

國立交通大學

機械工程學系

碩士論文

數位條紋投射法量測
胸腔外形三維點資料之處理與應用

Processing and application of 3D data points by digital fringe
projection method for the ribcage shape measurement

研究生：鍾宜達

指導教授：秦繼華 教授

中華民國九十六年六月



數位條紋投射法量測

胸腔外形三維點資料之處理與應用

Processing and application of 3D data points by digital fringe
projection method for the ribcage shape measurement

研究生：鍾宜達

Student : Yi-Ta Chung

指導教授：秦繼華

Advisor : Dr. Jih-Hua Chin

國立交通大學

機械工程學系



Submitted to Department of Mechanical Engineering

College of Engineering

National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of Master

in

Engineering

June 2007

Hsinchu, Taiwan, Republic of China

中華民國九十六年六月

國立交通大學

博碩士論文全文電子檔著作權授權書

(提供授權人裝訂於紙本論文書名頁之次頁用)

本授權書所授權之學位論文，為本人於國立交通大學機械工程系所機械設計組，95學年度第2學期取得碩士學位之論文。

論文題目：數位條紋投射法量測胸腔外形三維點資料之處理與應用

指導教授：秦繼華

同意 不同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：

本校及台灣聯合大學系統區域網路 中華民國96年7月31日公開

校外網際網路 中華民國96年7月31日公開

授權人：鍾宜達

親筆簽名：鍾宜達

中華民國 96 年 6 月 25 日

國立交通大學

博碩士紙本論文著作權授權書

(提供授權人裝訂於全文電子檔授權書之次頁用)

本授權書所授權之學位論文，為本人於國立交通大學機械工程系所機械設計組，95學年度第2學期取得碩士學位之論文。

論文題目：數位條紋投射法量測胸腔外形三維點資料之處理與應用

指導教授：秦繼華

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：_____，請將論文延至____年____月____日再公開。

授權人：鍾宜達

親筆簽名：鍾宜達

中華民國 96 年 6 月 25 日

國家圖書館博碩士論文電子檔案上網授權書

ID:GT009414519

本授權書所授權之論文為授權人在國立交通大學工學院機械工程系所機械設計組 95 學年度第 2 學期取得碩士學位之論文。

論文題目：數位條紋投射法量測胸腔外形三維點資料之處理與應用

指導教授：秦繼華

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

※ 讀者基於非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：鍾宜達

親筆簽名：鍾宜達



民國 96 年 6 月 25 日

1. 本授權書請以黑筆撰寫，並列印二份，其中一份影印裝訂於附錄三之二(博碩士紙本論文著作權授權書)之次頁；另一份於辦理離校時繳交給系所助理，由圖書館彙總寄交國家圖書館。

國立交通大學

論文口試委員會審定書

本校 機械工程學系 碩士班 鍾宜達 君

所提論文:

數位條紋投射法

量測胸腔外形三維點資料之處理與應用

合於碩士資格水準、業經本委員會評審認可。

口試委員:



楊文和

鍾文祥

秦繼華

指導教授:

秦繼華

系主任:

周以文

中華民國九十六年六月二十三日

數位條紋投射法


量測胸腔外形三維點資料之處理與應用

學生：鍾宜達

指導教授：秦繼華 教授

國立交通大學機械工程學系碩士班

摘 要



本論文研究漏斗胸病患胸腔外形量測三維點資料之處理技巧及其應用，三維點資料乃由兩台不同視角的相機分別利用數位條紋投射法取得。首先發展結合兩組不同視角點資料之自動對位與整合方法。在兩組點資料之對位時，經由指定兩組不同視角點資料之對應相同特徵點，本研究發展一套有效的演算法以找出最佳之座標系統平移及旋轉矩陣以結合兩組點資料。在資料點之整合時，發展平均混成、線性混成及二次混成等三種方法將兩組點資料整合成一組沒有縫隙及沒有多餘部分之平滑表面。接著發展一套軟體系統，可以進行兩組點資料之顯示、對位、整合與後處理等工作，此軟體系統使用 Visual C++ 程式語言發展而成。最後，將本文發展之系統應用於漏斗胸病人之胸腔外形量測，在置入固定桿件手術之前與之後分別量測病人之胸腔外形，經由比較手術前後之胸腔外形，可評估手術之效果。

關鍵字：數位條紋投射法、三維點資料整合、三維點資料對位、表面重建、漏斗胸

Processing and application of 3D data points by digital fringe projection method for the ribcage shape measurement

Student: Yi-Ta Chung

Advisor: Dr. Jih-Hua Chin

Department of Mechanical Engineering
National Chiao Tung University

ABSTRACT

This thesis studies the processing techniques and applications of 3D data points in the shape measurement of the ribcage for patients with funnel chest. The 3D data points are acquired by the digital fringe projection method from two different camera views. First, automatic registration and integration methods for combining two point sets of different views are developed. In registration of data points, points of the same feature in two different data sets are specified, and a robust algorithm is developed for finding the best coordinate translation and rotation matrices that combine two point sets. In integrating data points, average blending, linear blending, and quadratic blending methods are developed such that two data sets are integrated into one seamless and non-redundant smooth surface. Then, a software system is developed for data points display, registration, integration, and post processing for the two point sets. The software system is developed using Visual C++ language. Finally, the developed system is applied in the ribcage shape measurement for patients with funnel chest. The ribcage shapes of a patient before and after the surgery for placing a fixation bar inside the chest are measured. The effect of the surgery can be evaluated by comparing the ribcage shapes before and after the surgery.

Keywords: digital fringe projection method, 3D points registration, 3D points integration, surface reconstruction, funnel chest

誌 謝

回首過去兩年，研究的過程中遇到了各式各樣的困難，幸好身邊總是有著許多有能力給予我幫助並且也非常樂意給我幫助的人。首先，我要特別感謝的是我的指導教授秦繼華老師，從大三認識秦老師以來，經歷了四年的時間，非常感謝秦老師的諄諄指導與教誨；秦老師不只指導了我的專業學術知識、培養我獨立解決問題的能力，更在人生的方向與意義上，也讓我有了不少的瞭解與領悟。感謝長庚醫院兒童部主任兼任交通大學機械工程學系醫學工程授課教授張北葉醫師，沒有您的熱心指導並且給予醫工合作的機會，就不會有本論文的產生。並感謝口試委員 XXX、XXX 與 XXX 的細心指正與建議，另本論文更臻完善。也要感謝實驗室的學長權原、俊凱，同窗俊傑、永祥、威凱，學弟偉源、奕承、孟諭、彥男，謝謝你們在我奶奶重病與過逝的那段期間，給予我最大的幫助與慰問，讓我的研究與學校課業沒有因為家中有些負擔而中斷，得以用兩年的時間順利從實驗室畢業，與你們一同討論、切磋課業與研究的時光，我將會永遠放入最美好的記憶裡。感謝父親不分晝夜的從旁協助。感謝母親、姊姊與弟弟在我研究遇到瓶頸情緒低落時，總是給予我最多的包容與鼓勵。感謝台灣大學機械工程學系 CAD 實驗室的每一位成員，你們熱心不分彼此的協助，讓我論文得以從落後的進度中迎頭趕上。

最後感謝每一位曾經給我打氣的死黨、同學、朋友們，沒有你們一路的陪伴與支持，就沒有堅持到最後的我。在此感謝所有幫助、提攜過我的人，真的謝謝你們，以本論文獻給你們。

目 錄

	頁次
中文摘要	i
英文摘要	ii
誌謝	iii
目錄	iv
圖目錄	vii
一、 緒論	1
1.1 研究背景	1
1.2 文獻回顧	6
1.2.1 三維點資料對位回顧	7
1.2.2 三維點資料整合回顧	13
1.3 研究動機與目的	18
1.4 研究方法	23
二、 三維點資料處理之基本理論	26
2.1 使用最小平方法尋找最適合的平面	26
2.2 尋找一個最合適於多點三維點資料之拋物面	28
2.3 座標轉換矩陣	30
2.3.1 平移	30
2.3.2 旋轉	31
2.3.3 映像	32
三、 三維點資料之處理與軟體系統發展	34

3.1	讀取三維點資料	35
3.2	選取特徵點	38
3.3	進行三點對位	42
3.4	重新佈點	48
3.5	混成	50
3.6	網格化鋪面	54
3.7	剖面圖	58
四、	醫學工程應用之胸腔三維取像案例	60
4.1	漏斗胸男童病患手術前	63
4.2	漏斗胸男童病患手術矯正後	67
4.3	手術前後之比較	72
4.4	漏斗胸青春期女性病患手術矯正後	77
五、	結論與未來展望	82
5.1	結論	82
5.2	未來展望	83
參考文獻		84
附錄 A	程式 Color Fringe Projection System 使用步驟	88
附錄 B	程式 Data registration and integration for 3D shape measurement system 主程式碼	92
B.1	程式安裝說明	92
B.1.1	基本需求	92
B.1.2	ObjectARX 2002 安裝	92

B. 1. 3	環境設定	93
B. 2	程式碼檔案介紹	96
B. 2. 1	核心程式碼	96
B. 2. 2	系統設定檔	97
B. 2. 3	外部引用程式碼	97
B. 3	系統主程式介面與程式碼說明	98
B. 3. 1	主程式介面	98
B. 3. 2	對位區介面說明	99
B. 3. 3	整合區介面說明	101
B. 3. 4	其他功能說明	102
B. 4	系統函式說明	103
B. 4. 1	主程式附屬函式說明	103
B. 4. 2	數值方法函式說明	105
B. 4. 3	矩陣運算函式說明	106
B. 5	程式操作說明	106
B. 6	列出程式主程式碼	111
B. 6. 1	三點對位運算主程式碼	112
B. 6. 2	單點對位運算主程式碼	112
B. 6. 3	整合運算主程式碼	113
B. 6. 4	左側視角三維點資料網格化鋪面主程式碼	113
B. 6. 5	右側視角三維點資料網格化鋪面主程式碼	114
B. 6. 6	完成對位三維點資料網格化鋪面主程式碼	114
B. 6. 7	擷取剖面主程式碼	114
作者簡歷		116

圖目錄

	頁次
圖 1.1 不同圖案之結構光示意圖.....	3
圖 1.2 數位條紋投射法量測方式示意圖.....	5
圖 1.3 條紋結構光.....	5
圖 1.4 特徵對位示意圖.....	8
圖 1.5 校正樣本將兩個座標系統轉換單一座標系統示意圖.....	8
圖 1.6 Chua 自訂之幾何特徵示意圖.....	9
圖 1.7 各種特徵點的分析圖.....	10
圖 1.8 ICP 演算法示意圖	11
圖 1.9 校正框架.....	12
圖 1.10 校正塊.....	12
圖 1.11 十字記號圖樣.....	12
圖 1.12 線圖樣.....	12
圖 1.13 Sitnik 矩形圖樣.....	13
圖 1.14 Chen 矩形圖樣.....	13
圖 1.15 兩組點資料設定示意圖.....	14
圖 1.16 兩組點資料之掃描線與交界點示意圖.....	15
圖 1.17 進行混成的資料點示意圖.....	16
圖 1.18 線性混成與二次混成之結果比較.....	17
圖 1.19 整合示意圖.....	17
圖 1.20 嬰兒頭型異常資料照片.....	18
圖 1.21 頭盔製作過程 打模.....	19
圖 1.22 頭盔製作過程 打模.....	19
圖 1.23 頭盔製作過程 灌製石膏模.....	20

圖 1.24	頭盔製作過程 灌製石膏模	20
圖 1.25	頭盔製作過程 矯正帽完成	21
圖 1.26	漏斗胸兒童病患微創手術案例	22
圖 1.27	漏斗胸兒童病患傳統手術案例	23
圖 1.28	本論文之研究方法流程圖	24
圖 3.1	Color Fringe Projection System 硬體架設示意圖	34
圖 3.2	左邊相機原始圖像	36
圖 3.3	右邊相機原始圖像	36
圖 3.4	左邊相機條紋投射圖像	36
圖 3.5	右邊相機條紋投射圖像	36
圖 3.6	左邊視角相機取得之三維點資料	37
圖 3.7	左邊視角相機取得之三維點資料	37
圖 3.8	右邊視角相機取得之三維點資料	37
圖 3.9	右邊視角相機取得之三維點資料	37
圖 3.10	選取特徵點附近之任何一點	38
圖 3.11	距離選取點最近之 1400 點資料	39
圖 3.12	距離選取點最近之 1400 點資料	39
圖 3.13	找出特徵點資料集合之特徵平面法向量	40
圖 3.14	找出特徵點資料集合之特徵平面法向量	40
圖 3.15	距離選取點最近之 200 點資料	41
圖 3.16	距離選取點最近之 200 點資料	41
圖 3.17	求得拋物面，並尋找其頂點	42
圖 3.18	求得拋物面，並尋找其頂點	42
圖 3.19	對位結果	47
圖 3.20	對位結果-臉部特寫	47

圖 3.21	選取重新佈點之矩型範圍.....	48
圖 3.22	左邊視角三維點資料經重新佈點後之三維點點狀圖.....	49
圖 3.23	右邊視角三維點資料經重新佈點後之三維點點狀圖.....	50
圖 3.24	平均混成結果點圖.....	51
圖 3.25	線性混成結果點圖.....	52
圖 3.26	二次混成結果點圖.....	53
圖 3.27	左邊視角三維點資料之三維鋪面圖.....	55
圖 3.28	右邊視角三維點資料之三維鋪面圖.....	55
圖 3.29	平均混成結果鋪面圖.....	56
圖 3.30	線性混成結果鋪面圖.....	57
圖 3.31	二次混成結果鋪面圖.....	57
圖 3.32	點選得水平剖面圖之水平位置.....	58
圖 3.33	計算並繪出剖面圖.....	59
圖 3.34	剖面圖刪除網格化鋪面.....	59
圖 3.35	剖面圖結果呈現.....	59
圖 4.1	胸腔外形三維點資料之處理與應用軟體運算流程圖.....	61
圖 4.2	取得特徵點之運算流程圖.....	62
圖 4.3	拋物面所需點資料預設值 200 輸入改成 50.....	63
圖 4.4	依序點選對位依據特徵點.....	63
圖 4.5	特徵點之特徵方向.....	64
圖 4.6	特徵點之特徵方向.....	64
圖 4.7	合適之拋物面.....	64
圖 4.8	依序點選對位依據特徵點.....	64
圖 4.9	完成三點對位.....	65
圖 4.10	完成三點對位.....	65

圖 4.11	左側相機取得之三維點資料.....	66
圖 4.12	左側相機取得之三維點資料.....	66
圖 4.13	右側相機取得之三維點資料.....	66
圖 4.14	右側相機取得之三維點資料.....	66
圖 4.15	手術前胸腔三維外形展示.....	67
圖 4.16	手術前胸腔三維外形展示.....	67
圖 4.17	左視角條紋影像.....	68
圖 4.18	右視角條紋影像.....	68
圖 4.19	拋物面所需點資料預設值 200 輸入改成 100.....	68
圖 4.20	依序點選對位依據特徵.....	69
圖 4.21	依序點選對位依據特徵.....	69
圖 4.22	完成三點對位.....	70
圖 4.23	完成三點對位.....	70
圖 4.24	左側相機取得之三維點資料.....	71
圖 4.25	左側相機取得之三維點資料.....	71
圖 4.26	右側相機取得之三維點資料.....	71
圖 4.27	右側相機取得之三維點資料.....	71
圖 4.28	手術後胸腔三維外形展示.....	72
圖 4.29	手術後胸腔三維外形展示.....	72
圖 4.30	微創手術將手術器具穿過胸腔.....	73
圖 4.31	將固定桿件置入胸腔.....	73
圖 4.32	固定桿件旋轉將胸腔稱起.....	74
圖 4.33	乳頭位置圖.....	75
圖 4.34	手術前乳頭高度剖面圖.....	75
圖 4.35	乳頭位置圖.....	75

圖 4. 36	手術後乳頭高度剖面圖.....	75
圖 4. 37	胸部與腹腔交界位置圖.....	76
圖 4. 38	手術前胸部與腹腔交界高度剖面圖.....	76
圖 4. 39	胸部與腹腔交界位置圖.....	76
圖 4. 40	手術後胸部與腹腔交界高度剖面圖.....	76
圖 4. 41	乳頭與胸部、腹腔交界中點位置圖.....	77
圖 4. 42	手術前乳頭與胸部、腹腔交界中點高度剖面圖.....	77
圖 4. 43	乳頭與胸部、腹腔交界中點位置圖.....	77
圖 4. 44	手術後乳頭與胸部、腹腔交界中點高度剖面圖.....	77
圖 4. 45	手術治療後之條紋影像.....	78
圖 4. 46	左側視角依序點選對位依據特徵.....	79
圖 4. 47	右側視角依序點選對位依據特徵.....	79
圖 4. 48	完成三點對位.....	79
圖 4. 49	完成三點對位-下方視角.....	79
圖 4. 50	左側相機取得之三維點資料-左側視角.....	80
圖 4. 51	左側相機取得之三維點資料-右側視角.....	80
圖 4. 52	右側相機取得之三維點資料-左側視角.....	80
圖 4. 53	右側相機取得之三維點資料-右側視角.....	80
圖 4. 54	使用線性混成選項.....	81
圖 4. 55	女性病患手術後胸腔三維外形展示-左側視角.....	81
圖 4. 56	女性病患手術後胸腔三維外形展示-右側視角.....	81
圖 A. 1	漏斗胸兒童病患(左).....	88
圖 A. 2	漏斗胸兒童病患(右).....	88
圖 A. 3	擇條紋影像檔儲存資料夾位置.....	89
圖 A. 4	左邊相機所取得之圖檔(一).....	89

圖 A. 5	左邊相機所取得之圖檔 (二)	90
圖 A. 6	左邊相機所取得之圖檔 (三)	90
圖 A. 7	左邊相機所取得之圖檔 (四)	90
圖 A. 8	左邊相機所取得之圖檔 (五)	90
圖 A. 9	右邊相機所取得之圖檔 (一)	90
圖 A. 10	右邊相機所取得之圖檔 (二)	90
圖 A. 11	右邊相機所取得之圖檔 (三)	91
圖 A. 12	右邊相機所取得之圖檔 (四)	91
圖 A. 13	右邊相機所取得之圖檔 (五)	91
圖 A. 14	Load Image	91
圖 A. 15	Select Region	91
圖 A. 16	完成畫面	91
圖 B. 1	設定工作目錄至 D:\a2002\	93
圖 B. 2	選取『 ObjectARX 2000/2000i/2002 Add-in 』	94
圖 B. 3	選取『 Include files 』	95
圖 B. 4	選取『 Library files 』	95
圖 B. 5	設定 ObjectARX 2002 控制檔輸出路徑	96
圖 B. 6	主程式介面	98
圖 B. 7	對位區介面圖	99
圖 B. 8	整合區介面圖	101
圖 B. 9	其他功能介面圖	102

一、緒論

1.1 研究背景

隨著工業的不斷發展與進步，三維外形量測技術也隨之日益重要與成熟。由於三維外形量測技術的成熟發展，現今早已被廣泛使用於各種不同的應用，例如：逆向工程(RE, Reverse Engineering)、工業檢測、電腦視覺，多媒體動畫…等。尤其以逆向工程對於三維外形量測的依賴性為最高。傳統的產品開發過程：先由工程師構思產品需求，再來確定產品的功能與機構、規格，進而設計每一個單一零組件；設計完成後，經過分析，設計修正，直到設計結果合乎產品需求後，接著進行製造、組裝零組件、性能測試等步驟。以上的開發過程被稱為正向工程。逆向工程的程序並非從產品需求開始，而是從產品本身開始，由產品本體去回溯設計資料，並且接著進行分析與改良。所以取得產品三維外形便成為逆向工程相當重要的一項環節。一旦取得產品三維外形資料，便可使用電腦輔助設計系統(CAD, Computer Aided Design)建立實體模型，在系統中進行外形之編修後，再將修改後的三維外形資料傳至電腦輔助製造系統(CAM, Computer Aided Manufacturing)或快速成型機(RP, Rapid Prototyping)進行製造。相較於正向工程來說，逆向工程有生產速度快，成本低…等優點，不過精確度較低為其缺點。

三維外形量測的方法與理論非常多元化，以量測方式為區分主要可區分為『接觸式三維外形量測』與『非接觸式三維外形量測』兩種。

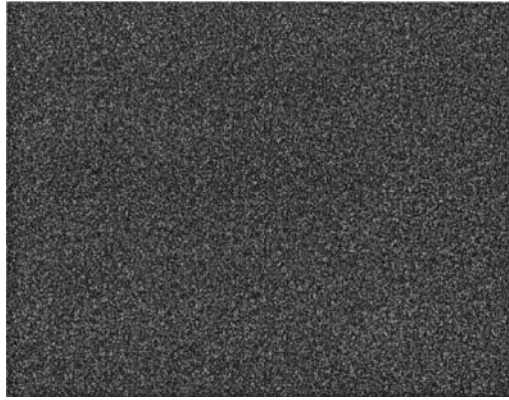
接觸式三維外形量測主要是由探針去接觸待量測物體的表面，進而根據探針的三維位子經由多次的碰觸與紀錄，經過各軸座標位置與實際量測空間座標位置的轉換，就可取得待測物體之三維點資料，經過處理即可還原待測物體的輪廓。接觸式量測的優點為精確性及可靠度高；但缺點為量測方式較費時，量測時有損害物件表面的可能，而且若待測物體為生物（如：人體）或表面過於柔軟的物件

時，接觸式量測的使用會更加的受限。相較之下，非接觸式量測利用各種光學原理作為量測基礎，不僅不需要與待測物體接觸，可以避免很多原先接觸式量測的限制，還隨著科技的日趨成熟，量測的時間也可以大幅度的縮減；由於以上關係，非接觸式三維外形量測的技術逐漸取代先前接觸式量測法的系統，成為三維外形量測系統發展的主流趨勢。

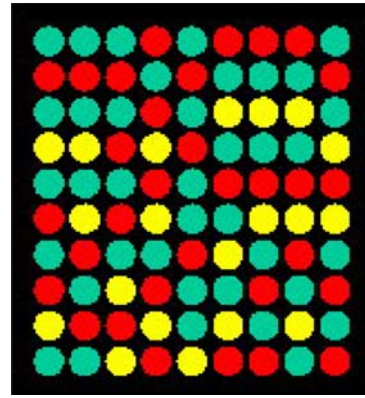
常見的非接觸式三維外形量測方法分別為以下四種：結構光法(structure lighting method)、雷射掃描法(laser scanner method)、立體視角法(stereoscopy method)與體積交集法(volume intersection method)。

1. 結構光法採取主動式光源，光源由儀器本身投射至物體表面，再以影像感測器取得投射於物體表面上的光之分佈影像，最後經由演算法重建出三維點資料。此三維量測法對於投射光源的架設位置需經過精確的調整，其優點為可以避免尋找特徵點對應(feature point correspondence)的問題且精確度較高，不過不同圖案的結構光(圖 1.1)所能應用的待測物體大小會有所不同[1]。

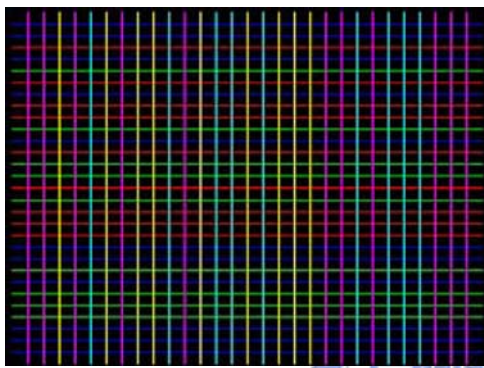




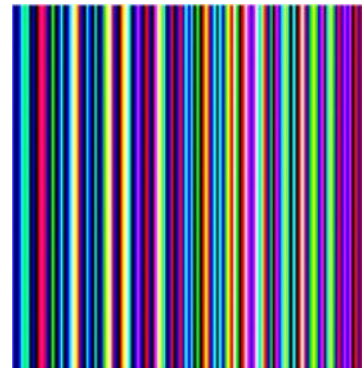
(a)Random speckle pattern



(b)Colored dots pattern



(c)Color grid pattern



(d)De Bruijn pattern

圖 1.1 不同圖案之結構光示意圖[1][2]

2. 雷射掃描法的原理，為投射點雷射光或者線雷射光於待測物體表面，利用影像感測器擷取因輪廓起伏而變形的雷射光形變影像，形變影像經過運算進而求得雷射光點或雷射光線上之輪廓分佈。藉由改變功率來調整雷射點或線的大小，便可以提高量測精度，為此量測法的最大優點；但量測速度緩慢，且容易受到環境光線的干擾；且此量測法應用於取得人體之三維外形圖像時，其必須投射雷射光於人體之流程會有對人體傷害之疑慮，為其應用於人體三維外形取像之最大缺點。
3. 立體視角法基於使用被動式光源之原理，擷取物體表面在自然光線下的反射光，不需投射主動式的光源為此三維外形量測方法最大的優點；但必須解決特徵點對應的問題；此外當物體為均勻表面時，量測法將無法

使用。

4. 體積交集法以物體投影至二維平面的輪廓線來運算出該物體之三維形狀，不需尋找對應之特徵點，只要物體表面的點出現在某個物體輪廓上，即可利用投射線的交集去計算實體輪廓。主要缺點為物品凹陷表面有時無法表現於輪廓線上，容易有量測死角。

而條紋投射法 (fringe projection method) 為依據相位干涉法 (phase-shifting interferometry) 所發展而成的三維非接觸量測方法。相位干涉法主要是量測待測物體表面光之波前相位 (wavefront phase)，再比較其與參考光線之相位差，利用感測器記錄的光強度 (intensity) 資料，計算出對應相位值；具有高精確度以及處理過程簡單的優點。

與其他主流非接觸三維量測方法最大的差異在於條紋投射法並沒有固定的硬體配置，對於各種不同類型的待測物體有著很高的適應性。投射亮度值為週期性的彩色條紋結構光於待測物體表面，利用影像感測器擷取隨著輪廓高低起伏而有著對應條紋變形的條紋影像，分別讀取每一個位置其條紋光強度變化並且計算其相位分佈 (phase distribution)，藉由以上步驟求得待測物體之三維外形。

條紋投射法之相位分佈，通常是使用相位移法 (phase-shifting method) 或者傅立葉轉換分析 (Fourier transformation analysis) 兩種方法擇其一求得，再從相位移分佈重建出輪廓高度。傳統的條紋投射法利用白色光光源置於光柵幻燈片 (grating slide) 後方，並且藉由配合平移機構的帶動，產生具週期性與相位移 (phase shift) 之弦波強度 (sinusoidal intensity) 條紋結構光，主要限制為弦波光柵製作較為困難，條紋特性不容易改變；或者是光源互相干涉產生條紋結構光，並改變光源相位使條紋產生相位移，但卻有光源不穩定及位移量不精確等缺點。為了克服上述問題，便發展出了「數位條紋投射法 (digital fringe projection)」 (圖 1.2)，簡稱 DFP [3]。

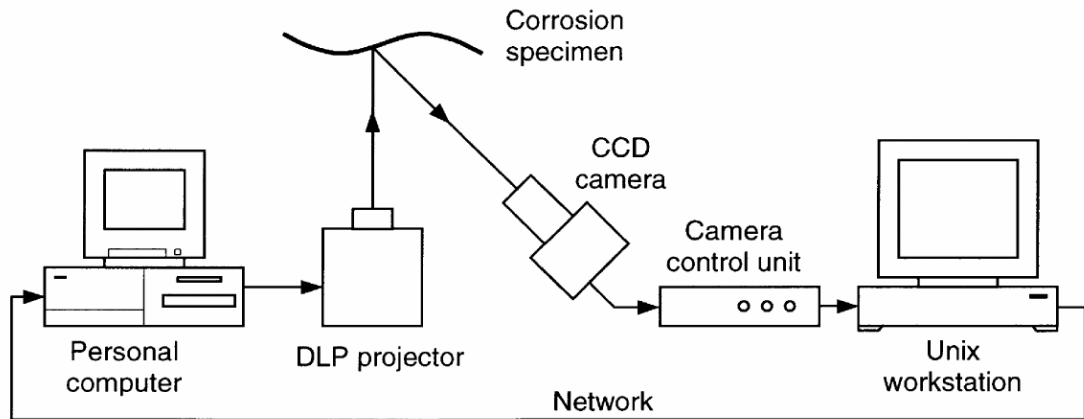


圖 1.2 數位條紋投射法量測方式示意圖[3]

數位條紋投射法主要是以投影機(projector)作為主動式的投射光源，例如 LCD(liquid crystal display)或 DLP (digital light processing)投影機等，必須先於電腦中設計好條紋結構光的圖案、週期及相位移量（圖 1.3），即可透過投影機將條紋圖案投射於待測物表面。此方式不僅方便改變條紋特性，投射出的條紋光亦擁有快速且精準之相位移，且條紋光的強度對於周遭環境的變化也較不敏感。

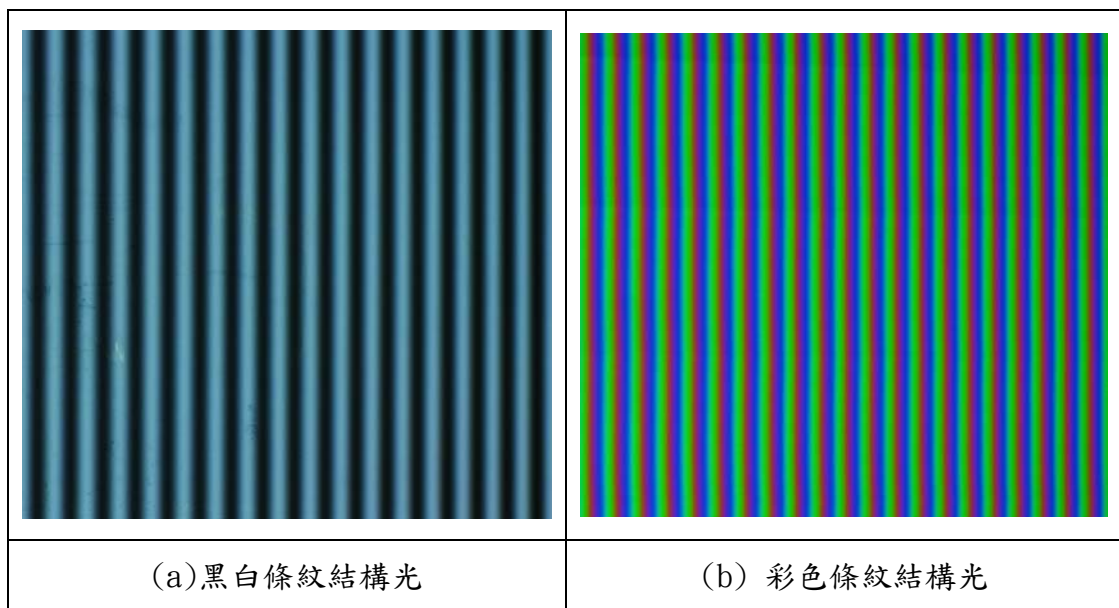


圖 1.3 條紋結構光

1.2 文獻回顧

架設一台數位相機作為影像感測器即可使用數位條紋投射法來進行三維外形非接觸式量測，取得該物體單一視角之三維形狀。不過使用單一相機來進行三維外形量測的動作，所取得之三維點資料有以下缺點：

1. 對於大型或者形狀不甚圓滑、規則之待測物體會因為遮蔽緣故，無法取得該視角完整正確之三維外形資料。
2. 單一相機能取得的三維外形點資料角度有所限制(必定小於一百八十度，依取相技術成熟與否有所不同)，而且誤差往往與取相角度的偏差程度有著相當的關係，在取相角度範圍的邊緣常常會因為計算誤差放大或者資料失真而出現三維外形點資料破裂的情況。

故使用條紋投射法進行三維外形量測時，本研究架設一台以上的數位相機，以不同的固定視角，來進行數位條紋投射法的三維外形量測以確保物體的三維外形點資料最接近待測物體的實際原貌。

以多台相機進行條紋投射法擷取條紋結構光的影像後，分別將各個固定視角的影像進行三維外形點資料的重建，如此會產生多筆的三維點資料；為了將多組的三維點資料經過分析、運算、整合成一組點資料，必須進行三維點資料對位(Data registration)與三維點資料整合(Data integration)。

各組的掃描點資料都有各自的座標系統，故需要統整各組點資料之座標系統，將各組點資料都置於單一的座標系統內；此種步驟稱之為三維點資料對位。對位完畢之後，將重疊的資料去除，並將接合處進行接合混成，重新排序成一組點資料，此步驟稱為三維點資料整合。以下將針對『三維點資料對位』、以及『三維點資料整合』兩部分來整理並回顧相關文獻。

1.2.1 三維點資料對位回顧

要進行點資料對位與點資料整合，多組三維點資料之間必須符合與滿足以下三個條件：

1. 各組三維點資料，必須來自不同的視角。最容易達成此目的的方法為將待測物件置於一旋轉盤上，藉由旋轉運動讓單一固定相機取得不同視角之影像。
2. 各組三維點資料所處的相對與絕對座標必須被定義對位（be registered）。如此方可去尋找其不同座標系統間的轉換矩陣。
3. 經過整合後之三維點資料檔，必須是一個沒有多餘部分的模型（non-redundant model）。

三維點資料對位的方式主要可分為以點資料的特徵(Characteristic)進行對位與以校正樣本(Calibration sample)進行對位，針對不同的情況使用不同的對位方法。點資料的特徵對位示意圖，如圖 1.4 所示。校正樣本對位的示意圖，如圖 1.5 所示。

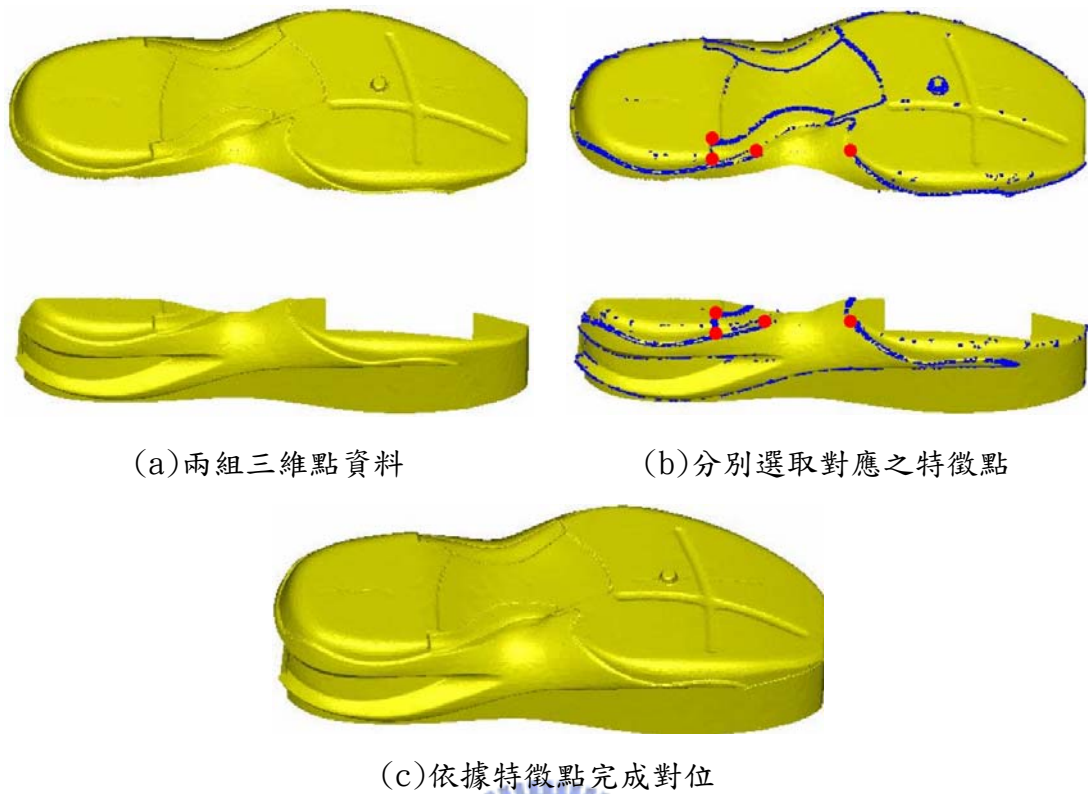


圖 1.4 特徵對位示意圖[19]

