

國立交通大學

多媒體工程研究所

碩士論文

六子棋自動棋譜產生系統



Automatic Record Generator for Connect Six

研究生：許諾白

指導教授：莊仁輝 教授

中華民國九十七年六月

六子棋自動棋譜產生系統
Automatic Record Generator for Connect Six

研究生：許諾白 Student：Nuo-Pai Hsu

指導教授：莊仁輝 Advisor：Jen-Hui Chuang

國立交通大學
多媒體工程研究所
碩士論文



A Thesis
Submitted to Institute of Multimedia Engineering
College of Computer Science
National Chiao Tung University

in partial Fulfillment of the Requirements

for the Degree of

Master

in

Computer Science

June 2008

Hsinchu, Taiwan, Republic of China

中華民國九十七年六月

六子棋自動棋譜產生系統

學生：許諾白

指導教授：莊仁輝 博士

國立交通大學

多媒體工程研究所

摘要

我們提出了一套自動棋譜產生系統，只要在下棋時利用攝影機拍攝棋盤，便可經由程式自動記錄每一步棋的位置及思考時間等資訊，而且只需要一般的攝影機及筆記型電腦即可執行此自動棋譜產生系統。我們先利用色彩資訊找出影像中的棋子區域，再找出影片中的關鍵影格，以及每一手新增之棋子。而在棋子的定位上我們提出了兩種方法，一種是「相對於棋盤之棋子定位」，另一種是「基於棋子相對位置之定位」。其中前者是利用投影轉換矩陣（Homography）把棋子的影像座標轉換至棋盤的座標；而後者則是先建立棋盤上所有棋子的相對關係，再藉此資訊重建棋盤的座標系，如此可以免除偵測棋盤格線所造成的誤差，為一種創新且更具彈性的作法。不管是哪一種定位方式，都可以正確定位每一手棋子的位置，達到即時記錄棋譜的目的，且本系統並不需要限制輸入的影片一定是俯視拍攝的，也針對在比賽過程中棋盤或攝影機可能被移動，而造成分析結果不正確的問題做了完善的處理，更增加了本系統在使用上的方便性。

Automatic Record Generator for Connect Six


Student : Nuo-Pai Hsu

Advisor : Dr. Jen-Hui Chuang

Institute of Multimedia Engineering

National Chiao Tung University

ABSTRACT



In this thesis, a system which can automatically extracts the game record, including each stone's position, color and ply interval, from video of "Connect Six" or "Go" is developed. Firstly, the system will identify all stone regions of each image frame by the color information so that the key frames and incremental stones of each ply can be detected. After identifying stone positions in the image, two approaches can be adopted to find their locations in the real world: (i) find them with respect to board by homographic transformation (as done in some previous works), and (ii) find them by reconstructing a coordinate system of the board based on relative positions among the stones in the image space but without using the board information (as proposed in this thesis). The proposed approach is not only more flexible, i.e., with less constraints imposed on the game video, but also more robust with respect to non-ideal image quality associated with the board image. Experimental results show that both methods can find stone location correctly and generate game record efficiently. Finally, the proposed system is easy to use since the system input is not

restricted to a top-view video, and the problem of board movement can also be resolved automatically.



致 謝

首先誠摯的感謝指導教授莊仁輝博士，老師的悉心教導帶領我進入電腦視覺的領域，讓我發現此領域的趣味及奧妙，有老師的不斷指引與討論，才能有我今日的研究成果，老師對學問的嚴謹態度更是我學習的典範。

感謝三位口試委員吳毅成教授、雷欽隆教授以及顏嗣鈞教授，你們對本論文的寶貴意見及指教，使得本論文能夠更充實且完善。感謝鈦象電子公司提供我獎助學金，對我是很大的鼓勵。

感謝實驗室的同學們寧玄、阿中、阿北、忠衛，以及所有學長姐及學弟妹，我們一起相互扶持成長，也一起吃喝玩樂，有你們的陪伴，讓這個實驗室總是充滿歡笑，讓我的研究生生活多采多姿。尤其是康樂股長寧玄，為實驗室辦了許多有趣的活動，而跟吉他手阿北一起組團，幫我圓了搖滾夢。特別感謝國華學長對我的細心指導，讓我能快速熟悉做研究所需的技能，跟學長無數次的開會及討論，讓我的研究過程能夠更加順利。還要感謝小馬在我寫論文的期間，一直陪伴在我身邊，你的支持總是讓我充滿能量，有了前進下去的動力。

最後，要感謝我親愛的父母親及外婆，提供我最溫暖的避風港，每次回家都能感受到滿滿的愛與關懷，讓我覺得我是最幸福的小孩，因為有你們才能有現在的我，希望你們會滿意我的表現。

目 錄

第一章 簡介	1
1.1 研究背景	1
1.2 相關研究	2
1.3 系統簡介	3
1.4 各章介紹	4
第二章 相對於棋盤之棋子定位	5
2.1 偵測棋盤格線角落	7
2.2 偵測新置放之棋子	7
2.2.1 偵測棋子區域.....	7
2.2.2 排除有下棋動作干擾之變動畫面.....	9
2.2.3 找出新置放之棋子.....	11
2.2.4 分析新置放棋子之資訊.....	12
2.3 相對於棋盤之棋子定位	12
2.3.1 平面投影轉換方法簡介.....	12
2.3.2 轉換棋子座標.....	15
2.4 排解棋盤移動問題	16
2.4.1 偵測棋盤移動.....	16
2.4.2 重新計算投影轉換矩陣.....	18
2.5 總結	18
第三章 基於棋子相對位置之定位	20
3.1 偵測新置放棋子	22
3.2 基於棋子相對位置之定位	22
3.2.1 建立棋子間的相對位置.....	22
3.2.2 重建棋子座標.....	29
3.3 排解棋盤移動問題	33
3.3.1 利用群組選出參考點.....	34
3.3.2 利用Cross-Ratio找出參考點之對應關係	37
3.3.3 轉換移動前紀錄.....	40
3.4 總結	41
第四章 實驗結果與討論	42
4.1 棋譜格式與呈現	42
4.2 棋譜分析結果與討論	43
第五章 結論與未來展望	47
參考文獻	48
附錄 棋子間距臨界值的決定	49

圖表目錄

圖 1-1	系統拍攝畫面。	3
圖 1-2	系統概略流程圖。	3
圖 2-1	「相對於棋盤之棋子定位」系統流程圖。	6
圖 2-2	棋盤格線角落 (紅色所標示之位置)。	7
圖 2-3	對於輸入影像之初步處理。	8
圖 2-4	系統所偵測之棋子區域 (STONE REGION)。	9
圖 2-5	判斷 $frame_i$ 是否為關鍵影格之流程圖。	10
圖 2-6	找出新置放之棋子。	11
圖 2-7	利用平面投影轉換做平面間點對點的轉換。	13
圖 2-8	系統中的轉換平面及參考點。	15
圖 2-9	新置放棋子之座標轉換。	16
圖 2-10	棋盤移動前後之影像及其棋子區域。	17
圖 2-11	棋盤移動前後之棋子區域比較。(A) $frame_i$ 及 $frame_{i-n}$ 之比較。綠色區域為 P_1 ，橘色區域為 P_2 ，藍色區域為 P_3 。(B)重疊區域之示意圖。	18
圖 3-1	「基於棋子相對位置之定位」系統流程圖。	21
圖 3-2	棋子之間的相鄰情形。	22
圖 3-3	兩顆棋子之間的距離。綠色線為水平相鄰棋子間的距離，紅色線為斜角相鄰棋子間的距離；深藍色標示棋子的輪廓，淺藍色標示棋子的質心。	23
圖 3-4	兩棋子距離的計算方式。圖中灰色線為 L ，紅色點為輪廓與 L 之交點。	24
圖 3-5	第二屆交通大學盃六子棋公開賽決賽之最後畫面。	25
圖 3-6	使用距離資訊建立棋子間相對位置的結果。橘色箭頭所指處為關係判斷錯誤之線段。	25
圖 3-7	改良後之使用距離資建立棋子間相對位置的結果。	26
圖 3-8	利用角度資訊判斷棋子關係的例子。灰色輪廓為所有與 S_1 及 S_2 相鄰的棋子。	27
圖 3-9	本例中棋盤格線的變形情況。棋盤格線的線條 (綠色線條) 依然是完全筆直，但夾角已經明顯不是 90° ；而棋盤格線的斜角連線 (紅色線條) 大致還是成一直線，其夾角偏離 90° 更多。	28
圖 3-10	使用距離及角度資建立棋子間相對位置的結果。	28
圖 3-11	建立棋子間相對位置的最後結果。	29
圖 3-12	建立投影轉換矩陣的參考點。	31
圖 3-13	本系統所記錄之棋譜的最後結果。	32
圖 3-14	棋盤移動前後之影像及其棋子區域。	34
圖 3-15	棋子區域中的單顆棋子區域。	35
圖 3-16	單顆棋子區域分群的結果。	36

圖 3-17	單顆棋子區域逆時針掃描的結果。	36
圖 3-18	最後被選擇當做參考點的棋子 (橘色所標示者)。	37
圖 3-19	平面上五點計算二維CROSS-RATIO。	38
圖 3-20	透視投影模型。	39
圖 3-21	所有移動前棋子輪廓轉換至移動後的結果。	41
圖 4-1	六子棋棋譜呈現介面。	42
圖 4-2	較大俯角拍攝棋盤之實驗結果。	43
圖 4-3	較小俯角拍攝棋盤之實驗結果。	44
圖 4-4	偏斜小方位角拍攝棋盤之實驗結果。	45
圖 4-5	偏斜大方位角拍攝棋盤之實驗結果。	46



第一章 簡介

1.1 研究背景

棋譜是一盤棋局發展的流程紀錄，棋盤遊戲（如六子棋、圍棋、象棋...等）常常需要記錄棋譜，以供事後的檢討或流傳，尤其在重要的賽局，記錄棋譜更是必要的步驟。不論是單純的練習到大型比賽，目前僅能依靠人工來記譜，若是人手不足的時候，棋手就得一邊下棋一邊分心記譜了。因此若是有一套方便使用的自動棋譜產生系統，對於棋手來說會是非常方便且期盼的事。

我們所提出的這套系統只要在下棋時利用攝影機拍攝棋盤，便可經由程式自動記錄每一步棋的位置及時間等資訊，而且只需要一般的攝影機及筆記型電腦即可執行此自動棋譜產生系統。除了節省人力之外，所產生的棋譜電子檔可以透過軟體呈現，方便閱讀及分析。

我們以六子棋為例，發展這套自動棋譜產生系統。六子棋是由國立交通大學資訊工程系吳毅成教授改良自五子棋而來，正式名稱訂為連六棋（英文名字是Connect6）。同時兼具「規則簡單」、「變化複雜」、「遊戲公平」三個很好的特性。它的規則是：除了黑的第一手下子外，黑白雙方輪流各下兩子，最後，連成六子者勝（類似五子棋）。且由於一次下兩子，組合非常多，可說是千變萬化。目前六子棋已成為一項非常重要的棋類遊戲項目。

在國科會心理人文智慧工程共構之「可塑型優質高齡化生活空間科技計畫」中，此自動棋譜產生系統將大量記錄居家年長者的下棋歷程，並且輸出為SGF(Smart Game Format)的棋譜。我們將此輸出棋譜交由吳毅成教授的程式呈現，並且加入棋型資訊變成有分析價值的資料，再由中央大學認知神經科學研究所張智宏教授根據每一手的棋型與思考時間之關係，做進一步的“年長者空間工作能力”之分析。

此自動棋譜產生系統不只可以記錄六子棋棋譜，圍棋或是任何使用黑白棋子

的棋盤遊戲皆適用，未來更希望可以發展出象棋、西洋棋...等棋譜的產生系統。

1.2 相關研究

自動記譜系統的相關研究其實並不多，直到今日似乎還沒有一套方便或廣泛應用的自動記譜系統。因為在下棋的現場，可能有各種的環境變因，如拍攝角度、棋盤或攝影機移動...等，而目前所被提出的自動棋譜產生系統，並沒有處理這類問題。

[1][2][5]提出了利用影像自動記錄圍棋棋譜的方法。[1][2]是針對數位電視的圍棋節目畫面做處理；在這類的節目中，攝影機會從正上方九十度角向下拍攝棋盤，所以他們不需要處理由於攝影機拍攝角度而造成的棋盤變形問題。[5]的系統有自己的攝影機，但仍然沒有提到拍攝角度的問題。在這三篇文獻中提到的棋子定位方式，都是先找出所有棋盤格線的交點（又稱棋點），再掃描所有棋點的位置，以判別是否有棋子在該棋點上。而找出棋點的方法：[1][2]先利用色彩資訊找出棋盤範圍，接著[1]將棋局的畫面做 Sobel filter 及 Hough transform，再用處理過的影像估算出所有棋點；[2]則是直接利用棋盤格線為 19×19 的特性，估算所有棋點；[5]是利用棋盤邊緣及棋盤格線找出盤面四個頂點，再根據該四個頂點，決定所有棋點的位置。

