預期的，第一組 3DMAX 資料因為是我們用電腦模擬，包括相機的參數皆有的情況下，算出來的誤差值較小，而第二組 LIDAR 資料因為在找對應點時遇到記憶體不足的問題，調整後已經造成些誤差，再來利用投影矩陣投影回相片上時，需要做正規化，這部份也產生了誤差，因此整體比較下來，誤差比 3DMAX 資料大，但是就 LIDAR 資料來說，我們提出的方法是有效率且誤差值較小的，有了這些資訊後，適合未來繼續進行相關研究。

第六章

結論

6.1 結論

在本研究中，我們成功將粒子群最佳化演算法應用在估測基礎矩陣上，在不同的物體模型中，透過多視角影像，重覆進行疊代次數相同的實驗，粒子群最佳化演算法估測基礎矩陣所需的時間，僅需要最小平方中值法的八分之一，同時粒子群最佳化演算法估測出來的基礎矩陣平均誤差值也優於最小平方中值法。

我們一開始先進行了兩個基礎實驗，了解粒子群最佳化演算法的特性後，進而轉換至估測基礎矩陣的問題，由於基礎矩陣的解題複雜度較高，同時也有多種限制，因此我們透過分群與隨機挑選以避免選取共平面之點，確保後續粒子群最佳化演算法中，不會因為含有較差的取樣，影響到實驗結果。

我們利用 SIFT 找尋大量對應點，可以有效的節省找尋對應點的時間，同時可以針對多種相機拍攝情況處理，例如透過多視角相機拍攝出來的多視角照片，在本研究的實驗中，不論是 3DMAX 資料或是 LIDAR 資料，相機的擺設都會有平移和旋轉，透過 SIFT 方法，可以快速找到大量的對應點，讓我們順利進行後續的實驗。

6.2 未來工作

未來我們可以考慮在演算法初始化步驟中加入挑選點的方法，田口方法是有機會加入並且結合的，由於粒子群最佳化演算法是會根據過去經驗不斷更新疊代的演算法，因此第一次挑選扮演著重要的角色，若初始選點可以分散並且具有規則，或許有機會在搜尋空間中做比較完整的搜尋，雖然我們已經利用分群與隨機取樣的方法儘可能將初始化的對應點分散和避免共平面問題，但仍是採用隨機取樣，並沒有一套機制分配初始化的點，這部份是我們未來可以更加探討的。

有了準確的基礎矩陣後，後續可以應用在找尋密集的對應點，過去有許多學者長期研究在找尋精確度高的對應點，本研究中曾提到，在幾何關係下，多視角影像的對應點是可以由準確的基礎矩陣計算出座標。

接著過濾誤差較大的雜訊點後，便可以建出三維模型空間，有了精確的三維模型空間後，進一步可以加入材質數貼，就形成了完整的三維模型，有機會可以應用在地理資訊系統(Geographic Information System)、數位典藏(Digital archive)、電影製作(filmmaking)。

參考文獻

- [1]. 尹邦嚴，柳依旻，江元傑，黃冠哲，陳映良，“粒子族群最佳化的視覺化及開發工具”，2005 銘傳大學國際學術研討會論文集，桃園，民國 94 年 11 月。
- [2]. 李宜靜，蔡賢亮，“以粒子群體最佳化為基礎之中文文字細線化演算法”，2009 資訊技術應用及管理研討會論文集，高雄，民國 98 年 6 月。
- [3]. 鄒慎財，“強健式估測基礎矩陣之研究”，碩士論文，華梵大學資訊管理系，台北，民國 94 年 1 月。
- [4]. 廖怡儂，“應用於電腦視覺強健式估測之研究”，碩士論文，華梵大學資訊管理系，台北，民國 94 年 5 月。
- [5]. 蔡介元，鍾佩潔，“建立一個以 PSO 求解多點最佳路徑的行動地理資訊系統”，2007 臺灣作業研究學會學術研討會論文集，花蓮，民國 96 年 10 月。
- [6]. 蔡賢亮，溫千慧，鄭永雄，“基於粒子族群最佳化之不完全資料處理”，2006 電子商務與數位生活研討會論文集，台北，民國 95 年 2 月。
- [7]. 賴岳宏，“利用演化式計算做最佳化之研究”，碩士論文，華梵大學資訊管理系，台北，民國 94 年 5 月。
- [8]. C. Tang, Y. Wu, and Y. Lai, “Fundamental Matrix Estimation using Evolutionary Algorithms with Multi-Objective Functions”, *Journal of Information Science and Engineering*, 2008, pp.785-800.
- [9]. F. Frenzel, “Genetic algorithms,” *Potentials, IEEE*, vol.12, no.3, pp.21-24, Oct 1993.
- [10]. L. George. 2004. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving (5th Edition)*. Pearson Addison Wesley.
- [11]. M. Lhuillier, Q. Long, “Match propagation for image-based modeling and rendering,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol.24, no.8, pp. 1140- 1146, Aug 2002.

- [12].G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- [13].J. Kennedy, R. Eberhart, "Particle swarm optimization," *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol.4, no., pp.1942-1948 vol.4, Nov/Dec 1995.
- [14].J. Phillip, "Taguchi Techniques for Quality Engineering", McGraw-Hill Professional, 2nd edition, 1995.
- [15].A. Ratnaweera, K. Halgamuge, C.Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *Evolutionary Computation, IEEE Transactions on*, vol.8, no.3, pp. 240- 255, June 2004.
- [16].H. Richard, Z. Andrew. 2003. *Multiple View Geometry in Computer Vision* (2 ed.). Cambridge University Press, New York, NY, USA.
- [17].H. Thomas, S. Clifford, L. Ronald, and E. Charles. 2001. *Introduction to Algorithms* (2nd ed.). McGraw-Hill Higher Education.
- [18].A. Xavier, S. Joaquim, "Overall view regarding fundamental matrix estimation" *Image and Vision Computing*, vol. 21, no. 2, pp. 205-220, 2003
- [19].Z. Zhengyou, "Parameter Estimation Technique: A Tutorial with Application to Conic Fitting", *Image and Vision Computing*, vol. 15, no. 1, pp. 59-76, 1997.
- [20].Ground Truth data, <http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html>