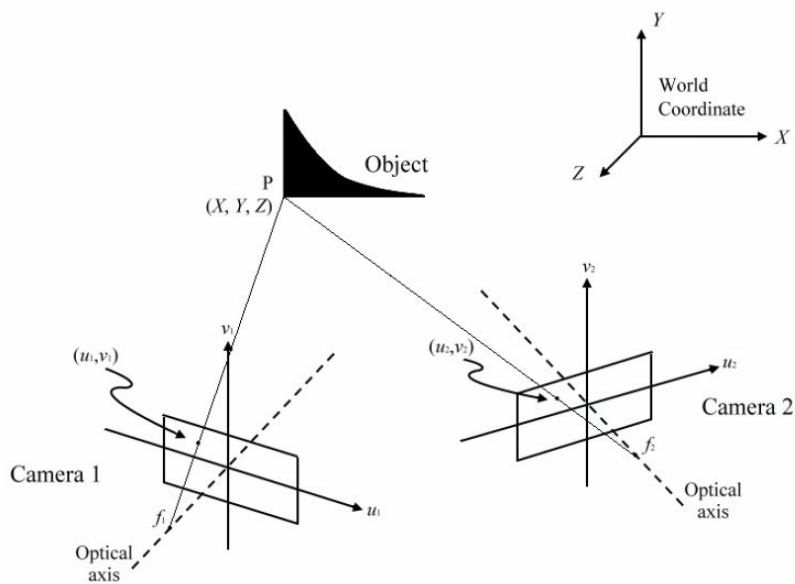
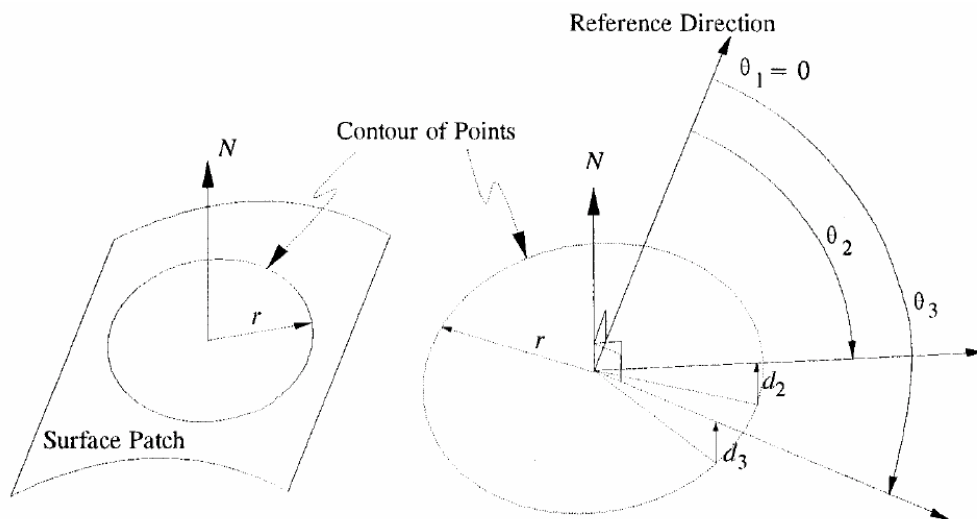


圖 1.5 校正樣本將兩個座標系統轉換單一座標系統示意圖[20]

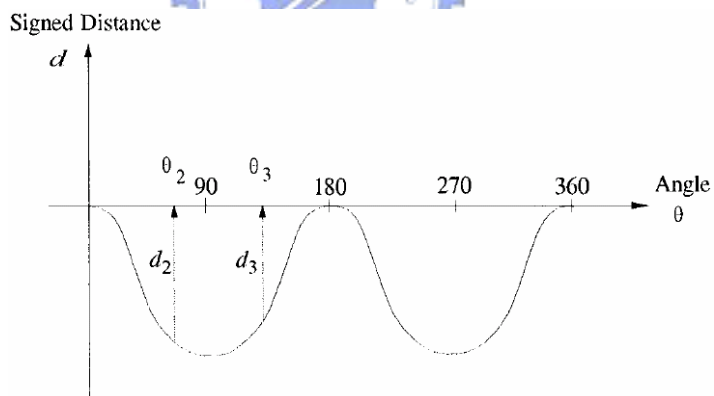
點資料特徵對位是以量測三維點資料的幾何特徵作為對位的依據，常用的幾何特徵為斜率(Slope)與曲率(Curvature)，也有許多學者自訂幾何特徵以增加對

位準確度與處理速度,如 Chua 等人[4]、Benjemaa 等人[5]、Hügli 等人[6]、Schütz 等人[7]皆有此方面的研究;Chua 的特徵點搜尋方式示意圖,如圖 1.6、圖 1.7 所示。



(a) 固定半徑之輪廓線

(b) 參考方位



(c) 特徵距離與轉換平面的角度關係

圖 1.6 Chua 自訂之幾何特徵示意圖[4]

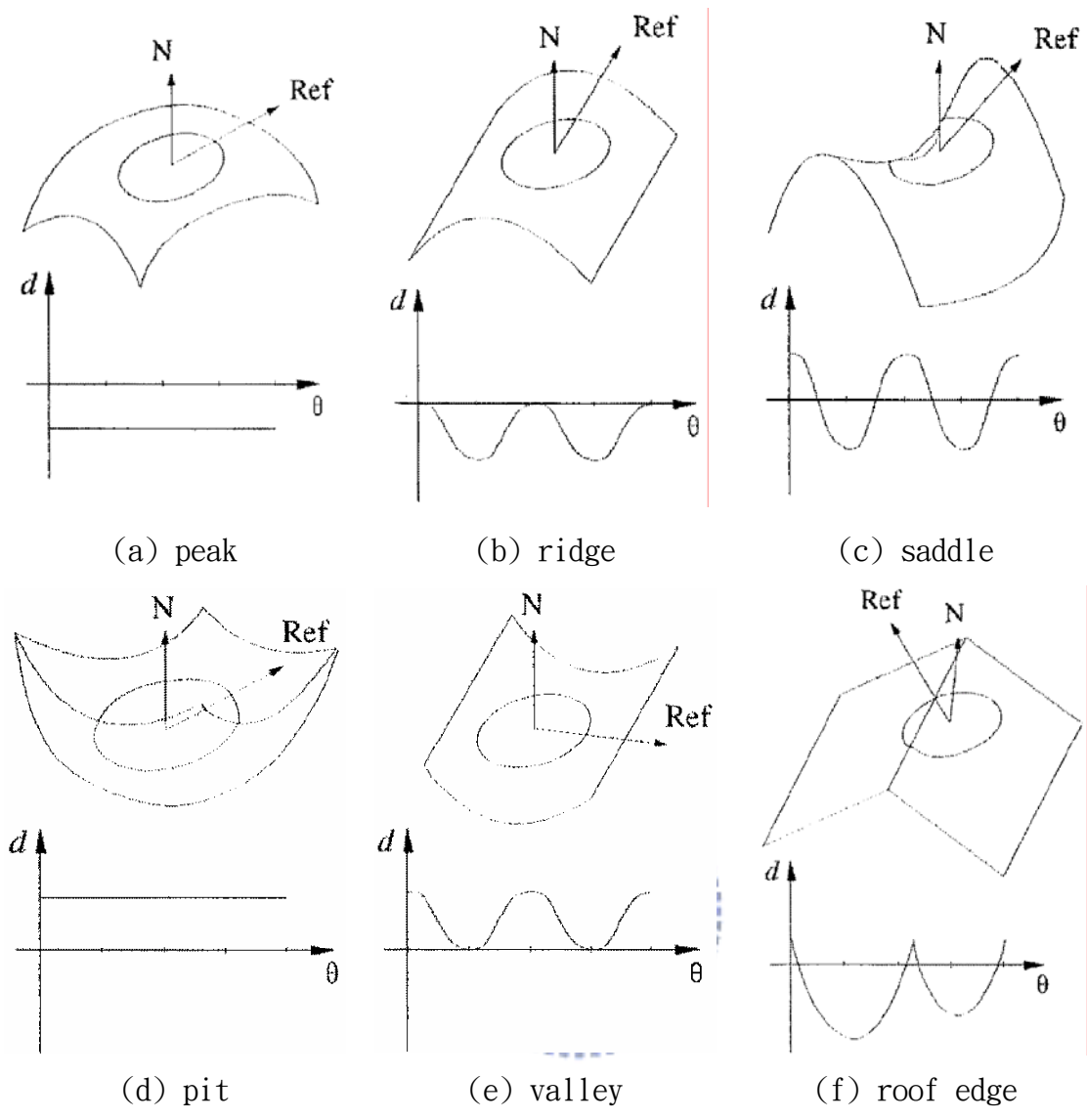


圖 1.7 各種特徵點的分析圖[4]

定義幾何特徵後，找出幾何特徵較為特別的點作為特徵點，並進行相關度分析找出特徵點在其他組點資料對應的點，將其紀錄為特徵點，由特徵點計算出座標轉換的矩陣，座標轉換的公式為：

$$P = RQ + T \tag{1.2.1}$$

Q：原始點資料，P：經過座標轉換矩陣轉換後之點資料

R：旋轉矩陣，T：位移矩陣

一般計算座標轉換矩陣的步驟可分為初始對位(Initial registration)與精確對位(Fine registration)。

初始對位常用 SVD(Singular Value Decomposition)演算法計算矩陣，使用數學上的奇異值分解計算旋轉矩陣(Rotation matrix)與位移矩陣(Translation matrix)，學者如 Arun 等人[8]、Yau 等人[9]皆進行此演算法的研究。

精確對位一般以 ICP(Iterative Closest Point)演算法進行之，如圖 1.8 所示，原理為使用資料點中最接近的點作為對應點，接著反覆地計算與套用座標轉換矩陣，由此慢慢地將資料點移動到相同的座標值，研究的學者有 Besl 等人[10]、Masuda 等人[11]、Rusinkiewicz 等人[12]。經由上述的過程可完成資料點特徵對位。

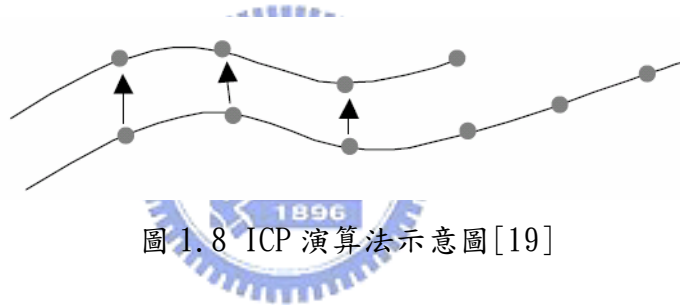


圖 1.8 ICP 演算法示意圖[19]

校正樣本對位是以校正樣本的特徵作為對位的依據，特徵的需求為具有幾何形狀簡單且容易辨識的特點，如此才能夠方便地挑選出校正樣本中的特徵。學者們使用各種物體作為校正樣本，Zagorchev 等人[13]使用校正框架，如圖 1.9 所示，Parasnis[14]使用校正塊，如圖 1.10 所示，Hebert[15]使用十字記號圖樣，如圖 1.11 所示，Niem 等人[16]使用線圖樣，如圖 1.12 所示，Sitnik 等人[17]、Chen 等人[18]使用矩形圖樣，如圖 1.13、圖 1.14 所示。



圖 1.9 校正框架[13]

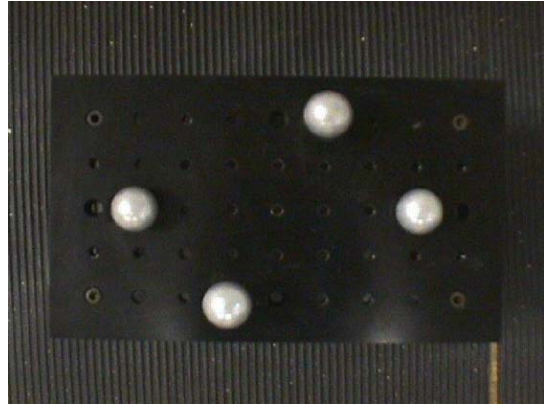


圖 1.10 校正塊[14]

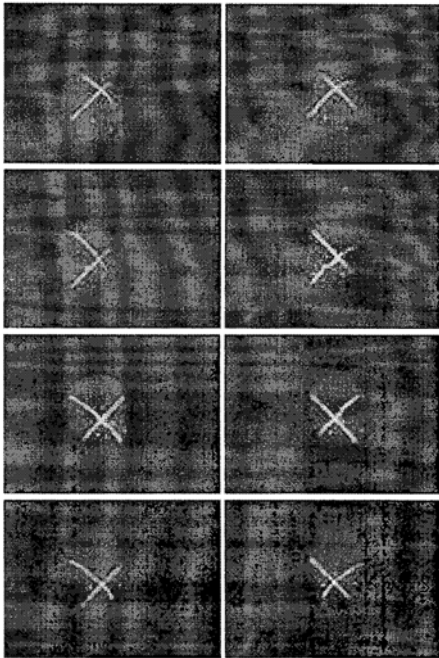


圖 1.11 十字記號圖樣[15]

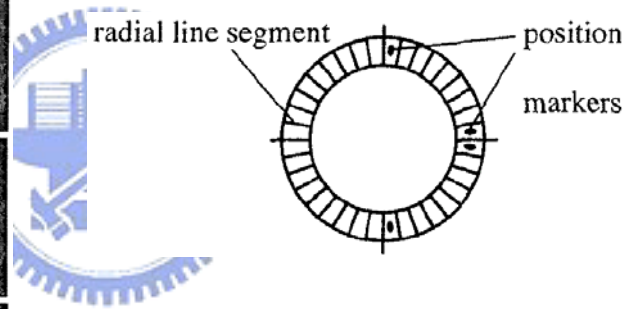


圖 1.12 線圖樣[16]

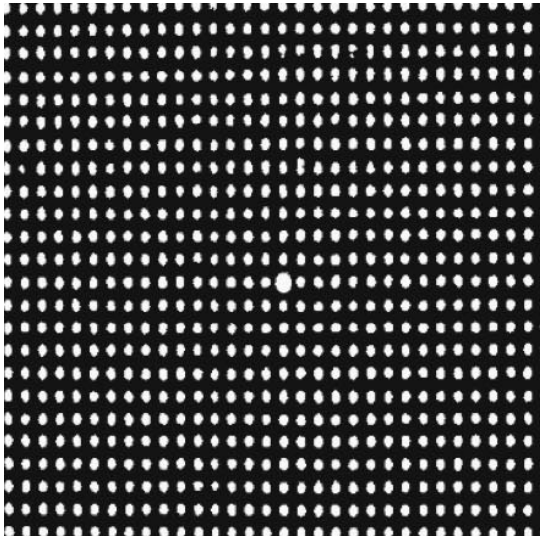


圖 1.13 Sitnik 矩形圖樣[17]

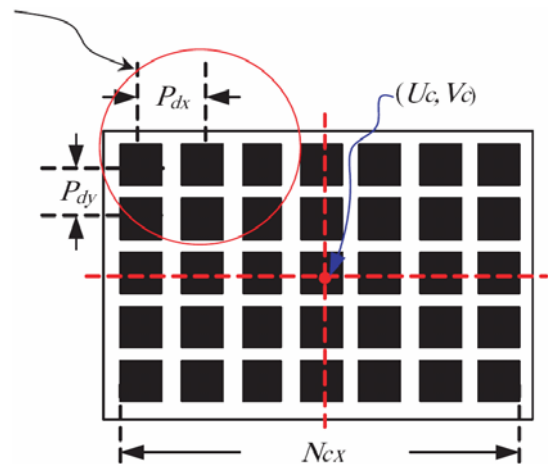


圖 1.14 Chen 矩形圖樣[18]

決定校正樣本後，以相機拍攝樣本的影像，接著必須選取樣本中的特徵點，為了達到自動選取特徵的需求，常用影像二值化(Image binarization)的處理方式讓影像中的特徵點變得明顯。挑選特徵點之後，搜尋特徵點對應到的空間三維座標，由此計算座標轉換的矩陣。計算矩陣的方法，一般採取最小平方法計算之，以多組的特徵點計算出最準確的矩陣元素。

點資料特徵對位的優點在於操作步驟簡單、流程較為自動化，對位全程皆可交給電腦運算，但缺點在於特徵點比對所需的運算時間長、量測準確度不足與各組點資料點重合部份過少時對位容易失敗，適合準確度佳與無法進行校正樣本對位的量測系統使用。一般來說，多台相機的量測系統使用校正樣本方法來解決對位問題的案例較多，儘管校正過程所需人工步驟較多，但電腦運算所需的時間較短且不受限於準確度。

1.2.2 三維點資料整合回顧

將各組點資料對位完畢後，接著必須進行點資料整合。點資料整合的方式主

要可分為掃描線(Scan line)為基礎與網格(Mesh)為基礎的方式，研究掃描線為基礎的學者有 Masuda 等人[21]、Lai 等人[22]，研究網格為基礎的學者有 Yau 等人[9]、Soucy 等人[23]、Turk 等人[24]。

掃描線一般為水平方向組成的點結構，通常由數十到數百個點組成。網格一般為三角形組成的點結構，由三個點組成。掃描線為基礎的方法優點在於處理方式簡單，電腦運算所需的時間較短，缺點為因點資料的處理單位較大，對於整合的效果較不穩定。與網格為基礎的方式相比較，若各組點資料的掃描線結構差異較大，則整合的效果較差，反之則整合的效果差不多。網格為基礎的優點則是因點資料的處理單位較小，對於整合的效果較穩當，缺點則是需要耗費較多的時間。因為本研究背景使用兩台數位相機作為影像擷取器，故兩邊點資料的掃描線結構差異不大，且考量到資料處理的效率，故選擇以掃描線為基礎的方式進行資料點整合。以下為以掃描線為基礎的點資料整合方式[21]：

1. 首先將兩筆資料點分別定義為來源點資料 (Source)與目的點資料 (Destination)。
2. 以來源點資料的掃描線架構為主，將目的點資料嵌入來源資料點的掃描線架構，依照掃描線的儲存方向設定點資料為來源點資料或目的點資料。由於本研究的掃描線儲存方向為由左向右，故設定左邊的點資料為來源點資料，右邊的點資料為目的點資料，如圖 1.15 所示。

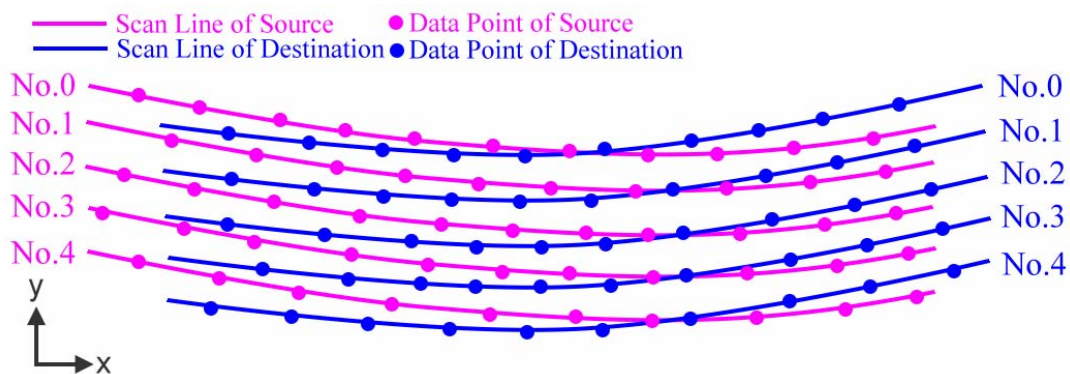


圖 1.15 兩組點資料設定示意圖[21]

3. 定義兩組點資料的屬性後，接著找出兩組點資料中各掃描線的交界點 (Intersection point)，交界點定義為最接近兩組點資料的掃描線相交處之點。圖 1.16 為兩組點資料之掃描線與交界點示意圖。

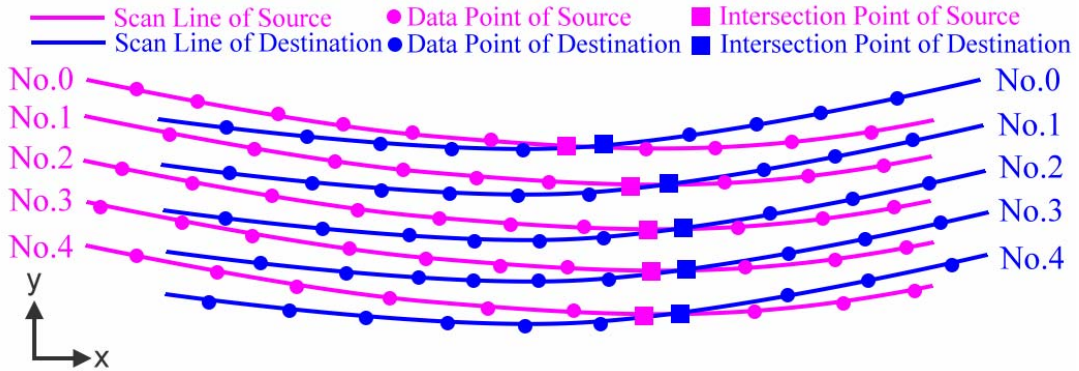


圖 1.16 兩組點資料之掃描線與交界點示意圖[21]

4. 交界點的搜尋方式為在兩組點資料的同編號掃描線中，找出點距離最近的點對(Point-pair)，並規定點對中的兩點，目的點資料的 x 軸座標必須大於來源點資料的 x 軸座標，如此才不會出現排序混亂的情況。為了加速搜尋速度與確保搜尋結果為正確，通常會設定一個搜尋範圍，在此範圍內定義點對出現的位置。這裡以來源資料點為基準，設定其搜尋範圍，定義搜尋範圍起始比率 (Rst) 與搜尋範圍結束比率 (Red)，此比率為點編號與掃描線總點數之比，也就是說點對出現的位置在(起始比率×掃描線之總點數)至(結束比率×掃描線之總點數)間。一般設定起始比率為 0.47，結束比率為 0.53。
5. 找出點對後，將此兩點分別定義為來源資料點與目的資料點的交界點，接著進行資料點混成(Blending)的步驟，由此步驟能夠將兩筆資料點平滑地接合。
6. 首先必須定義出兩組點資料的點資料對應，這裡使用前面定義過的交

界點作為點資料對應的標準，兩組點資料的交界點擁有相同的點編號，之後再依排列順序定義其他點的點編號。

- 接著定義出混成點數目 N_{blend} ，判別以幾個點進行混成的處理，一般設定 N_{blend} 為 15。圖 1.17 為進行混成的點資料示意圖，其中 S_i 為來源點資料中進行混成的點， D_i 為目的點資料中進行混成的點， i 為點編號，這裡使用 9 個點進行混成，則 $i=0, 1, \dots, 8$ 。為了能讓點資料混成的效果較佳，故設定以交界點為混成點中的中心點，如此可讓混成的結果能夠平均地從左右兩邊的點資料取樣之。

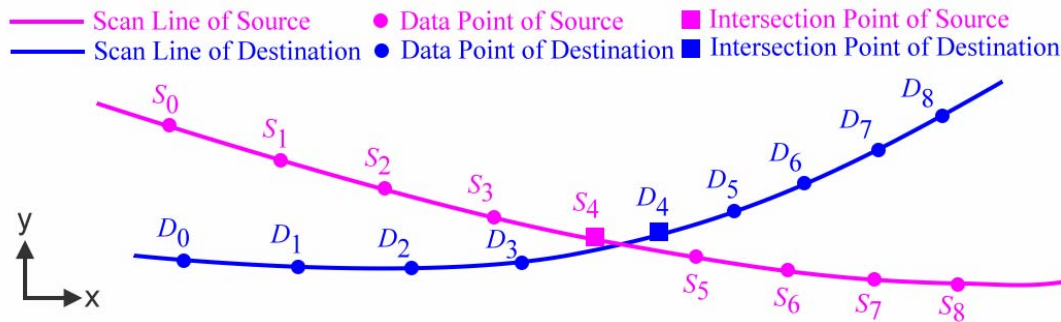


圖 1.17 進行混成的資料點示意圖[21]

- 點資料混成一般可分為線性混成與二次混成，其公式分別如式(1.2.2)與式(1.2.3)所示，其中 B_i 為混成出的點資料。

$$i = 0, 1, 2, \dots, N_{blend} - 1$$

$$B_i = ((N_{blend} - 1 - i) \times S_i + i \times D_i) / (N_{blend} - 1) \quad (1.2.2)$$

$$B_i = [(N_{blend} - 1 - i)^2 \times S_i + i^2 \times D_i] / [(N_{blend} - 1 - i)^2 + i^2] \quad (1.2.3)$$

- 使用線性混成(Linear blending)得到的結果較不平滑，故本論文選擇使用二次混成(Quadratic blending)進行混成。線性混成與二次混成之結果比較如圖 1.18 所示。

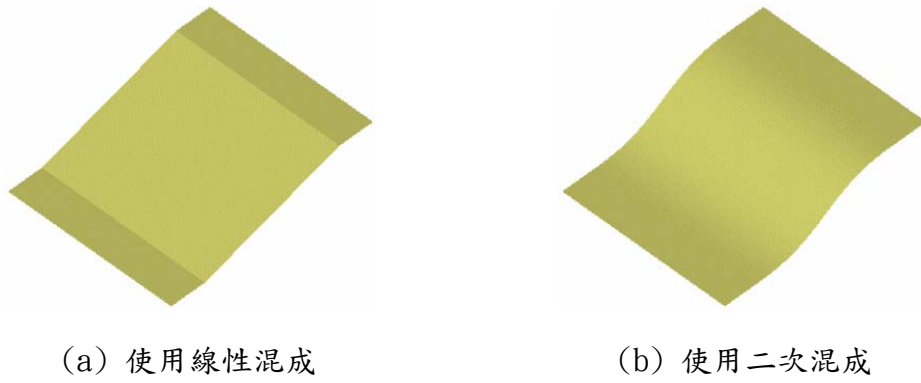


圖 1.18 線性混成與二次混成之結果比較[19]

- 經過點資料混成的步驟後，刪除位於混成點之前的目的點資料以及位於混成點之後的來源點資料，再將目的點資料嵌入來源點資料的線架構，則來源點資料即為兩組點資料經過點資料整合後產生一組新的點資料。

圖 1.19 為整個整合流程之示意圖。

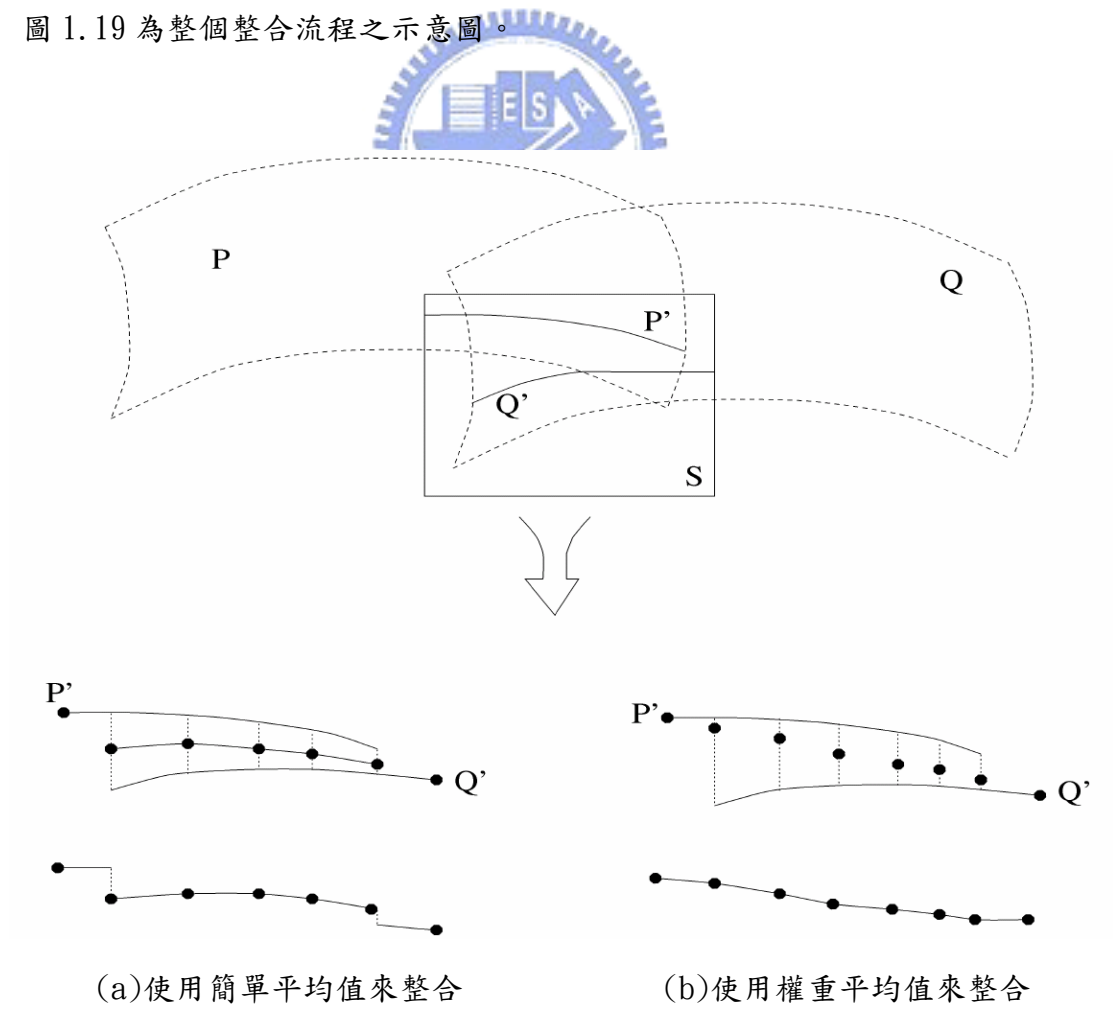


圖 1.19 整合示意圖[21]

1.3 研究動機與目的

日前與長庚醫院兒童部進行醫學工程合作的發展與研究，將數位條紋投射法應用在嬰兒畸形頭型矯正之嬰兒頭型的取得，與漏斗胸（Funnel Chest）兒童病患手術前後胸腔形狀改善的比對上面。

圖 1.20 為各種嬰兒頭型異常之案例[25]，醫療行為的步驟依序為：取得嬰兒頭型、製作適合個別頭型之頭型矯正帽，在需要壓迫生長的地方予以施加壓力，需要加速生長的地方將矯正帽該區塊進行挖空的動作，以利嬰兒頭型在該區塊之生長。



圖 1.20 嬰兒頭型異常資料照片[25]

圖 1.21 與圖 1.22 為醫院原先使用取得嬰兒頭型的方法，用石膏在嬰兒頭型上直接進行接觸打模，藉以此步驟取得該個案嬰兒頭型，待石膏定形之後所呈現之頭型模型如圖 1.22 所示。

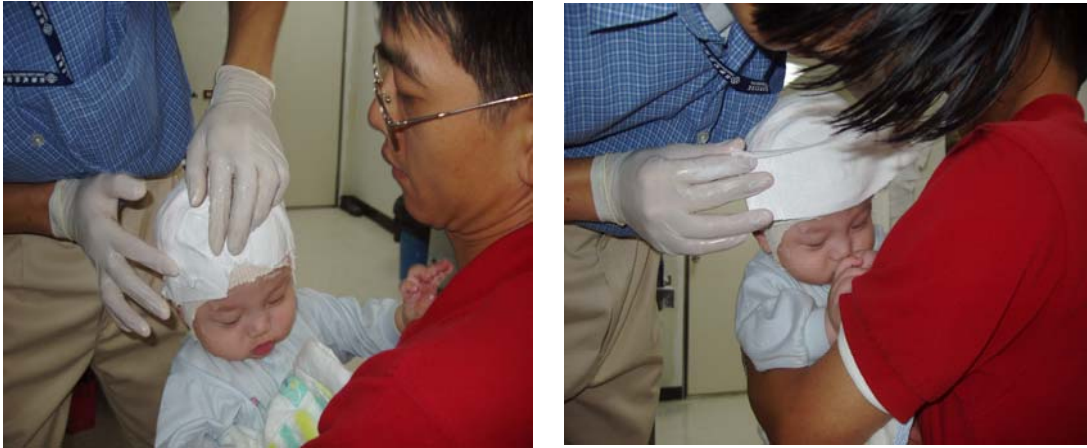


圖 1.21 頭盔製作過程 打模[25]



圖1.22 頭盔製作過程 打模[25]

打模完成後，將石膏填入模內，即便完成了與嬰兒頭型相似之一石膏頭型模型，灌製石膏模過程如圖 1.23 與圖 1.24 所示。



圖 1.23 頭盔製作過程 灌製石膏模[25]



圖 1.24 頭盔製作過程 灌製石膏模[25]

圖 1.25 為矯正帽完成後之樣式。



圖 1.25 頭盔製作過程 矯正帽完成[25]

但由於因為嬰兒自制能力較為薄弱，即使數位條紋投射法取相時間目前已經縮短至二十秒以內，但嬰兒依舊沒有辦法維持二十秒的時間靜止不動，故此技術目前除非再縮短取相的時間，否則無法直接應用於嬰兒頭型的取得上面；但是對於已經取得並且製成石膏像之嬰兒頭型模型，則可以進行取相存檔的動作，可於矯正後期比較矯正前後的成果差異與比較矯正後跟正常嬰兒頭型的差異性。

漏斗胸(Funnel chest)的兒童病患由於自制力已到達一定的水準，除了手術後立即取相會因為疼痛而造成的晃動以致取相偶爾失敗之外，基本上以數位條紋投射法來取得漏斗胸兒童病患的胸部三維外形點資料不會有太大的困難或者誤差，可以藉由此三維外形點資料以比較手術前、手術後、與手術後多日已康復之醫療行為改善情況。圖 1.26 與圖 1.27 為漏斗胸兒童病患案例。



(a)傳統手術失敗後，漏斗胸復發



(b)微創手術後兩個月，疤痕在兩側



(c)手術前



(d)手術後兩個月

圖 1.26 漏斗胸兒童病患微創手術案例[26]



(a)傳統手術成功者，
仍有大疤痕（男生）



(b)傳統手術成功者，
仍有大疤痕（女生）

圖 1.27 漏斗胸兒童病患傳統手術案例[26]

由上述兩個案例可看出，此套系統幾乎已能為醫院所接受並且使用；不過目前由於對位與整合的技術尚未成熟，一切都是藉由緩慢複雜的手動程序完成整合與對位；對於非理工背景的人要操作使用數位條紋投射法理論建構出來的設備過於困難；另一方面，於市面上具有三維點資料對位與整合之電腦輔助設計系統價格接過於昂貴而且並非能適用於分析處理研究室儀器取得之三維點資料，故期望本研究能發展出一套數位條紋投射法取得之三維外形點資料的對位與整合軟體，並且配合簡單操作的介面。