在找出棋子座標的過程中，棋盤位置是很重要的參考資訊，[1][2][5]都有提出偵測棋盤的方法，但都只針對較單純的場景，若攝影機不是從正上方拍攝，或是場景較複雜，棋盤就不容易被正確偵測了。[3]提出了利用基因演算法找出真實影像中的棋盤邊緣，可將四邊形的模型正確調整至影像中的棋盤位置，是一種限制較小的棋盤偵測方法。

除了圍棋之外，[4]提出了日本將棋的自動記譜方法，同樣是利用影像處理技術，但此篇文獻的重點僅放在不同種類棋子的偵測，離實際記錄棋譜還有一段距離。另外，[6]提出利用檢知器感應下棋時，棋子敲響棋盤之聲音及棋盤之振

動，但此系統僅有計時的功能，且需要特殊設備。

1.3 系統簡介

本篇論文是以一般可見光攝影機影像為來源，且攝影機不需架設在棋盤正上方，可在一定範圍內以任意角度拍攝棋盤，拍攝畫面如圖 1-1所示。我們所建之系統首先將影片輸出為連續多張的影像，以這些影像作為系統的輸入，接著找出影像中新置放之棋子，針對該棋子分析它的顏色、座標等相關資訊，再存入棋譜中。系統的概略流程如圖 1-2 所示。



圖 1-1 系統拍攝畫面。

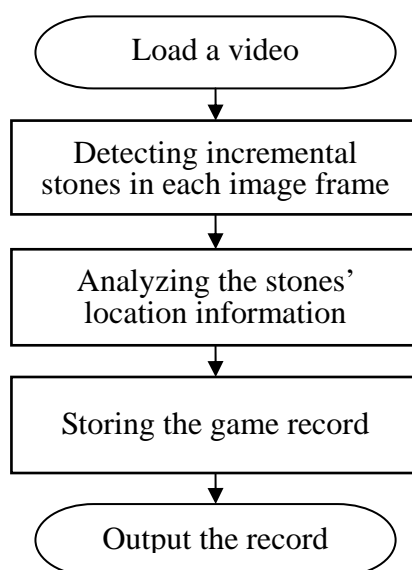


圖 1-2 系統概略流程圖。

在棋子座標的定位上，我們提出了兩種方法：第一種「相對於棋盤之棋子定位」類似[1][2][5]，利用棋盤邊緣資訊定位所有棋子，且容許棋盤變形；另一種「基於棋子相對位置之定位」則是完全創新的方法，不需要棋盤資訊，而是利用棋子之間的相對位置重建出整個座標系。另外針對棋盤或攝影機在拍攝途中移動所造成的棋子座標判斷錯誤，我們也提出了解決的辦法。希望能在最少的使用限制之下，完成最高正確率的自動棋譜產生系統。

1.4 各章介紹

本篇論文全文共分為五章。除本章為簡介外，第二章詳細介紹「相對於棋盤之棋子定位」系統，包含如何偵測棋盤上之新置放棋子，以及利用 Homography 將棋子質心由影像中的座標，轉換到棋盤上的座標。第三章詳細介紹「基於棋子相對位置之定位」系統，包含利用棋子之間的相對位置定位棋子座標，以及藉由找出位移前後棋盤的對應關係，解決棋盤移動問題。第四章介紹系統所產生之棋譜，以及棋譜的呈現與應用。第五章為結論並說明本研究未來的一些研究方向。

第二章 相對於棋盤之棋子定位

本章介紹「相對於棋盤之棋子定位」系統，此方法是較為直覺的棋子定位方法，也是與相關研究較相似的作法；但利用投影轉換矩陣（Homography）把棋子的影像座標轉換至相對於棋盤格線的座標，則是我們首先提出，因此我們的系統對於攝影機的拍攝角度有較大的容忍範圍。系統依序執行以下流程：

- (1) 擷取影片中的一張影像。
- (2) 若此張影像為第一張影像，則跳到 (6)。
- (3) 把影像轉為灰階，並由灰階強度找出影像中的棋子區域。
- (4) 排除有下棋動作干擾之變動畫面，若畫面被干擾則跳到 (9)。
- (5) 偵測棋盤是否移動，若未移動則跳到 (7)。
- (6) 偵測棋盤格線的四個角落，並根據此四個點計算出投影轉換矩陣。然後跳到 (9)。
- (7) 把影像之棋子部份與上一手之棋子狀態比較，找出新放置之棋子。若沒有新放置之棋子則跳到 (9)。
- (8) 分析新放置之棋子的顏色、思考時間、影像中之座標，並且藉由投影轉換矩陣把棋子的影像座標轉換至棋盤的座標。然後將所得到之資訊加到棋譜記錄中。
- (9) 若棋局未結束則跳回 (1)，讀入下一個張影像。否則輸出棋譜記錄。

圖 2-1 為「相對於棋盤之棋子定位」系統流程圖，我們將在以下各節逐一介紹各個步驟的詳細作法與原理。

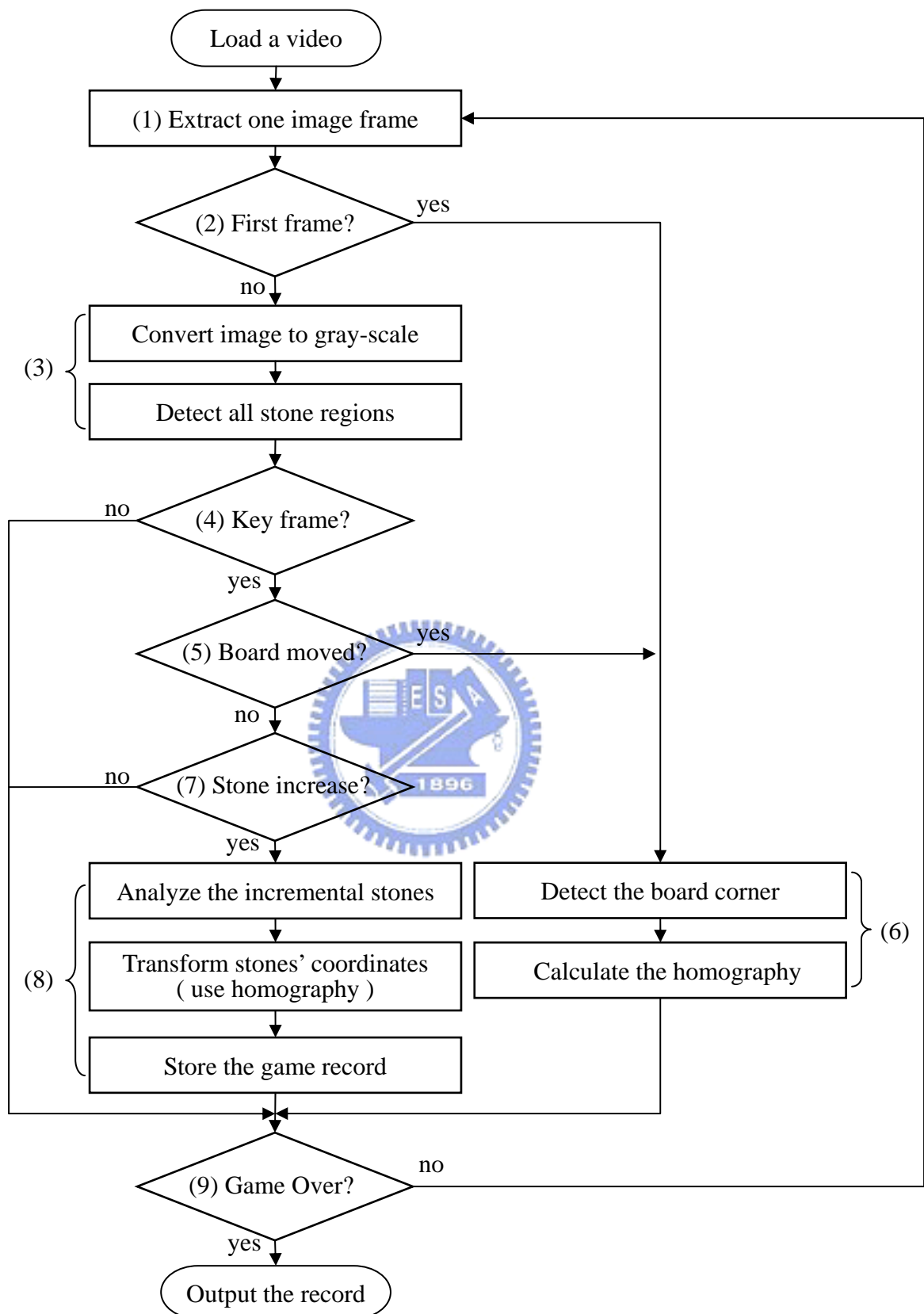


圖 2-1 「相對於棋盤之棋子定位」系統流程圖。

2.1 偵測棋盤格線角落

「相對於棋盤之棋子定位」系統藉由棋盤的位置定位棋子，所以首先我們必須找出棋盤格線的四個角落當作座標轉換之參考點，此四個點的位置如圖 2-2 所示。要準確的找出這四個參考點不容易，[3]提出了一個可行的方法，不過本篇論文的重點並不在此，所以我們直接在棋盤格線的四個角落貼上紅色標記，如此我們就可以利用 RGB 色彩資訊，輕易的找出此四個參考點的位置。



圖 2-2 棋盤格線角落（紅色所標示之位置）。

2.2 偵測新置放之棋子

要分析每一手的棋子資訊，首先須分別找出每一手新置放之棋子。我們的作法是先偵測影像中所有黑白棋子區域，再排除掉有下棋動作干擾之變動畫面，最後找出其中新置放的棋子。以下介紹各步驟的詳細作法。

2.2.1 偵測棋子區域

在利用 RGB 色彩資訊找出畫面中的棋盤區域之後，接下來的所有處理都只針對此棋盤區域，以排除棋盤外其他物品的干擾。由於棋子的顏色非黑即白，對比非常強烈，所以我們直接把輸入系統的彩色影像轉為灰階，由灰階影像偵測棋

子，如圖 2-3(a)(b)所示。如果把輸入影像的灰階強度畫成直方圖，會發現棋子的部份會集中分布在頭尾兩端，而棋盤等非棋子部分則是分布在中間，如圖 2-3(c)所示。於是我們分別對黑子及白子訂下兩個臨界值 K_1 及 K_2 ，像素之影像強度比 0 大 K_1 之內判定為黑子，像素之影像強度比 255 小 K_2 之內判定為白子，我們系統的初始值設 K_1 為 40， K_2 設為 90。處理的結果如圖 2-3(d)所示，藍色的區域為影像中被判定為棋子之像素。