1.4 研究方法

本論文研究方法流程可用圖 1.28 表示之。

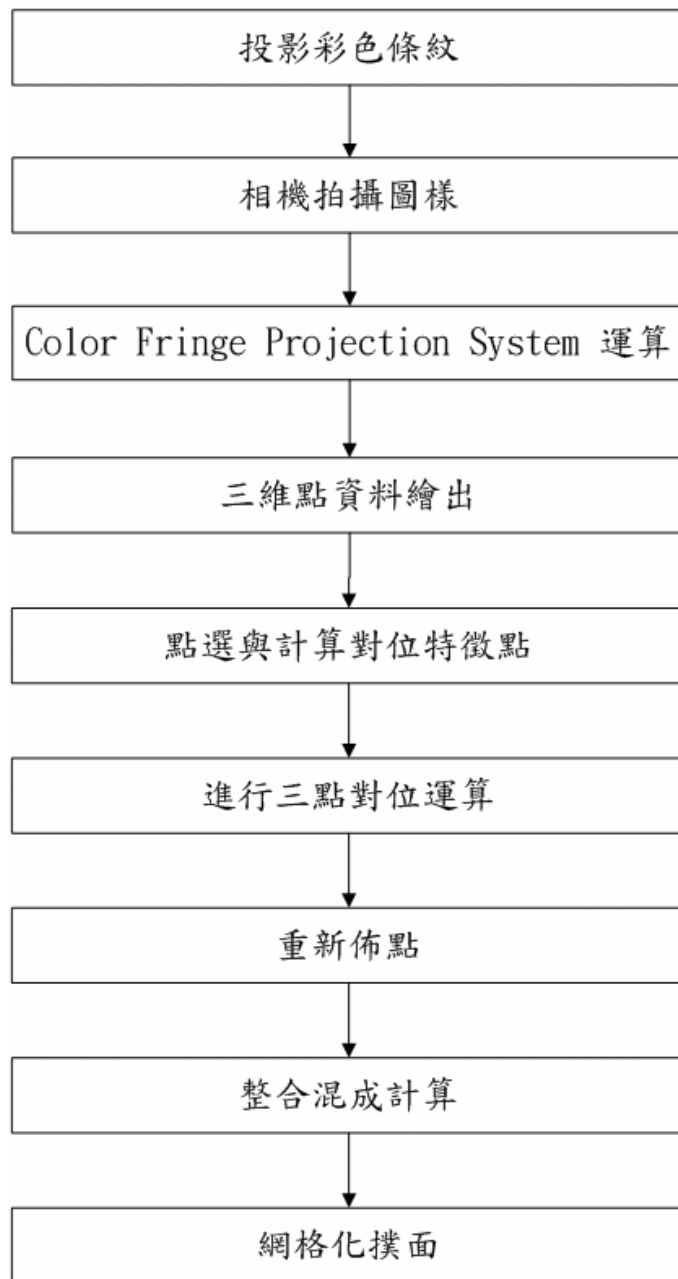


圖 1.28 本論文之研究方法流程圖

流程之步驟詳述如下：

1. 投射一張或多張之數位彩色條紋圖樣至物體表面。
2. 使用相機拍攝投射於物體表面之彩色條紋圖樣。
3. 使用軟體 Color Fringe Projection System 進行各項運算，其中包含讀取影像中的像素質、以相位移法將相位求出、對求出的相位進行相位補償、進行

相位重建等等。最終將連續的相位值轉換成三維座標，儲存為三維點資料檔。

4. 將儲存之三維點資料檔讀取並於 CAD 軟體中繪出。
5. 於繪出之圖像上，點選三維對位所需之特徵點，並且透過計算求得對位之特徵頂點。
6. 進行三維點資料之對位運算。
7. 將三維對位完成之三維點資料進行重新佈點。
8. 將重新佈點後之平面透過混成計算，還原其高度值。
9. 將其完成對位與整合之三維點資料進行網格化鋪面。

經由上述的步驟，只需要兩台相機，一台投影機，與一台電腦即可獲得物體之三維資訊，並且透過本研究的處理，將兩組三維資訊因應各種情況做不同計算的接合，使得三維外形能最清晰正確的被表達。



二、三維點資料處理與應用之基本理論

本章說明本論文所使用之三維點資料處理與應用之基本理論，首先說明於一堆隨機三維點資料中，如何使用最小平方法尋找其最合適之平面，使得所有隨機三維點資料到該平面之距離總和為最小值。接著介紹於一堆隨機三維點資料中，如何找出其最合適之拋物面，使得所有隨機三維點資料到該拋物面之距離總和為最小值。最後介紹如何於座標系統轉換中，計算出可達到各種目的之座標系統轉換矩陣。

三維點資料之各種對位與整合理論各有特色與應用限制，本研究針對數位彩色條紋投射法之資料結構特性選取各方理論進行整合，期望透過各種理論之整合與程式建立，可發展出一套最適用於數位彩色條紋投射法所求得之三維外形點資料處理與應用軟體。



2.1 使用最小平方法尋找最適合的平面

首先要找出一個合適的平面，所有特徵點到該平面之距離為最小值；則該平面之法向量 (normal vector) 為特徵點資料集合之特徵方向。使用最小平方法 (Least squares fitting) 尋找最適合的平面[27][28][29][30][31][32][33] 是為了找出特徵點資料集合的特徵方向。

Given a set of points $p_i, 1 \leq i \leq k$

定義一個平面，經過 \bar{x}_0 ，其法線向量為 \bar{a} ，則其平面公式為：

$$\bar{a} \cdot (\bar{x} - \bar{x}_0) = 0 \rightarrow \bar{a} \cdot \bar{x} = \bar{a} \cdot \bar{x}_0 = c \rightarrow \bar{a}^T \bar{x} - c = 0 \quad (2.1.1)$$

with $\bar{a}^T \bar{a} = 1$

$$E_i = \bar{a}^T \bar{p}_i - c \quad (2.1.2)$$

$$\begin{aligned} \text{let } f(\bar{a}, c) &= \sum_{i=1}^k E_i^2 \\ &= \frac{1}{2k} \sum_{i=1}^k (\bar{a}^T \bar{p}_i - c)^2 \\ &= \frac{1}{2k} \sum_{i=1}^k [(\bar{a}^T \bar{p}_i - c)(\bar{a}^T \bar{p}_i - c)] \\ &= \frac{1}{2k} \sum_{i=1}^k (\bar{a}^T \bar{p}_i \bar{p}_i^T \bar{a} - 2c \bar{a}^T \bar{p}_i + c^2) \\ &= \frac{1}{2k} [\bar{a}^T (\sum_{i=1}^k \bar{p}_i \bar{p}_i^T) \bar{a} - 2c \bar{a}^T \sum_{i=1}^k \bar{p}_i + kc^2] \\ &= \frac{1}{2} \bar{a}^T (\sum_{i=1}^k \bar{p}_i \bar{p}_i^T) \bar{a} - c \bar{a}^T \bar{\bar{p}} + \frac{1}{2} c^2 \end{aligned} \quad (2.1.3)$$

$$\text{where } \bar{\bar{p}} = \frac{1}{k} \sum_{i=1}^k \bar{p}_i$$



解決此最佳化問題

$$\text{Find } \bar{a}, c \text{ such that } f(\bar{a}, c) = \frac{1}{2} \bar{a}^T (\sum_{i=1}^k \bar{p}_i \bar{p}_i^T) \bar{a} - c \bar{a}^T \bar{\bar{p}} + \frac{1}{2} c^2 \rightarrow \min. \quad (2.1.4)$$

$$\text{subject to } \bar{a}^T \bar{a} - 1 = 0$$

上述之最佳化問題可用 Lagrange Multipliers Method 求得理論解，其求解過程

先假設 Lagrange 函數：

$$\text{Define } L(\bar{a}, c, \lambda) = f(\bar{a}, c) + \frac{1}{2} \lambda (1 - \bar{a}^T \bar{a}) \quad (2.1.5)$$

$$\frac{\partial L}{\partial a_i} = 0, i=1,2,3 \rightarrow \frac{\partial f(\bar{a}, c)}{\partial a_i} - \lambda a_i = 0, i=1,2,3 \rightarrow \frac{\partial f(\bar{a}, c)}{\partial \bar{a}} - \lambda \bar{a} = 0 \quad (2.1.6)$$

$$\frac{\partial f(\bar{a}, c)}{\partial \bar{a}} = (\frac{1}{k} \sum_{i=1}^k \bar{p}_i \bar{p}_i^T) \bar{a} - c \bar{\bar{p}} \quad (2.1.7)$$

$$\frac{\partial L}{\partial c} = 0 \rightarrow \frac{\partial f(\bar{a}, c)}{\partial c} = 0 \rightarrow -\bar{a}^T \bar{\bar{p}} + c = 0 \rightarrow \bar{a}^T \bar{\bar{p}} = c \quad (2.1.8)$$

將公式(2.1.7)、(2.1.8)帶入公式(2.1.6)，可得：

$$\left(\frac{1}{k} \sum_{i=1}^k \bar{p}_i^T \bar{p}_i\right) \bar{a} - \bar{\bar{p}} \bar{\bar{p}}^T \bar{a} - \lambda \bar{a} = 0 \quad (2.1.9)$$

$$\left[\left(\frac{1}{k} \sum_{i=1}^k \bar{p}_i^T \bar{p}_i\right) - \bar{\bar{p}} \bar{\bar{p}}^T\right] \bar{a} = \lambda \bar{a} \rightarrow [M] \bar{a} = \lambda \bar{a} \quad (2.1.10)$$

$$[M] = \left(\frac{1}{k} \sum_{i=1}^k \bar{p}_i^T \bar{p}_i\right) - \bar{\bar{p}} \bar{\bar{p}}^T \quad (2.1.11)$$

由公式(2.1.9)可得：

$$\left(\frac{1}{k} \sum_{i=1}^k \bar{p}_i^T \bar{p}_i\right) \bar{a} = \bar{\bar{p}} \bar{\bar{p}}^T \bar{a} + \lambda \bar{a} = c \bar{\bar{p}} + \lambda \bar{a} \quad (2.1.12)$$

利用公式(2.1.8)、(2.1.12)可將公式(2.1.3)整理如下：

$$\begin{aligned} f(\bar{a}, c) &= \frac{1}{2} \bar{a}^T \left(\sum_{i=1}^k \bar{p}_i \bar{p}_i^T\right) \bar{a} - c \bar{a}^T \bar{\bar{p}} + \frac{1}{2} c^2 \\ &= \frac{1}{2} \bar{a}^T (\lambda \bar{a} + c \bar{\bar{p}}) - c \bar{a}^T \bar{\bar{p}} + \frac{1}{2} c^2 \\ &= \frac{1}{2} \lambda \bar{a}^T \bar{a} + \frac{1}{2} c \bar{a}^T \bar{\bar{p}} - c \bar{a}^T \bar{\bar{p}} + \frac{1}{2} c^2 \\ &= \frac{1}{2} \lambda + \frac{1}{2} c^2 - c^2 + \frac{1}{2} c^2 \\ &= \frac{1}{2} \lambda \end{aligned} \quad (2.1.13)$$

由公式(2.1.4)(2.1.10)(2.1.13)可知最佳化問題之解為[M]矩陣之最小特徵值之特徵向量，該特徵向量即為所求平面之法向量。

2.2 尋找一個最適合多點三維點資料之拋物面

尋找出一個拋物面(Paraboloid)，三維點資料樣本到該拋物面的距離為最小值。

給一組三維點資料樣本 $\{(x_i, y_i, z_i)\}_{i=1}^m$ ，並且假設其確實位於拋物面上。

$$z = f(x, y) = p_1x^2 + p_2xy + p_3y^2 + p_4x + p_5y + p_6 = \vec{P} \cdot \vec{Q}(x, y) \quad (2.2.1)$$

$$\text{where } \vec{P} = (p_1, p_2, p_3, p_4, p_5, p_6)$$

$$\text{and } \vec{Q} = (x^2, xy, y^2, x, y, 1)$$

找出 \vec{P} 讓誤差平方和為最小值：

$$E(\vec{P}) = \sum_{i=1}^m (\vec{P} \cdot \vec{Q}_i - z_i)^2 \quad (2.2.2)$$

$$\text{where } \vec{Q}_i = \vec{Q}(x_i, y_i)$$

當 E 的梯度為零的時候，誤差為最小值

$$\rightarrow \nabla E = 2 \sum_{i=1}^m (\vec{P} \cdot \vec{Q}_i - z_i) \vec{Q}_i = \vec{0} \quad (2.2.3)$$

$$\rightarrow \left(\sum_{i=1}^m \vec{Q}_i \vec{Q}_i^t \right) \vec{P} = \sum_{i=1}^m z_i \vec{Q}_i$$

定義一個 6×6 的對稱矩陣 $A = \sum_{i=1}^m \vec{Q}_i \vec{Q}_i^t$ 與一個 6×1 的向量 $B = \sum_{i=1}^m z_i \vec{Q}_i$

線性系統方程式 $A\vec{P} = \vec{B}$ 之解答 \vec{P} 即為所求拋物面方程式之係數。

$$\rightarrow \begin{bmatrix} \sum_{i=1}^m x_i^4 & \sum_{i=1}^m x_i^3 y_i & \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i^2 \\ \sum_{i=1}^m x_i^3 y_i & \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i y_i \\ \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m y_i^4 & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m y_i^2 \\ \sum_{i=1}^m x_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m z_i x_i^2 \\ \sum_{i=1}^m z_i x_i y_i \\ \sum_{i=1}^m z_i y_i^2 \\ \sum_{i=1}^m z_i x_i \\ \sum_{i=1}^m z_i y_i \\ \sum_{i=1}^m z_i \end{bmatrix}$$

[29](2.2.4)

求已知拋物面之頂點：

$$z = f(x, y) = p_1x^2 + p_2xy + p_3y^2 + p_4x + p_5y + p_6$$

$$\frac{\partial f(x, y)}{\partial x} = 2p_1x + p_2y + p_4 = 0 \quad (2.2.5)$$

$$\frac{\partial f(x, y)}{\partial y} = p_2x + 2p_3y + p_5 = 0$$

$$\rightarrow x = \frac{\left(\frac{2p_3p_4 - p_5}{p_2}\right)}{\left(p_2 - \frac{4p_3p_1}{p_2}\right)}, y = \frac{-(2p_1 + p_4)}{p_2}, z = p_1x^2 + p_2xy + p_3y^2 + p_4x + p_5y + p_6$$

$$(2.2.6)$$

2.3 座標轉換矩陣(transformation matrix)

於各種座標系統中的轉換可以透過簡單的矩陣運算達到其目的，本研究使用的座標轉換矩陣可分為三種：1、平移 (translation)；2、旋轉 (rotation)；3、映像 (mapping) [34][35][36]。

2.3.1 平移

當一個物件要在 x, y, 和 z 方向分別的被平移 a, b, 和 c，從他的初始位置模型座標系統轉至世界座標系統的時候，於世界座標系統中他會有一個新的位置，新位置座標 (X_w, Y_w, Z_w) 會有以下的關係：

$$\begin{aligned} X_w &= X_m + a \\ Y_w &= Y_m + b \\ Z_w &= Z_m + c \end{aligned} \quad (2.3.1)$$

其中 X_m, Y_m, Z_m 乃是在模型座標系統中的相同點位置

使用矩陣形式，公式 (2.3.1) 可以表示成下列形式：

$$\text{Trans}(a,b,c) \Rightarrow \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad [34] \quad (2.3.2)$$

2.3.2 旋轉

假設一個物件要對於其模型座標系統之 x 軸旋轉角度 θ 至世界座標系統中，從原始座標系統中的 (X_m, Y_m, Z_m) 轉至世界座標系統中的 (X_w, Y_w, Z_w) 。

(X_w, Y_w, Z_w) 與 (X_m, Y_m, Z_m) 之間的關係可以用以下公式簡單的表示出來：

$$X_w = X_m \quad (2.3.3)$$

$$\begin{aligned} Y_w &= l \cos(\theta + \alpha) \\ &= l(\cos \theta \cos \alpha - \sin \theta \sin \alpha) \\ &= l \cos \alpha \cos \theta - l \sin \alpha \sin \theta \\ &= Y_m \cos \theta - Z_m \sin \theta \end{aligned} \quad (2.3.4)$$

$$\begin{aligned} Z_w &= l \sin(\theta + \alpha) \\ &= l(\sin \theta \cos \alpha + \cos \theta \sin \alpha) \\ &= l \cos \alpha \sin \theta + l \sin \alpha \cos \theta \\ &= Y_m \sin \theta + Z_m \cos \theta \end{aligned} \quad (2.3.5)$$

由公式 (2.3.3)、(2.3.4) 和 (2.3.5) 可以表示成矩陣形式如下：

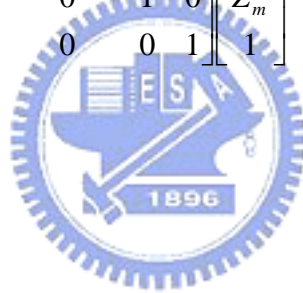
$$Rot(x, \theta) \Rightarrow \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad [34] \quad (2.3.6)$$

同理可得依據 y 軸或者 z 軸旋轉的齊次轉換矩陣可以表示成：

$$Rot(y, \theta) \Rightarrow \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad [34] \quad (2.3.7)$$

$$Rot(z, \theta) \Rightarrow \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad [34] \quad (2.3.8)$$

2.3.3 映像



假設要將一個依據 $x_1y_1z_1$ 座標系統位置為 (X_1, Y_1, Z_1) 的一點 P，經過計算轉換成依據 $x_2y_2z_2$ 座標系統位置為 (X_2, Y_2, Z_2) 的點，可使用下列座標系統轉換矩陣：

$$T_{1-2} \Rightarrow \begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} \quad [34] \quad (2.3.9)$$

其中 p_x, p_y, p_z 為依據 $x_2y_2z_2$ 座標系統看 $x_1y_1z_1$ 座標系統原點之位置座標； n_x, n_y, n_z 分別為元素 x_2, y_2, z_2 依據 $x_2y_2z_2$ 座標系統看 $x_1y_1z_1$ 座標系統 x 軸方向之單位向量； o_x, o_y, o_z 分別為元素 x_2, y_2, z_2 依據 $x_2y_2z_2$ 座標系統

看 $x_1y_1z_1$ 座標系統 y 軸方向之單位向量； a_x, a_y, a_z 分別為元素 x_2, y_2, z_2 依據 $x_2y_2z_2$ 座標系統看 $x_1y_1z_1$ 座標系統 z 軸方向之單位向量。



三、三維點資料處理與應用之軟體系統發展

隨著數位條紋投射法發展之日益成熟，為了改進設備取得之便利性與購買設備所需之經費，Color Fringe Projection System[37]（附錄 A）將以往數位條紋投射法使用之 DLP 投影機改成 LCD 數位投影機，將 CCD 相機以大眾市場導向之數位相機取代之，這樣便可以拿掉相機控制裝置與 Unix 系統工作站，直接使用個人電腦進行控制。Color Fringe Projection System 硬體架設示意圖如圖 3.1 所示。

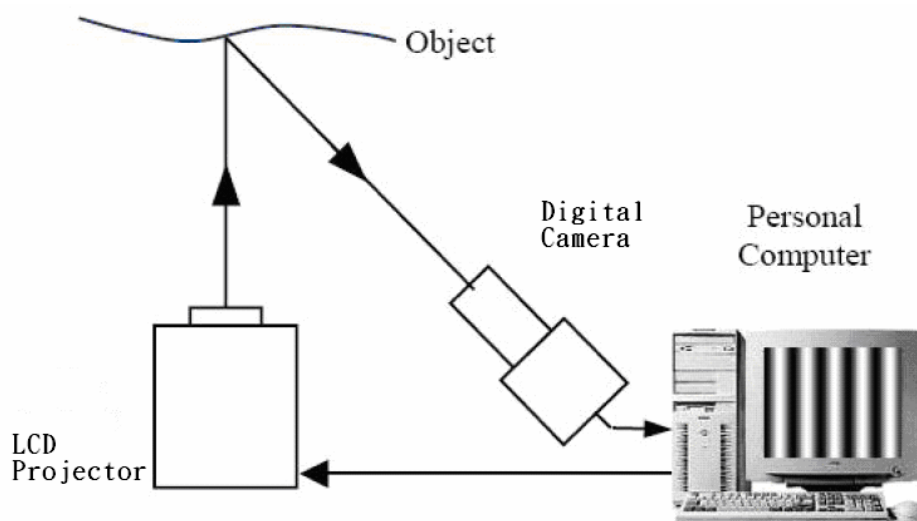


圖 3.1 Color Fringe Projection System 硬體架設示意圖

以數位條紋投射法擷取待測物體之條紋影像並且使用程式 Color Fringe Projection System 運算取得左右相機視角之三維點資料後，便可使用三維點資料進行對位與整合等後處理的工作。三維點資料對位可分為三步驟：

- 一、讀取三維點資料。
- 二、選取特徵點。
- 三、進行三點對位。

三維點資料整合可分為三步驟：


一、重新佈點。

二、混成。計算每一點混成之高度值。

三、網格化鋪面。輸出一組三維點資料，並且完成網格化鋪面。

最後可以藉由剖面圖取得的運算來取得待測物體三維外形之剖面圖，得以對物體之三維外形進行分析與比較。

3.1 讀取三維點資料



分別讀取左右相機視角擷取之三維點資料檔案，其附檔名為“.tst”；由於本研究是採用掃瞄線形式作為點資料的儲存格式，故在讀取點資料之時，將同一條掃瞄線上之所有點資料使用 AutoCAD 2002 之內建指令“三維多點連線(3dpolyline)”連成一條線。左右相機視角取得之原始石膏像模型如圖 3.2 與 3.3 所示。左右相機視角取得之石膏像模型條紋圖像如圖 3.4 與 3.5 所示。左右相機視角取得之三維點資料檔案以 3D 掃瞄線圖像形式表現，如圖 3.6~圖 3.9 所示。



圖 3.2 左邊相機原始圖像



圖 3.3 右邊相機原始圖像

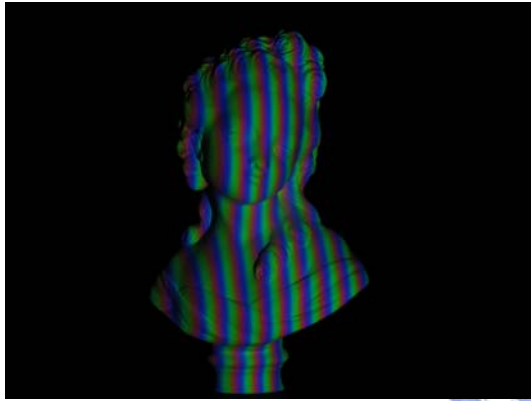


圖 3.4 左邊相機條紋投射圖像



圖 3.5 右邊相機條紋投射圖像

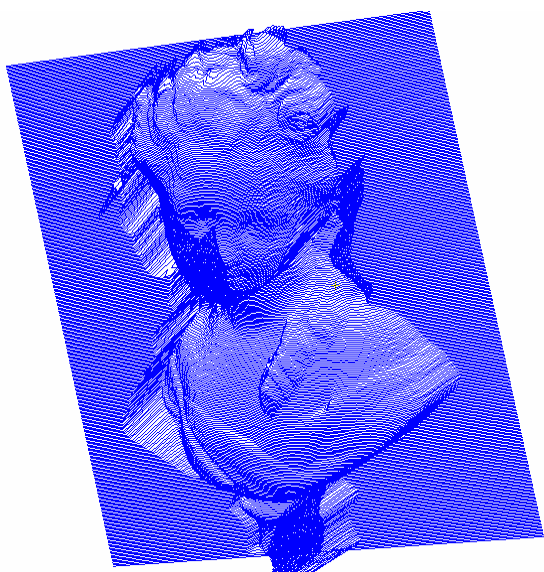


圖 3.6

左邊視角相機取得之三維點資料

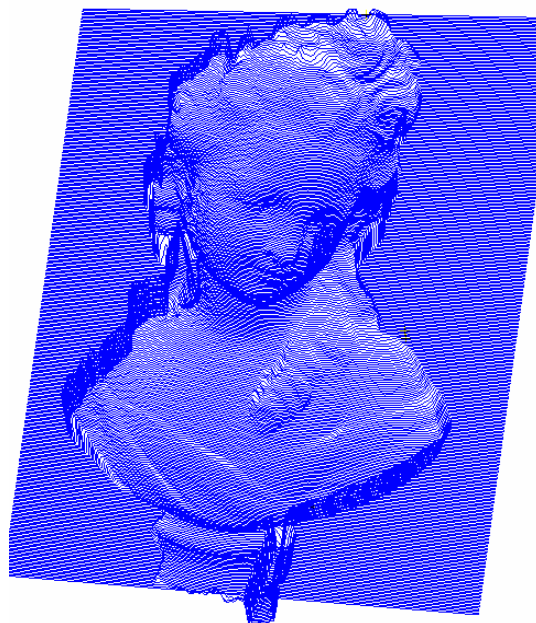


圖 3.7

左邊視角相機取得之三維點資料

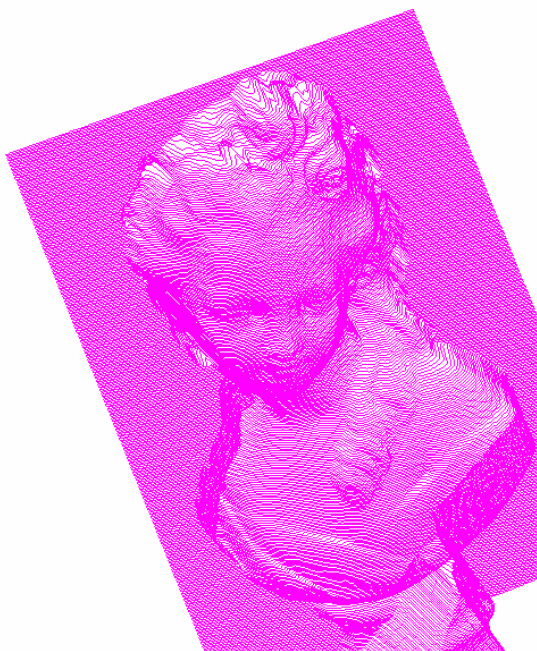


圖 3.8

右邊視角相機取得之三維點資料



圖 3.9

右邊視角相機取得之三維點資料

3.2 選取特徵點

選擇方便使用於特徵對位之特徵點。挑選三維圖像中具有強烈特徵之單點三維點資料；若該特徵點為周遭區域三維外形起伏之頂點，可使用下列方式選得欲選取之三維頂點特徵點資料。

1. 先點選該特徵頂點附近之任何一點 P，如圖 3.10 所示。

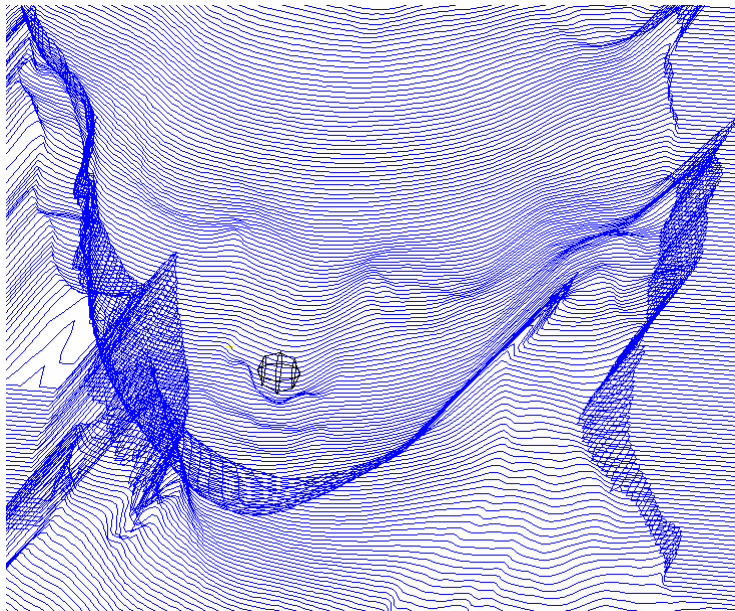


圖 3.10 選取特徵點附近之任何一點

2. 使用距離比較法找出整組三維點資料中距離點 P 最近之 1000~1500 個點，如圖 3.11 與圖 3.12 所示。

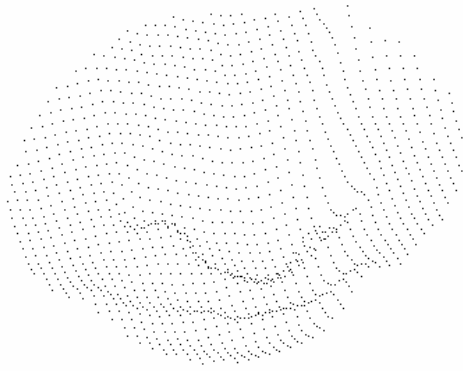


圖 3.11

距離選取點最近之 1400 點資料



圖 3.12

距離選取點最近之 1400 點資料

3. 找出一個合適的平面，可以使步驟 2 所取得之距離點 P 最為相近的所有三維點資料到欲求平面之距離總和為最小值；求得平面之法向量為步驟 2

所取得之所有特徵點資料集合之特徵方向。將其所有特徵點資料帶入矩陣

$[M] = \left(\frac{1}{k} \sum_{i=1}^k \vec{p}_i^T \vec{p}_i \right) - \vec{p}\vec{p}^T$ 之中，並且求解該矩陣之特徵值與特徵向量；選

取最小特徵值之特徵向量即為選取特徵點資料集合之特徵方向其單位向

量。該特徵向量方向與選取特徵點資料集合之關係會如圖 3.13 與圖 3.14

所示。

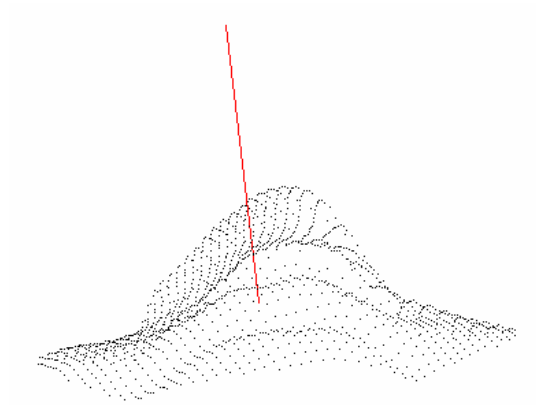


圖 3.13 找出特徵點資料集合
之特徵平面法向量

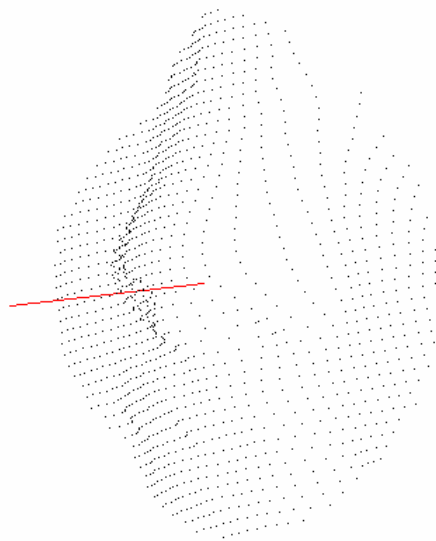


圖 3.14 找出特徵點資料集合
之特徵平面法向量

4. 根據步驟 3 所求得之單位法向量，利用座標轉換矩陣依序將所有步驟 3 選取之特徵點資料集合從原先位於之世界座標系統 XYZ 中轉換至新定義的模型座標系統 xyz 中，其中新定義的模型座標系統 xyz 之 z 軸方向於座標系統 XYZ 中之單位方向向量必須與求得之單位法向量相同，並且其原點必須位於原先世界座標系統中所有選取之特徵點資料之 (x, y, z) 平均值上面。
5. 在模型座標系統 xyz 中，使用距離比較法找出距離點 P 最近之 50~250 個點作為拋物面特徵點資料集合，如圖 3.15 與圖 3.16 所示。

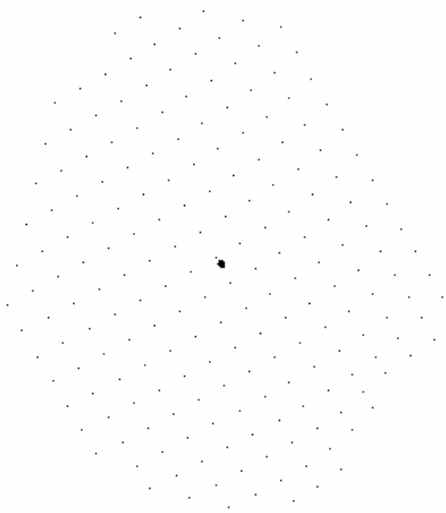


圖 3.15

距離選取點最近之 200 點資料

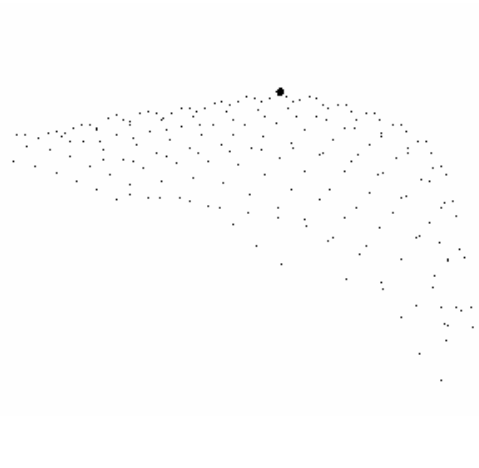


圖 3.16

距離選取點最近之 200 點資料

6. 於模型座標系統 xyz 中，尋找出一個合適的拋物面，使得拋物面特徵點資料集合內之所有三維點到該拋物面之距離總和為最小值。將所有點依序帶入方程式 (3.2.1) 與方程式 (3.2.2) 中，可求得拋物面未知常數 $(p_1, p_2, p_3, p_4, p_5, p_6)$ 。

$$\begin{bmatrix} \sum_{i=1}^m x_i^4 & \sum_{i=1}^m x_i^3 y_i & \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i^2 \\ \sum_{i=1}^m x_i^3 y_i & \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i y_i \\ \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m y_i^4 & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m y_i^2 \\ \sum_{i=1}^m x_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i \\ \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i \\ \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m 1 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m z_i x_i^2 \\ \sum_{i=1}^m z_i x_i y_i \\ \sum_{i=1}^m z_i y_i^2 \\ \sum_{i=1}^m z_i x_i \\ \sum_{i=1}^m z_i y_i \\ \sum_{i=1}^m z_i \end{bmatrix} \quad (3.2.1)$$

$$\begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m x_i^4 & \sum_{i=1}^m x_i^3 y_i & \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m z_i x_i^2 \\ \sum_{i=1}^m x_i^3 y_i & \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m z_i x_i y_i \\ \sum_{i=1}^m x_i^2 y_i^2 & \sum_{i=1}^m x_i y_i^3 & \sum_{i=1}^m y_i^4 & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m z_i y_i^2 \\ \sum_{i=1}^m x_i^3 & \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m x_i & \sum_{i=1}^m z_i x_i \\ \sum_{i=1}^m x_i^2 y_i & \sum_{i=1}^m x_i y_i^2 & \sum_{i=1}^m y_i^3 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m y_i & \sum_{i=1}^m z_i y_i \\ \sum_{i=1}^m x_i^2 & \sum_{i=1}^m x_i y_i & \sum_{i=1}^m y_i^2 & \sum_{i=1}^m x_i & \sum_{i=1}^m y_i & \sum_{i=1}^m 1 & \sum_{i=1}^m z_i \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^m z_i x_i^2 \\ \sum_{i=1}^m z_i x_i y_i \\ \sum_{i=1}^m z_i y_i^2 \\ \sum_{i=1}^m z_i x_i \\ \sum_{i=1}^m z_i y_i \\ \sum_{i=1}^m z_i \end{bmatrix} \quad (3.2.2)$$

7. 求得該拋物面後，於模型座標系統 xyz 中，再求該拋物面在其 z 方向的極值，並且求得產生極值時該點之三維點資料，此三維點資料即為我們所要尋找之特徵頂點，如圖 3.17 與圖 3.18 所示。再透過座標轉換矩陣，將此特徵頂點從模型座標系統 xyz 中轉回世界座標系統 XYZ 中。

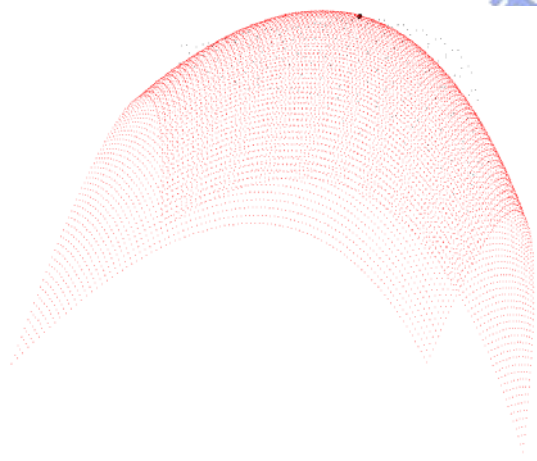


圖 3.17 求得拋物面，並尋找其頂點

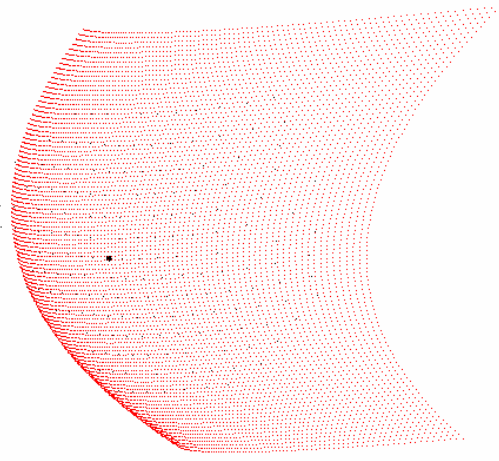


圖 3.18 求得拋物面，並尋找其頂點

3.3 進行三點對位

三維點資料之對位方式非常多樣化，根據數位條紋投射法取得的三維點資料特性，於此採用三點對位流程：

1. 定義廣義座標系統 XYZ ，並且分別自選定的兩組點資料 A、B 中，點選找出特徵對位所需要的三個點 $\vec{A}_1, \vec{A}_2, \vec{A}_3$ 與 $\vec{B}_1, \vec{B}_2, \vec{B}_3$ 。
2. let $\vec{P} = \overrightarrow{B_1A_1} = \vec{A}_1 - \vec{B}_1$ 將所有的 B 組點資料依據 \vec{P} 移動，分別於 (x, y, z) 方向平移 (P_x, P_y, P_z) ；此時所有的 B 組點資料依然位於 XYZ 座標系統之中，並且透過平移的運算使得 \vec{A}_1 、 \vec{B}_1 兩點重合。

平移的矩陣為：

$$\begin{bmatrix} 1 & 0 & 0 & P_x \\ 0 & 1 & 0 & P_y \\ 0 & 0 & 1 & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



3. 依據 $\Delta A_1B_2A_2$ 所在的平面定義一個新的座標系統 xyz ：

$$\vec{n}_2 = \frac{\overrightarrow{A_1A_2}}{\|\overrightarrow{A_1A_2}\|}$$

$$\vec{n}_1 = \frac{\overrightarrow{B_1B_2}}{\|\overrightarrow{B_1B_2}\|}$$

$$\vec{n}_x = \frac{\overrightarrow{B_1 B_2}}{\|\overrightarrow{B_1 B_2}\|}$$

$$\vec{n}_y = \frac{\vec{n}_z \times \vec{n}_x}{\|\vec{n}_z \times \vec{n}_x\|}$$

$$\vec{n}_z = \frac{\vec{n}_x \times \vec{n}_2}{\|\vec{n}_x \times \vec{n}_2\|}$$

origin point : (P_x, P_y, P_z)

4. 列出 XYZ 座標系統與 xyz 座標系統之間座標轉換的轉換矩陣，並且將 B 組點資料座標系從 XYZ 座標系轉換至 xyz 座標系。

a. let: $\vec{P} = \vec{A}_1$

b. $\vec{P}, \vec{n}_x, \vec{n}_y, \vec{n}_z$ are all w.r.t XYZ axis

$$\rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} n_{xx} & n_{yx} & n_{zx} & P_x \\ n_{xy} & n_{yy} & n_{zy} & P_y \\ n_{xz} & n_{yz} & n_{zz} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.3.1)$$

$$\rightarrow \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} n_{xx} & n_{yx} & n_{zx} & P_x \\ n_{xy} & n_{yy} & n_{zy} & P_y \\ n_{xz} & n_{yz} & n_{zz} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.3.2)$$

經過轉換之後的 B 組點資料已經全部位於 xyz 座標系之中。

5. 將所有的 B 組點資料依據 Z 軸旋轉一個角度 θ_z 。

$$\Rightarrow Rot(z, \theta_z) \quad (\theta_z = \cos^{-1}(\vec{n}_1 \cdot \vec{n}_2))$$

旋轉的矩陣為：

$$\begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 & 0 \\ \sin \theta_z & \cos \theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

此時的 B 組點資料依舊全部位於 xyz 座標系中。

6. 利用座標轉換矩陣(3.3.1)將 B 組點資料從 xyz 座標系轉回 XYZ 座標系。

此時 A、B 組點資料都存在於 XYZ 座標系中。

7. 找出第三個座標系統 $x'y'z'$ 。

$$\begin{aligned} \vec{n}_{x'} &= \vec{n}_2 \\ \vec{n}_{y'} &= \frac{\vec{n}_z \times \vec{n}_{x'}}{\|\vec{n}_z \times \vec{n}_{x'}\|} \\ \vec{n}_{z'} &= \vec{n}_z \\ \text{origin point} &: (P_x, P_y, P_z) \end{aligned}$$



8. 列出 XYZ 座標系統與 $x'y'z'$ 座標系統之間座標轉換的轉換矩陣。

$\vec{P}, \vec{n}_{x'}, \vec{n}_{y'}, \vec{n}_{z'}$ are all with respect to XYZ axis

$$\rightarrow \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} n_{x'x} & n_{y'x} & n_{z'x} & P_x \\ n_{x'y} & n_{y'y} & n_{z'y} & P_y \\ n_{x'z} & n_{y'z} & n_{z'z} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \quad (3.3.3)$$

$$\rightarrow \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} n_{x'x} & n_{y'x} & n_{z'x} & P_x \\ n_{x'y} & n_{y'y} & n_{z'y} & P_y \\ n_{x'z} & n_{y'z} & n_{z'z} & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (3.3.4)$$

9. 將 B 組點資料從 XYZ 座標系統轉換至 $x'y'z'$ 座標系。

$$\vec{b}_3 = \vec{B}_3 - \vec{B}_1 \quad (\vec{b}_3 \text{ 位於 } x'y'z' \text{ 座標系統中})$$

$\vec{a}_3 = \vec{A}_3 - \vec{A}_1$ (\vec{a}_3 本來位於 XYZ 座標系統中，將其從 XYZ 座標系統轉至 $x'y'z'$ 座標系統)

投影 \vec{a}_3 與 \vec{b}_3 至 $y'z'$ 平面 (忽略 x' 方向之值)。

利用方程式 $\vec{a}_3 \cdot \vec{b}_3 = \|\vec{a}_3\| \|\vec{b}_3\| \cos \theta_x$ 可求得 θ_x 。

10. 將 B 組點資料 (位於 $x'y'z'$ 座標系統中) 依據 x' 軸旋轉 θ_x 。

$\Rightarrow Rot(x', \theta_x)$ (旋轉後 B 組點資料依舊位於 $x'y'z'$ 座標系統中)

11. 將 B 組點資料依據公式 (3.3.3) 從 $x'y'z'$ 座標系統轉至 XYZ 座標系統中；此時 A、B 兩組點資料皆位於 XYZ 座標系統中。

對位結果以第一對位點為分割線，以左暫時取左邊相機之三維點資料顯示，以右暫時取右邊相機經過轉換對位之三維點資料顯示，如圖 3.19 與 3.20 所示。



圖 3.19 對位結果

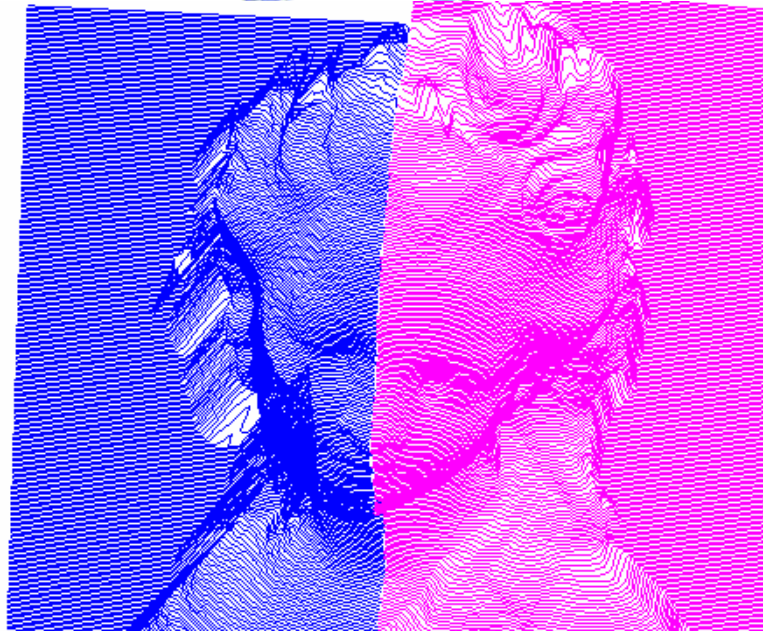


圖 3.20 對位結果-臉部特寫

3.4 重新佈點 (resampling)

由於兩組原先由掃瞄線形式組成的三維點資料，經過三維對位過程的多次平移與旋轉之後，兩組三維點資料間之關係近乎等於兩組隨機三維點資料間之關係；為了兩組三維點資料間的點資料格式有其共通性，用以方便進行整合混成的運算，必須進行重新佈點的運算。

先點選一矩型之斜對角兩點定義重新佈點之範圍 (圖 3.21)，將所有的三維點資料投影至 xy 平面，於選取之矩形範圍內自行定義佈點數量，總佈點數目

$$N_p = n \times n \text{ 。}$$

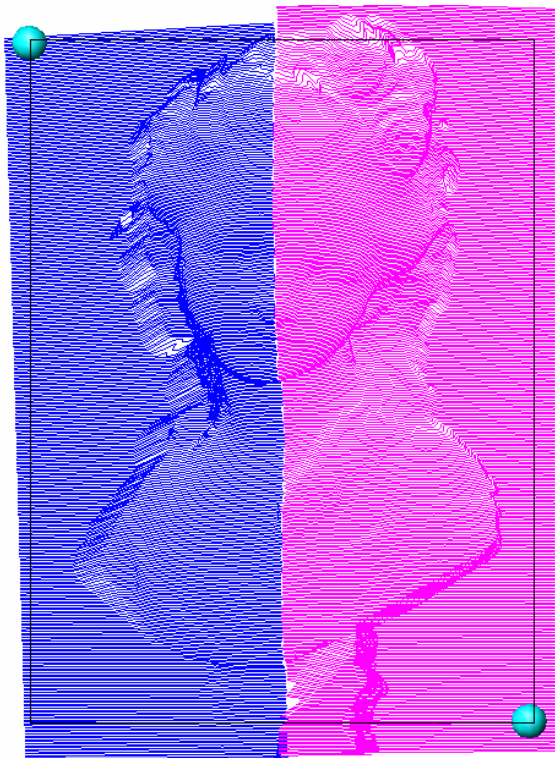
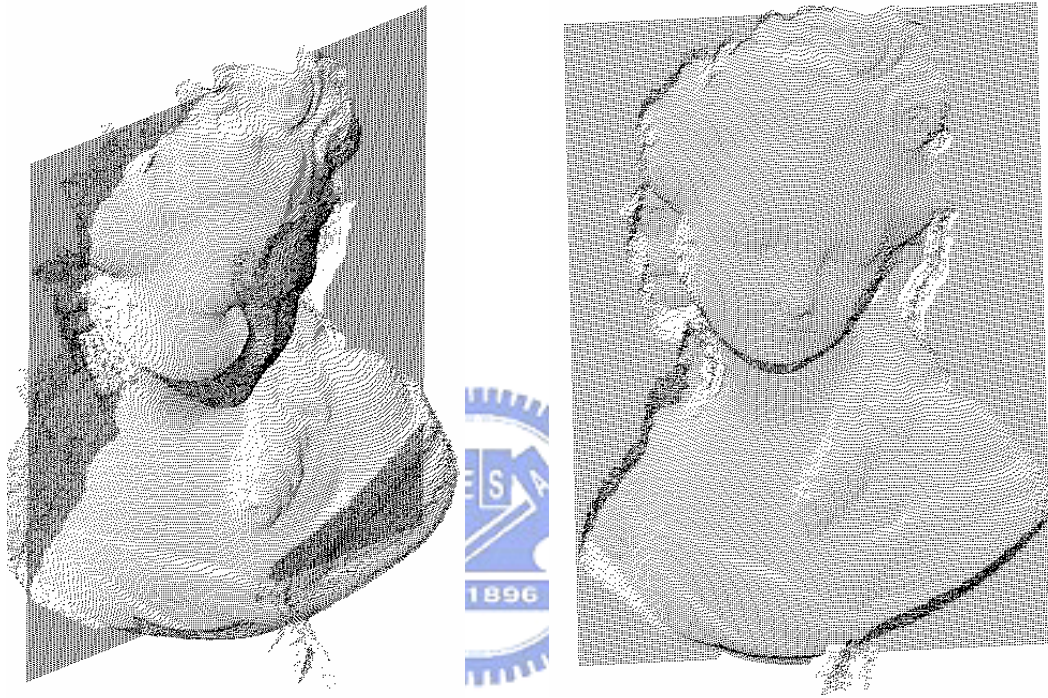


圖 3.21 選取重新佈點之矩型範圍

搜尋定義點距離投影至 xy 平面點資料之最近三點，並且透過該三點三維點資料原本之三維位置，定義出一平面；將其平面方程式之 (x, y) 帶入定義點之

(x, y) 值，藉以求得該定義點之高度值 z 。

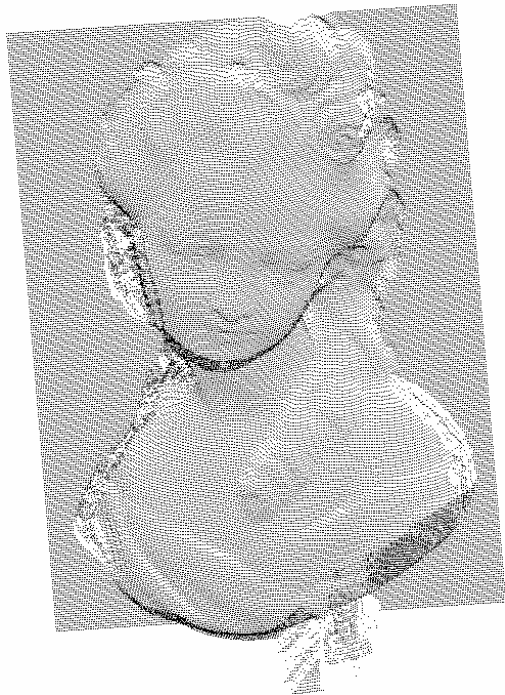
重複同樣的步驟 N_p 次，即可得到經過重新佈點之三維點資料，如圖 3.22、圖 3.23 所示。由於最後使用 AutoCAD 指令 3dmesh 來進行網格化鋪面之緣故，自行定義佈點之數目不得大於 255×255 。



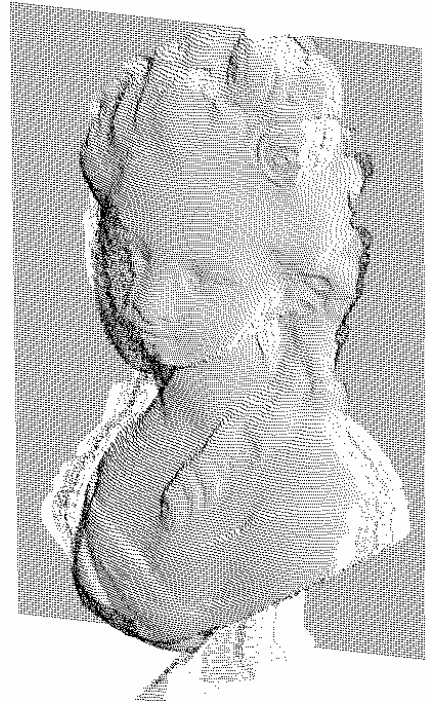
(a) 左側視角

(b) 右側視角

圖 3.22 左邊視角三維點資料經重新佈點後之三維點點狀圖



(a) 左側視角



(b) 右側視角

圖 3.23 右邊視角三維點資料經重新佈點後之三維點點狀圖



3.5 混成

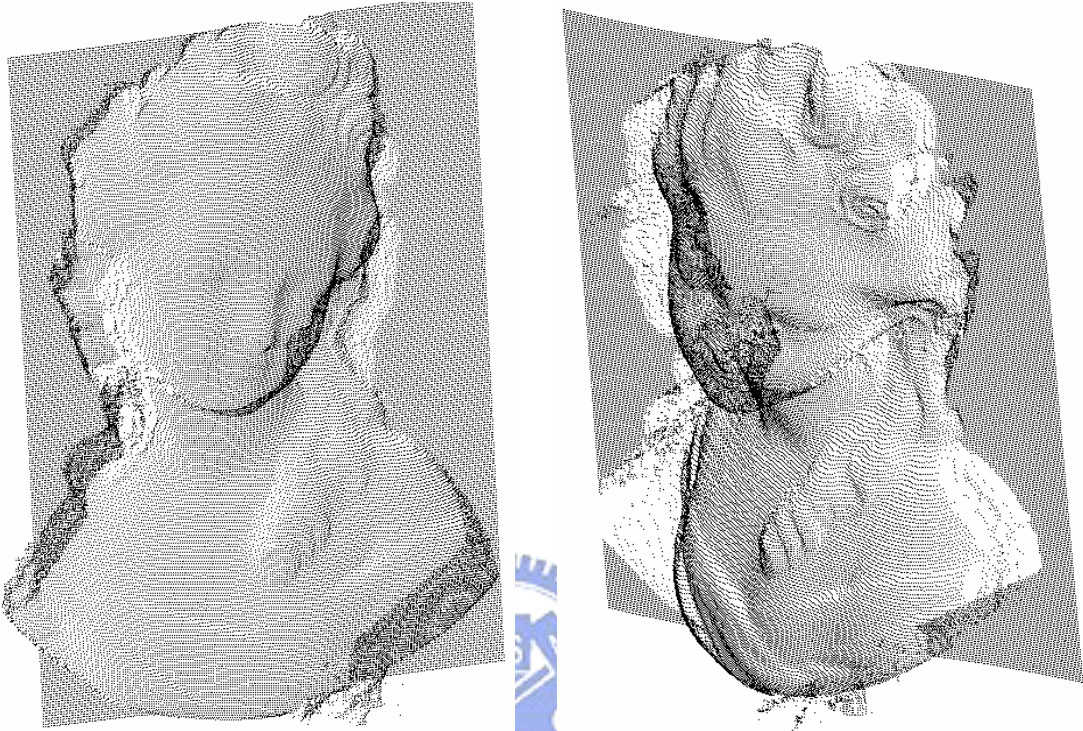
混成步驟最主要的意義在於使兩組三維點資料整合成一組三維點資料，並且不單只是做結合的動作，而是期望完成一個使整合後之三維點資料既平滑又不失真的接合運算。

依照三維點資料特性之不同，可選擇不同之混成公式；本研究於對位整合軟體中提供三種混成函數以供選擇，分別如以下所示：

(其中 S_i 為左邊視角相機取得之三維點資料中進行混成的點， D_i 為右邊視角相機取得之三維點資料中進行混成的點， B_i 為混成出的點資料， i 為點編號， $i = 0, 1, 2, \dots, N_{blend} - 1$ 。)

1. 平均混成 (Average Blending), 如圖 3.24 所示:

$$B_i = ((S_i + D_i) / 2)$$



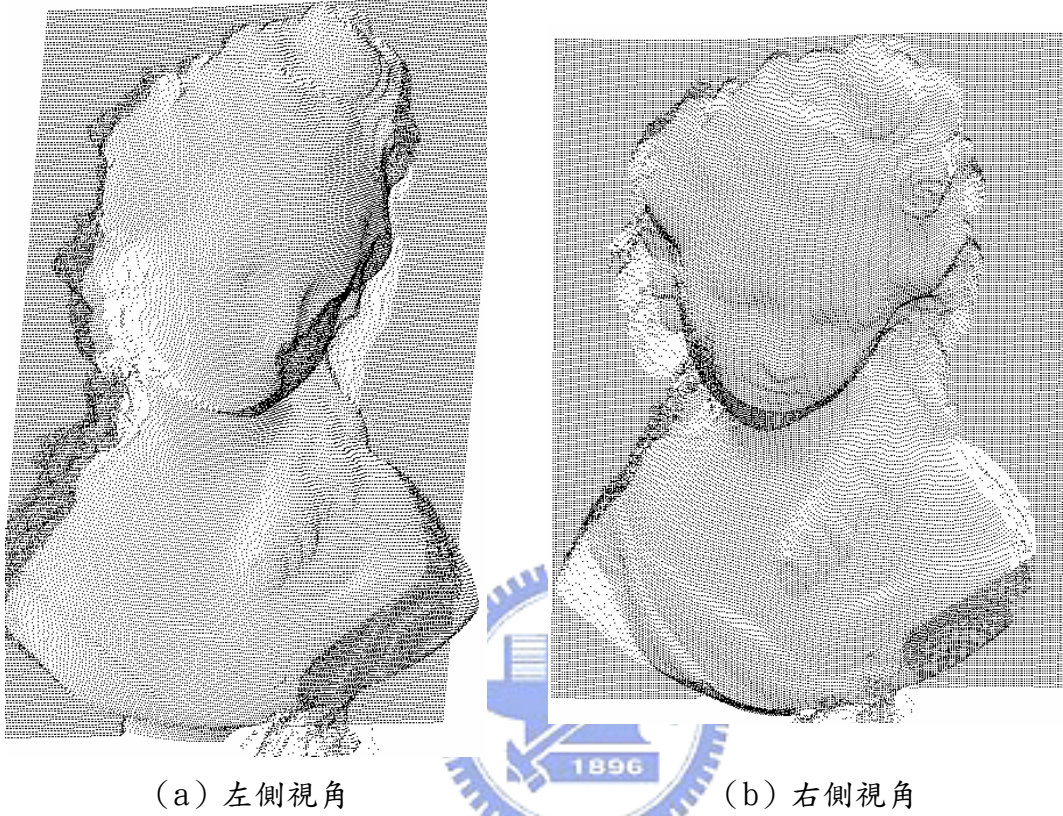
(a) 左側視角

(b) 右側視角

圖 3.24 平均混成結果點圖

2. 線性混成 (Linear blending), 如圖 3.25 所示:

$$B_i = ((N_{blend} - 1 - i) \times S_i + i \times D_i) / (N_{blend} - 1)$$



(a) 左側視角

(b) 右側視角

圖 3.25 線性混成結果點圖

3. 二次混成 (Quadratic blending) ，如圖 3.26 所示：

$$B_i = [(N_{blend} - 1 - i)^2 \times S_i + i^2 \times D_i] / [(N_{blend} - 1 - i)^2 + i^2]$$

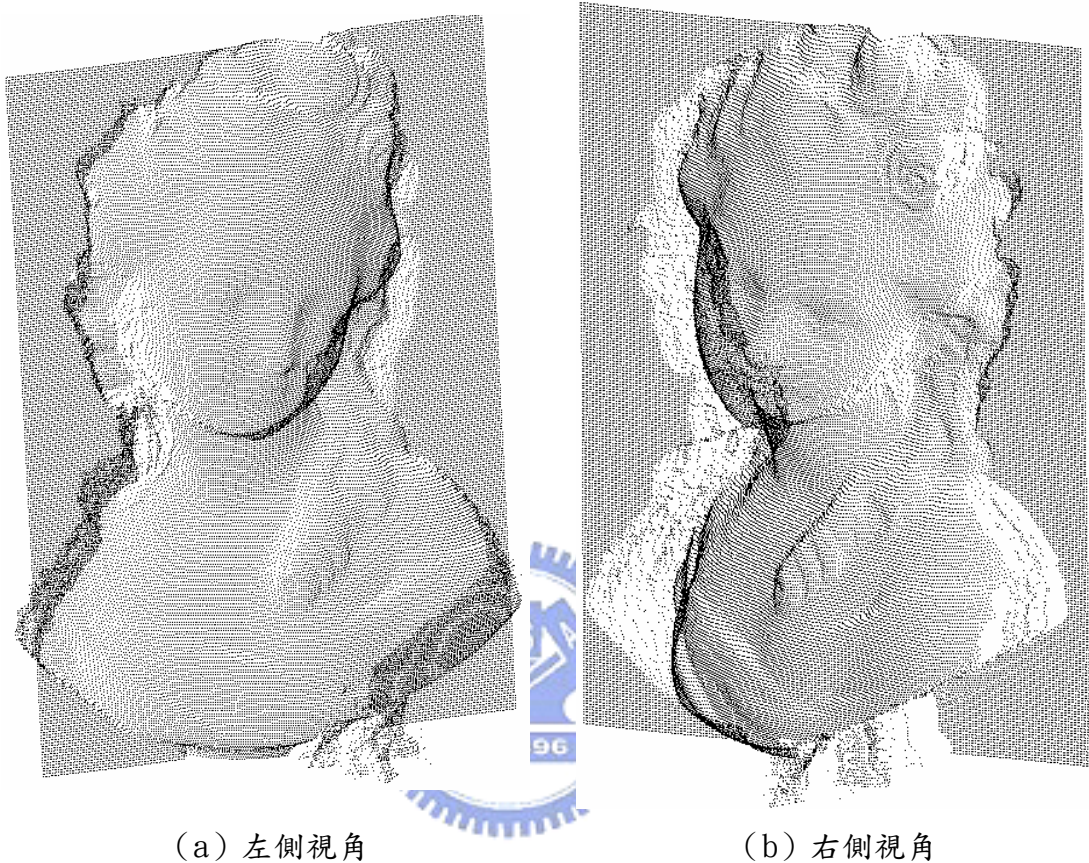


圖 3.26 二次混成結果點圖

三種函數運算結果最主要的差異點在於結果三維點資料之左右兩側與左右兩相機取得之兩組三維點資料之相關度。使用越高次的混成公式，左側之形狀與左側相機取得之三維點資料之相關度就越高，右側亦然；相關度的大小依序為二次混成 > 線性混成 > 平均混成。但若使用越高次的混成公式，兩組點資料銜接點的落差會越明顯，故必須依據三維點資料的特性來選擇使用混成之公式；對於三維點資料兩側失真較嚴重的三維點資料來說，一般是建議使用二次混成，分別取兩組點資料最準確的部分來作為最終點資料的資料點。

3.6 網格化鋪面

將三維點資料以點的形式並無法很平順的表達出真實物體的三維曲線，為了方便瀏覽物體的三維外形，必須進行網格化鋪面的運算。本研究以 AutoCAD 內建參數” 3dmesh” 進行網格化鋪面的運算，已降低網格化鋪面運算所需要之時間。

3dmesh 指令使用流程為：

1. 鍵入指令 3dmesh。
2. 輸入在 m 方向之 mesh 尺寸。
3. 輸入在 n 方向之 mesh 尺寸。由於 3dmesh 指令所能允許之尺寸大小為 2~255 之間，所以若三維點資料於 m 或 n 方向大於 255 的時候，必須將點分次輸入，依序網格化鋪面，再將其組合起來。
4. 從 m=0, n=0 開始，接著 m=0, n=1 依序將目標點資料之三維座標依序輸入，直到 m=m size, n=n size 為止，就結束 3dmesh 指令。

本研究之三維點資料整合於對位形式將以上流程寫成『Mesh Integration』、『Mesh Left』、『Mesh Right』等三個指令；『Mesh Left』為針對左邊視角相機所取得之三維點資料進行網格化鋪面，如圖 3.27 所示；『Mesh Right』為針對右邊視角相機所取得之三維點資料進行網格化鋪面，如圖 3.28 所示。『Mesh Integration』為針對整合完成之三維點資料進行網格化鋪面，分別呈現平均混成、線性混成與二次混成之網格化鋪面圖，如圖 3.29、圖 3.30 與圖 3.31 所示。

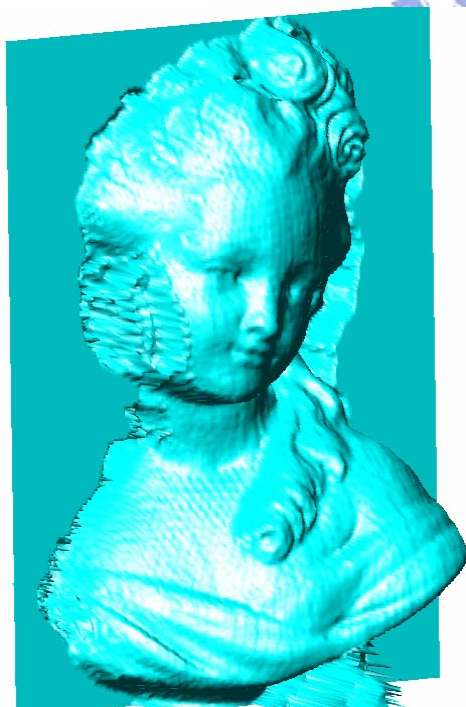


(a) 左側視角

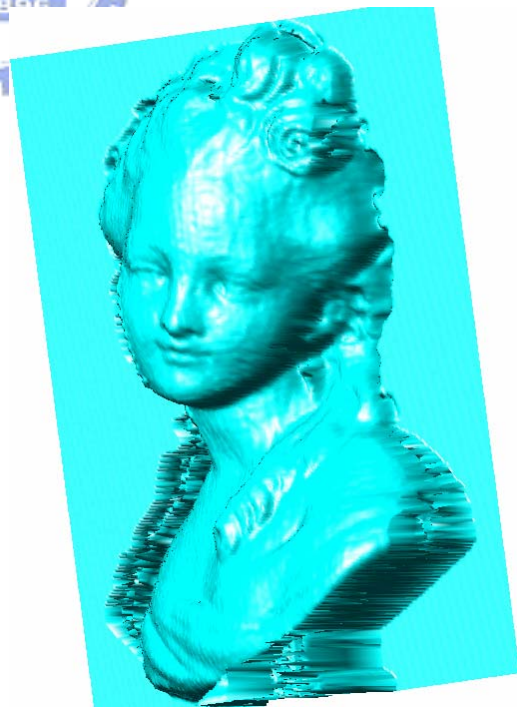


(b) 右側視角

圖 3.27 左邊視角三維點資料之三維鋪面圖

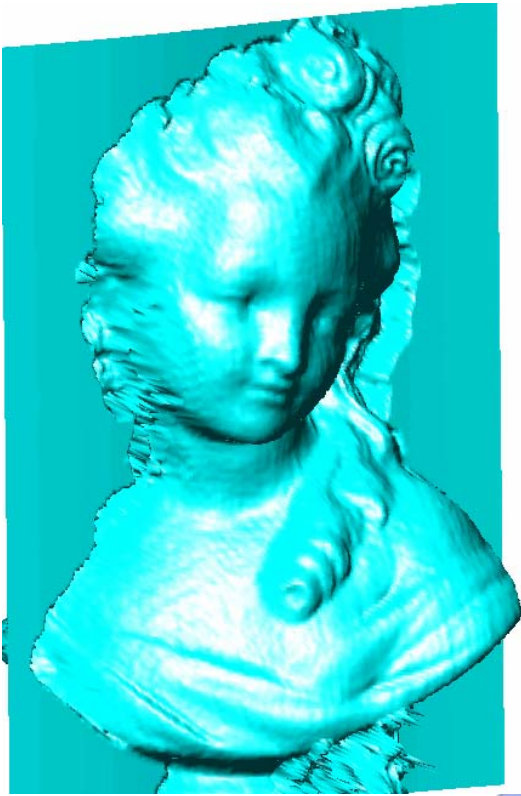


(a) 左側視角



(b) 右側視角

圖 3.28 右邊視角三維點資料之三維鋪面圖

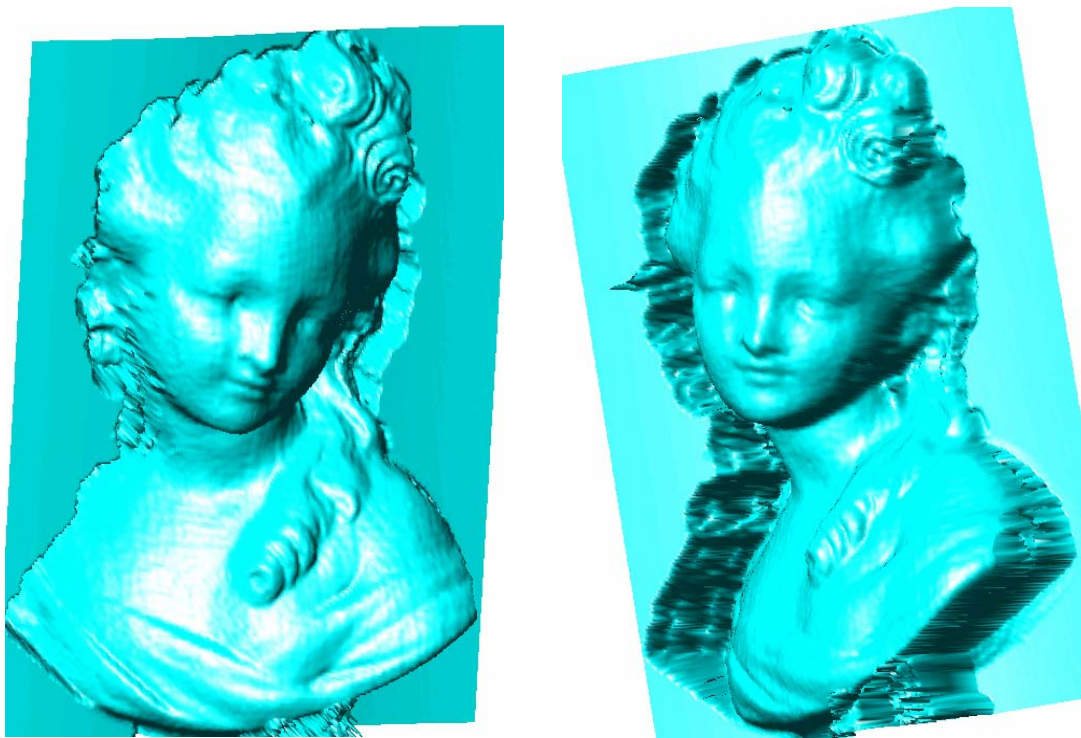


(a) 左側視角



(b) 右側視角

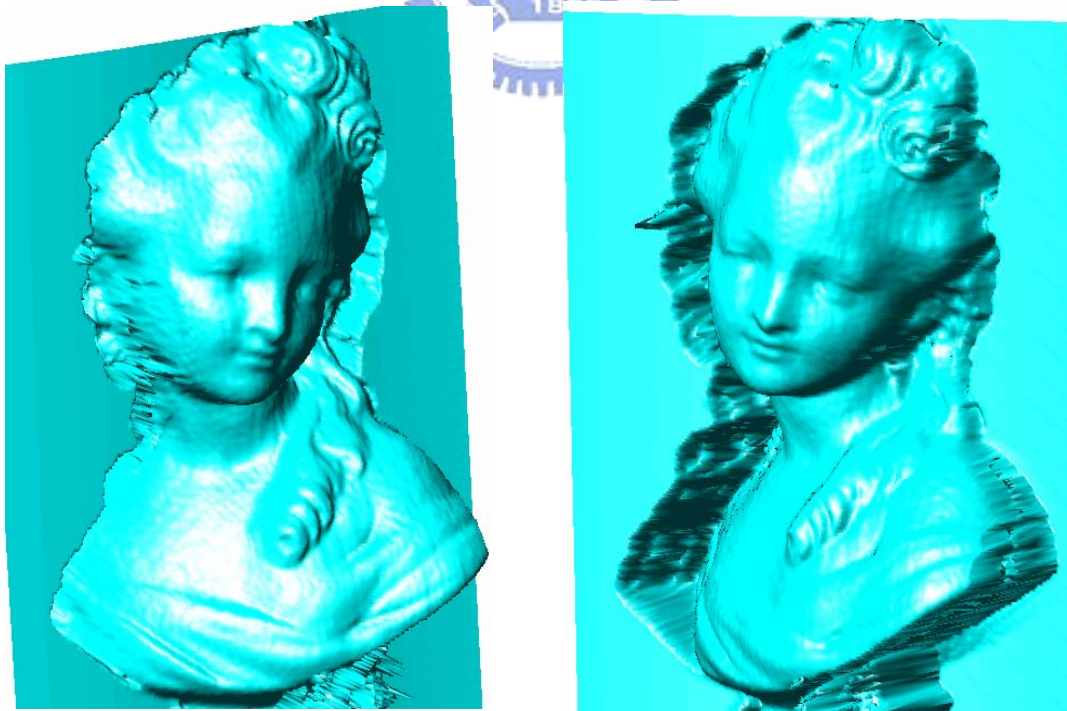
圖 3.29 平均混成結果鋪面圖



(a) 左側視角

(b) 右側視角

圖 3.30 線性混成結果鋪面圖



(a) 左側視角

(b) 右側視角

圖 3.31 二次混成結果鋪面圖

3.7 剖面圖

左右兩側視角分別藉由數位條紋投射法取得之三維外形點資料經過對位、整合與網格化鋪面等運算後，已可完全於 CAD 軟體中呈現出待測物體之三維外形，不過若要對於待測物體進行分析比較，將其進行剖面的流程則不可缺少。