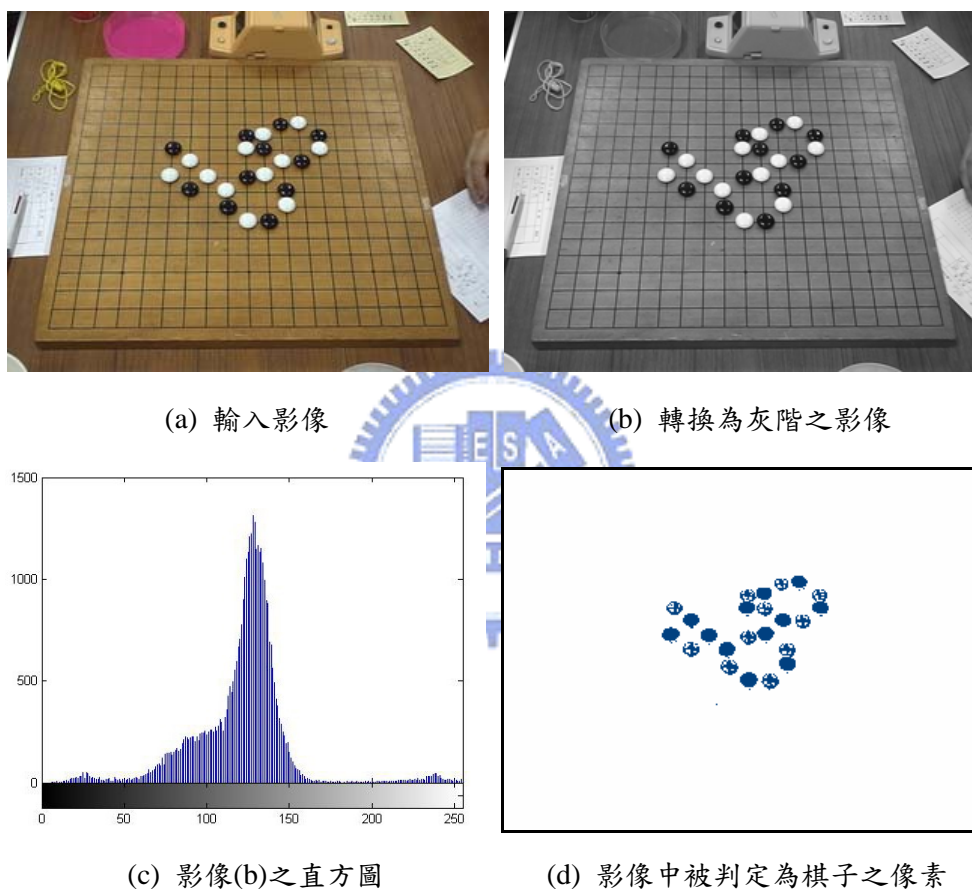


圖 2-3 對於輸入影像之初步處理。

我們會發現圖 2-3(d)中藍色區域有些並不完整，這是因為黑色棋子上面的反光沒有被偵測為棋子區域；此外還有許多小碎點，這是因為少許的棋盤格線被偵測為棋子所造成的。為了排除這些干擾，我們對圖 2-3(d)中藍色區域先進行一個像素的擴張(Dilation)來填補黑色棋子的反光，再進行兩個像素的侵襲(Erosion)排除掉棋盤格線或其他雜訊，最後再進行一個像素的擴張(Dilation)補回藍色

區域原本的大小，如此就可以得到畫面中完整且平滑的棋子區域了，我們把此區域叫做“Stone region”，如圖 2-4 所示。

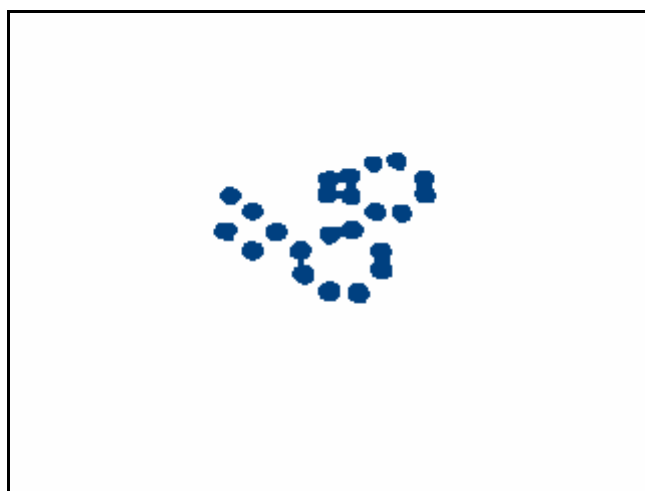
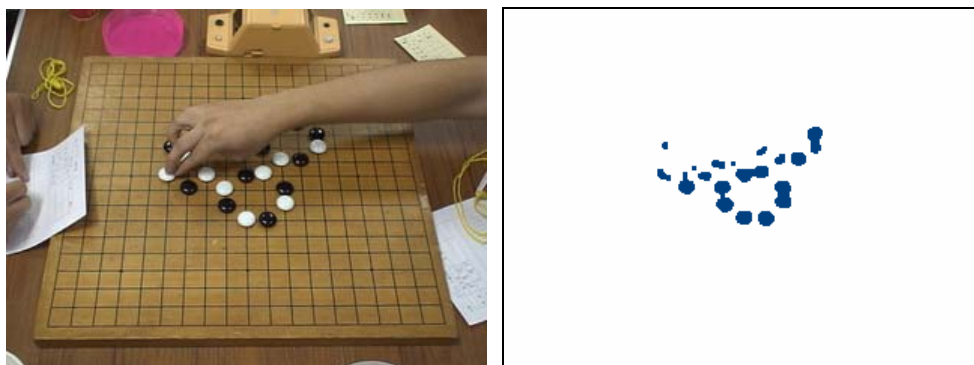


圖 2-4 系統所偵測之棋子區域 (Stone region)。

2.2.2 排除有下棋動作干擾之變動畫面

對於自動棋譜產生系統來說，棋盤上所有棋子以外的東西都是雜訊，尤其是當下棋者的手伸入棋盤時，會使棋子區域偵測錯誤，所以我們必須排除有下棋動作干擾之變動畫面，找出棋盤上只有棋子之穩定畫面，我們稱此畫面為關鍵影格 (Key frame)，若畫面不是關鍵影格，我們就不用對它作後續的處理及分析了。圖 2-5 表示當下棋者正在下棋時的畫面及其棋子區域，可以發現部分棋子被手所遮蔽，而且手的陰影也會影響棋子區域的判定。



(a) 輸入影像

(b) (a)之棋子區域

圖 2-5 有下棋動作干擾之變動畫面。

現在我們找出關鍵影格，也就是已經穩定一段時間之畫面，此處只需偵測棋子區域的穩定度，因為若是有手或其他東西在棋盤上移動，棋子區域一定是會不停變動的。實際作法是把目前畫面的棋子區域跟上一個畫面的棋子區域相減，得到兩張影像之棋子區域的變動像素數目，若此變動數目大於一個臨界值 Th_1 ，我們認為此畫面不是穩定的畫面；否則持續追蹤 Th_2 個畫面，在這期間畫面的變動值都要小於 Th_1 ，我們才認為此畫面為“關鍵影格”。在我們的系統中定 Th_2 為輸入影像之影格速率的一半，也就是約半秒的時間。圖 2-5 表示系統判斷 $frame_i$ 是否為關鍵影格之流程，其中令 $frame_i$ 為影片中的第 i 個影格， S_i 為 $frame_i$ 之 Stone region，而 $count$ 記錄系統追蹤了幾個穩定的影格。

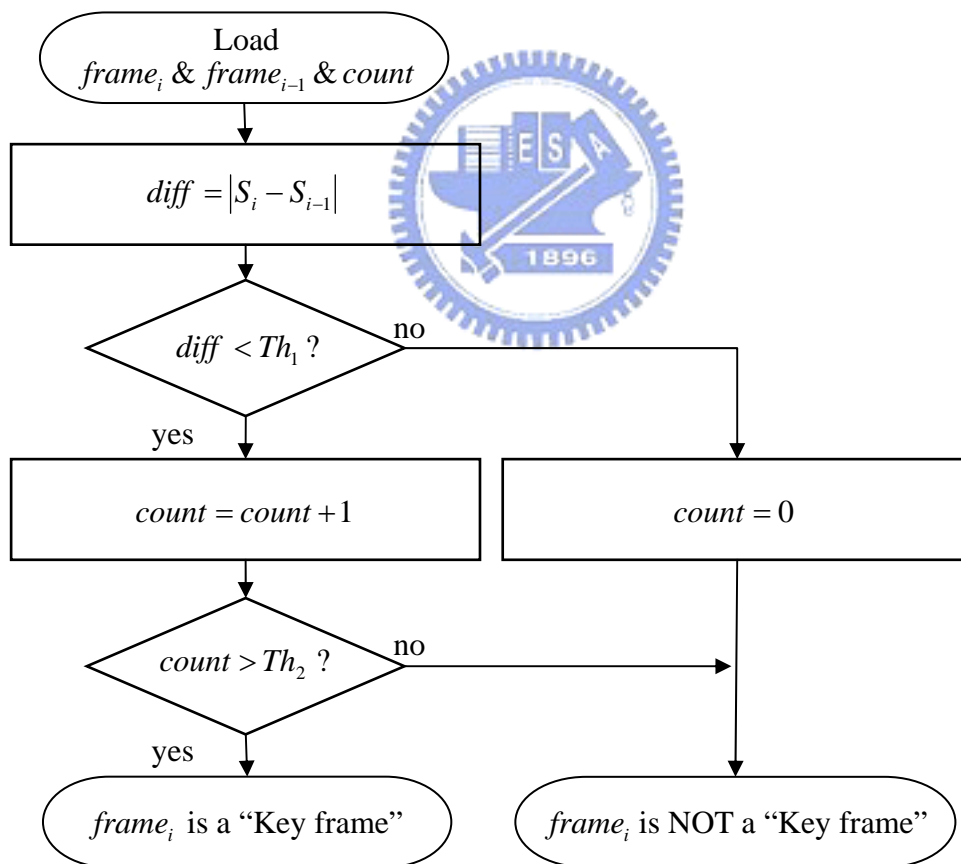


圖 2-5 判斷 $frame_i$ 是否為關鍵影格之流程圖。

2.2.3 找出新置放之棋子

找到關鍵影格之後，我們要進一步找出此關鍵影格中新置放之棋子，並且記錄此棋子的輪廓及質心。若 $frame_i$ 為關鍵影格， $frame_{i-n}$ 為前一個關鍵影格，如圖 2-6(a)(b)，則我們把這兩個影格之棋子區域相減，會得到 $frame_i$ 之新增的棋子區域，稱之為 NS_i ，如圖 2-6(c)。由於六子棋一次下兩子的特殊規則，有可能在一個關鍵影格中有兩顆新置放之棋子，所以我們先找出 NS_i 中兩個最大的區域，以去除大部分非棋子的雜訊，若此兩個區域都大於最小棋子大小 S_{TH} ，則知道這個關鍵影格有兩顆新置放之棋子；若只有其中一個區域大於 S_{TH} ，則這個關鍵影格只有一顆新置放之棋子；若此兩個區域都小於 S_{TH} ，那麼這個關鍵影格沒有任何新置放之棋子。若此關鍵影格有新置放之棋子，則分別找其輪廓及質心，把此資訊存在記錄器中，等待後續的處理，結果如圖 2-6(d)所示。

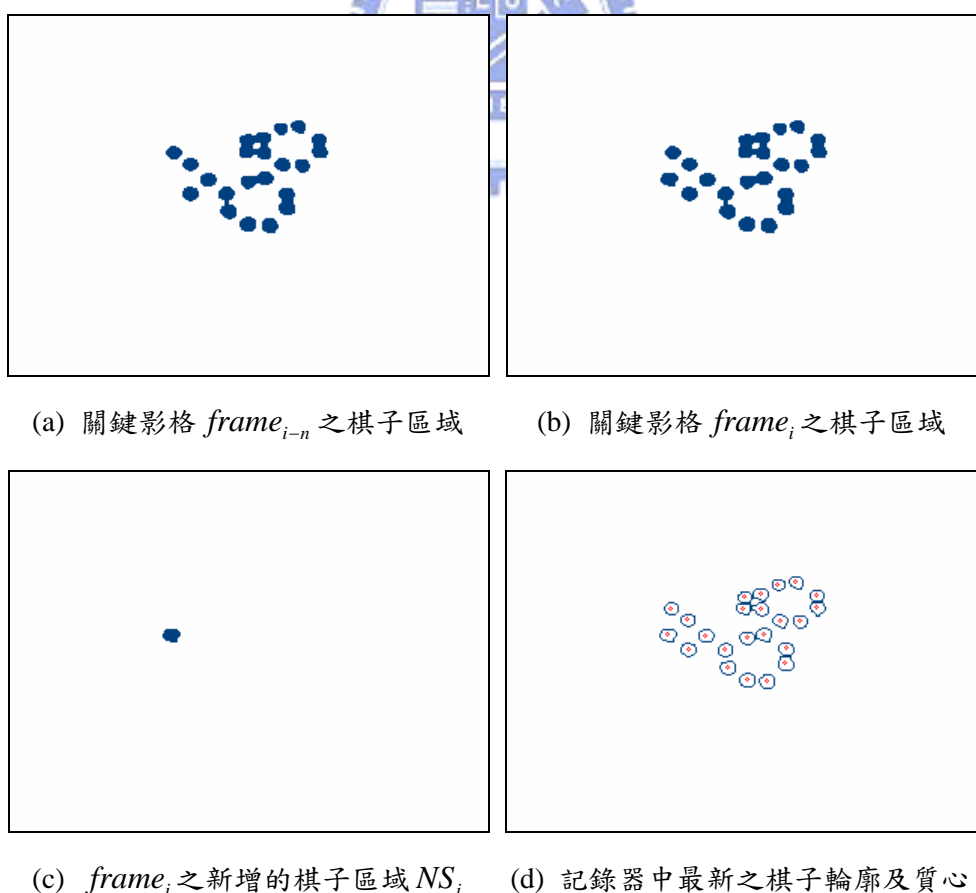


圖 2-6 找出新置放之棋子。

2.2.4 分析新置放棋子之資訊

對於每一手新置放之棋子，我們所需要的資訊有棋子顏色、思考時間以及棋子相對於棋盤格線的座標，除了棋子相對於棋盤格線的座標之定位將在下一章介紹外，以下介紹如何找出棋子顏色及思考時間。