將物體網格化鋪面之圖形繪出後，點選欲取得水平剖面圖之水平位置，如圖 3.32 所示；利用程式運算找出該水平位置之上下掃描線，並且由內插法求出欲求之水平位置掃描線，如圖 3.33 所示；將原本之網格化鋪面圖刪除，如圖 3.34 所示，；再將選取點特徵與上下兩條掃描線刪除，即得所求之剖面圖，如圖 3.35 所示。



圖 3.32 點選得水平剖面圖之水平位置

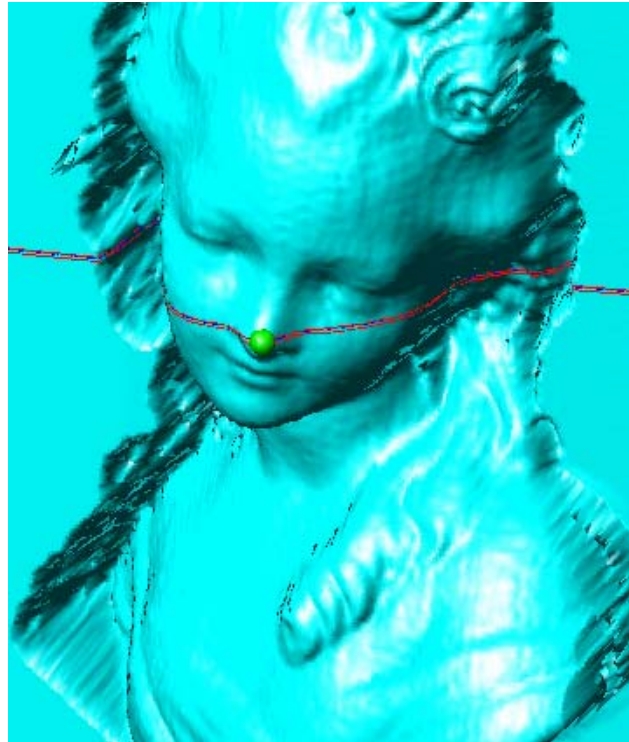


圖 3.33 計算並繪出剖面圖

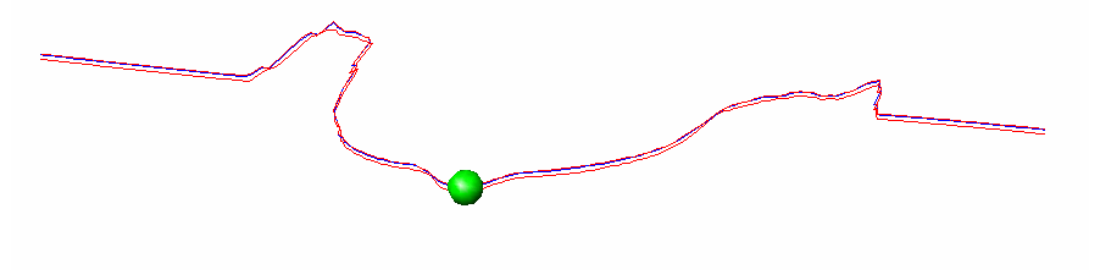


圖 3.34 剖面圖刪除網格化鋪面

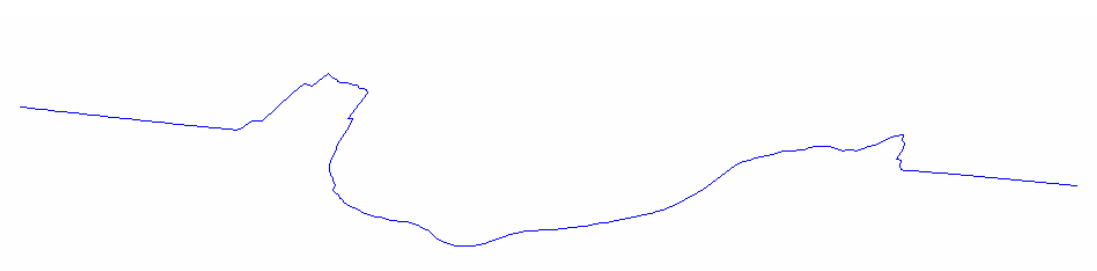


圖 3.35 剖面圖結果呈現

四、醫學工程應用之胸腔三維取像案例

本論文研究與長庚醫院進行醫學工程合作，對於漏斗胸的兒童病患進行手術治療前後的胸腔量測，藉由此來觀察手術治療後漏斗胸兒童病患胸腔凹陷改良的程度與情況。其中影像中頸部附近的白色半球狀物體為因應三維點資料後處理所黏貼之定位標記。通常於胸腔三維外形取像案例之對位中，除了頸部附近的白色半球狀物體為明顯對位依據之外，本研究多半選用胸部尖端突出之乳頭來作為完成三點對位所需之額外兩點，不過實際的三維點資料對位特徵依據點選之部位，會因每一個個案案例不同而有所差異。

本論文發展之胸腔外形三維點資料之處理與應用軟體，可以將分別以左右相機為取像視角經過條紋投射法所擷取之三維外形點資料進行處理與應用，將之整合成一組視角較為寬廣，取像結果較為可靠之三維外形網格化鋪面圖。

此套軟體之運算處理流程圖如圖 4.1 所示，其中流程『取得對位特徵點』由於內部流程判別稍微複雜，將其流程圖額外使用圖 4.2 表示之。

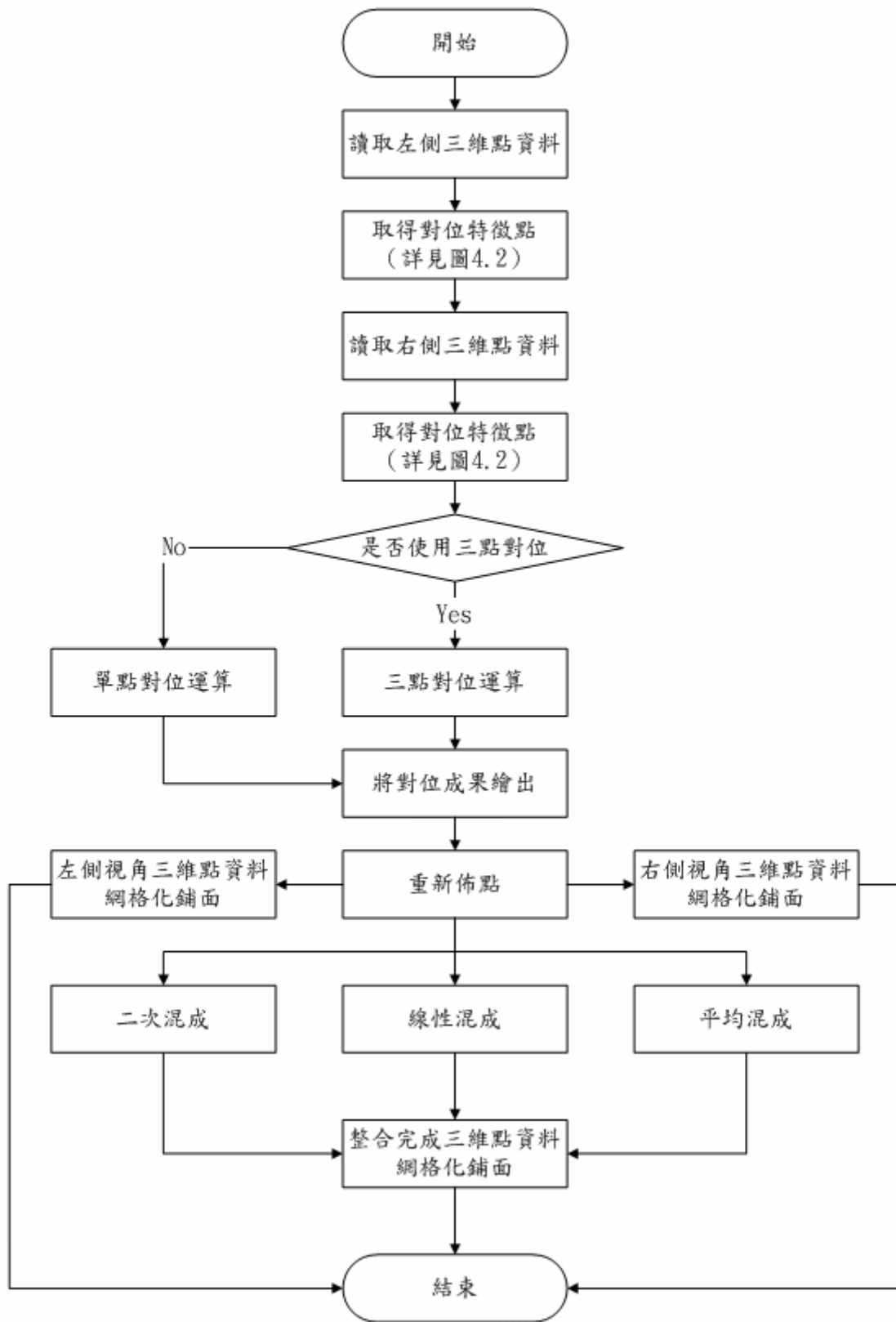


圖 4.1 胸腔外形三維點資料之處理與應用軟體運算流程圖

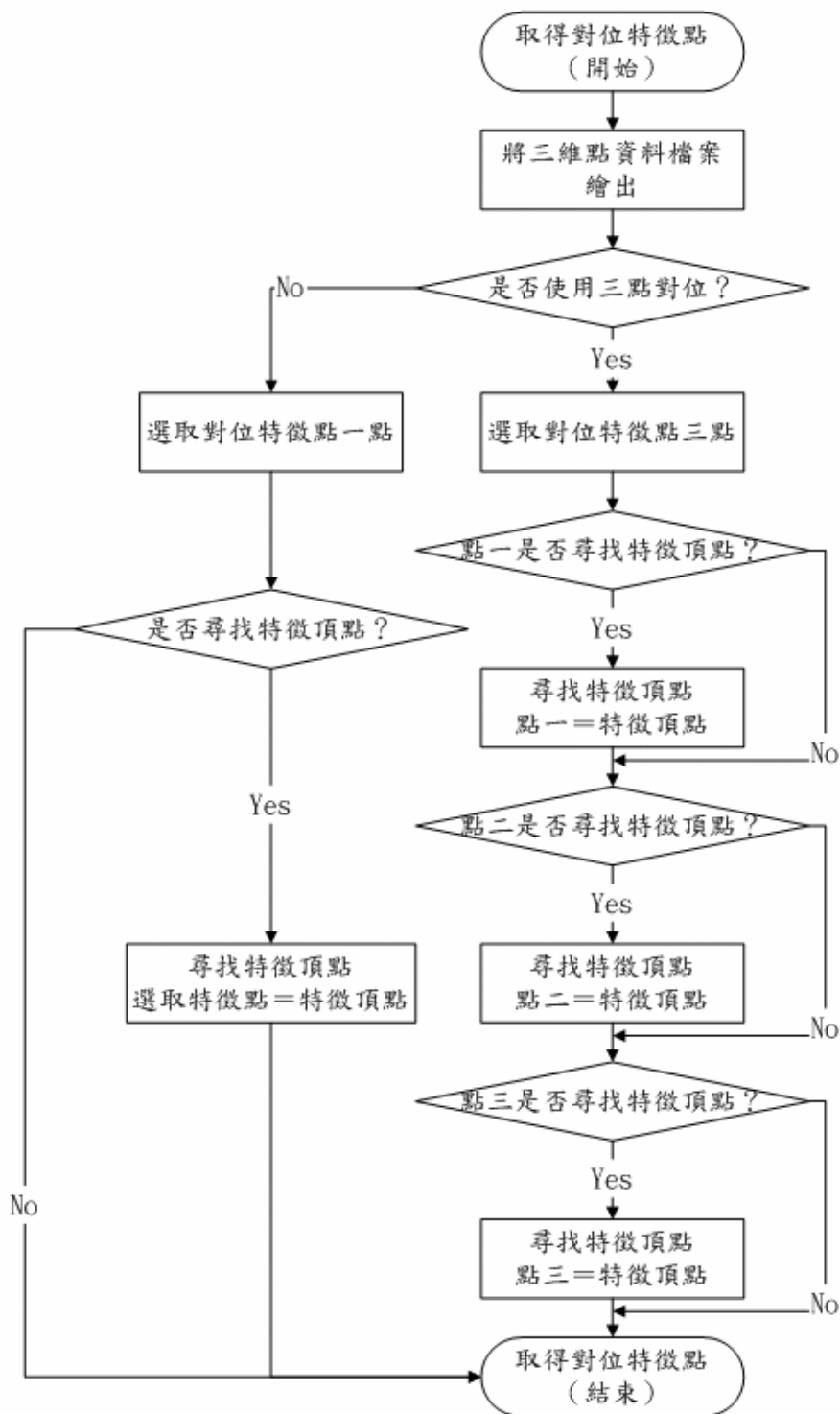


圖 4.2 取得特徵點之運算流程圖

本論文研究與長庚醫院之醫學工程合作上處於合作初期，目前為止一共取得了二十三組案例，透過這二十三組案例的處理，針對其應用特性來發展三維取像的環境，希望此技術發展成熟後能於醫院經過簡單的軟體教學為醫院員工所使用。

取其中同一位漏斗胸男童病患手術前後的兩組案例與漏斗胸青春期女性病患矯正後為展示案例：

4.1 漏斗胸男童病患手術前

- 由於至於頸部下方用於定位之特徵物尺寸比起以往之對位特徵點相對狹小，故必須調降尋找一個合適拋物面所需要的點資料數量，從預設值 200 輸入改成 50，如圖 4.3 所示。
- 在左側相機取得之三維點資料上，依序點選特徵白球、左邊乳頭與右邊乳頭三點為對位依據特徵點，如圖 4.4 所示。

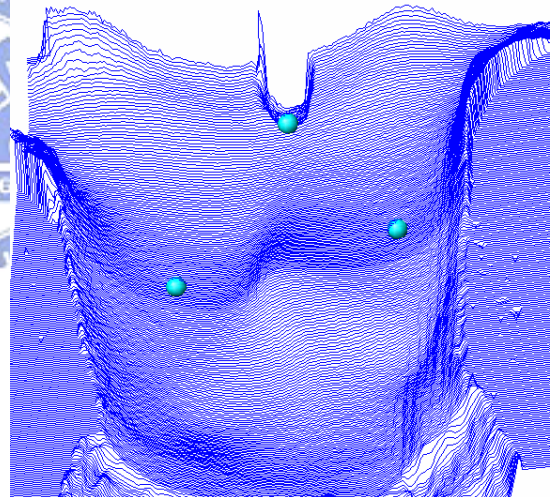
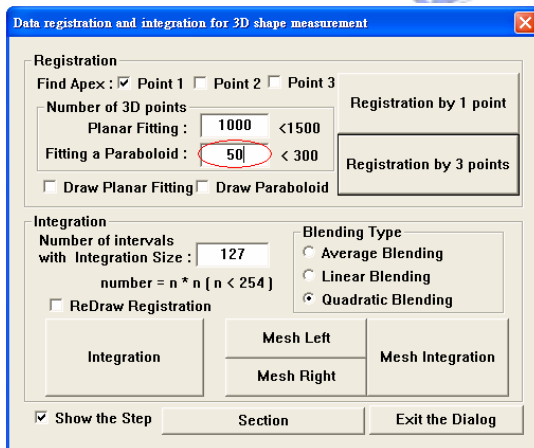


圖 4.3 拋物面所需點資料
預設值 200 輸入改成 50

圖 4.4 依序點選對位依據特徵點

- 特徵點之特徵方向，如圖 4.5 與圖 4.6 所示。

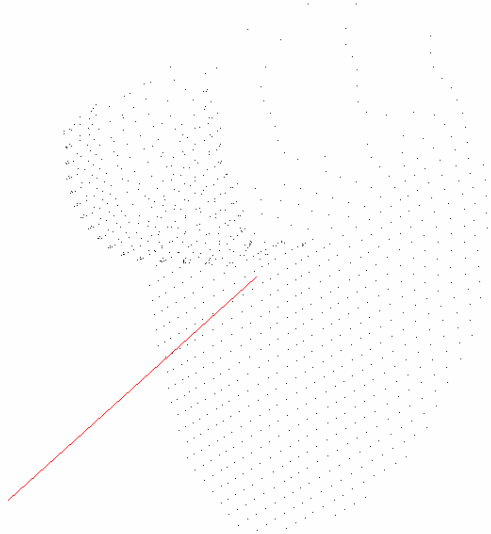


圖 4.5 特徵點之特徵方向

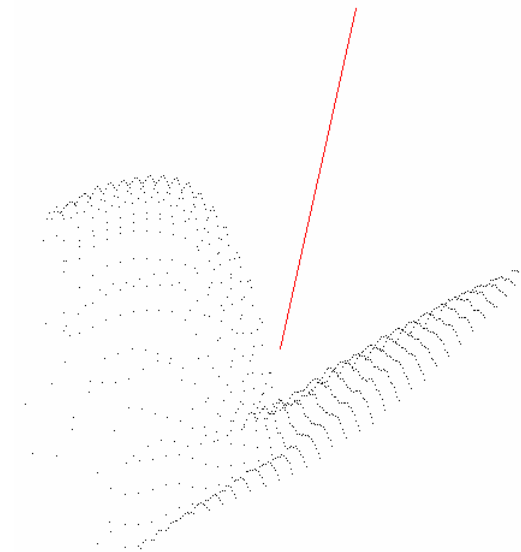


圖 4.6 特徵點之特徵方向

- d. 找出合適之拋物面並且計算出拋物面於 Z 方向之極值，如圖 4.7 所示。
- e. 在右側相機取得之三維點資料上，如處理左側向機之三維點資料一般，依序點選特徵白球、右邊乳頭與左邊乳頭三點為對位依據特徵點，如圖 4.8 所示。並且重複以上步驟。



圖 4.7 合適之拋物面

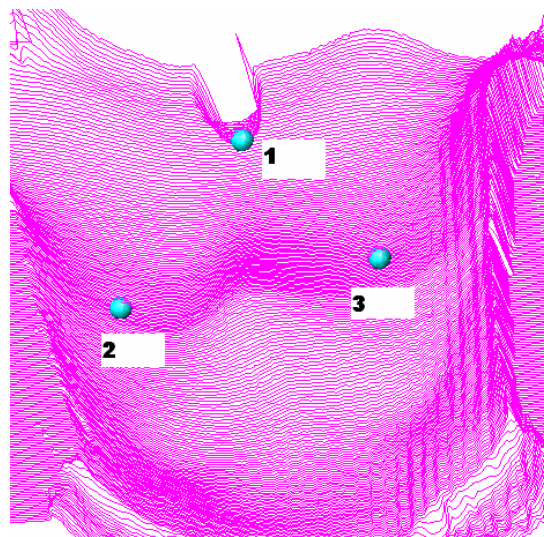


圖 4.8 依序點選對位依據特徵點

f. 完成三點對位，如圖 4.9 與圖 4.10 所示。

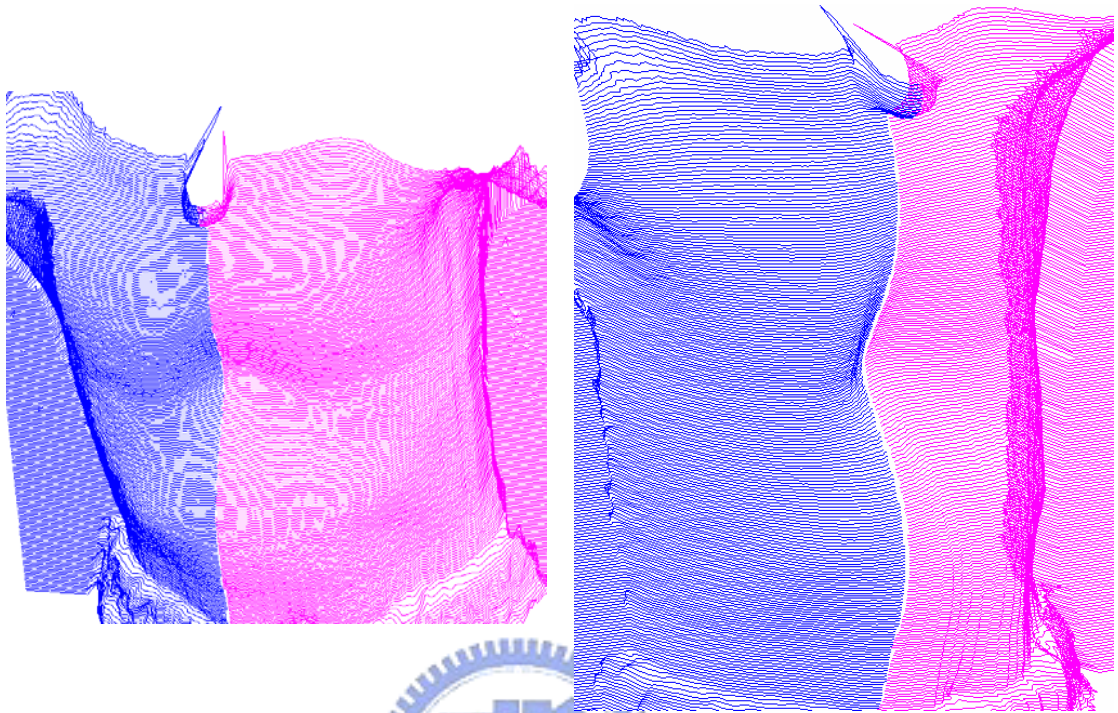


圖 4.9 完成三點對位

圖 4.10 完成三點對位

g. 對於左右相機分別取得之三維點資料進行個別鋪面。左側相機取得之三維點資料胸腔右側明顯有些許的模糊與皺摺，如圖 4.11 與圖 4.12 圈圈處所示一般；而右側相機亦然，胸腔左側之三維點資料有明顯之不平滑，如圖 4.13 與圖 4.14 所示；於條紋投射法中我們稱之為圖像破裂，多半是由於該二維視角無法讀取該區域之三維點資料，或者投射光被該區域特殊色彩、材質所吸收而造成。

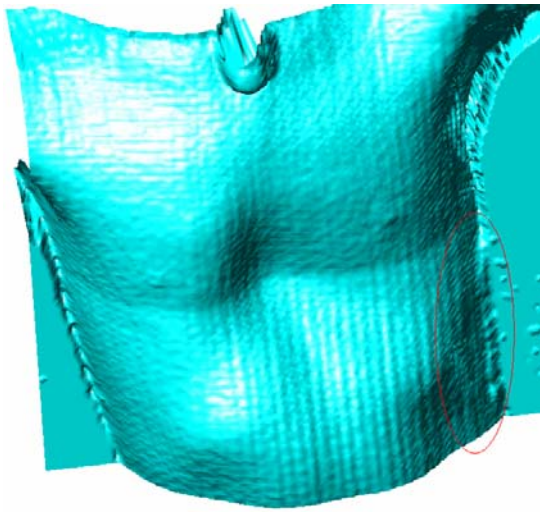


圖 4.11 左側相機取得之三維點資料

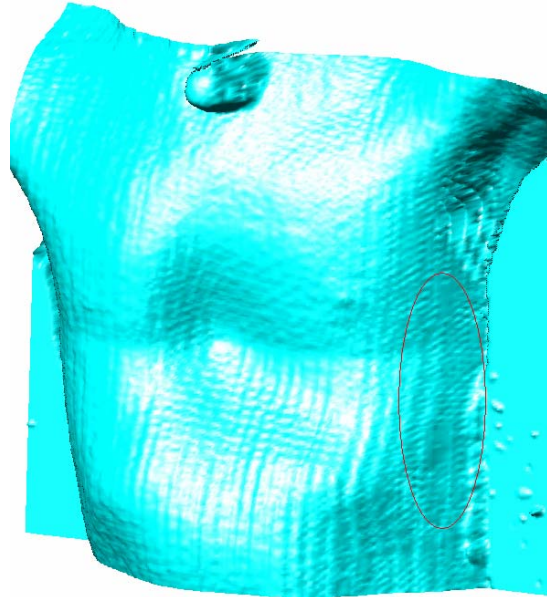


圖 4.12 左側相機取得之三維點資料

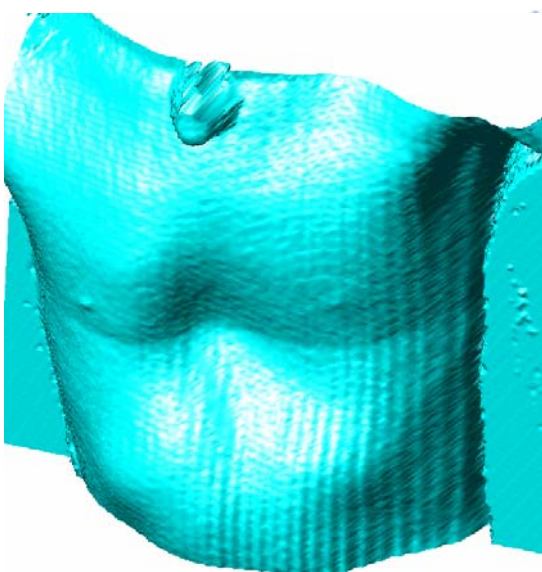


圖 4.13 右側相機取得之三維點資料

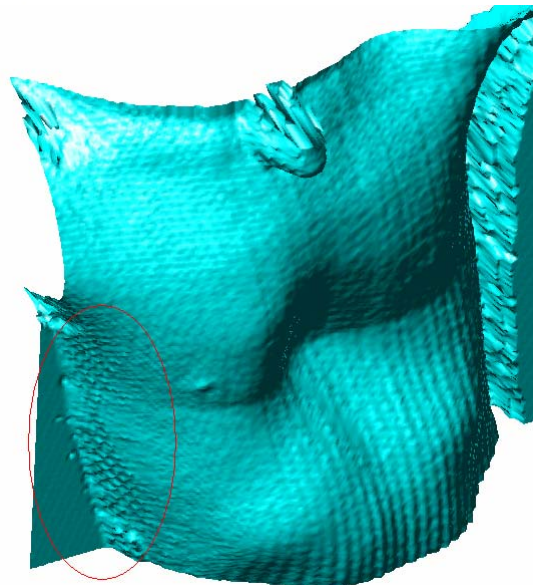


圖 4.14 右側相機取得之三維點資料

h. 由於兩側相機所取得之三維點資料皆各自有其破損或取向模糊之區域，故本範例於整合時使用二次混成，分別取其準確寫實之三維點資料區域來進行整合，整合結果如圖 4.15 與圖 4.16 所示，明顯於胸腔兩側三維外形都沒有出現破裂或失真的情形。

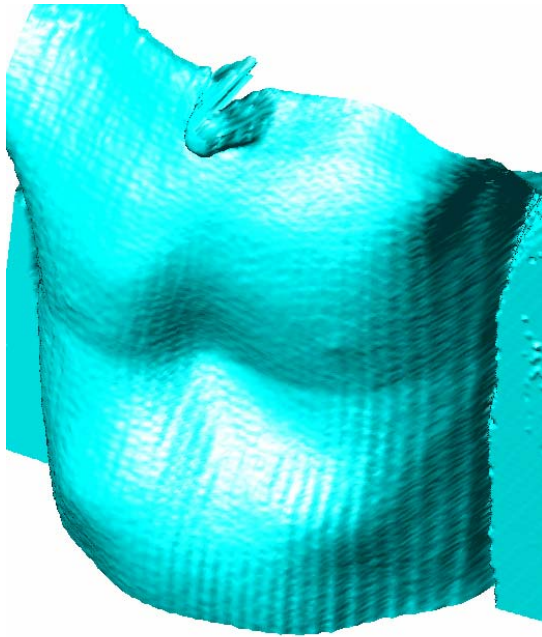


圖 4.15 手術前胸腔三維外形展示

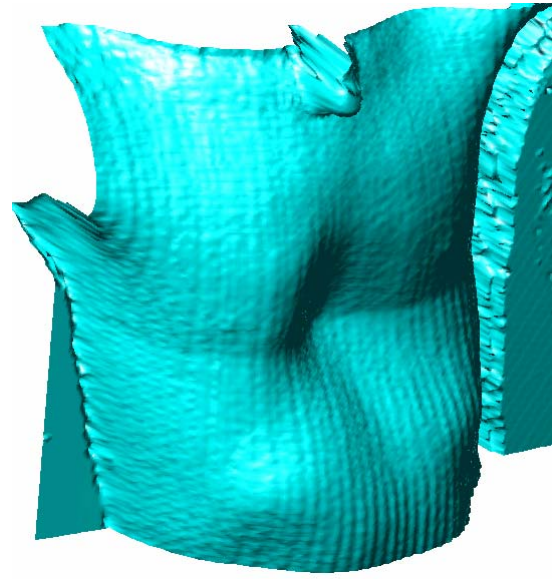


圖 4.16 手術前胸腔三維外形展示

4.2 漏斗胸男童病患手術矯正後



- a. 圖 4.17 與圖 4.18 為手術治療後之條紋影像，其中影像中頸部附近的白色物體為定位標記，而胸腔附近的白色物體為手術治療後使用的膠帶，照片左側之黃色與透明管線為點滴管線。此兒童病患於手術後三天隨即進行胸腔形狀改良之三維外形取像，由於手術傷口依舊劇痛之緣故，此兒童病患並無法處於完全靜止不動的狀態進行取像，故所得之三維點資料檔有些許的誤差，不過仍可看出手術明顯的改善幅度。

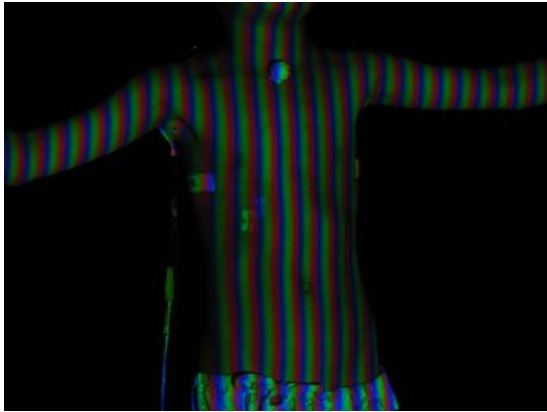


圖 4.17 左視角條紋影像



圖 4.18 右視角條紋影像

- b. 由於至於頸部下方用於定位之特徵物尺寸比起手術前較為大些，但對於以往之對位特徵點仍相對狹小，故仍須調降尋找一個合適拋物面所需要的點資料數量，從預設值 200 輸入改成 100，如圖 4.19 所示。

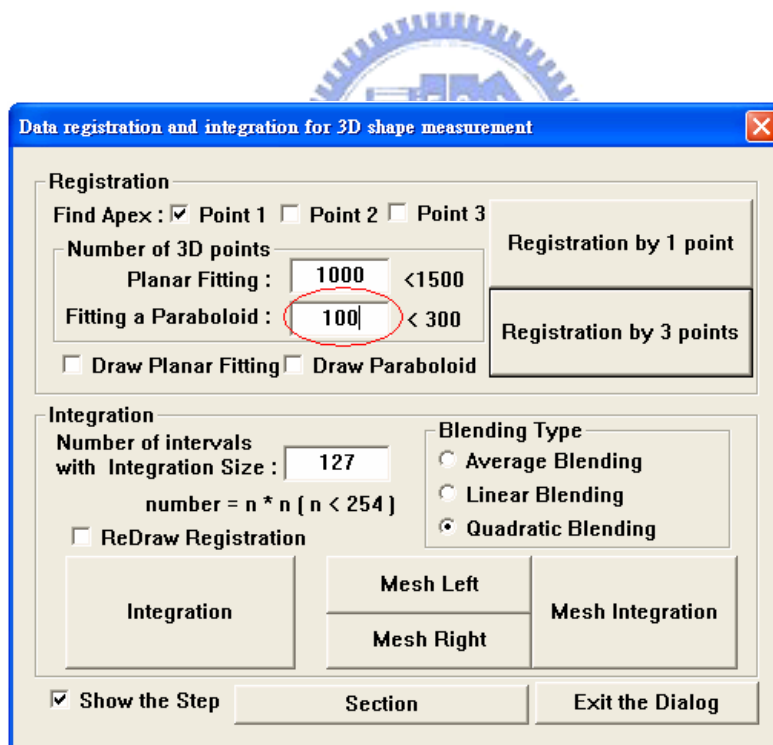


圖 4.19 拋物面所需點資料預設值 200 輸入改成 100

- c. 在左側相機取得之三維點資料上，依序點選特徵點；由於兒童病患於取像時有因為手術疼痛而造成晃動之緣故，左側乳頭之特徵並不如其他組胸腔三維點資

料來的明顯，故選擇肚臍作為第二對位特徵點，依序點選特徵白球、肚臍與右邊乳頭三點為對位依據特徵點，如圖 4.20 所示。而右邊向機取得之三維點資料亦然，重複上述步驟，如圖 4.21 所示。

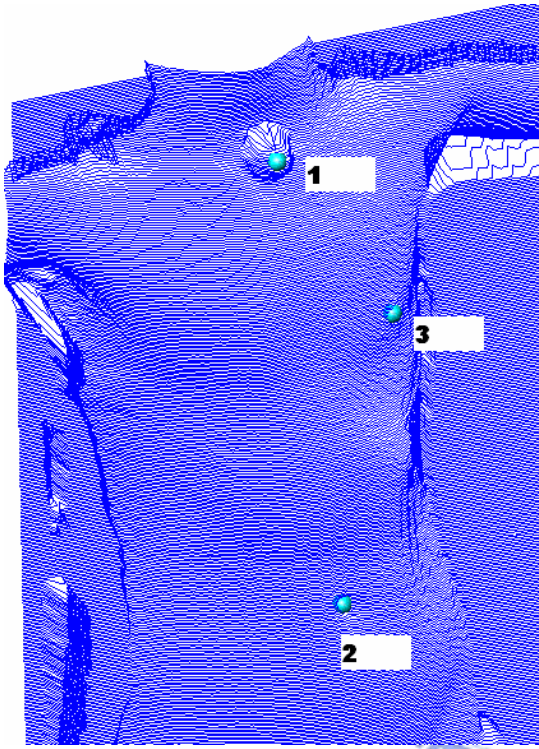


圖 4.20 依序點選對位依據特徵

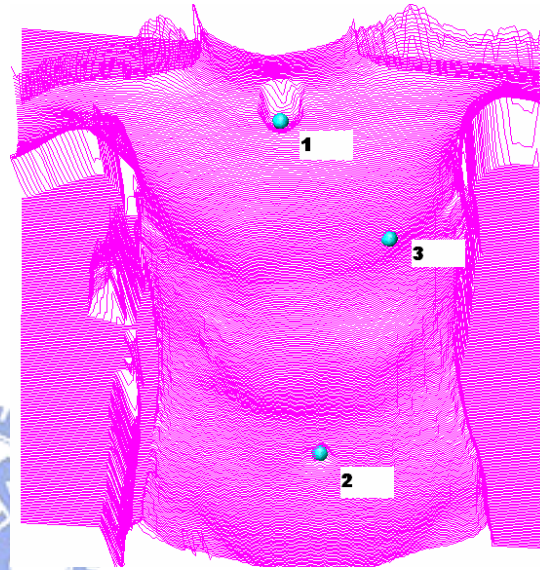


圖 4.21 依序點選對位依據特徵

d. 完成三點對位，如圖 4.22 與圖 4.23 所示。

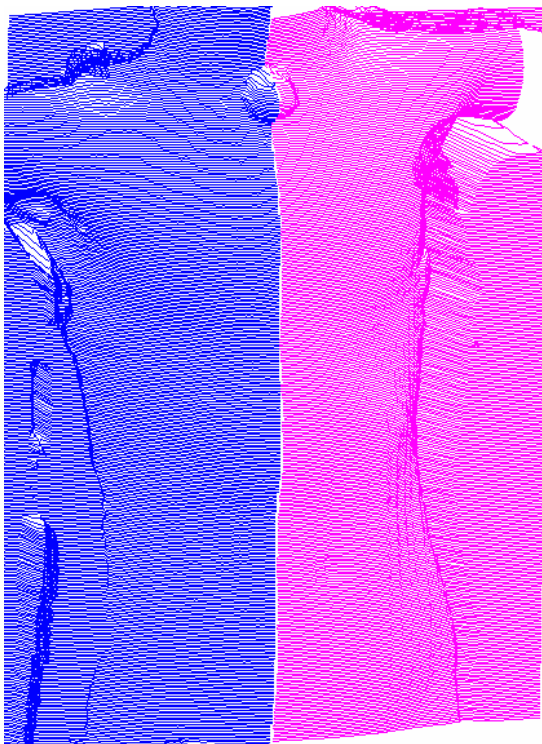


圖 4.22 完成三點對位

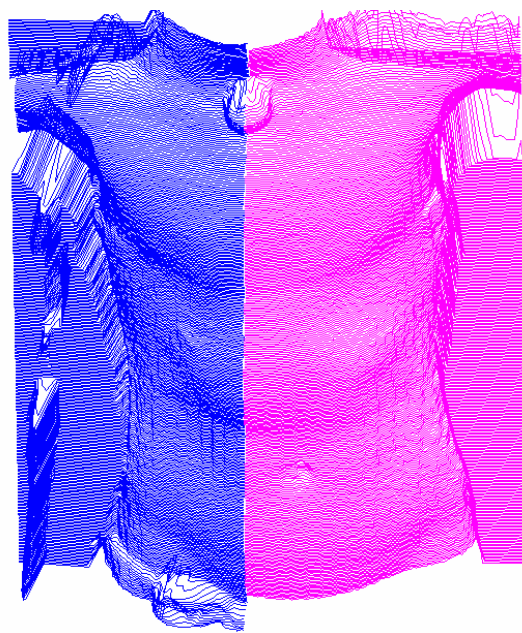


圖 4.23 完成三點對位

- e. 對於左右相機分別取得之三維點資料進行個別鋪面。左側相機取得之三維點資料胸腔右側基本上只有些許的模糊，如圖 4.24 與圖 4.25 圈圈處所示一般；但右側相機則不然，胸腔左側之三維點資料有明顯之不平滑，甚至是大篇幅的破裂，如圖 4.26 與圖 4.27 所示。

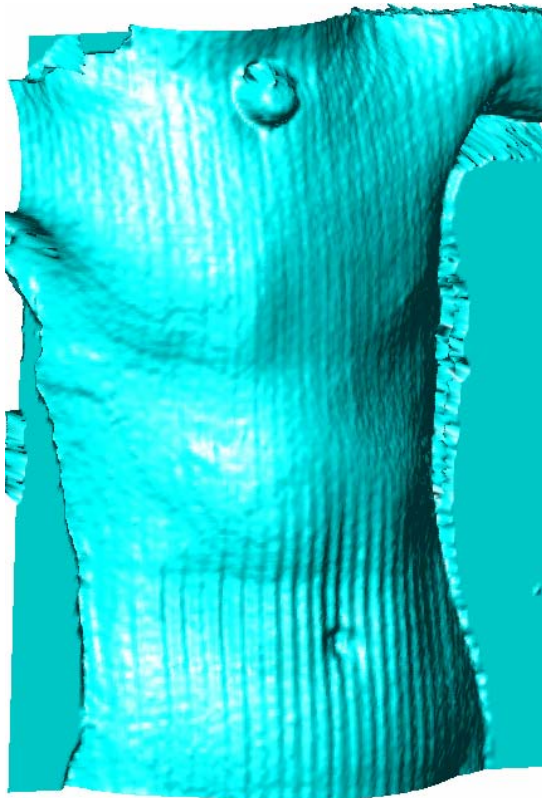


圖 4.24 左側相機取得之三維點資料

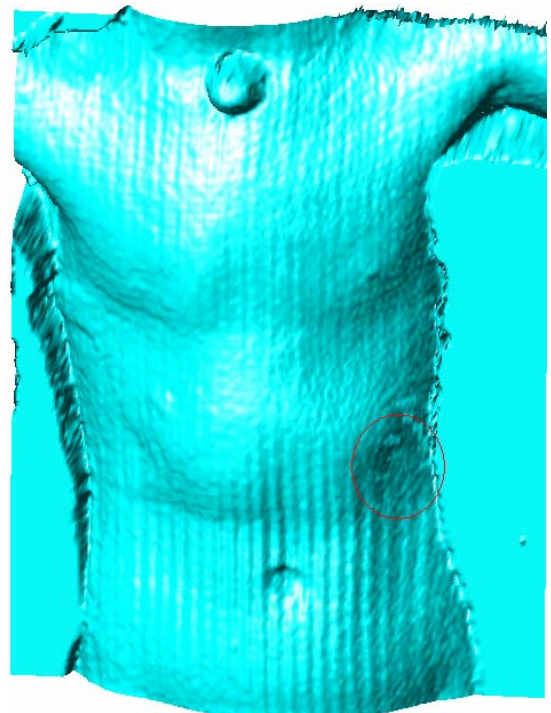


圖 4.25 左側相機取得之三維點資料

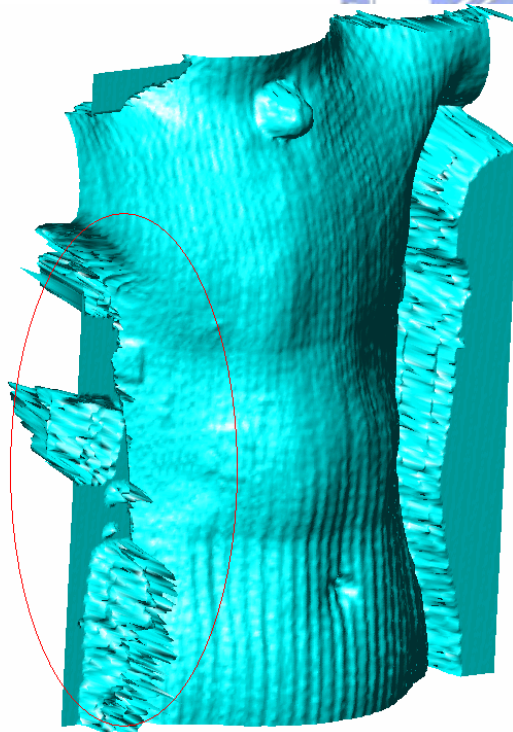


圖 4.26 右側相機取得之三維點資料

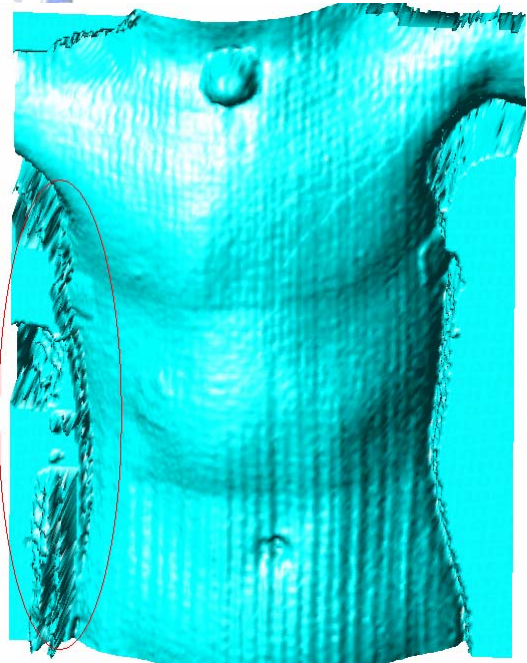


圖 4.27 右側相機取得之三維點資料

f. 由於右側相機所取得之三維點資料左側破裂情況過於嚴重，為了避免右側相機取得之三維點資料左側部分被呈現於結果上，我們依舊使用二次混成來降低右側相機點資料於胸腔左側所佔之權重；只取其準確寫實之三維點資料區域來進行整合，整合結果如圖 4.28 與圖 4.29 所示，明顯於胸腔左側三維外形沒有出現原先破裂或失真的情形。

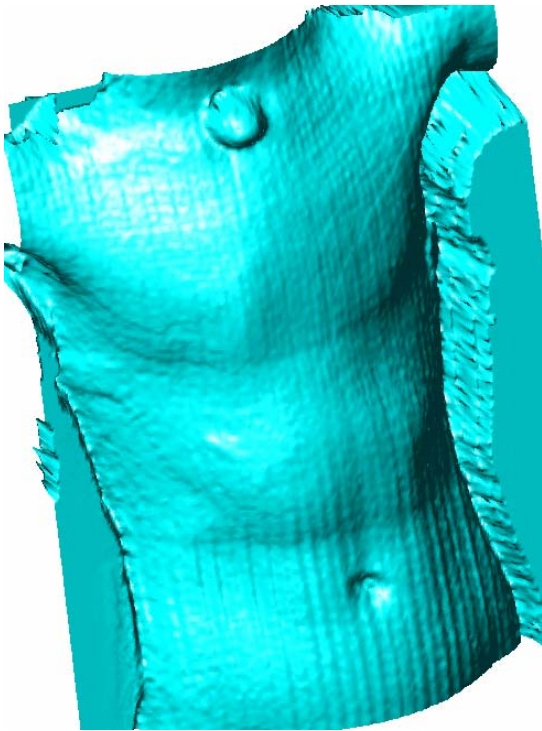


圖 4.28 手術後胸腔三維外形展示

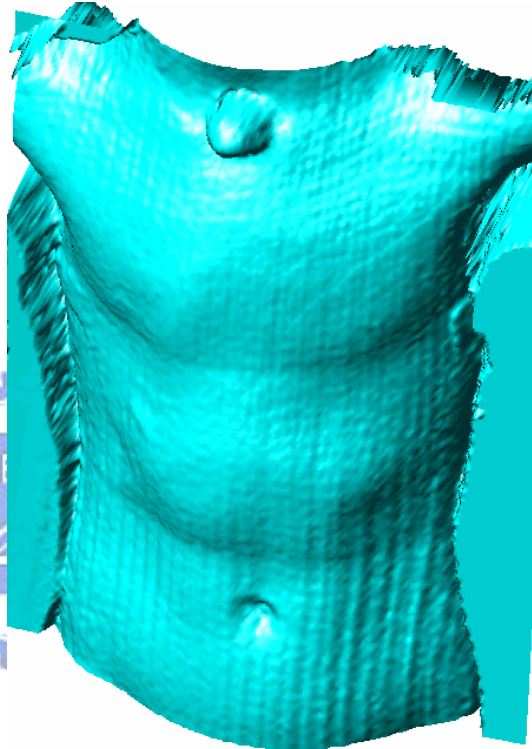


圖 4.29 手術後胸腔三維外形展示

4.3 手術前後之比較

醫學工程治療漏斗胸病患之手術流程簡述如下：

一、利用微創手術將手術器具從胸腔右側穿入，左側穿出，如圖 4.30 所示。

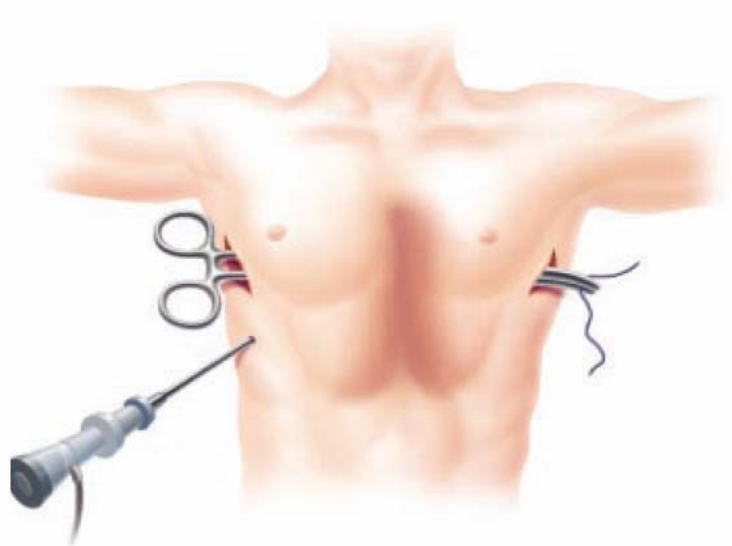


圖 4.30 微創手術將手術器具穿過胸腔[38]

二、利用已穿過胸腔之手術器具將固定桿件以牽引之方式置入胸腔，如圖 4.31 所示。

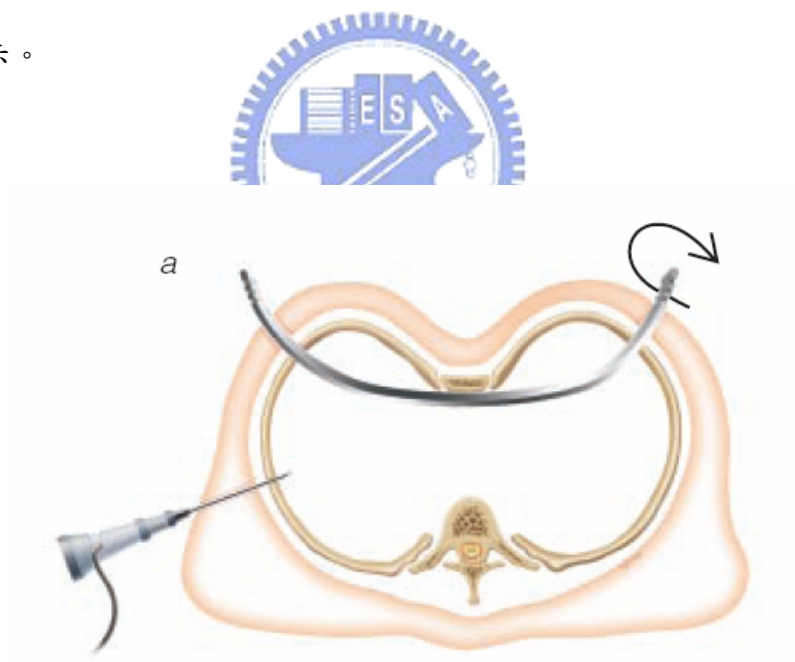


圖 4.31 將固定桿件置入胸腔[38]

三、將固定桿件做 180 度之旋轉，利用固定桿件將胸腔稱起，如圖 4.32 所示，手術完成。



圖 4.32 固定桿件旋轉將胸腔稱起[38]

觀察手術治療成效之最簡易方法即為比較手術前後胸腔之剖面圖，於手術前後分別取相對應之三剖面圖進行比較，已檢視手術之成效。由於手術前後取像範圍皆有所差異，故以乳頭、胸部與腹腔交界與胸部與腹腔交界與乳頭之中點三點分別為三剖面圖位置之比較依據。圖 4.33 與圖 4.35 為乳頭位置圖，圖 4.34 為手術前乳頭高度剖面圖，圖 4.36 為手術後乳頭高度剖面圖，圖 4.37 與圖 4.39 為胸部與腹腔交界位置圖，圖 4.38 為手術前胸部與腹腔交界高度剖面圖，圖 4.40 為手術後胸部與腹腔交界高度剖面圖，圖 4.41 與圖 4.43 為乳頭與胸部、腹腔交界中點位置圖，圖 4.42 為手術後乳頭與胸部、腹腔交界中點高度剖面圖，圖 4.44 為手術後乳頭與胸部、腹腔交界中點高度剖面圖。

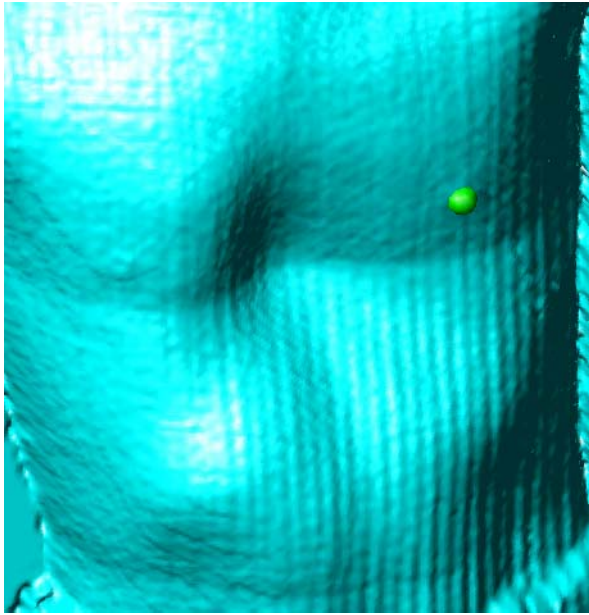


圖 4.33 乳頭位置圖

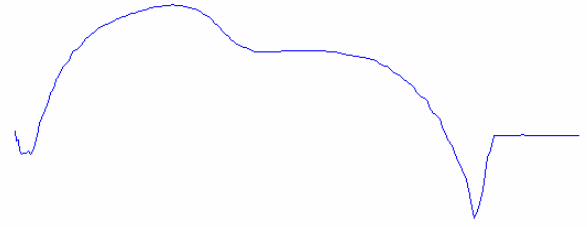


圖 4.34 手術前乳頭高度剖面圖

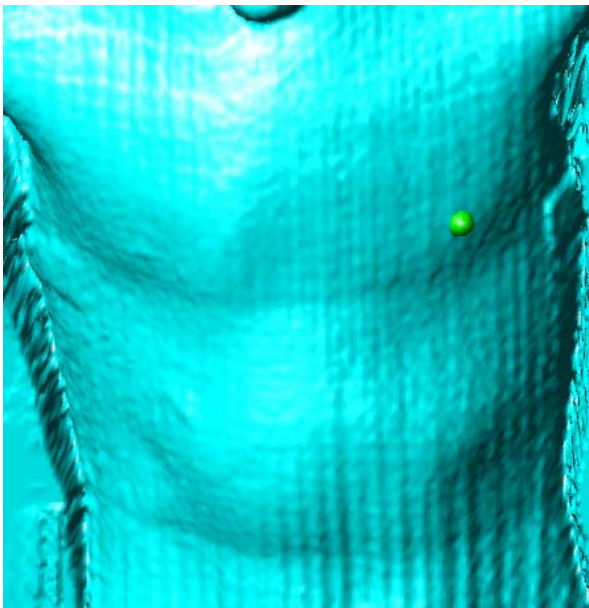


圖 4.35 乳頭位置圖

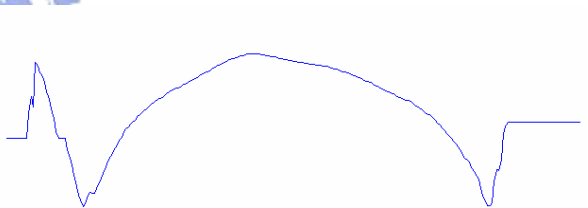


圖 4.36 手術後乳頭高度剖面圖

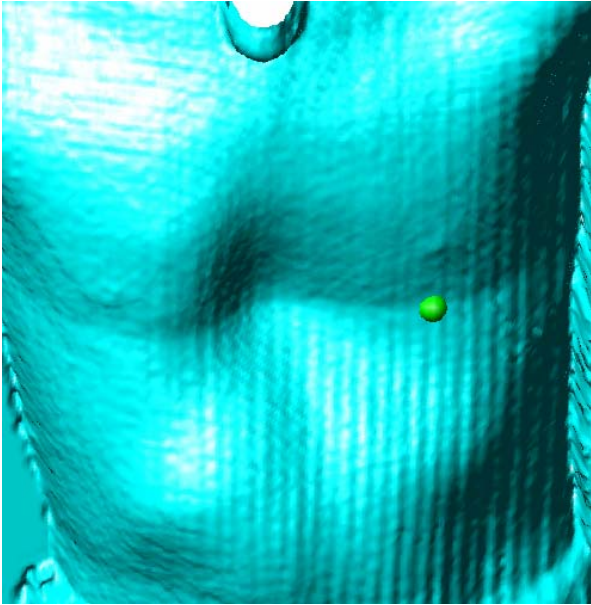


圖 4.37 胸部與腹腔交界位置圖

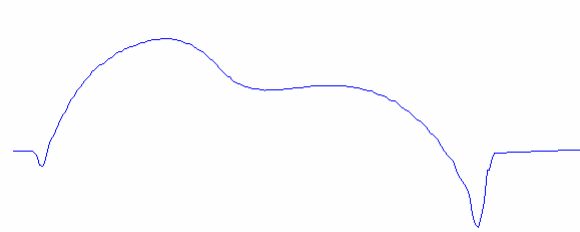


圖 4.38

手術前胸部與腹腔交界高度剖面圖

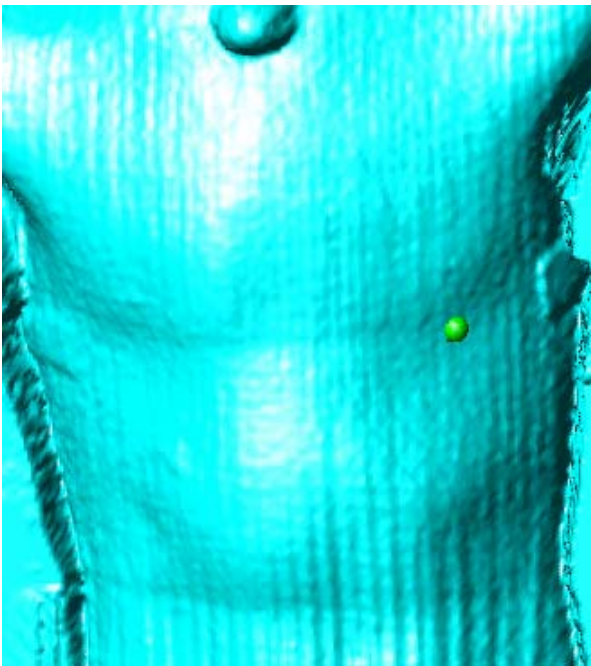


圖 4.39 胸部與腹腔交界位置圖

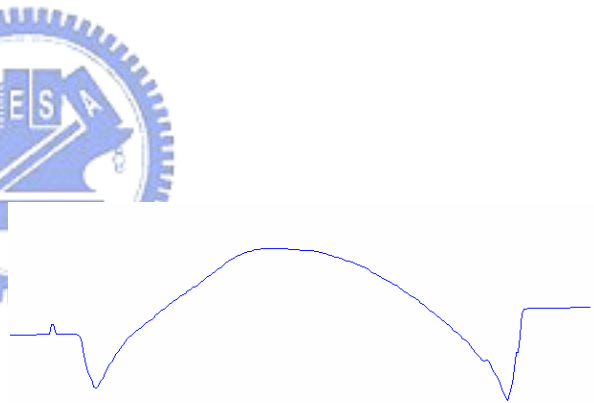


圖 4.40

手術後胸部與腹腔交界高度剖面圖

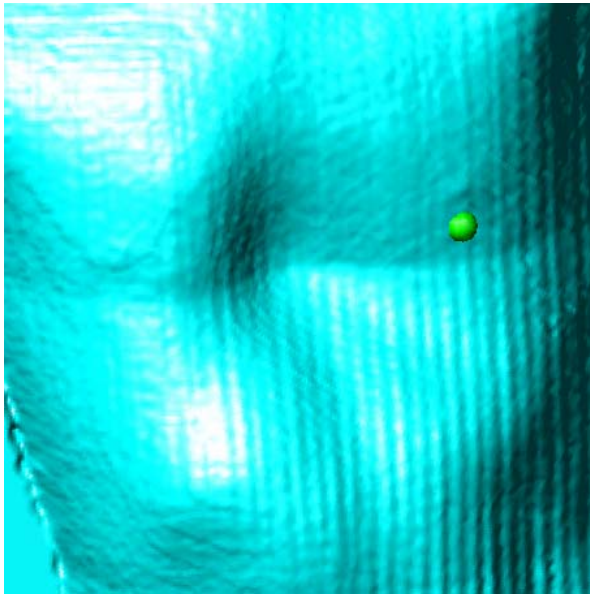


圖 4.41

乳頭與胸部、腹腔交界中點位置圖

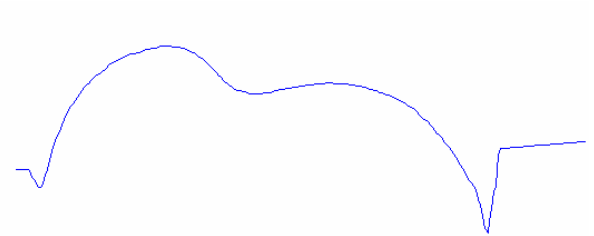


圖 4.42 手術前乳頭與

胸部、腹腔交界中點高度剖面圖

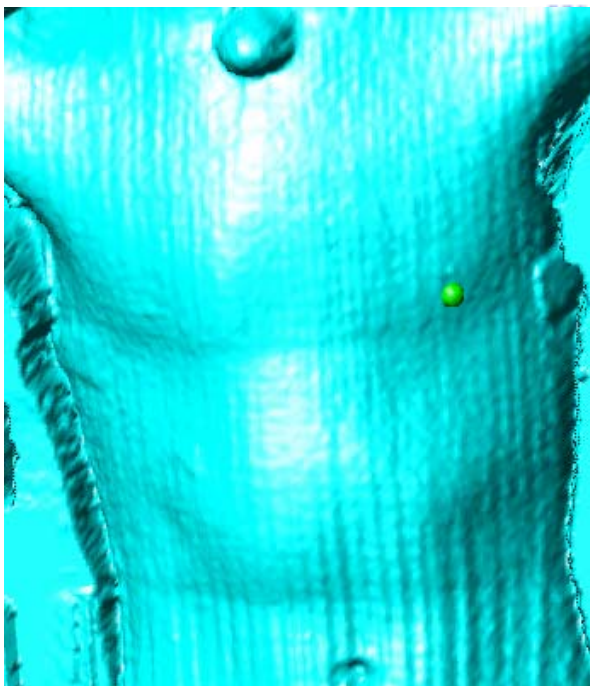


圖 4.43

乳頭與胸部、腹腔交界中點位置圖

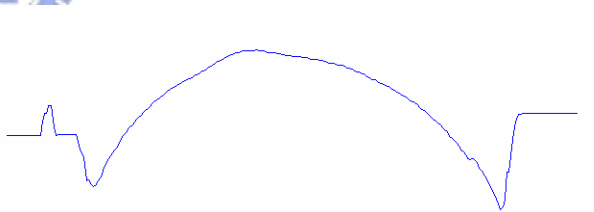


圖 4.44 手術後乳頭與

胸部、腹腔交界中點高度剖面圖

4.4 漏斗胸青春期女性病患手術矯正後

1. 圖 4.45 為手術治療後之條紋影像，其中影像中頸部附近的白色物體為定位標記，胸口正中央有明顯之手術傷口，不過傷口痕跡因為沒有劇烈之三維起伏，故不影響三維外形量測。此女性病患於手術後三天隨即進行胸腔形狀改良之三維外形取像，由於手術傷口依舊劇痛之緣故，此病患並無法處於完全靜止不動的狀態進行取像，加上病患性別為女性，取像過程係由護士小姐進行操作，站立位置有些許偏差，故所得之三維點資料檔有若干的誤差。不過利用女性第二特徵明顯作為對位依據，加上線性混成依舊可得距離實際外形相似之三維外形數據。

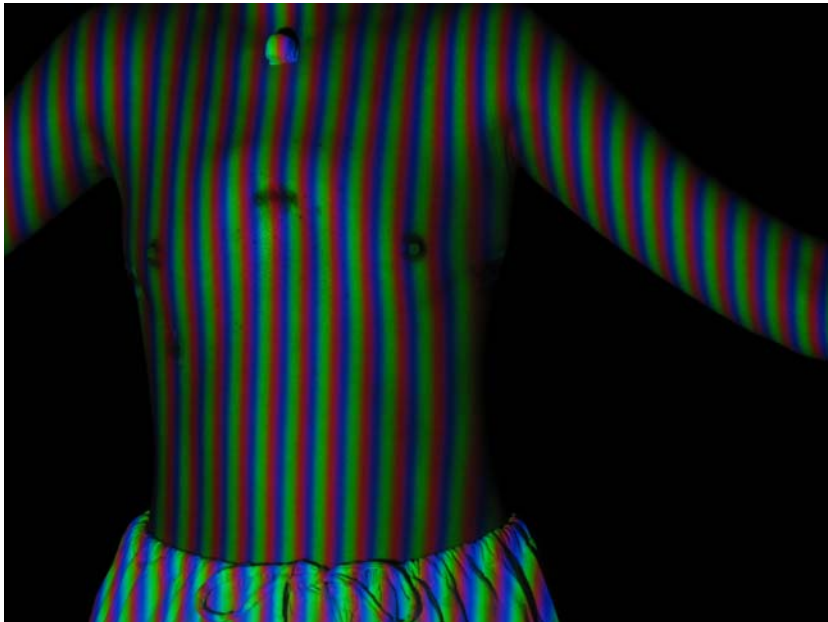


圖 4.45 手術治療後之條紋影像

- a. 由於至於頸部下方用於定位之特徵物尺寸大小之緣故，須調降尋找一個合適拋物面所需要的點資料數量，從預設值 200 輸入改成 100，如上述案例一般，如圖 4.19 所示。
- b. 在左側相機取得之三維點資料上，依序點選特徵點；由於女性病患第二特徵明顯的關係，依序選取左右兩側乳頭作為特徵對位之第二與第三特徵

點。如圖 4.46 所示。而右邊向機取得之三維點資料亦然，重複上述步驟，如圖 4.47 所示。

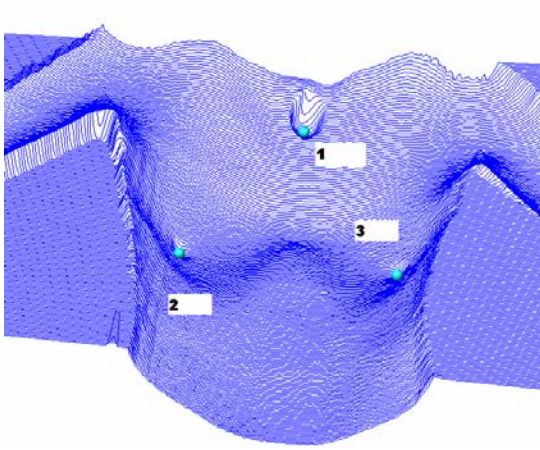


圖 4.46

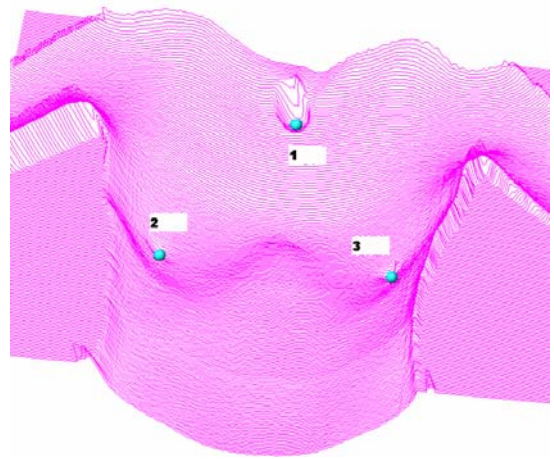


圖 4.47

左側視角依序點選對位依據特徵

右側視角依序點選對位依據特徵

- c. 完成三點對位。明顯於接合處因為取像誤差的關係透過平移與旋轉等運算依舊無法使兩圖案密切接合，如圖 4.48 與圖 4.49 所示。

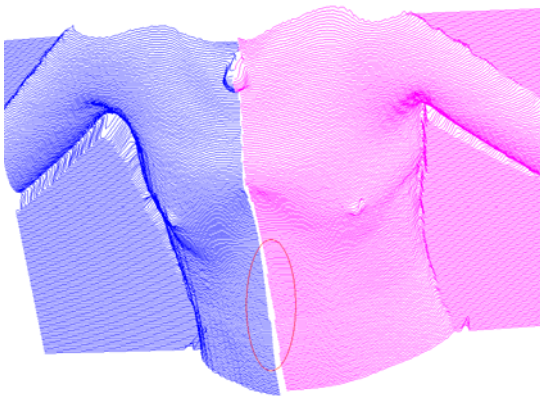


圖 4.48 完成三點對位

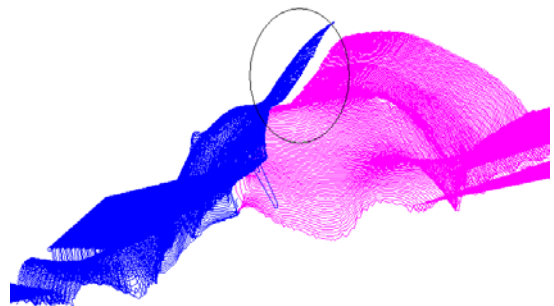


圖 4.49 完成三點對位-下方視角

對於左右相機分別取得之三維點資料進行個別鋪面。左側相機取得之三維點資料胸腔右側仍有些許的模糊，如圖 4.50 與圖 4.51 圈圈處所示一般；右側相機於胸腔左側之

三維點資料仍有些許之不平滑，不過不算太嚴重的破裂，如圖 4.52 與圖 4.53 所示。由於兩邊破裂情形皆算是輕微，為了使中間接合部分能夠平順，沒有突然的突起造成誤導的特徵，使用線性混運算成來進行整合。

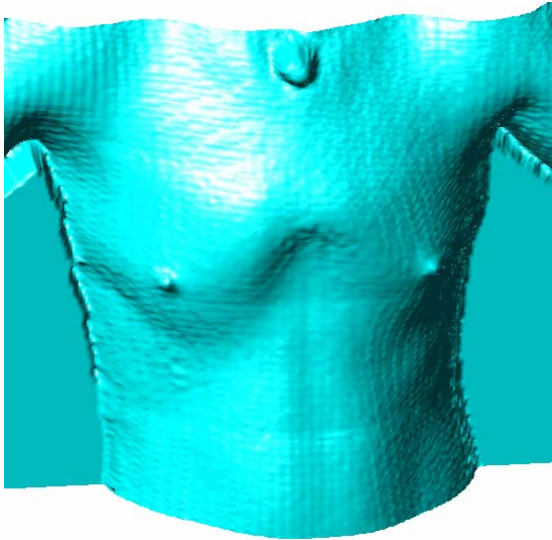


圖 4.50

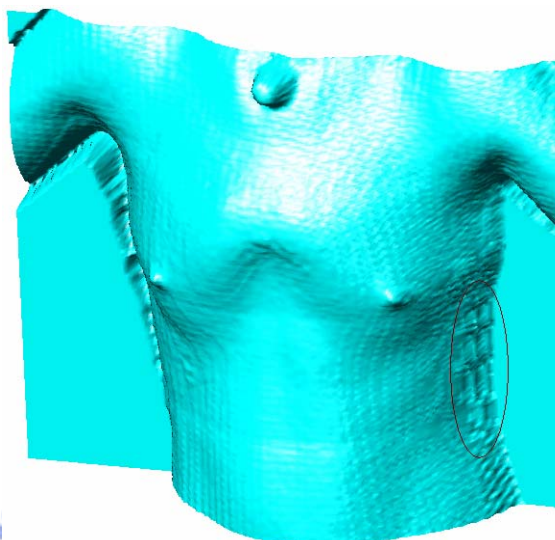


圖 4.51

左側相機取得之三維點資料-左側視角

左側相機取得之三維點資料-右側視角

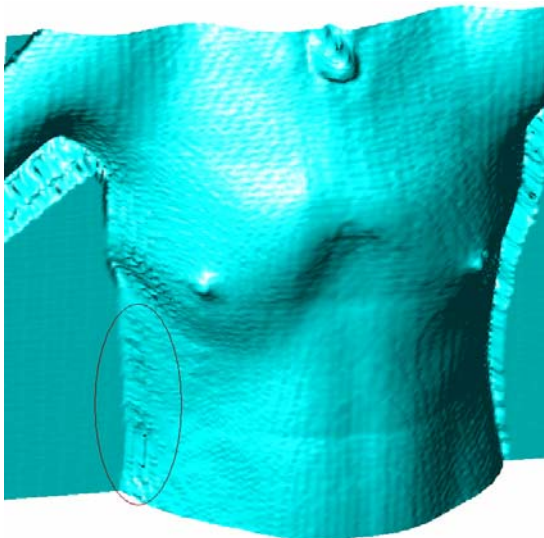


圖 4.52

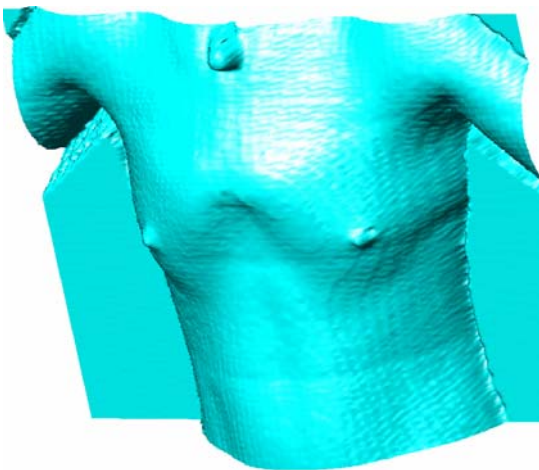


圖 4.53

右側相機取得之三維點資料-左側視角

右側相機取得之三維點資料-右側視角

- d. 更改預設值，如圖 4.54 所示，使用線性混成來處理這組邊緣破裂輕微但中間對位不密合之三維點資料，整合結果如圖 4.55 與圖 4.56 所示，明顯於胸腔左側三維外形沒有出現原先破裂或失真的情形。