假設 $frame_i$ 為有新置放棋子的關鍵影格， $frame_{i-n}$ 為前一個有新置放棋子的關鍵影格。要知道 $frame_i$ 中新置放棋子的顏色，我們把 $frame_i$ 之新增的棋子區域 NS_i 當作遮罩，偵測 $frame_i$ 之灰階影像中 NS_i 區域的灰階強度，並依灰階強度將每個像素分類為黑子或白子，最後被歸類為黑子的像素多則新置放之棋子為黑子，反之則為白子。要知道 $frame_i$ 中新置放棋子的思考時間，則計算前後兩個有新置放棋子的關鍵影格的相差影格數 n ，再把 n 除以影格速率 (frame rate)，即為此手棋之思考時間了。



2.3 相對於棋盤之棋子定位

「相對於棋盤之棋子定位」是使用平面投影轉換 (Homography) 的觀念。所以在此節，我們先對平面投影轉換的理論做簡單的介紹，再說明如何藉由此理論定位棋子相對於棋盤格線的座標。

2.3.1 平面投影轉換方法簡介

平面投影轉換的簡單定義為，有兩平面 π_A 、 π_B ，要將一群位在 π_A 平面上的點集合 x_i ，透過透視投影對應到另一平面 π_B 上的點集合 x_i' ，如圖 2-7 所示。其間存在一個 3×3 的投影轉換矩陣 H ，可將 x_i 中的每個點轉換至 x_i' 中對應的點。如果以齊次座標表示法表示，而 x_i 與 x_i' 分別表示為 $x_i = (u_i, v_i, w_i)^T$ 、 $x_i' = (u_i', v_i', w_i')^T$ ，則 x_i 、 x_i' 與 H 間的關係為：

$$Hx_i = \lambda x_i' \quad (2.1)$$

其中 λ 是不為 0 的比例常數， H 則是可逆 (non-singular) 矩陣。

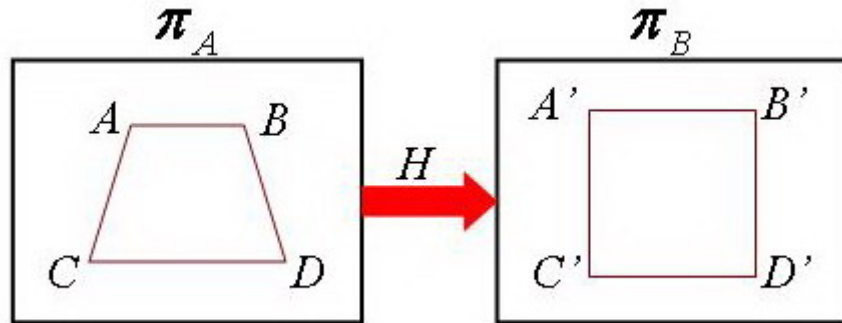


圖 2-7 利用平面投影轉換做平面間點對點的轉換。

2.3.1.1 齊次線性解

若以外積的形式來表示，則(2.1)可以寫成：

$$x_i' \times Hx_i = 0, \quad H = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \quad (2.2)$$

若將 H 的第 j 列以 h^{jT} 表示，再將(2.2)展開，則可以推導出：

$$x_i' \times Hx_i = \begin{bmatrix} v_i' h^{3T} x_i - w_i' h^{2T} x_i \\ w_i' h^{1T} x_i - u_i' h^{3T} x_i \\ u_i' h^{2T} x_i - v_i' h^{1T} x_i \end{bmatrix} \quad (2.3)$$

因為 $h^{jT} x_i = x_i^T h^j$, $\forall j \in N, 1 \leq j \leq 3$ ，藉由此關係式化簡後，(2.3)可以寫成以 H 列

向量為變數的式子：

$$\begin{bmatrix} 0^T & -w_i' x_i^T & v_i' x_i^T \\ w_i' x_i^T & 0^T & -u_i' x_i^T \\ -v_i' x_i^T & u_i' x_i^T & 0^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (2.4)$$

(2.4)雖然是由三個方程式構成，但只有前兩個方程式是線性獨立的，所以只需要利用(2.4)的前兩個方程式即可，亦即：

$$\begin{bmatrix} 0^T & -w_i'x_i^T & v_i'x_i^T \\ w_i'x_i^T & 0^T & -u_i'x_i^T \end{bmatrix} \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix} = 0 \quad (2.5)$$

至此已經推導出以 h 為變數的方程式，接下來則要把 $h_i, 1 \leq i \leq 9$ 解出，即可解出 H 。

2.3.1.2 求解平面投影轉換矩陣

投影轉換矩陣 H 有九個未知數，但因為其中一個為縮放係數 (up to scale)，故其自由度為 8，也就是至少需要八條方程式才能解出 H 。而因為二維平面有 x 和 y 座標，所以給予一組對應點，就可以得到兩條方程式，故需要四組對應點，即能求出 H ，會完全符合這四組點的對應關係。而求解 H 的方法不只一種，我們只介紹在本系統中的作法：

- (1) 每組 x_i 與 x_i' 中的對應點，可從(2.5)中得到一個 2×9 的矩陣 A_i 。
- (2) 因為有 4 組對應點，則可得到 4 個 2×9 的矩陣 A_i ，將這些 A_i 結合成一個 8×9 的矩陣 A 。
- (3) 將 A 做 SVD 分解，得到 $A = UDV^T$ ，其中 D 為 A 的 singular value 所構成的對角矩陣(diagonal matrix)、 V 為 singular vector 所構成的正交矩陣(orthogonal matrix)。我們所要求的 H 是最小的 singular value 所對應的 singular vector，一般而言，SVD 分解後的 D 會將 singular value 由大到小排列，意即 V 中第 9 行的九個元素： $[h_1 \ h_2 \ h_3 \ h_4 \ h_5 \ h_6 \ h_7 \ h_8 \ h_9]^T$ ，為構成 H 的元素。
- (4) $h_i, 1 \leq i \leq 9$ 已計算出，則根據(2.2)中 H 的定義，即能求出 H 。

2.3.2 轉換棋子座標

由上一小節的說明可以得知，兩個不同的平面上只要有四個相對應的參考點，就可以求出它們之間的投影轉換矩陣，有了此投影轉換矩陣，就可把一個平面中的任意點，轉換到另一個平面中。在「相對於棋盤之棋子定位」的系統中，我們由影像中的棋盤平面，轉換到 19×19 的真實棋盤的座標平面，而棋盤格線的四個角落就是其參考點，我們由此參考點求得兩個棋盤平面間的投影轉換矩陣 H ，如圖 2-8 所示。

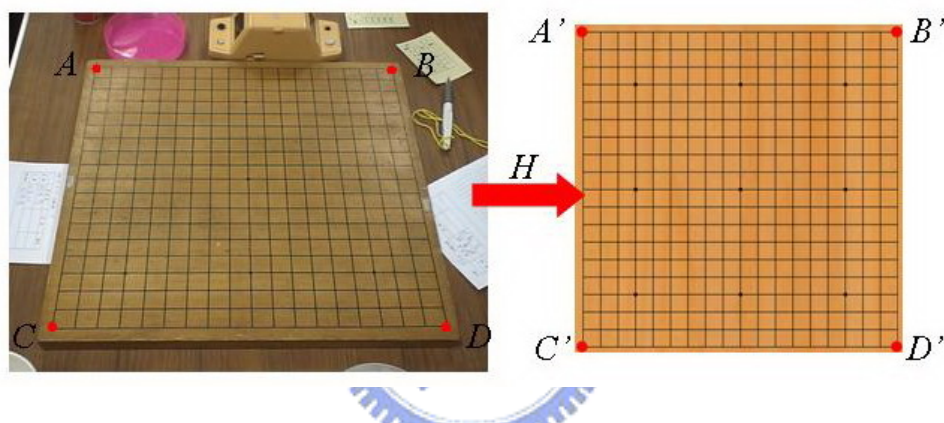


圖 2-8 系統中的轉換平面及參考點。

另外，利用在 2.2 節中介紹的方法，我們已經知道每一手新置放棋子的質心，也就是該棋子在影像平面上的座標位置 x_i ，乘上投影轉換矩陣 H ，可得到 $Hx_i = x_i'$ ， x_i' 即是該新置放棋子在真實棋盤平面的座標位置，如圖 2-9 所示。但是由於棋子放置的偏斜或是轉換造成的誤差，我們所得到 x_i' 不一定是整數座標，而棋子一定要放在真實棋盤平面的格線交點（即整數座標）才有意義，所以我們對 x_i' 四捨五入，藉此找到最接近 x_i' 之真實棋盤平面的格線交點 x_i'' ， x_i'' 即為此新置放棋子最後被記錄至棋譜內的座標位置。

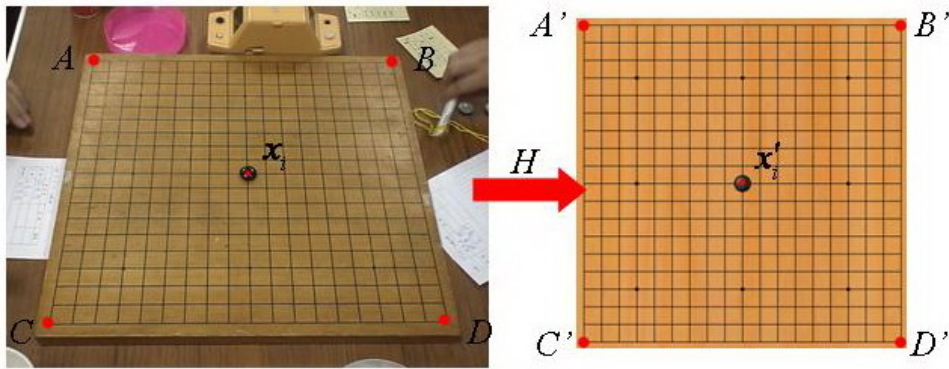


圖 2-9 新置放棋子之座標轉換。

2.4 排解棋盤移動問題

在下棋的過程中，棋盤或攝影機偶而會被不小心撞到而移動，此時攝影機拍攝到的畫面就會改變，而造成新置放的棋子偵測錯誤，以及棋子座標轉換錯誤。因此在此節中，我們將會探討如何偵測棋盤移動，以及棋盤被移動後的處理對策。



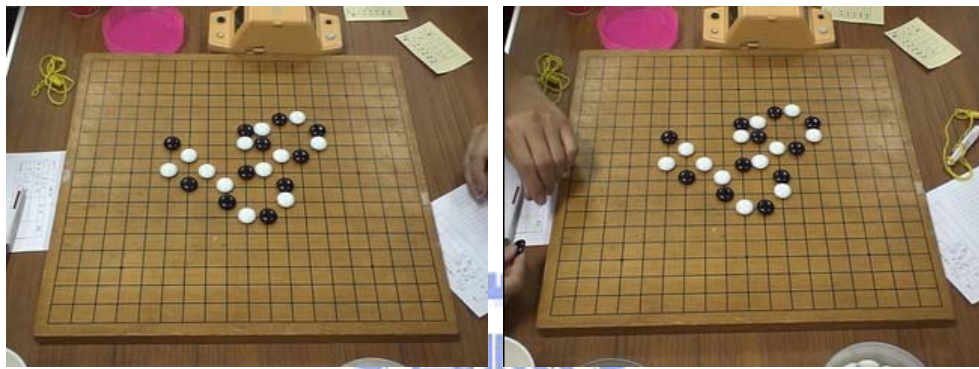
2.4.1 偵測棋盤移動

跟偵測新置放之棋子一樣，我們利用前後兩個關鍵影格的棋子區域來判定棋盤是否移動。假設 $frame_i$ 為一個關鍵影格， $frame_{i-n}$ 為前一個關鍵影格，他們的棋子區域為 S_i 及 S_{i-n} ，圖 2-10 展示了一個棋盤移動的例子；令 P_1 為 $S_i \cap \bar{S}_{i-n}$ ，即原本有棋子後來沒有的區域， P_2 為 $S_{i-n} \cap \bar{S}_i$ ，即原本沒有棋子後來有的區域， P_3 為 $S_i \cap S_{i-n}$ ，即一直都有棋子的區域，如圖 2-11 所示。若在 $frame_i$ 與 $frame_{i-n}$ 之間發生棋盤移動，我們可以預想 P_1 與 P_2 大小應該相當，此外，為了避免 P_1 與 P_2 都是雜訊而不是真的棋子位移， P_1 與 P_2 加起來應該要跟 P_3 有一定比例的關係。以上的條件可整理如下：

$$\left| \frac{P_1}{P_2} - 1 \right| < K_1 \quad (2.6)$$

$$(P_1 + P_2) > K_2 \times P_3 \quad (2.7)$$

其中 K_1 及 K_2 為比例常數，在我們的系統中定 K_1 為 0.5，意即 P_1 與 P_2 大小不能相差超過一半；定 K_2 為 0.3，意即棋子至少要位移 15%。若同時符合(2.6)(2.7)，系統才會判定為棋盤移動。



(a) $frame_{i-n}$ 之原始影像 (棋盤移動前) (b) $frame_i$ 之原始影像 (棋盤移動後)



(c) $frame_{i-n}$ 之棋子區域 S_{i-1}

(d) $frame_i$ 之棋子區域 S_i

圖 2-10 棋盤移動前後之影像及其棋子區域。

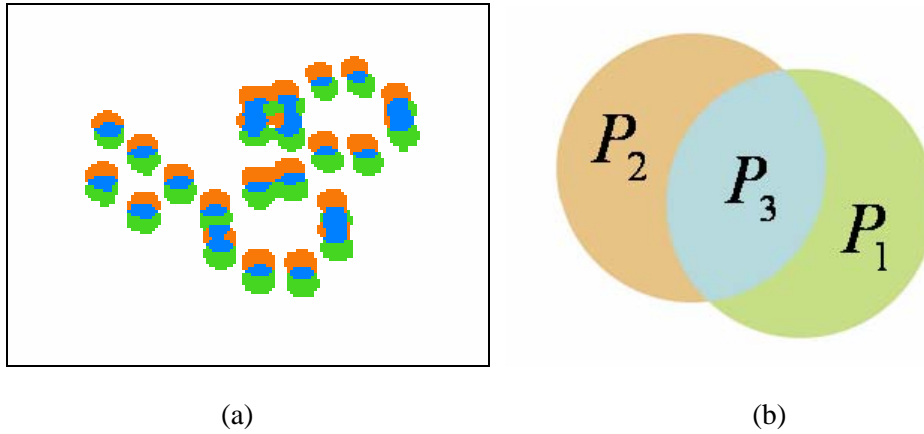


圖 2-11 棋盤移動前後之棋子區域比較。(a) $frame_i$ 及 $frame_{i-n}$ 之比較。綠色區域為 P_1 ，橘色區域為 P_2 ，藍色區域為 P_3 。(b)重疊區域之示意圖。

2.4.2 重新計算投影轉換矩陣

對於「相對於棋盤之棋子定位」的系統來說，棋盤移動不是太難解決的問題，我們只需要重新執行一次 2.1 及 2.3 節介紹的程序，就是重新偵測棋盤格線的四個角落，以及計算出新的投影轉換矩陣。其實對於每一個影格或是關鍵影格都重新偵測棋盤交點也可以避免棋盤移動造成的棋子定位錯誤，就像[2][5]的作法，但是這樣會加重系統的負擔，影響處理效能，而我們提出了一個偵測棋盤移動的機制，只有在棋盤移動後才重新偵測棋盤交點以及計算投影轉換矩陣，節省了許多系統資源。

2.5 總結

在本章中，我們完整介紹了「相對於棋盤之棋子定位」的自動棋譜產生系統，包含了找出新置放棋子、利用投影轉換矩陣定位棋子座標、棋盤移動的偵測及處理。我們拍攝了數段影片當作系統輸入，發現除非棋子放置的位置偏斜非常多，系統都能正確的判斷出每一手新置放棋子的座標。

「相對於棋盤之棋子定位」的方法直覺且較簡單，缺點就是棋盤格線的四個

角落不易正確偵測，所以需要貼上紅點當作標記，稍微降低了此系統的方便性。在下一章中，我們將介紹「基於棋子相對位置之定位」系統，此方法完全不需要棋盤的資訊，所以可以免除在棋盤貼標記的麻煩，是一個創新且更具彈性的棋子座標定位方法。



第三章 基於棋子相對位置之定位

本章介紹「基於棋子相對位置之定位」系統，由於相關研究中並無類似方法定位棋子座標，因此是一個完全創新的作法。此方法完全不需要棋盤的資訊，而是藉由建立棋子之間的相對位置關係來計算棋子在棋盤上之座標，並且容許攝影機在一定範圍內以任意角度拍攝棋盤。系統依序執行以下流程：

- (1) 擷取影片中的一張影像，把影像轉為灰階，並由灰階強度找出影像中的棋子區域。
- (2) 排除有下棋動作干擾之變動畫面，若畫面被干擾則跳到 (7)。
- (3) 偵測棋盤是否移動，若發現移動則跳到 (6)。
- (4) 把影像之棋子部份與上一手之棋子狀態比較，找出新放置之棋子。若沒有新放置之棋子則跳到 (7)。
- (5) 分析新放置之棋子的顏色、思考時間、影像中之座標。然後將所得到之資訊加到棋譜記錄中。然後跳到 (7)。
- (6) 找出棋盤移動前後至少四組對應棋子，並計算投影轉換矩陣，以轉換棋盤移動前之紀錄到移動後的狀態。
- (7) 若棋局未結束則跳回 (1)，讀入下一個張影像。
- (8) 建立棋子之間的相對位置，並藉由此關係重建出所有棋子之相對於棋盤格線的座標。

圖 3-1 為「基於棋子相對位置之定位」系統流程圖，我們將在以下各節逐一介紹各個步驟的詳細做法與原理。

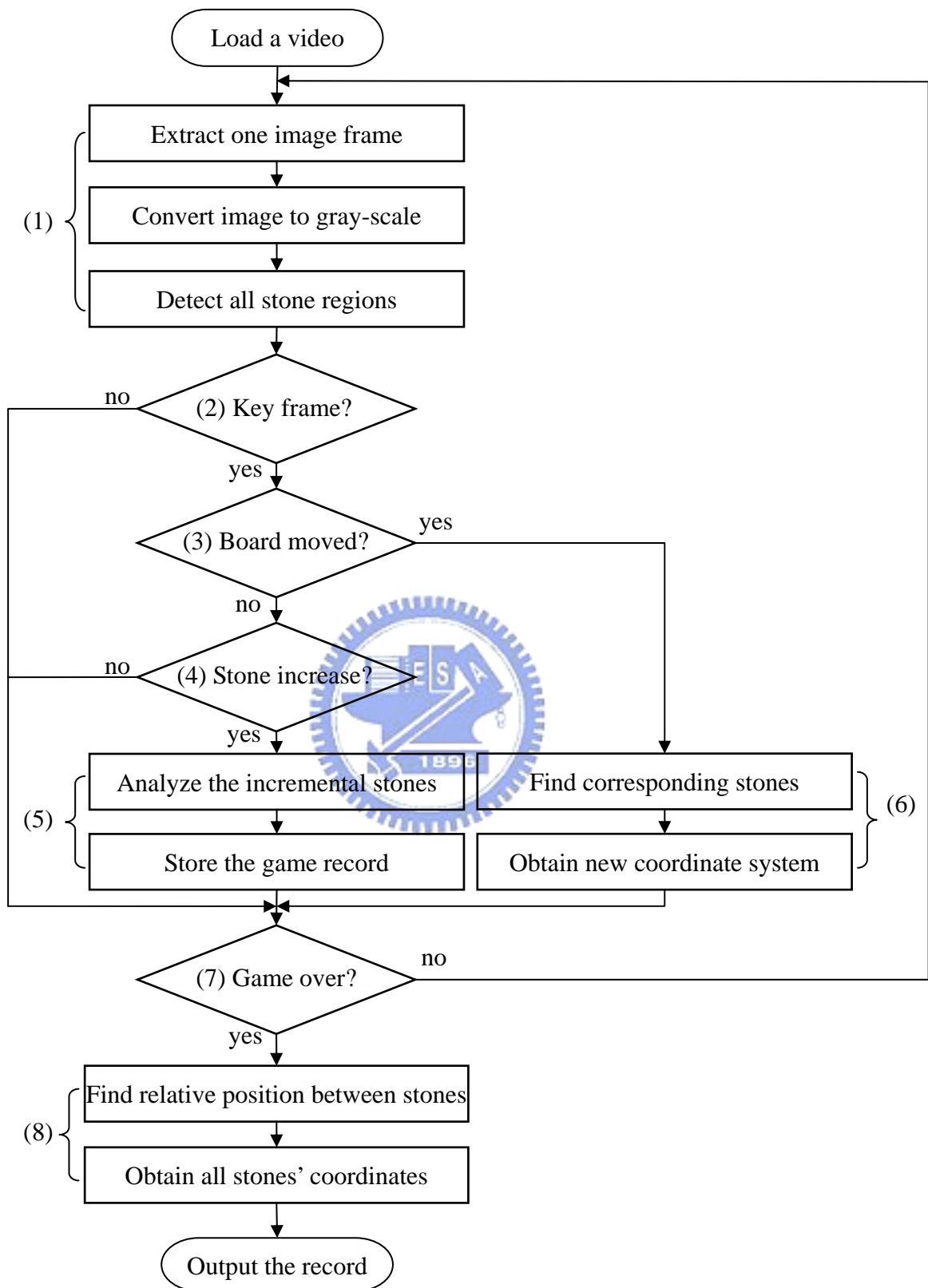


圖 3-1 「基於棋子相對位置之定位」系統流程圖。

3.1 偵測新置放棋子

由於此處有關「基於棋子相對位置之定位」系統偵測新置放棋子的作法，與先前「相對於棋盤之棋子定位」系統一模一樣，所以在此就不再重新說明一次。基本上是沿用 2.2 節所提出的方法找出每一手新置放之棋子，並分析出它的棋子顏色、思考時間以及棋子在影像平面中之座標及輪廓。下一節將繼續介紹如何利用不同於 2.3 節之方法，僅利用棋子間的相對位置，來定位棋子之相對於棋盤格線的座標。

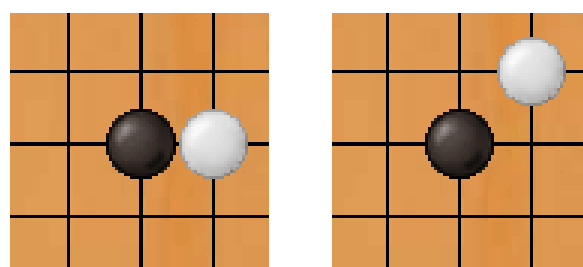
3.2 基於棋子相對位置之定位

我們在棋局結束後，才進行「基於棋子相對位置之定位」。此定位方法有兩個步驟，首先需建立起所有棋子間的相對位置，再基於此相對位置，來決定棋子相對於棋盤格線的座標。以下將介紹這兩個步驟的詳細作法。



3.2.1 建立棋子間的相對位置

建立棋子間相對位置的實際作法，是針對每一顆棋子 S ，找到所有與 S 相鄰的棋子，並判斷每一顆相鄰棋子與 S 的關係，是水平或垂直的相鄰，還是斜角的相鄰，如圖 3-2 所示。若是水平或垂直的相鄰，則我們可以知道該相鄰棋子與 S 在相對於棋盤格線的座標上， x 或 y 座標相差 1；若是斜角的相鄰，則 x 及 y 座標均相差 1。



(a) 水平相鄰

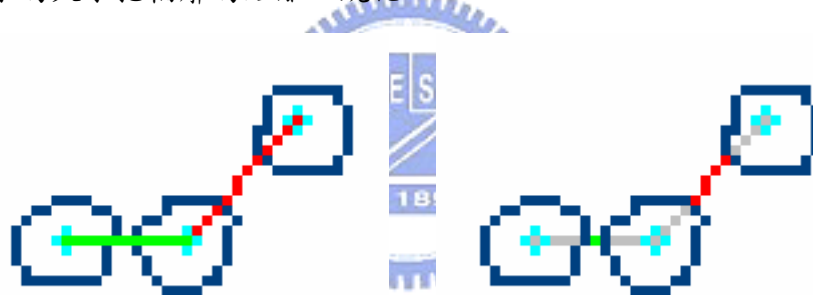
(b) 斜角相鄰

圖 3-2 棋子之間的相鄰情形。

建立棋子間相對位置所需要用到的資訊，是記錄器中每一手棋子在影像平面中之“座標（即質心）”及“輪廓”，在 3.1 節中我們已經得知這些資訊，這些資訊可以幫助我們得到棋子之間的“距離”及“角度”。接下來我們就要介紹如何利用“距離”及“角度”，一步一步建立起棋子間的相對位置。