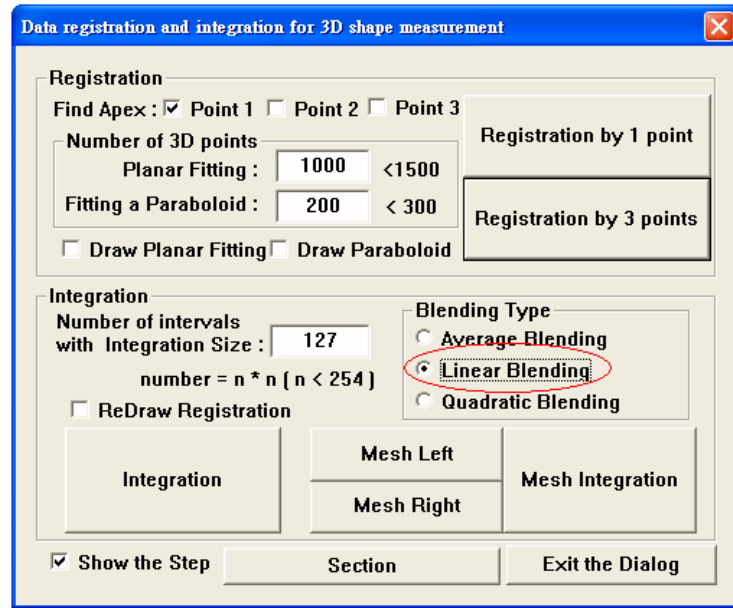


圖 4.54 使用線性混成選項

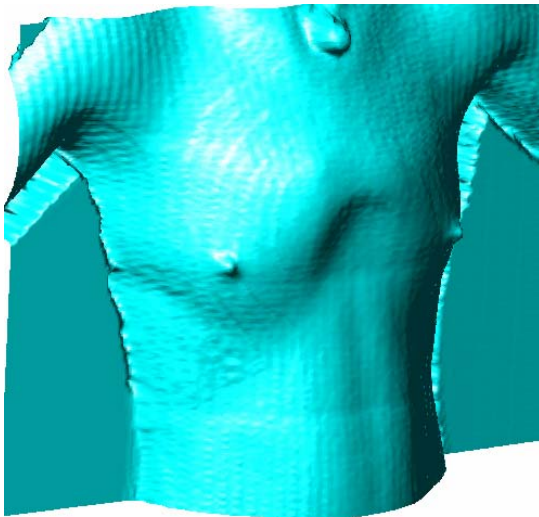


圖 4.55 女性病患

手術後胸腔三維外形展示-左側視角

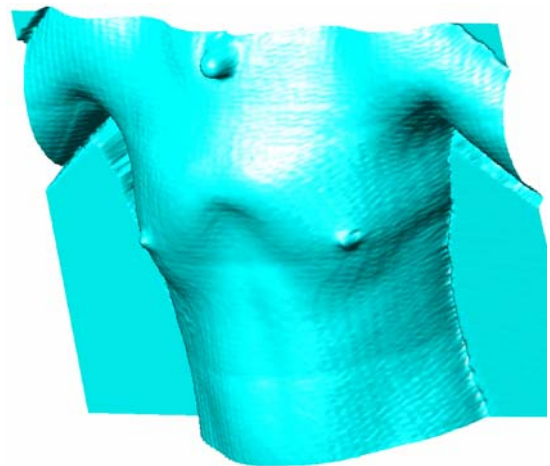


圖 4.56 女性病患

手術後胸腔三維外形展示-右側視角

五、結論與未來展望

5.1 結論

本論文發展出條紋投射法取得之三維點資料整合與對位系統，能以少量的時間與便捷的步驟完成條紋投射法三維點資料之後處理與應用。本論文之研究成果如下：

1. 針對條紋投射法三維點資料之特有特性，發展出與其配合成效相當良好之三點對位運算模式。
2. 將計算特徵方向已求得特徵點之方式，應用於三點對位中之定位點取得運算。
3. 針對條紋投射法三維點資料之特有特性，發展出先將三維點投影至平面再進行還原之混成方式。
4. 將其應用於胸腔三維外形取得之醫療用途上，以不對病人造成任何可能性傷害為前提大，方便與低成本的大量記錄病人之胸腔三維外形；此病歷之建立可明確呈現每一病患之醫療成果與改善幅度。
5. 使用 Visual C++ 撰寫程式介面與友善的對話視窗，並且控制其於 AutoCAD 2002 中展示其三維資料與運算成果。整合三維點資料之對位、三維點資料之整合、左右視角取得之三維點資料分別進行網格化鋪面、整合完畢之三維點資料進行網格化鋪面等功能方便使用此套三維取像系統之使用者操作之。

5.2 未來展望

經過本論文之研究，可以快速、準確的完成三維點資料之整合與對位，對於條紋投射法取得之三維點資料以可以非常有效率的進行後處理的動作，期望本論文研究能夠在逆向工程之三維外形量測與醫學工程之胸腔外形取得方面盡一份心力。然而本研究尚有許多值得發展的方向與需要改進的地方，未來展望如下：

1. 本論文使用 Visual C++ 控制 AutoCAD 2002 之方法，礙於記憶體空間宣告限制問題，宣告指令 ads_point 僅能宣告小於[200][200]之變數，必須將變數寫入硬碟中，再將其讀取出來，於整合計算過程中需要的運算時間仍過於冗長；若能破除記憶體宣告之限制，將其所有三維點資料寫入隨機存取記憶體中，便可大幅改善對位過程所需之等待時間。
2. 可以嘗試多拍攝一張不投射條紋圖樣之影像，並且由此影像擷取物體表面之顏色資訊，於後處理部分重建三維外形之色彩貼圖，使得量測系統能有更廣泛的應用範圍。
3. 目前本論文使用兩台相機進行拍攝，可量測角度大約 180° ，進行整合與對位之角度也大約是 180° ；若於取像技術與對位技術兩方面一同進行突破，則有機會達到 360° 量測之要求。
4. 可針對條紋投射法之特色於校正樣本的部分著手進行改良初步對位之流程。
5. 將取得之三維點資料與剖面圖與實際尺寸之間作一個尺寸比例換算與實際尺寸標記，以便將手術改良成果量化。

參考文獻

- [1] J. Salvi, J. Pages, J. Battle, “Pattern codification strategies in structured light systems,” *Pattern Recognition*, 37(4), pp. 827-849, April 2004.
- [2] Y. S. Chen, B. T. Chen, “Measuring of a three-dimensional surface by use of a spatial distance computation,” *Applied Optics*, 42(11), pp. 1958-1972, 2003.
- [3] P. S. Huang, F. Jin, and F. P. Chiang, “Quantitative evaluation of corrosion by a digital fringe projection technique,” *Optics and Lasers in Engineering*, 31(5), pp. 371-380, 1999.
- [4] C. S. Chua, and R. Jarvis, “Point signatures: a new representation for 3D object recognition,” *International Journal of Computer Vision*, Vol. 25, No. 1, pp. 63-85, 1997.
- [5] R. Benjemaa, and F. Schmitt, “Fast global registration of 3D sampled surfaces using a multi-z-buffer technique,” *International Conference on 3-D Digital Imaging and Modeling*, pp. 113-120, 1997.
- [6] H. Hügli, and C. Schütz, “Geometric matching of 3D Objects: assessing the range of successful initial configurations,” *International Conference on 3-D Digital Imaging and Modeling*, pp. 101-106, 1997.
- [7] C. Schütz, T. Jost, and H. Hügli, “Multi-feature matching algorithm for free-form 3D surface registration,” *International Conference on Pattern Recognition*, Vol. 2, pp. 982-984, 1998.
- [8] K. S. Arun, T. S. Huang, and S. D. Blostein, “Least-squares fitting of two 3-D point sets,” *IEEE Transactions on Pattern Analysis and Machine*

- Intelligence, Vol. 9, No. 5, pp. 587-700, 1987.
- [9] H. T. Yau, C. Y. Chen, and R. G. Wilhelm, "Registration and integration of multiple laser scanned data for reverse engineering of complex 3D models," International Journal of Production Research, Vol. 38, No. 2, pp. 269-285, 2000.
- [10] P. J. Besl, and N. D. McKay, "A method for registration of 3-D shapes," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 2, pp. 239-256, 1992.
- [11] T. Masuda, and N. Yokoya, "A robust method for registration and segmentation of multiple range images," Computer Vision and Image Understanding, Vol. 61, No. 3, pp. 295-307, 1995.
- [12] S. Rusinkiewicz, and M. Levoy, "Efficient variants of the ICP algorithm," International Conference on 3D Digital Imaging and Modeling, 2001.
- [13] L. Zagorchev, and A. Goshtasby, "A mechanism for range image integration without image registration," 3-D Digital Imaging and Modeling (3DIM), pp. 126-133, 2005.
- [14] S. S. Parasnis, "Four point calibration and a comparison of optical modeling and neural networks for robot guidance," Master of Science thesis, University of Cincinnati, 1999.
- [15] P. Hebert, "A self-referenced hand-held range sensor," 3-D Digital Imaging and Modeling (3DIM), pp. 5-12, 2001.
- [16] W. Niem, and J. Wingbermuehle, "Automatic reconstruction of 3D objects using a mobile monoscopic camera," 3-D Digital Imaging and Modeling (3DIM), pp. 173-180, 1997.
- [17] R. Sitnik, M. Kujawiska, and Jerzy Wonicki, "Digital fringe projection system for large-volume 360-deg shape measurement," Optical Engineering,

- Vol. 41, Issue 2, pp.443-449, 2002.
- [18]L. C. Chen, and C. C. Liao, "Calibration of 3D surface profilometry using digital fringe projection," *Measurement Science and Technology*, Vol.16, No.8, pp.1554-1566, 2005.
- [19]陳俊諺, "利用 3D 多重掃描資料建構多面體架構之實體模型," 國立中正大學機械工程研究所碩士論文, 2000.
- [20]黃文昭, "三維外形量測之非線性數位條文誤差校正," 國立台灣大學機械工程學研究所碩士論文, 2006.
- [21]T. Masuda, K. Sakaue, and N. Yokoya, "Registration and integration of multiple range images for 3-D model construction," *International Conference on Pattern Recognition*, Vol.1, No.1, pp.879-885, 1996.
- [22]J. Y. Lai, W. D. Ueng, and C. Y. Yao, "Registration and data merging for multiple sets of scan data," *The International Journal of Advanced Manufacturing Technology*, Vol.15, pp.54-63, 1999.
- [23]M. Soucy, and D. Laurendeau, "A general surface approach to the integration of a set of range views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.17, No.4, pp.344-358, 1995.
- [24]G. Turk, and M. Levoy, "Zippered polygon meshes from range images," *Annual Conference Series on Computer Graphics*, pp.311-318, 1994.
- [25]張北葉, "嬰兒頭型維護—長庚兒童醫院", 國立交通大學機械工程學系醫學工程課堂講義, 2005.
- [26]賴勁堯, "漏斗胸簡介", 長庚醫院兒童外科病例簡介, 2005.
- [27]Jasbir S. Arora, "Introduction to optimum design-Second Edition", Elsevier Academic Press, pp.143-148, 2004.
- [28]Niloy J. Mitra, An Nguyen, "Estimating Surface Normals in Noisy Point Cloud Data", SoCG' 03, June 8-10, San Diego, California, USA, 2003.
- [29]David Eberly, "Least Squares Fitting of Data", Geometric Tools, Inc.

- <http://www.geometrictools.com>, 2001.
- [30] Adam Huang, Ronald M. Summers, and Amy K. Hara, "Surface curvature estimation for automatic colonic polyp detection", Proceedings of SPIE Vol. 5746, Bellingham, WA, pp.393-402, 2005.
- [31] Tatiana Surazhsky, Evgeny Magid, Octavian Soldea, Gershon Elber, and Ehud Rivlin, "A Comparison of Gaussian and Mean Curvatures Estimation Methods on Triangular Meshes", Center for Graphics and Geometric Computing, Technion, Israel Institute of Technology, Haifa 32000, Israel, 2003.
- [32] Craig M. Shakarji, "Least-Squares Fitting Algorithms of the NIST Algorithm Testing System", Journal of Research of the National Institute of Standards and Technology, Volume 103, Number 6, November - December 1998.
- [33] D. L. Page, Y. Sun, A. F. Koschan, J. Paik, and M. A. Abidi, "Normal Vector Voting: Crease Detection and Curvature Estimation on Large, Noisy Meshes", Graphical Models, Volume 64, pp.199-229, 2002.
- [34] Kunwoo Lee, "Principles of CAD/CAM/CAE Systems," Addison-Wesley, pp. 54-65, 1999.
- [35] Ibrahim Zeid, "CAD/CAM Theory and Practice", McGraw-Hill, pp. 481-525, 1991.
- [36] Ibrahim Zeid, "Mastering CAD/CAM", McGraw-Hill, pp. 461-500, 2005.
- [37] 徐偉盛, "三維外形量測之N步彩色相位移法," 國立台灣大學機械工程學研究所碩士論文, 2006.
- [38] Seth D. Force, "ACS Surgery: Principles and Practice CH 6 CHEST WALL PROCEDURES", WedMD, Inc, 2005.

附錄 A 程式 Color Fringe Projection System 使用步驟

取相技術使用『彩色數位條紋投射十五步相位移法』，由於彩色數位條紋投射法一次可投射 RGB（紅、綠、藍）三種顏色之條紋光柵，故只需擷取五張圖像即可滿足十五步相位移法之處理與運算所需。

執行使用程式 Color Fringe Projection System 開始進行三維取相。圖 A.1 與圖 A.2 分別為左邊視角相機與右邊視角相機對於漏斗胸兒童病患胸腔取相之照片。

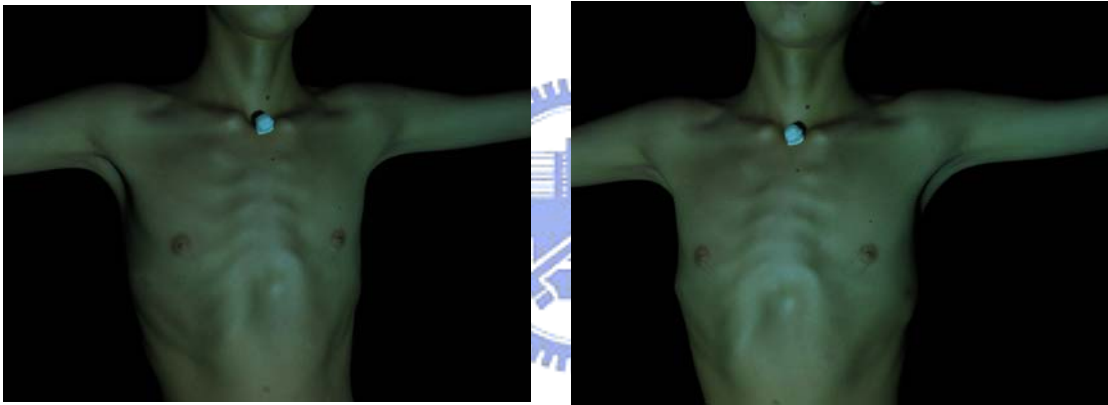


圖 A.1 漏斗胸兒童病患（左）

圖 A.2 漏斗胸兒童病患（右）

使用軟體 Color Fringe Projection System 步驟為：

1. 點選按鈕『Capture Model』，並選擇條紋影像檔儲存資料夾位置。如圖 A.3 所示。
2. 按『確定』開始取相。取相時間約 15 秒，其間投射五組不同相位移角度之彩色條紋光柵，並分別於左、右兩側進行左、右側相機取相。
3. 取相完畢。左、右側視角相機分別取得五張彩色數位條紋投射之不同相位移角度照片。圖 A.4～圖 A.8 為左側相機所取得之五張圖像檔；圖 A.9～圖 A.13 為右側相機所取得之五張圖像檔。

4. 『Data Set』先選至『R』的位置，點選按鈕『Load Image』，載入右邊相機取得的五張照片，如圖 A.14 所示。
5. 點選按鈕『Select Region』，選擇要需要轉換相位值計算三維外形點資料之資料範圍，如圖 A.15 所示。
6. 點選按鈕『Auto Execut』，自動完成下列步驟，並將右側視角相機經過運算取得之三維點資料儲存於 Test0.tst 資料檔中。完成後畫面如圖 A.16 所示。
7. 更動『Data Set』選項，選至『L』的位置，重複步驟 4~6。將左側相機視角取得之三維點資料儲存於 Test1.tst 資料檔中。
8. 完成使用軟體 Color Fringe Projection System 取得漏斗胸兒童病患胸腔外形之三維點資料步驟。

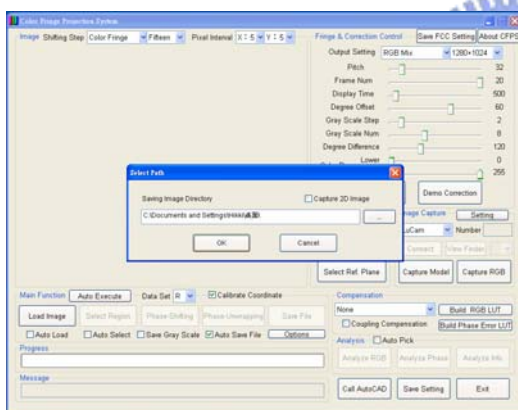


圖 A.3 擇條紋影像檔儲存資料夾位置



圖 A.4 左邊相機所取得之圖檔（一）

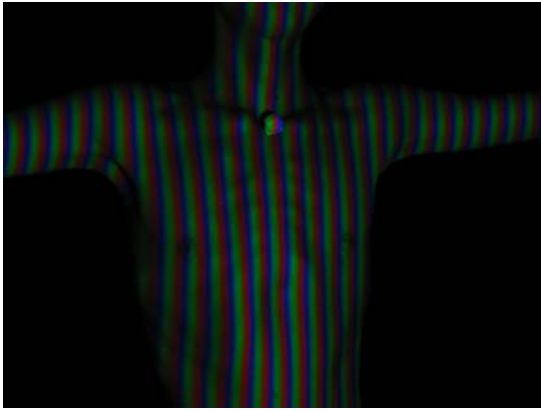


圖 A.5 左邊相機所取得之圖檔 (二)

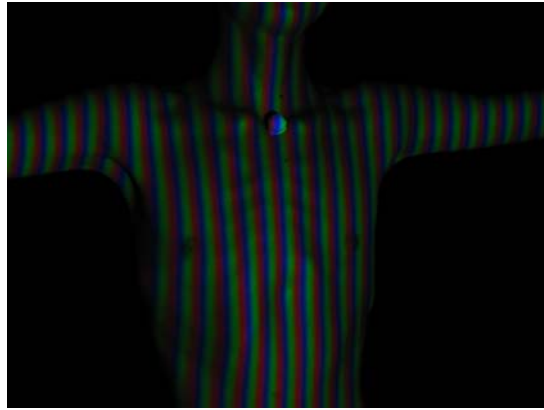


圖 A.6 左邊相機所取得之圖檔 (三)

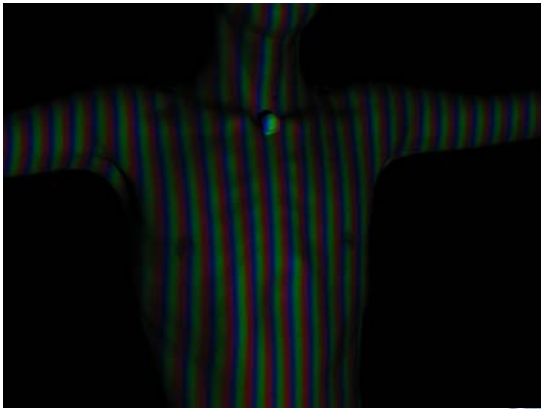


圖 A.7 左邊相機所取得之圖檔 (四)



圖 A.8 左邊相機所取得之圖檔 (五)



圖 A.9 右邊相機所取得之圖檔 (一)

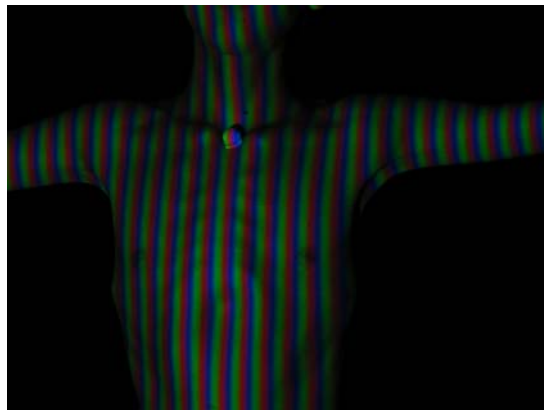


圖 A.10 右邊相機所取得之圖檔 (二)

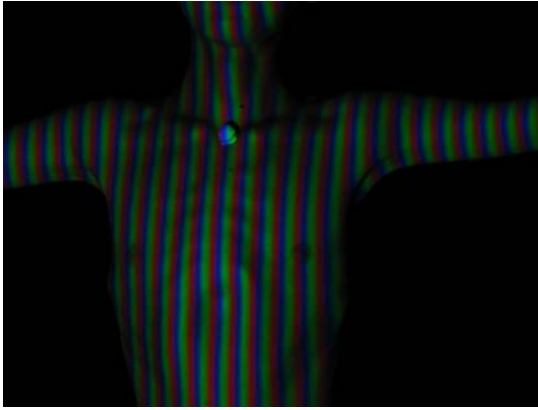


圖 A.11 右邊相機所取得之圖檔 (三)

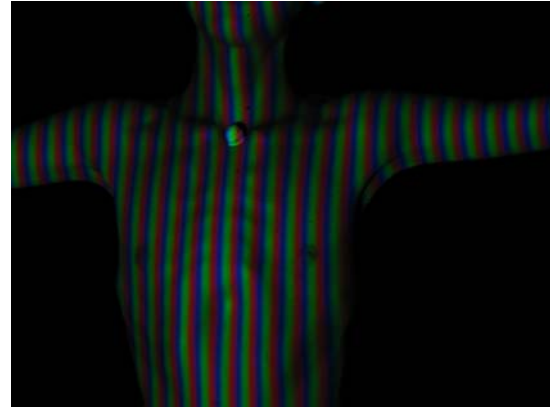


圖 A.12 右邊相機所取得之圖檔 (四)

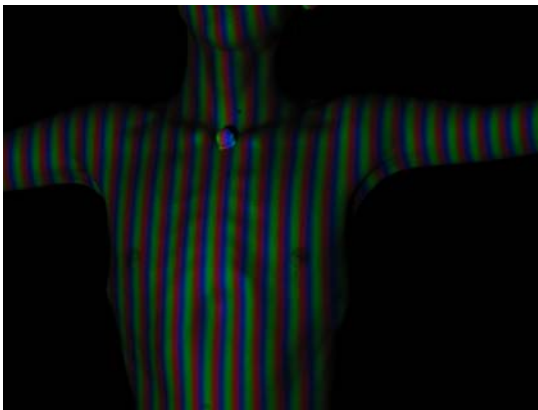


圖 A.13 右邊相機所取得之圖檔 (五)

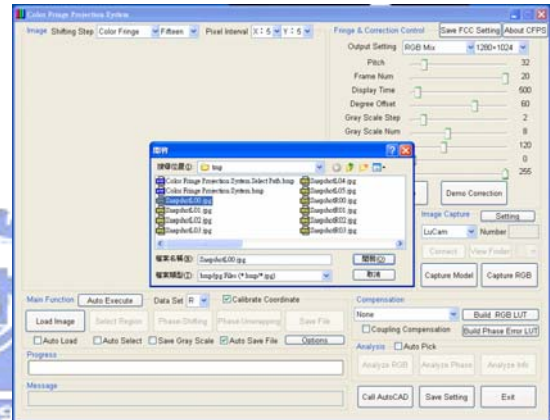


圖 A.14 Load Image

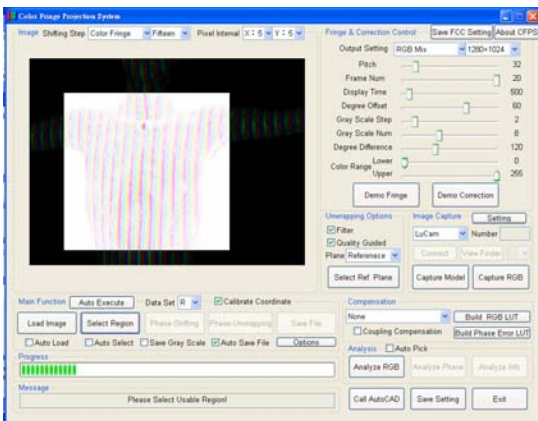


圖 A.15 Select Region

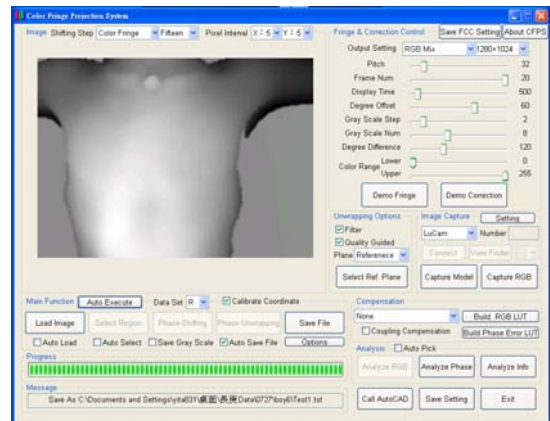


圖 A.16 完成畫面

附錄 B 程式 Data registration and integration for 3D shape measurement system

主程式碼

B.1 程式安裝說明

B.1.1 基本需求

必須具有下列軟體方能進程式安裝：

1. 程式編譯軟體：Microsoft Visual C++ 6.0。
2. 繪圖軟體：AutoCAD 2002。
3. 程式編譯軟體控制繪圖軟體之工具：ObjectARX 2002。
4. 三維資料點處理軟體：本論文研究成果開發之介面，名為 Data registration and integration for 3D shape measurement system (RI)。

B.1.2 ObjectARX 2002 安裝

於安裝程式編譯軟體 Microsoft Visual C++ 6.0 與繪圖軟體 AutoCAD 2002 安裝完成後，開始進行 ObjectARX 2002 之安裝程序。

1. 解壓縮 ObjectARX2002.zip 至 C:\ObjectARX2002。
2. 至 C:\ObjectARX2002\utils\ObjARXWiz 資料夾中，將執行檔 wizards.exe 雙擊滑鼠左鍵開啟解壓縮視窗，更改解壓縮路徑至 C:\ARX2002 後點選『Unzip』進行解壓縮，解壓縮完畢後，點選『Close』用以關閉該解壓縮視窗。
3. 至 C:\ARX2002 資料夾中，於執行檔 WizardSetup.exe 雙擊滑鼠右鍵

後，點選『 Install 』執行安裝，安裝完畢後點選『 確定 』結束安裝視窗。

B.1.3 環境設定

首先設定程式安裝路徑，使用者可根據需求自行更改：

AutoCAD 2002：C:\Program Files\AutoCAD 2002。

ObjectARX 2002：C:\ObjectARX 2002。

依照上述的安裝路徑進行環境之設定，若使用者更改安裝路徑，在下述的步驟中必須將所提到的路徑改更至對應的路徑，以免執行時出現路徑尋找錯誤之問題。

1. 設定 AutoCAD 2002 之工作目錄至 D:\a2002\。於 AutoCAD 2002 之啟動捷徑上方單擊滑鼠右鍵，點選『 內容 』，於『 捷徑 』『 開始位置 』中，更改工作目錄為 D:\a2002\，如圖 B.1 所示。



圖 B.1 設定工作目錄至 D:\a2002\

2. 將 RI.zip 解壓縮至 D:\ARX2002\RI 。
3. 將 NumericalRecipies2(C++).zip 解壓縮至 C:\NumericalRecipies2(C++) 。
4. 開啟 Microsoft Visual C++ 6.0 ，至『 Tools 』 『 Customize 』 『 Add-ins and Macro Files 』 視窗中，選取『 ObjectARX 2000/2000i/2002 Add-in 』，如圖 B.2 所示。

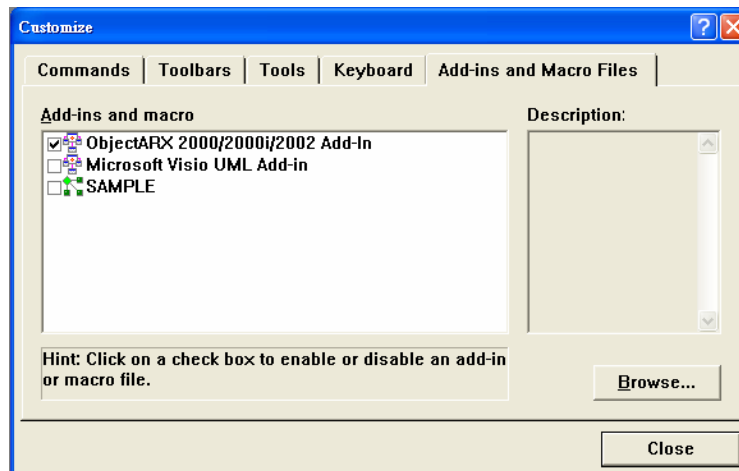


圖 B.2 選取『 ObjectARX 2000/2000i/2002 Add-in 』

5. 至『 Tools 』 『 Options 』 『 Directories 』中，於『 Show directories for: 』 選項欄內選取『 Include files 』；並且加入路徑 C:\OBJECTARX2002\INC 與 C:\NUMERICALRECIPIES2(C++)\CPP\OTHER ，如圖 B.3 所示。加入完成後按『 OK 』 鍵結束該視窗路徑設定。

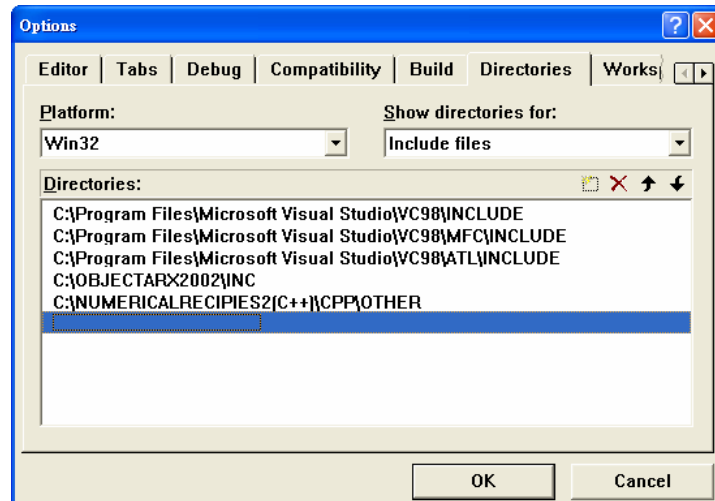


圖 B.3 選取『 Include files 』

- 至『 Tools 』『 Options 』『 Directories 』中，於『 Show directories for: 』選項欄內選取『 Library files 』；並且加入路徑 C:\OBJECTARX2002\LIB，如圖 B.4 所示。加入完成後按『 OK 』鍵結束該視窗路徑設定。

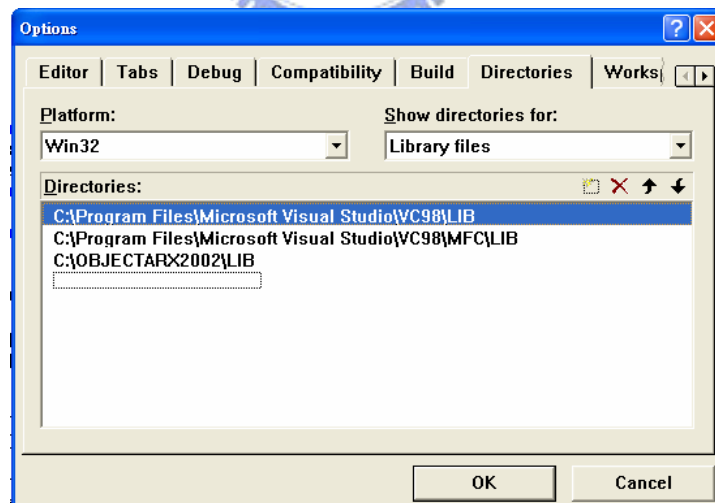


圖 B.4 選取『 Library files 』

- 至『 File 』『 Open Workspace 』中開啟 D:\ARX2002\RI\ri.dsw。
- 至『 Project 』『 settings 』『 Link 』中，在『 Output file name: 』

欄中設定 ObjectARX 2002 控制檔輸出路徑於
D:/a2002/yitari.arx ，如圖 B.5 所示。

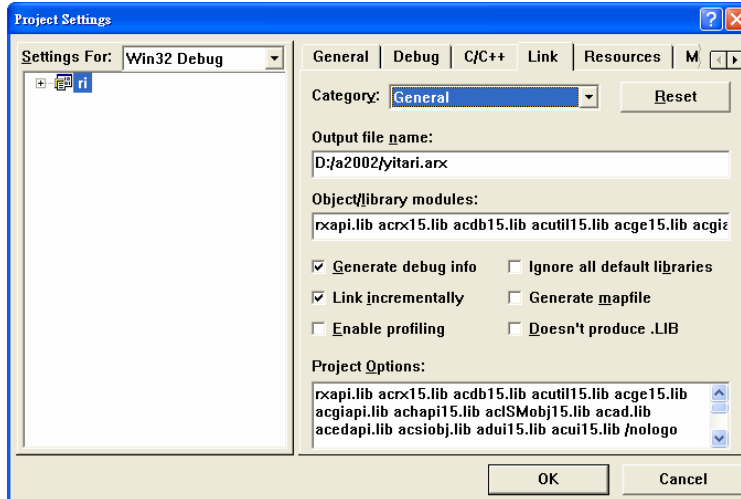


圖 B.5 設定 ObjectARX 2002 控制檔輸出路徑

9. 至『 Build 』中點選『 Rebuild All 』建立『 .arx 』檔案。若建立成功則會顯示訊息『 yitari.arx -- 0 error(s), 18 warning(s) 』表示成功。
10. 可直接複製檔案『 yitari.arx 』至 AutoCAD2002 工作目錄 D:\a2002\ 中，用來替代以上開啟 Microsoft Visual C++ 6.0 重新編譯程式碼取得『 yitari.arx 』之步驟。

B.2 程式碼檔案介紹

B.2.1 核心程式碼 (位於 D:\ARX2002\RI\)

檔名	說明
ri.rc	資源檔。
resource.h	資源檔。
ri.cpp	主程式基本設定檔。
dialog.cpp	主程式介面控制檔。
dialog.h	主程式介面控制檔。
riCommands.cpp	AutoCAD 指令控制檔
ObjectARX.prj	ObjectARX 控制檔
NRmatLib.CPP	矩陣運算處理檔
NRmatLib.h	矩陣運算處理檔