3.2.1.1 利用距離資訊判斷關係

我們的系統容許攝影機在一定範圍內以任意角度拍攝棋盤，所以畫面中的棋盤會有變形的情況，相鄰的兩顆棋子在棋盤上不同的位置，在影像平面中會有不同的質心距離。為了能更明確的分辨兩棋子是否相鄰，這裡所使用的距離資訊，並不是兩顆棋子之質心距離，而是兩對應輪廓之間的最近距離，如圖 3-3 所示，並且用棋子的大小把輪廓的距離正規化。



(a) 棋子質心的距離

(b) 棋子輪廓的距離

圖 3-3 兩顆棋子之間的距離。綠色線為水平相鄰棋子間的距離，紅色線為斜角相鄰棋子間的距離；深藍色標示棋子的輪廓，淺藍色標示棋子的質心。

假設我們要求出相鄰棋子 S_1 與 S_2 之間的距離，令通過兩棋子質心的直線為 l ， l 與 S_i 之輪廓的交點為 C_i^1 與 C_i^2 ， i 為 1 或 2，如圖 3-4 所示，而兩棋子輪廓間的距離指的是輪廓間最小的距離，即為：

$$contourdist' = \min \{d(C_i, C_j) | C_i \in \{C_1^1, C_1^2\}, C_j \in \{C_2^1, C_2^2\}\} \quad (3.2)$$

我們訂一顆棋子的直徑為 R ，由於棋盤平面在畫面中變形，所以這 R 在畫中的大小並不固定，因此需要知道 R 在棋盤上不同位置的大小，藉此對棋子輪廓的距

離正規化。對於 S_1 與 S_2 來說，我們把 R 的大小定為兩棋子的平均大小，且 S_i 的大小為 C_i^1 與 C_i^2 的距離 $d(C_i^1, C_i^2)$ ，亦即：

$$R = (d(C_1^1, C_1^2) + d(C_2^1, C_2^2)) / 2 \quad (3.1)$$

最後，把兩棋子間的輪廓距離正規化，即為：

$$\text{contourdist} = \text{contourdist}' / R \quad (3.3)$$

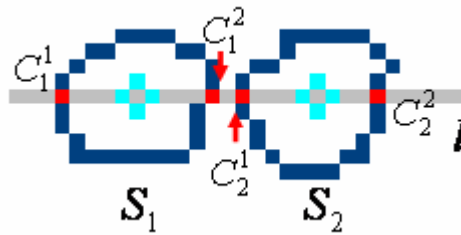


圖 3-4 兩棋子距離的計算方式。圖中灰色線為 l ，紅色點為輪廓與 l 之交點。

接著我們就用兩顆棋子正規化後的輪廓距離，來判斷兩棋子之間的關係。已知棋盤格線一格的寬度，約為棋子的尺寸 R ，所以，若兩顆棋子為水平或垂直相鄰，則理論上輪廓距離接近 0，若為斜角相鄰，則理論上輪廓距離為 $(\sqrt{2}-1)R$ 。於是，我們定棋子之間的距離小於 TH_1 為水平或垂直相鄰， TH_1 以上到 TH_2 為斜角相鄰， TH_2 以上不相鄰。由於 R 在畫面中的大小約為 10 個像素，且系統偵測棋子區域會有少許誤差，所以在實作上，我們定 TH_1 為 0.35，定 TH_2 為 0.8，定下此數值的相關討論請見附錄。

我們以 2007 第二屆交通大學盃六子棋公開賽之決賽的實況錄影為例，實作所提出的“建立棋子間相對位置”方法。圖 3-5 為決賽之最後畫面，圖 3-6 表示用距離資訊建立棋子間相對位置的結果，若兩棋子被判定為水平或垂直相鄰，則其質心以綠色線段連接，若被判定為斜角相鄰，則以紅色線段連接。



圖 3-5 第二屆交通大學盃六子棋公開賽決賽之最後畫面。

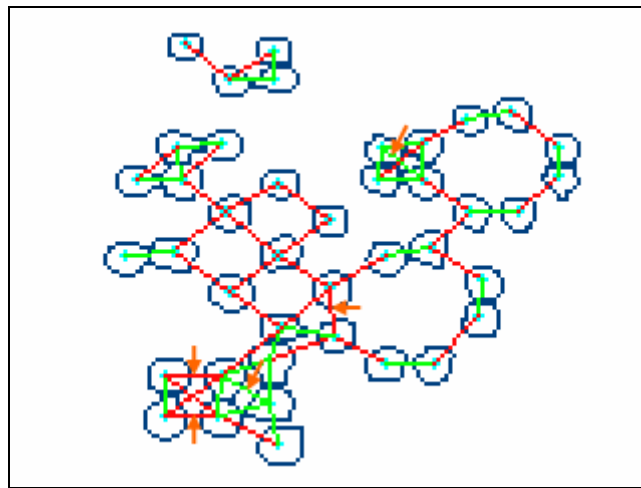


圖 3-6 使用距離資訊建立棋子間相對位置的結果。橘色箭頭所指處為關係判斷錯誤之線段。

由圖 3-6 的結果，我們發現有幾處發生位置關係之判斷錯誤，這是因為棋子放置在棋盤上的位置會有些許偏斜，以及偵測棋子區域的少許誤差所造成的。因此，我們不能完全只憑棋子間的距離資訊，建立棋子間的相對位置，於是，我們針對不易使用距離資訊判斷關係的棋子，也就是距離很接近臨界值 TH_1 者，再利用角度資訊判斷他們的關係。我們稍微改變上述的作法，定棋子之間的距離小於 TH'_1 為水平或垂直相鄰， TH'_1 以上到 TH'_2 為不易以距離判斷關係， TH'_2 以上到 TH'_3 為斜角相鄰， TH'_3 以上不相鄰。我們的系統定 TH'_1 為 0.3， TH'_2 為 0.45， TH'_3 為 0.8，相關討論請見附錄。改良後的判斷結果如圖 3-7 所示，若兩棋子被判定

為“不易以距離判斷關係”者，其質心以黃色線段相連，會發現原本判斷錯誤的線段，幾乎會被判斷為“不易以距離判斷關係”。

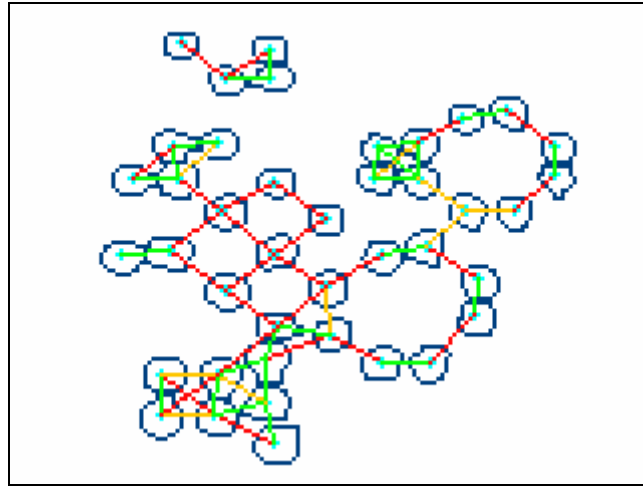


圖 3-7 改良後之使用距離資訊建立棋子間相對位置的結果。

3.2.1.2 利用角度資訊判斷關係

針對在上一小節中被判定為“不易以距離判斷關係”之棋子，此小節將介紹如何利用角度資訊，來判斷它們之間的關係。假設要判定相鄰棋子 S_1 與 S_2 之間的關係，所謂利用角度資訊判斷，看的是兩顆棋子的質心連線線段 l ，但由於畫面中的棋盤會因為拍攝角度變形，所以不管兩棋子是水平、垂直或是斜角相鄰， l 都不會有一個固定的斜率，於是我們由 S_1 及 S_2 與附近棋子的關係，來協助判斷 S_1 與 S_2 的關係。

我們以圖 3-8 的情況為例子，說明如何利用角度資訊判斷 S_1 與 S_2 之關係。由圖中可以看出， S_1 除了與 S_2 相鄰之外，也與另外三顆棋子相鄰，我們令 S_1 與這三顆棋子的質心連線線段，分別為 l_1^1, l_1^2 及 l_1^3 ；而 S_2 除了與 S_1 相鄰之外，也與另外一顆棋子相鄰，令 S_2 與這顆棋子的質心連線線段為 l_2^1 ；再令 $\theta(l_i, l_j)$ 為 l_i 與 l_j 之夾角的弧度，且 $\theta(l_i, l_j) \leq \pi$ 。接下來，我們要分別由 l 與 l_1^1, l_1^2, l_1^3 及 l_2^1 所夾的弧

度，判斷 S_1 與 S_2 的關係。

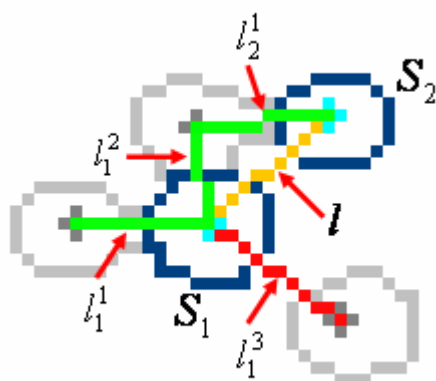


圖 3-8 利用角度資訊判斷棋子關係的例子。灰色輪廓為所有與 S_1 及 S_2 相鄰的棋子。

首先由 l_1^1 開始看起， l_1^1 為綠色線段，表示 l_1^1 的斜率代表的是水平或垂直相鄰，於是我們算出 l 與 l_1^1 之夾角的弧度 $\theta(l, l_1^1)$ ，若 l 與 l_1^1 約成一直線，即 $\pi - TH_1^* \leq \theta(l, l_1^1) \leq \pi$ ，或是 l 與 l_1^1 相差約 90° ，即 $\frac{\pi}{2} - TH_2^* \leq \theta(l, l_1^1) \leq \frac{\pi}{2} + TH_2^*$ ，則 l 與 l_1^1 代表著相同的關係；否則， l 與 l_1^1 代表著不同的關係，本例中 l 與 l_1^1 就代表著不同關係，也就是斜角相鄰。對於其餘線段 l_1^2, l_1^3 及 l_2^1 ，也是進行相同的作法，最後，統計 l 被判斷為水平或垂直相鄰的次數多，還是被判斷斜角相鄰的次數多，即為 l 最後所代表關係，本例中 l 最後被判定為斜角相鄰。

另外，我們要給予 TH_1^* 與 TH_2^* 不同的值， TH_2^* 會比 TH_1^* 大，在我們的系統中定 TH_1^* 為 $\frac{4}{45}\pi$ ，即 16° ，而 TH_2^* 為 $\frac{6}{45}\pi$ ，即 24° 。這是因為，但原本在棋盤上成一直線的格線線條，在棋盤變形後依然是筆直的，而原本成一直線的斜角連線，在棋盤變形後也不會有太大的角度偏斜，如圖 3-8 所示，所以在判斷線段角度相同上，給予較小的彈性 TH_1^* ；但是原本在棋盤上是垂直相交的格線線條及斜角連線，在棋盤變形後他們的夾角會出現較大的變化，如圖 3-9 所示，所以在判斷線段垂直上，給予較大的彈性 TH_2^* 。

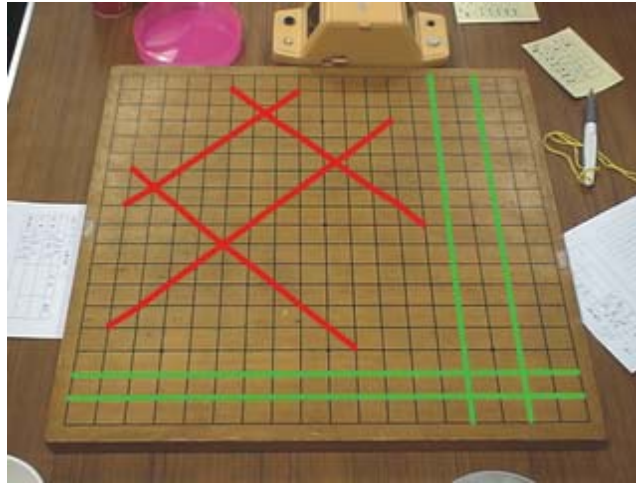


圖 3-9 本例中棋盤格線的變形情況。棋盤格線的線條（綠色線條）依然是完全筆直，但夾角已經明顯不是 90° ；而棋盤格線的斜角連線（紅色線條）大致還是成一直線，其夾角偏離 90° 更多。