B. 2.2 系統設定檔 (位於 D:\a2002\)



檔名	說明
Test0.tst	左側相機取得之三維外形點資料。
Test1.tst	右側相機取得之三維外形點資料。
tmp.tst	完成對位之右側相機取得之三維外形點資料。
int_L.tst	完成重新佈點之左側相機取得之三維外形點資料。
int_R.tst	完成重新佈點之右側相機取得之三維外形點資料。
final.tst	完成重新佈點與整合之三維外形點資料。

B. 2.3 外部引用程式碼 (位於 D:\ARX2002\RI\與

C:\NumericalRecipies2(C++)\cpp\other\)

檔名	說明
gaussj.cpp	高斯運算處理檔
jacobi.cpp	特徵值運算處理檔
nr.h	NumericalRecipies2(C++)處理檔

B.3 系統主程式介面與程式碼說明

B.3.1 主程式介面

主程式介面如圖 B.6 所示，主要可區分為對位區(Registration)與整合區(Integration)兩部分。各區域中的各項功能於下文介紹說明之。

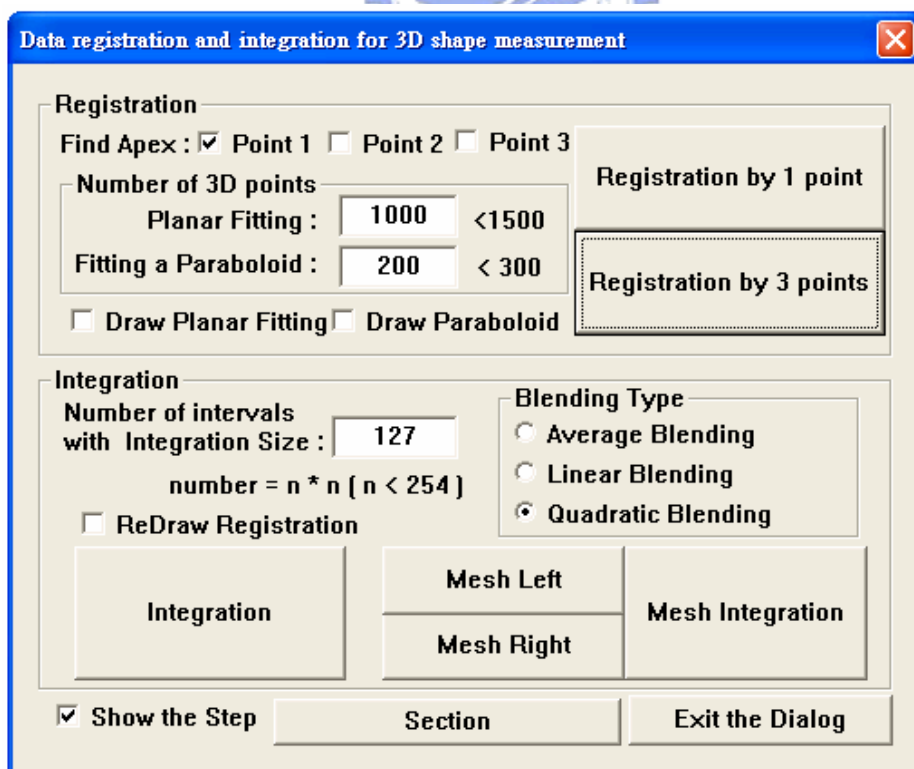


圖 B.6 主程式介面

B.3.2 對位區(Registration)介面說明

對位區的介面圖如圖 B.7 所示。

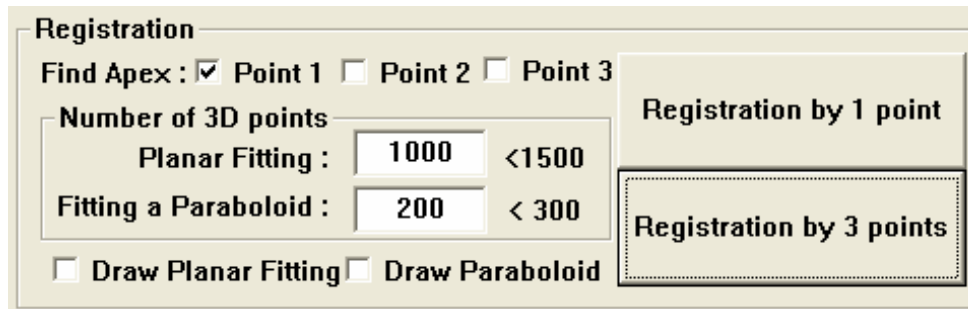


圖 B.7 對位區介面圖

介面名稱	Point 1	參數名稱	m_bFAPoint1
說明	特徵對位所需之特徵對位點一是否需要進行尋找特徵頂點之運算。 選項為是（打溝選取）或否（不打溝選取）。 初始預設值為 TRUE（是）。		
介面名稱	Point 2	參數名稱	m_bFAPoint2
說明	特徵對位所需之特徵對位點二是否需要進行尋找特徵頂點之運算。 選項為是（打溝選取）或否（不打溝選取）。 初始預設值為 FALSE（否）。		
介面名稱	Point 3	參數名稱	m_bFAPoint3
說明	特徵對位所需之特徵對位點三是否需要進行尋找特徵頂點之運算。 選項為是（打溝選取）或否（不打溝選取）。		

	初始預設值為 FALSE (否)。		
介面名稱	Planar Fitting	參數名稱	m_nnei
說明	<p>尋找一個最適合的平面所需要之三維資料點。</p> <p>三維資料點數量必須大於三個，小於一千五百個。</p> <p>初始預設值為 1000。</p>		
介面名稱	Fitting a Paraboloid	參數名稱	m_npara
說明	<p>尋找一個最適合的拋物面所需要之三維資料點。</p> <p>三維資料點數量必須大於六個，小於三百個。</p> <p>初始預設值為 200。</p>		
介面名稱	Draw Planar Fitting	參數名稱	m_bPlanar
說明	<p>是否要將尋找一個最適合的平面之過程繪出並且於過程中要求確認繼續流程。</p> <p>選項為是 (打溝選取) 或否 (不打溝選取)。</p> <p>初始預設值為 FALSE (否)。</p>		
介面名稱	Draw Paraboloid	參數名稱	m_bPara
說明	<p>是否要將尋找一個最適合的拋物面之過程繪出並且於過程中要求確認繼續流程。</p> <p>選項為是 (打溝選取) 或否 (不打溝選取)。</p> <p>初始預設值為 FALSE (否)。</p>		
介面名稱	Registration by 1 point	參數名稱	OnRegisl
說明	特徵對位使用選取單一特徵點運作模組。		
介面名稱	Registration by 3 points	參數名稱	OnOK
說明	特徵對位使用選取三個特徵點運作模組。		

B. 3.3 整合區(Integration)介面說明

整合區的介面圖如圖 B. 8 所示。

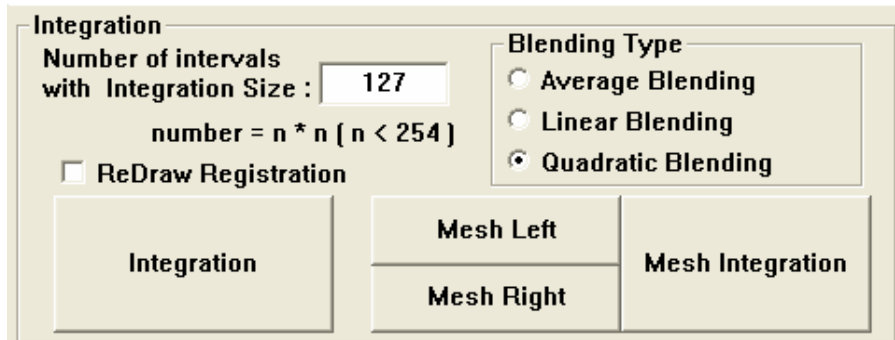


圖 B. 8 整合區介面圖

介面名稱	Number of intervals with Integration Size	參數名稱	m_nint
說明	<p>為了三維點資料整合進行之重新佈點步驟所設定的佈點區間數量大小。</p> <p>重新佈點區間數量必須大於十，小於等於兩百五十四。</p> <p>初始預設值為 127。</p>		
介面名稱	ReDraw Registration	參數名稱	m_bReDraw
說明	<p>是否要重新繪製完成三維點資料對位後之資料檔。</p> <p>選項為是（打溝選取）或否（不打溝選取）。</p> <p>初始預設值為 FALSE（否）。</p>		
介面名稱	Blending Type	參數名稱	m_iBlending
說明	<p>進行三維點資料整合運算之點資料混成形式，分為 0，1，2 三</p>		

	種。 0：Average Blending（平均混成） 1：Linear Blending（線性混成） 2：Quadratic Blending（二次混成） 選項為 0，1，2 三種選擇一種運算形式進行混成。 初始預設值為 2。		
介面名稱	Integration	參數名稱	OnIntegration
說明	進行三維點資料整合運算。		
介面名稱	Mesh Left	參數名稱	OnButton4
說明	進行左邊視角三維點資料網格化鋪面運算。		
介面名稱	Mesh Right	參數名稱	OnButton
說明	進行右邊視角三維點資料網格化鋪面運算。		
介面名稱	Mesh Integration	參數名稱	OnButton2
說明	進行整合後三維點資料網格化鋪面運算。		

B. 3.4 其他功能說明

其他功能的介面圖如圖 B. 9 所示。



圖 B. 9 其他功能介面圖

介面名稱	Show the Step	參數名稱	m_bStep
說明	是否要將所有運算流程繪出並且於過程中要求確認繼續流程。 選項為是（打溝選取）或否（不打溝選取）。		

	初始預設值為 TRUE (是)。		
介面名稱	Section	參數名稱	OnButton6
說明	擷取剖面圖。		
介面名稱	Exit the Dialog	參數名稱	OnCancel
說明	關閉對話視窗。		

B.4 系統函式說明

B.4.1 主程式附屬函式說明 (位於 D:\ARX2002\RI\dialog.cpp)

函式名稱	NRdmatprinc(char *str, Mat_DP &m, int nr, int nc)
說明	於 AutoCAD 訊息欄中列印出矩陣。
函式名稱	parafunc(Mat_DP &p, double x, double y)
說明	找出拋物面方程式之 Z 值。
函式名稱	double sqr(double x)
說明	算平方值。
函式名稱	double sqr2(double x)
說明	算四次方值。
函式名稱	paraboloid_fit(ads_point *inp, int np, double *neidist, Mat_DP &B, ads_point apex)
說明	找出合適之拋物面。
函式名稱	planar_fit(ads_point *inp, double *dist, int np, ads_point un, bool m_bPlanar)

說明	找出合適之平面。
函式名稱	translation_to_z(ads_point *inp, ads_point un, int np, Mat_DP &pf2)
說明	將特徵方向轉至 Z 方向。
函式名稱	draw_fn(char *fn)
說明	繪出檔案三維點資料。
函式名稱	find_neighbor(char *fn, int np, ads_point gp, ads_point *outp, double *outd)
說明	找出鄰近的點。
函式名稱	get_point(ads_point gp)
說明	擷取選取點資料。
函式名稱	find_parapoint(Mat_DP &tm, int np, ads_point *inp, ads_point *outp)
說明	求得尋找合適拋物面所需之點。
函式名稱	draw_paraboloid(Mat_DP &X, int np, double *dist)
說明	繪出拋物面。
函式名稱	trans_apex(Mat_DP &tm, ads_point c, double *neidist, double *cr)
說明	將特徵頂點做座標轉換。
函式名稱	find_apex(char *fn, int nnei, int npara, ads_point gpoint1, ads_point center, double *cr, bool m_bPlanar, bool m_bPara)
說明	尋找特徵頂點。
函式名稱	draw_apex(ads_point c, double cr)
說明	繪出特徵頂點。

函式名稱	corss(ads_point n1, ads_point n2, ads_point n3)
說明	向量外積。
函式名稱	registration(char *fn0, char *fn1, char *fnout, ads_point A1, ads_point B1, ads_point A2, ads_point B2, ads_point A3, ads_point B3)
說明	三點對位。
函式名稱	integration_tmp2(char *fn0, char *fntmp, int nint, ads_point c1, ads_point c2)
說明	重新佈點。
函式名稱	integration(char *fn0, char *fnout, int nint, int blending, bool m_bStep)
說明	混成計算。
函式名稱	draw_int()
說明	繪出重新佈點點資料
函式名稱	draw_inp_tmp(char *fntmp)
說明	繪出對位結果。
函式名稱	Regis1(char *fn0, char *fn1, char *fnout, ads_point A1, ads_point B1)
說明	單點對位。

B. 4.2 數值方法函式說明 (位於 D:\ARX2002\RI\ gaussj. cpp
與 D:\ARX2002\RI\ jacobi. cpp)

函式名稱	NR::gaussj(Mat_IO_DP &a, Mat_IO_DP &b)
說明	計算反矩陣與解線性方程式。

函式名稱	NR::jacobi(Mat_IO_DP &a, Vec_O_DP &d, Mat_O_DP &v, int &nrot)
說明	計算矩陣之特徵值與其對應之特徵向量。

B. 4. 3 矩陣運算函式說明 (位於 D:\ARX2002\RI\ NRmatLib.H 與 D:\ARX2002\RI\ NRmatLib.cpp)

函式名稱	NRdmatmul(Mat_DP &c, Mat_DP &a, Mat_DP &b, int m, int n, int p)
說明	矩陣相乘。
函式名稱	NRdmatunit(Mat_DP &m, int n)
說明	製造單位矩陣。
函式名稱	NRdmatinit(Mat_DP &m, double v, int nr, int nc)
說明	製造元素全部相同之矩陣。
函式名稱	NRtrans(Mat_DP &t, double *pt)
說明	製造平移矩陣。
函式名稱	NRrot(Mat_DP &t, int axis, double ang)
說明	製造旋轉矩陣。
函式名稱	NRdmatprint(char *str, Mat_DP &m, int nr, int nc)
說明	矩陣列印。

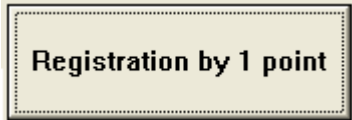
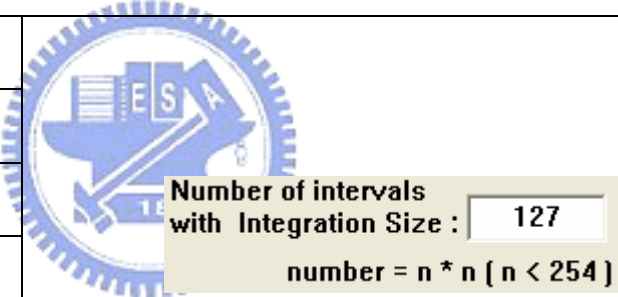
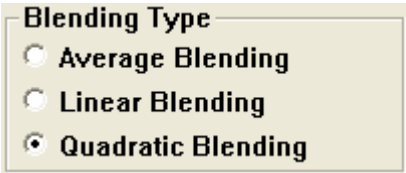
B. 5 程式操作說明

在 AutoCAD 指令列中依序輸入 『 arx 』 『 laod 』 『 yitari 』 『 ri 』 即可呼叫程式介面。

Command: arx [enter]
Enter an option [?/Load/Unload/Commands/Options]: Load [enter]
File name: yitari [enter]
Command: ri

1.	選項位於區域	
	【Registration】	
	操作選項	
	【Find Apex】	
<p>設定是否要進行特徵頂點運算，分別於【Point 1】、【Point 2】、【Point 3】打勾處打溝與否用以表示該特徵點是否需要進行特徵頂點運算；若使用單點特徵對位，只需要於【Point 1】打溝處做選擇即可。</p>		
2.	選項位於區域	
	【Registration】	
	操作選項	
	【Number of 3D points】	
<p>設定尋找鄰近特徵點之三維點資料數量。分別設定【Planar Fitting】與【Fitting a Paraboloid】兩種整數值；其中【Planar Fitting】尋找鄰近之三維資料點數量必須大於三個，小於一千五百個；而【Fitting a Paraboloid】尋找鄰近之三維資料點數量必須大於六個，小於三百個。</p> <p>【Planar Fitting】尋找鄰近之三維點資料數量多寡決定於特徵之大小與特徵起伏之程度，若特徵越大或者起伏程度越高，則需要較多的鄰近點確保求得之特徵方向正確。【Fitting a Paraboloid】尋找鄰近之三維點資</p>		

	料數量多寡決定於特徵之大小，以涵蓋特徵但不超出特徵為設定原則。	
3.	選項位於區域	<input type="checkbox"/> Draw Planar Fitting
	【Registration】	
	操作選項	
	【Draw Planar Fitting】	
勾選【Draw Planar Fitting】是否要將尋找一個最適合的平面之過程繪出並且於過程中要求確認繼續流程。		
4.	選項位於區域	<input type="checkbox"/> Draw Paraboloid
	【Registration】	
	操作選項	
	【Draw Paraboloid】	
勾選【Draw Paraboloid】是否要將尋找一個最適合的拋物面之過程繪出並且於過程中要求確認繼續流程。		
5.	選項位於區域	<input checked="" type="checkbox"/> Show the Step
	【others】	
	操作選項	
	【Show the Step】	
勾選【Show the Step】否要將所有運算流程繪出並且於過程中要求確認繼續流程。		
6.	選項位於區域	Registration by 3 points
	【Registration】	
	操作選項	
	【Registration by 3 points】	

	<p>點選【Registration by 3 points】特徵對位時使用選取三個特徵點運作模組進行對位計算，【Registration by 3 points】適合用於兩組三維外形點資料完成對位需要有旋轉步驟方能對位完成之圖檔。</p>	
7.	選項位於區域	
	【Registration】	
	操作選項	
	【Registration by 1 point】	
<p>點選【Registration by 1 point】特徵對位時使用選取單一特徵點運作模組進行對位計算，【Registration by 1 point】適合用於兩組三維外形點資料完成對位需要有旋轉步驟方能對位完成之圖檔。</p>		
8.	選項位於區域	
	【Integration】	
	操作選項	
	【Number of intervals with Integration Size】	
<p>為了三維點資料整合進行之重新佈點步驟所設定的佈點區間數量大小。 【Number of intervals with Integration Size】重新佈點區間數量必須大於十，小於等於兩百五十四。</p>		
9.	選項位於區域	
	【Integration】	
	操作選項	
	【Blending Type】	
<p>進行三維點資料整合運算之點資料混成形式，分為【Average Blending】，</p>		

		<p>【Linear Blending】、【Quadratic Blending】三種。使用越高次的混成公式，左側之形狀與左側相機取得之三維點資料之相關度就越高，右側亦然，依照圖檔誤差形式分別從【Average Blending】(平均混成)、【Linear Blending】(線性混成)與【Quadratic Blending】(二次混成)三種混成公式則一使用。</p>
10.	選項位於區域	
	【Integration】	
	操作選項	
	【ReDraw Registration】	
	<p>勾選【ReDraw Registration】是否要重新繪製完成三維點資料對位後之資料檔。</p>	
11.	選項位於區域	
	【Integration】	
	操作選項	
	【Integration】	
	<p>點選【Integration】進行三維點資料整合運算。點選後需先選擇重新佈點範圍，務必先點選左側之矩型直角，在點選右側之矩型直角，以免混成計算時造成左右顛倒的情況。</p>	
12.	選項位於區域	
	【Integration】	
	操作選項	
	【Mesh Left】	
	<p>點選【Mesh Left】進行左邊視角三維點資料網格化鋪面運算。</p>	
13.	選項位於區域	

	【Integration】	
	操作選項	
	【Mesh Right】	
	點選【Mesh Right】進行右邊視角三維點資料網格化鋪面運算。	
14.	選項位於區域	
	【Integration】	
	操作選項	
	【Mesh Integration】	
	點選【Mesh Integration】進行整合後三維點資料網格化鋪面運算。	
15.	選項位於區域	
	【others】	
	操作選項	
	【Section】	
	點選【Section】開始進行剖面擷取運算。執行後必須先給入欲擷取剖面之高度點，取得高度點後立即進行剖面圖之運算與繪製。	
16.	選項位於區域	
	【others】	
	操作選項	
	【Exit the Dialog】	
	點選【Exit the Dialog】關閉對話視窗。	

B.6 列出程式主程式碼

列出程式 Data registration and integration for 3D shape measurement 之主程式碼。由於整體程式碼過於龐大，因此只列出主程式碼表示其結構，副程

式與眾多函式庫不於此列出。

B. 6. 1 三點對位運算主程式碼

```
void Cdialog::OnOK()
{
    // TODO: Add extra validation here
    char fn0[200], fn1[200], fnout[80];
    UpdateData(TRUE);
    int nnei, npara, nint, i;
    double pi, cr1, cr2;
    cr1 = 1.3; cr2 = 1.3;
    ads_point
    gpoin1, center1, gpoin2, center2, tmp, gpoin3, gpoin4, gpoin5, gpoin6,
    center3, center4, center5, center6;
    nnei=m_nnei; npara = m_npara; nint=254; pi=4.0*atan(1.0);
    Cdialog::OnOK();
    //nnei<1500 npara<300
    acedCommand(RTSTR, "osnap", RTSTR, "none", 0);
    acedCommand(RTSTR, "cmdecho", RTSTR, "0", 0);

    // ***** draw left *****
    acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
    acedCommand(RTSTR, "layer", RTSTR, "Make", RTSTR, "left", RTSTR, "color",
    RTSTR, "blue", RTSTR, "", RTSTR, "", 0);
    strcpy(fn0, "d:/a2002/test0. tst"); draw_fn(fn0);
    acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0);
;
    get_point(gpoin1);
    get_point(gpoin3);
    get_point(gpoin5);
    if(m_bStep == TRUE)acedGetPoint(NULL, "\n Get 3 Points!! Get
Point to Continue", tmp);
    acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
    if(m_bFAPoint1 == FALSE) for(i=0;i<3;i++) center1[i] =
gpoin1[i];
    if(m_bFAPoint2 == FALSE) for(i=0;i<3;i++) center3[i] =
gpoin3[i];
    if(m_bFAPoint3 == FALSE) for(i=0;i<3;i++) center5[i] =
gpoin5[i];
    // ***** find apex *****
    if(m_bFAPoint1 == TRUE)
    find_apex(fn0, nnei, npara, gpoin1, center1, &cr1, m_bPlanar, m_bPara)
;
    if(m_bFAPoint2 == TRUE)
    find_apex(fn0, nnei, npara, gpoin3, center3, &cr1, m_bPlanar, m_bPara)
;
    if(m_bFAPoint3 == TRUE)
    find_apex(fn0, nnei, npara, gpoin5, center5, &cr1, m_bPlanar, m_bPara)
;
    // ***** re draw left *****
    if(m_bStep == TRUE)
    {
        draw_apex(center1, cr1);
        draw_apex(center3, cr1);
        draw_apex(center5, cr1);

    }

    acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "left", RTSTR, "",
, 0);
    draw_fn(fn0);

    acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
    acedGetPoint(NULL, "\n Get 3 Points!! Get Point to
Continue", tmp);
}
```

```
    } //end if

    // ***** draw right *****
    acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
    acedCommand(RTSTR, "layer", RTSTR, "Make", RTSTR, "right", RTSTR,
"color", RTSTR, "magenta", RTSTR, "", RTSTR, "", 0);
    strcpy(fn1, "d:/a2002/test1. tst"); draw_fn(fn1);
    acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
    get_point(gpoin2);
    get_point(gpoin4);
    get_point(gpoin6);
    if(m_bStep == TRUE)acedGetPoint(NULL, "\n Get 3 Points!! Get
Point to Continue", tmp);
    acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
    if(m_bFAPoint1 == FALSE) for(i=0;i<3;i++) center2[i] =
gpoin2[i];
    if(m_bFAPoint2 == FALSE) for(i=0;i<3;i++) center4[i] =
gpoin4[i];
    if(m_bFAPoint3 == FALSE) for(i=0;i<3;i++) center6[i] =
gpoin6[i];
    // ***** find apex *****
    if(m_bFAPoint1 == TRUE)
    find_apex(fn1, nnei, npara, gpoin2, center2, &cr2, m_bPlanar, m_bPara)
;
    if(m_bFAPoint2 == TRUE)
    find_apex(fn1, nnei, npara, gpoin4, center4, &cr2, m_bPlanar, m_bPara)
;
    if(m_bFAPoint3 == TRUE)
    find_apex(fn1, nnei, npara, gpoin6, center6, &cr2, m_bPlanar, m_bPara)
;
    // ***** re draw right *****
    if(m_bStep == TRUE)
    {
        draw_apex(center2, cr2);
        draw_apex(center4, cr2);
        draw_apex(center6, cr2);

    }

    acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "right", RTSTR, "
", 0);
    draw_fn(fn1);

    acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
    acedGetPoint(NULL, "\n Get 3 Points!! Get Point to
Continue", tmp);
    } //end if
    acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
    strcpy(fnout, "d:/a2002/tmp. tst");
    registration(fn0, fn1, fnout, center1, center2, center3, center4,
center5, center6);
    if(m_bStep == FALSE) acedCommand(RTSTR, "zoom", RTSTR, "e", 0);
    acedCommand(RTSTR, "regen", 0);
}
}
```

B. 6. 2 單點對位運算主程式碼

```
void Cdialog::OnRegis1()
{
    char fn0[200], fn1[200], fnout[80];
```

```

UpdateData(TRUE);
int nnei, npara, nint, i;
double pi, cr1, cr2;
cr1 = 1.3; cr2 = 1.3;
ads_point gpoint1, center1, gpoint2, center2, tmp;
nnei=m_nnei; npara = m_npara; nint=254;
CDialog::OnOK();
//nnei<1500 npara<300
acedCommand(RTSTR, "osnap", RTSTR, "none", 0);
acedCommand(RTSTR, "cmdecho", RTSTR, "0", 0);

// ***** draw left *****
acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
acedCommand(RTSTR, "layer", RTSTR, "Make", RTSTR, "left", RTSTR, "
color", RTSTR, "blue", RTSTR, "", RTSTR, "", 0);
strcpy(fn0, "d:/a2002/test0. tst"); draw_fn(fn0);
acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
get_point(gpoint1);
if(m_bStep == TRUE)acedGetPoint(NULL, "\n Get 1 Point!! Get
Point to Continue Registration", tmp);
acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
if(m_bFAPoint1 == FALSE) for(i=0; i<3; i++) center1[i] =
gpoint1[i];
// ***** find apex *****
if(m_bFAPoint1 == TRUE)
find_apex(fn0, nnei, npara, gpoint1, center1, &cr1, m_bPlanar, m_bPara)
;
// ***** re draw left *****
if(m_bStep == TRUE)
{
draw_apex(center1, cr1);
}
acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "left", RTSTR, "
, 0);
draw_fn(fn0);

acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
acedGetPoint(NULL, "\n Get 1 Apex Point!! Get Point to
Continue Registration", tmp);
} //end if
// ***** draw right *****
acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
acedCommand(RTSTR, "layer", RTSTR, "Make", RTSTR, "right", RTSTR,
"color", RTSTR, "magenta", RTSTR, "", RTSTR, "", 0);
strcpy(fn1, "d:/a2002/test1. tst"); draw_fn(fn1);
acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
get_point(gpoint2);
if(m_bStep == TRUE)acedGetPoint(NULL, "\n Get 1 Point!! Get
Point to Continue Registration", tmp);
acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
if(m_bFAPoint1 == FALSE) for(i=0; i<3; i++) center2[i] =
gpoint2[i];
// ***** find apex *****
if(m_bFAPoint1 == TRUE)
find_apex(fn1, nnei, npara, gpoint2, center2, &cr2, m_bPlanar, m_bPara)
;
// ***** re draw right *****
if(m_bStep == TRUE)
{
draw_apex(center2, cr2);
}
acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "right", RTSTR, "
", 0);
draw_fn(fn1);

```

```

acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
acedGetPoint(NULL, "\n Get 1 Apex Point!! Get Point to
Continue Registration", tmp);
} //end if
acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
strcpy(fnout, "d:/a2002/tmp. tst");
Regis1(fn0, fn1, fnout, center1, center2);
if(m_bStep == FALSE) acedCommand(RTSTR, "zoom", RTSTR, "e", 0);
acedCommand(RTSTR, "regen", 0);
}

```

B. 6. 3 整合運算主程式碼

```

void Cdialog::OnIntegration()
{
char fn0[200], fnout[200];
int nint, blending;
UpdateData(TRUE);
CDialog::OnOK();
nint=m_nint;
blending=m_iBlending;
strcpy(fn0, "d:/a2002/test0. tst");
strcpy(fnout, "d:/a2002/tmp. tst");
if(m_bReDraw == TRUE) draw_int();
integration(fn0, fnout, nint, blending, m_bStep);
}

```

B. 6. 4 左側視角三維點資料 網格化鋪面主程式碼

```

void Cdialog::OnButton4()
{
char ffin[80];
strcpy(ffin, "d:/a2002/int_L. tst");
FILE *fpin;
int n, m, i, j;
ads_point pt;
CDialog::OnOK();
acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
fpin=fopen(ffin, "r");
fscanf(fpin, "%d %d", &m, &n);
acedCommand(RTSTR, "3dmesh", RTSHORT, m, RTSHORT, n, 0);
for(i=0; i<m; i++)
for(j=0; j<n; j++)
{
fscanf(fpin, "%lf %lf %lf", &pt[0], &pt[1],
&pt[2]);
acedCommand(RT3DPOINT, pt, 0);
}
acedCommand(RTSTR, "zoom", RTSTR, "e", 0);
acedCommand(RTSTR, "regen", 0);
return;
}

```

B. 6. 5 右側視角三維點資料 網格化鋪面主程式碼

```
void Cdialog::OnButton3()
{
    char fnin[80];
    strcpy(fnin, "d:/a2002/int_r.tst");
    FILE *fpin;
    int n, m, i, j;
    ads_point pt;
    CDialog::OnOK();
    acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
    fpin=fopen(fnin, "r");
    fscanf(fpin, "%d %d", &m, &n);
    acedCommand(RTSTR, "3dmesh", RTSHORT, m, RTSHORT, n, 0);
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
        {
            fscanf(fpin, "%lf %lf %lf", &pt[0], &pt[1],
&pt[2]);
            acedCommand(RT3DPOINT, pt, 0);
        }
    acedCommand(RTSTR, "zoom", RTSTR, "e", 0);
    acedCommand(RTSTR, "regen", 0);
    return;
}
```

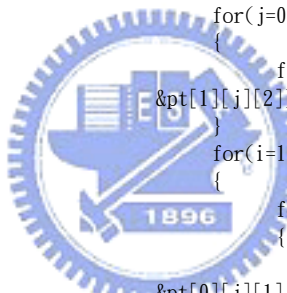
B. 6. 6 完成對位三維點資料 網格化鋪面主程式碼

```
void Cdialog::OnButton2()
{
    char fnin[80];
    strcpy(fnin, "d:/a2002/final.tst");
    CDialog::OnOK();
    FILE *fpin;
    int n, m, i, j;
    ads_point pt;
    CDialog::OnOK();
    acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
    fpin=fopen(fnin, "r");
    fscanf(fpin, "%d %d", &m, &n);
    acedCommand(RTSTR, "3dmesh", RTSHORT, m, RTSHORT, n, 0);
    for(i=0; i<m; i++)
        for(j=0; j<n; j++)
        {
            fscanf(fpin, "%lf %lf %lf", &pt[0], &pt[1],
&pt[2]);
            acedCommand(RT3DPOINT, pt, 0);
        }
    acedCommand(RTSTR, "zoom", RTSTR, "e", 0);
    acedCommand(RTSTR, "regen", 0);
    return;
}
```

B. 6. 7 擷取剖面主程式碼

```
void Cdialog::OnButton6()
{
```

```
// TODO: Add your control notification handler code here
char fnin[80];
strcpy(fnin, "d:/a2002/final.tst");
CDialog::OnOK();
FILE *fpin;
int n, m, i, j;
ads_point pt[2][255], pt1[255], gp, tmp;
double YYY, c, r;
r=5.0;
CDialog::OnOK();
acedCommand(RTSTR, "layer", RTSTR, "Make", RTSTR, "Section0", RTS
TR, "color", RTSTR, "green", RTSTR, "", RTSTR, "", 0);
acedCommand(RTSTR, "osnap", RTSTR, "near", 0);
acedGetPoint(NULL, "\n Get a nearest Point ", gp);
YYY = gp[1];
acedCommand(RTSTR, "sphere", RT3DPOINT, gp, RTREAL, r, 0);
acedGetPoint(NULL, "\n Get Point to Continue!! ", tmp);
acedCommand(RTSTR, "osnap", RTSTR, "none", 0);
//acedGetReal(" Input Y :", &YYY);
//acedCommand(RTSTR, "erase", RTSTR, "all", RTSTR, "", 0);
fpin=fopen(fnin, "r");
fscanf(fpin, "%d %d", &m, &n);
for(j=0; j<n; j++)
{
    pt[0][n][0]=0;
    pt[0][n][1]=0;
    pt[0][n][2]=0;
}
for(j=0; j<n; j++) //read frist line
{
    fscanf(fpin, "%lf %lf %lf", &pt[1][j][0], &pt[1][j][1],
&pt[1][j][2]);
}
for(i=1; i<m; i++)
{
    for(j=0; j<n; j++)
    {
        fscanf(fpin, "%lf %lf %lf", &pt[0][j][0],
&pt[0][j][1], &pt[0][j][2]);
    }
    if(YYY<pt[1][0][1] && YYY>pt[0][0][1]) break;
    if(YYY>pt[1][0][1] && YYY<pt[0][0][1]) break;
    if(YYY==pt[0][0][1]) break;
    for(j=0; j<n; j++)
    {
        pt[1][j][0]=pt[0][j][0];
        pt[1][j][1]=pt[0][j][1];
        pt[1][j][2]=pt[0][j][2];
    }
    if(i== m-1)acutPrintf("Not Found!");
}
c = (YYY-pt[0][0][1])/(pt[1][0][1]-pt[0][0][1]);
acutPrintf("i = %d", i);
acedCommand(RTSTR, "layer", RTSTR, "Make", RTSTR, "Section1", RTS
TR, "color", RTSTR, "red", RTSTR, "", RTSTR, "", 0);
acedCommand(RTSTR, "3Dpoly", 0);
for(j=0; j<n; j++)
{
    acedCommand(RT3DPOINT, pt[0][j], 0);
}
acedCommand(RTSTR, "", 0);
acedCommand(RTSTR, "3Dpoly", 0);
for(j=0; j<n; j++)
{
    acedCommand(RT3DPOINT, pt[1][j], 0);
}
acedCommand(RTSTR, "", 0);
```



```

    acedCommand(RTSTR, "layer", RTSTR, "Make", RTSTR, "Section2", RTS
TR, "color", RTSTR, "blue", RTSTR, "", RTSTR, "", 0);
    acedCommand(RTSTR, "3Dpoly", 0);
    for(j=0; j<n; j++)
    {
        pt1[j][0]=(pt[1][j][0]-pt[0][j][0])*c+pt[0][j][0];
        pt1[j][2]=(pt[1][j][2]-pt[0][j][2])*c+pt[0][j][2];
        pt1[j][1]=YYY;
        acedCommand(RT3DPOINT, pt1[j], 0);
    }
    acedCommand(RTSTR, "", 0);
    acedCommand(RTSTR, "zoom", RTSTR, "e", 0);
    acedCommand(RTSTR, "regen", 0);
    acedCommand(RTSTR, "layer", RTSTR, "set", RTSTR, "0", RTSTR, "", 0)
;
return;
}

```



作者簡歷

姓 名 : 鍾宜達

出生日期 : 中華民國 72 年 8 月 31 日

出生地 : 台北市

學 經 歷 : 民國 78~84 年 台北市立公館國民小學

民國 84~87 年 台北市立民族國民中學

民國 87~90 年 台北市立大同高級中學

民國 90~94 年 國立交通大學機械工程學系

民國 94~96 年 國立交通大學機械工程學系碩士班

通訊地址 : 台北市文山區萬美街二段八十一號

電 話 : 0958-080-052

E m a i l : Yita.me94g@nctu.edu.tw