執行了上述以角度判斷其關係的方法之後，結果如圖 3-10 所示，我們可以發現，圖 3-7 中原本不易以距離判斷關係的黃色線段，全都藉由棋子間的角度資訊，被標記上正確的顏色。尤其是，圖 3-10 中圓圈範圍內的棋子，原本就擺放雜亂，連肉眼都不易正確判斷它們之間的關係，也可以藉由棋子間距離及角度資訊，建立起正確的關係。

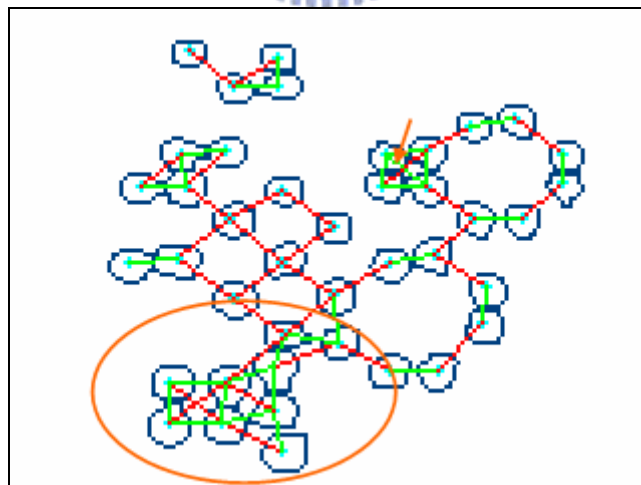


圖 3-10 使用距離及角度資建立棋子間相對位置的結果。

但是，圖 3-10 結果仍然存在一處關係錯誤，就在箭頭所指處。雖然這兩顆棋子為斜角相鄰，但由於它們距離過近，所以在第一階段用距離資訊判斷關係時，就被判定為水平或垂直相鄰。於是為了解決這類問題，在做完距離及角度兩階段的判斷之後，我們最後再對於每一條棋子關係連線，做一次利用角度資訊的判斷。這樣一來所有的關係又再被角度檢查一次，可以排除掉在第一階段用距離判斷時發生的錯誤。最後結果如圖 3-11 所示，棋盤上的棋子已經建立起完全正確的相對位置。在下一節，我們將介紹如何使用棋子間的相對位置，重建出棋子相對於棋盤格線的座標。

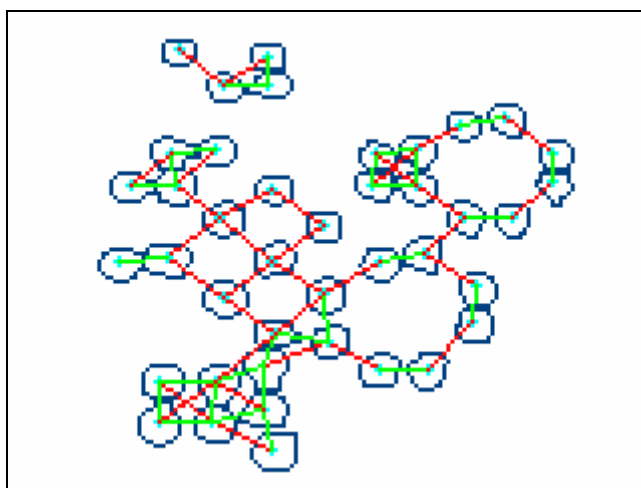


圖 3-11 建立棋子間相對位置的最後結果。

3.2.2 重建棋子座標

建立起棋子間相對位置後，我們可以把棋子及其關係連線視為一個無向圖形 (Undirected Graph)，棋子的質心為頂點 (Vertex)，關係連線為邊 (Edge)，但是由於棋盤上所有棋子不一定只會形成一個連通單元 (Connected Component)，擺放較遠的棋子可能會形成“孤島”，如圖 3-11 左上方的四顆棋子，於是我們分為兩階段重建棋子座標。第一階段，由第一手棋子開始進行圖形追蹤 (Graph Traversal)，建立起所有和第一手棋子連通之棋子的座標；第二階段，針對第一階段無法建立座標的棋子，利用投影轉換矩陣 (Homography) 建立其座標。以

上建立起的是棋子相對於棋盤格線的相對座標，最後我們還要將坐標平移，得到相對於棋盤格線的絕對座標。以下將介紹座標重建各步驟的詳細做法。

3.2.2.1 利用圖形追蹤

圖形追蹤有兩種方法：縱向優先搜尋（Depth First Search, DFS）以及橫向優先搜尋（Breadth First Search, BFS），對於本系統來說使用哪種方式都沒有差別，而我們採用橫向優先搜尋。我們將逐一檢視每一棋子，一邊建立起棋子相對於棋盤格線的相對座標。

首先從第一手棋子 S 開始，並且訂 S 的座標為 $(0,0)$ ，接著檢視與 S 相鄰的任一棋子 S' ，並根據兩棋子的關係及方位，訂出該棋子相對於棋盤格線的相對座標。若是水平或垂直的相鄰，則 S' 的 x 或 y 座標與 S 相差 1；若是斜角的相鄰，則 S' 的 x 及 y 座標與 S 各相差 1；至於是 +1 還是 -1，端視 S' 是在 S 的哪個方位來決定。至於所有相鄰 S 而尚未檢視過的棋子，都將在下一個步驟裡一一檢視；且相鄰至這些檢視過的棋子又尚未檢視過的頂點，又將一一被檢視；重複上述，直到所有棋子都被檢視過為止。

執行完 BFS 的追蹤，我們已經建立起所有和第一手棋子 S 連通之棋子的相對座標，至於未與 S 連通的棋子，我們就在下一階段，利用投影轉換矩陣建立其相對座標。

3.2.2.2 利用投影轉換矩陣

這邊的作法相當類似 2.4 節中所介紹的方法，即利用四組參考點，求出影像平面對應到棋盤平面的投影轉換矩陣，再用此投影轉換矩陣把棋子質心在影像平面的座標，轉換至相對於棋盤格線的相對座標。唯一不同的就是這裡不使用棋盤格線的四個端點當作參考點，而是從已經建立起相對座標的棋子中，選出四顆棋子當作參考點，來求出投影轉換矩陣。

選擇參考點的原則是四個點不要太靠近，於是我們從已經建立起相對座標的棋子中，選擇最靠近四個邊緣的四顆棋子當作參考點。實際的作法相當簡單，就是分別找 x 座標及 y 座標最大及最小的四顆棋子；若是邊緣的棋子不只一顆，則選擇後下的一顆；若是被選擇的四顆棋子中有重複的，則再選擇次靠近邊緣的一顆棋子。本例的結果如圖 3-12 所示，圖中填上橘色的即為被選擇當做參考點的棋子。

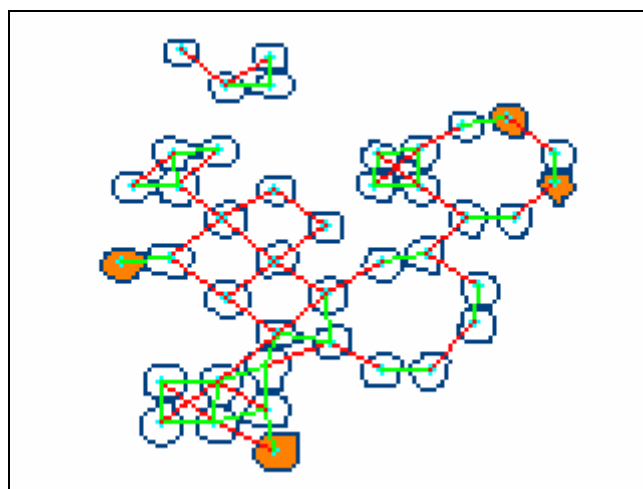


圖 3-12 建立投影轉換矩陣的參考點。

我們已知當做參考點的四顆棋子之影像平面的座標，以及相對於棋盤格線的相對座標，所以可以由 2.3.1 節中介紹的方法，建立這兩個平面的投影轉換矩陣。再來針對第一階段無法建立座標的棋子，利用該投影轉換矩陣建立其座標。至此，我們已經建立起棋盤上所有棋子之相對於棋盤格線的相對座標了，接下來就要將坐標平移，得到相對於棋盤格線的絕對座標。

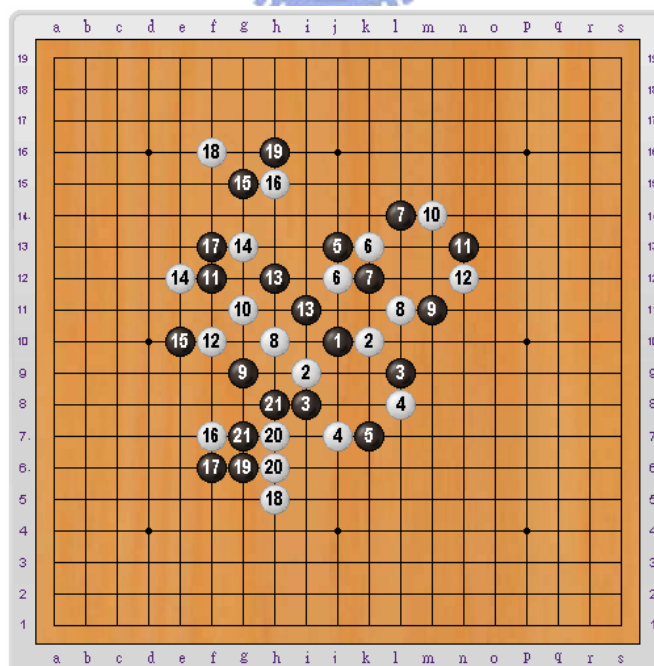
3.2.2.3 轉換相對座標為絕對座標

剛才我們所建立起來的座標，是相對於棋盤格線的相對座標，以第一手棋子 S 為原點，事實上棋盤格線的座標應該以左上角的端點為原點，所以我們要進行座標平移，把相對座標轉為相對於棋盤格線的絕對座標。進行座標平移前，有一

件事我們未知，就是 S 的位置，但六子棋有一個很重要的特性，就是第一手棋子絕大多數是下在中元，也就是棋盤正中間的位置，因此我們預設 S 的座標為 $(10,10)$ 。在本例中，第一手棋子也是下在中元，座標平移後的結果如圖 3-13 所示。當然，我們也可以人工給予第一手棋子 S 的位置，這樣就能保證平移後的位置與棋盤上真實的位置一致了。介紹到此，我們已經利用棋子間相對位置，定出所有棋子相對於棋盤格線的座標了。



(a) 第二屆交通大學盃六子棋公開賽決賽之最後畫面



(b) 棋譜最後結果的呈現畫面

圖 3-13 本系統所記錄之棋譜的最後結果。

3.3 排解棋盤移動問題

「基於棋子相對位置之定位」系統偵測棋盤移動的方法跟「相對於棋盤之棋子定位」系統一模一樣，所以在此就不再重新說明一次，我們直接用 2.4.1 節所介紹的作法，偵測棋盤被移動後的關鍵影格。以下繼續介紹偵測到棋盤移動後的處理方法。

當畫面中棋盤的位置移動，所有的棋子也都跟著偏移，會造成棋盤移動前後之棋子間相對位置的錯亂，因此我們必須把記錄器中，所有棋子之移動前的質心及輪廓資訊，都轉換到移動後的位置。我們可以用平面投影轉換來達成此目的，為此則先要在棋盤移動前後的關鍵影格中，找到四組以上相對應的參考點。

假設 $frame_{i-n}$ 與 $frame_i$ 為棋盤移動前後的兩個關鍵影格，而 S_{i-n} 與 S_i 為兩關鍵影格之棋子區域，當發生棋盤移動後，先分別針對 S_{i-n} 與 S_i 執行以下步驟：

- (1) 利用棋子區域的大小作篩選，找出所有單顆棋子區域。
- (2) 相鄰的單顆棋子區域當作一個群組，把單顆棋子區域分為數群。
- (3) 逆時針掃描所有單顆棋子區域，各群組中第一個被掃描到的棋子，即為我們所選擇的棋子。若被選擇的棋子不到五顆，則從較大的群組內選擇最後被掃描到的棋子，總共要選出五顆棋子。

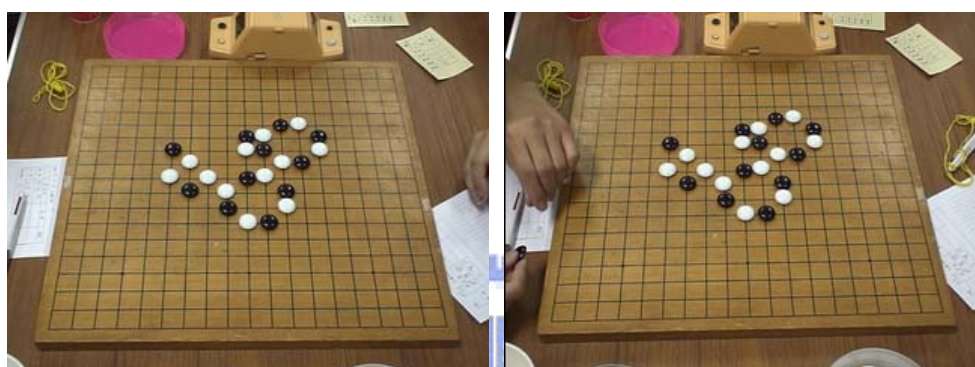
從 S_{i-n} 與 S_i 各自選出五顆棋子之後，再執行以下步驟：

- (1) 利用 Cross-Ratio 方法，找出 S_{i-n} 與 S_i 中五個棋子的對應關係。
- (2) 利用這五組相對應的參考點，計算出投影轉換矩陣，並且把記錄器中，所有棋子之移動前的質心及輪廓資訊，都轉換到移動後的位置。

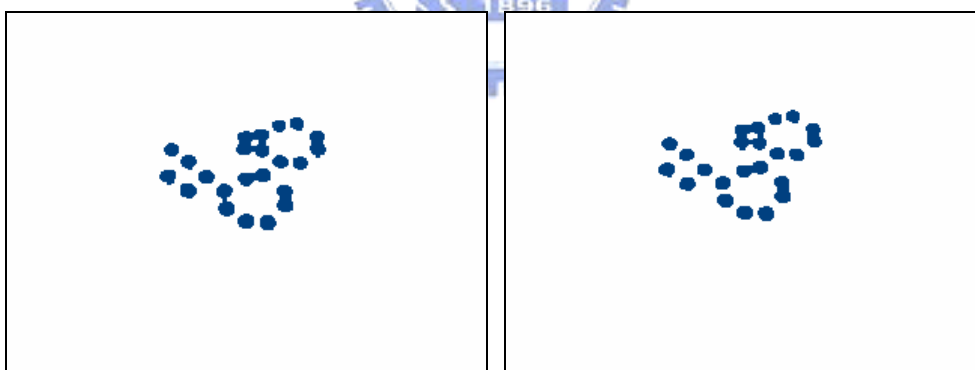
我們將在以下各小節逐一介紹各個步驟的詳細做法與原理。

3.3.1 利用群組選出參考點

圖 3-14 為一個發生棋盤移動的情況，我們將以這個例子說明棋盤移動問題的處理。首先我們要利用棋子區域的大小作篩選，找出 S_{i-n} 與 S_i 中所有單顆棋子區域，即一個棋子區域只包含一顆棋子。因為只有單顆棋子區域，我們才能夠藉由此棋子區域的質心，得知這一顆棋子的位置，若是棋子區域包含兩顆以上棋子，我們無法直接得知此區域內各棋子的位置。圖 3-15 之橘色區域，即為 S_{i-n} 與 S_i 中之單顆棋子區域，我們把它表示為兩個集合 $U(S_{i-n})$ 及 $U(S_i)$ 。



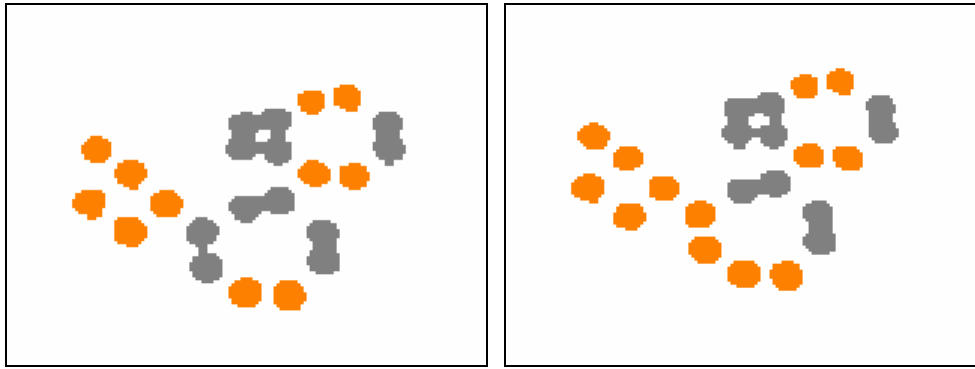
(a) $frame_{i-n}$ 之原始影像 (棋盤移動前) (b) $frame_i$ 之原始影像 (棋盤移動後)



(c) $frame_{i-n}$ 之棋子區域 S_{i-1}

(d) $frame_i$ 之棋子區域 S_i

圖 3-14 棋盤移動前後之影像及其棋子區域。



(a) S_{i-n} 之單顆棋子區域 $U(S_{i-n})$

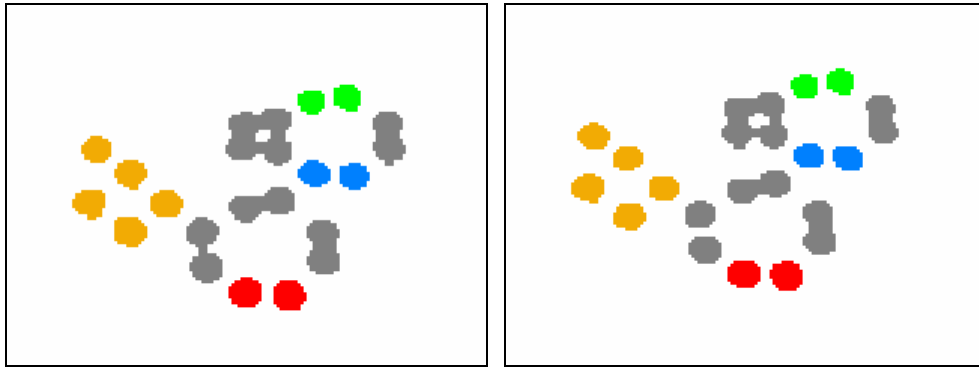
(b) S_i 之單顆棋子區域 $U(S_i)$

圖 3-15 棋子區域中的單顆棋子區域。

找到 $U(S_{i-n})$ 與 $U(S_i)$ 之後，我們要進一步的縮小範圍，找出五顆當作對應點的棋子。由於攝影機或棋盤移動後，棋盤平面可能會發生任何距離或角度的移動或旋轉，所以我們不能再用棋子在畫面中位置，來判定是否為相同棋子。但是棋子之間的相對位置是不會變的，所以我們把相鄰的單顆棋子區域當作一個子群，於是 $U(S_{i-n})$ 與 $U(S_i)$ 各會被分為數個子群，而各群組內的棋子數目即為該群組的特徵。

但是在此例子中， $U(S_{i-n})$ 與 $U(S_i)$ 中單顆棋子的數目根本不一樣，因為棋盤移動前 S_{i-n} 有兩顆棋子是連在一起的，而在棋盤移動後 S_i 這兩顆棋子分開了，因此在分群之前，我們還要有一個確認 $U(S_{i-n})$ 與 $U(S_i)$ 中棋子為相同的機制。首先比對 $U(S_{i-n})$ 與 $U(S_i)$ 中的棋子數目，本例中 $|U(S_{i-n})| = 11$ 而 $|U(S_i)| = 13$ ，所以我們必須刪減掉 $S(S_i)$ 中的兩顆棋子，我們無法知道要刪減哪兩顆棋子，唯一的線索就是這兩顆棋子一定是相鄰的，所以我們任意刪減 $U(S_i)$ 中相鄰的兩顆棋子，再把 $S(S_{i-n})$ 與 $S(S_i)$ 中的棋子分群組。分群組的結果，若是 $U(S_{i-n})$ 與 $U(S_i)$ 包含不同數目的群組，或是 $U(S_{i-n})$ 與 $U(S_i)$ 的各群組包含的棋子數目不一樣，則我們知道剛才所刪減的棋子是錯誤的，於是重新刪減 $U(S_i)$ 中相鄰的另外兩顆棋子，直到 $U(S_{i-n})$ 與 $U(S_i)$ 包含相同數目的群組，且 $U(S_{i-n})$ 與 $U(S_i)$ 的各群組包含的棋子數目一樣為止。圖 3-16 為 $U(S_{i-n})$ 與 $U(S_i)$ 刪減棋子及分群後的結果，不同群組用

不同顏色標記。

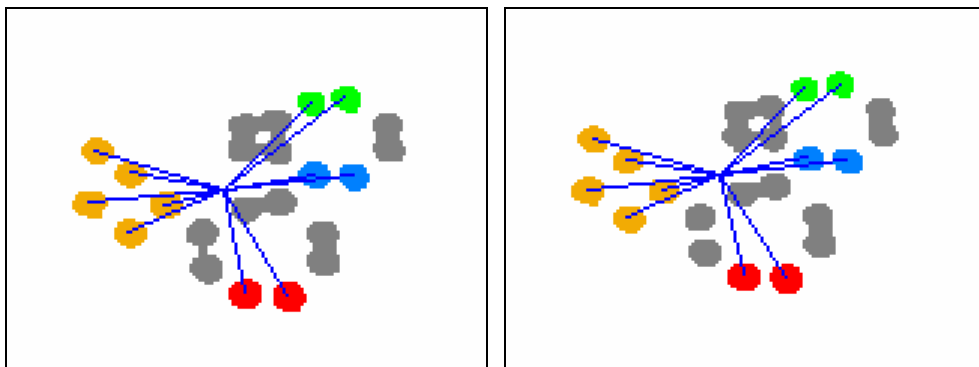


(a) $U(S_{i-n})$ 內棋子分群的結果

(b) $U(S_i)$ 內棋子分群的結果

圖 3-16 單顆棋子區域分群的結果。

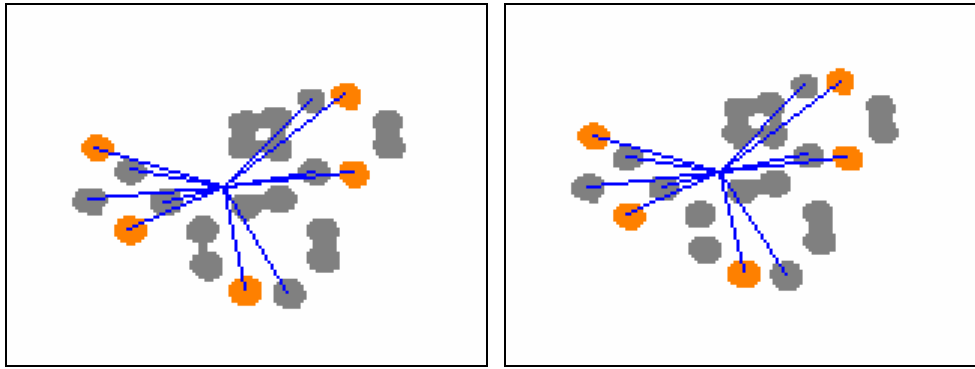
把 $U(S_{i-n})$ 與 $U(S_i)$ 分成群組之後，我要進一步在 $U(S_{i-n})$ 與 $U(S_i)$ 的各群組之中挑選出相同的棋子。由於棋盤平面的移動，棋子可能被位移任何地方，棋子之間的方位順序已經不具參考性，唯有棋子之間的角度順序是不會變的。因此我們要分別逆時針掃描 $U(S_{i-n})$ 與 $U(S_i)$ 內所有棋子，各群組中第一個被掃描到的棋子，即為我們所選擇的棋子。逆時針掃描的圓心必須在所有棋子的中間位置，我們使用 $U(S_{i-n})$ 與 $U(S_i)$ 中所有棋子質心的中心點，當作時針掃描的圓心，掃描的結果如圖 3-17 所示。若被選擇的棋子不到五顆，則從較大的群組內選擇最後被掃描到的棋子，總共要選出五顆棋子，最後被選擇的棋子如圖 3-18 所示。



(a) $U(S_{i-n})$ 內棋子掃描的結果

(b) $U(S_i)$ 內棋子掃描的結果

圖 3-17 單顆棋子區域逆時針掃描的結果。



(a) $U(S_{i-n})$ 中被選擇的棋子

(b) $U(S_i)$ 中被選擇的棋子

圖 3-18 最後被選擇當做參考點的棋子（橘色所標示者）。

3.3.2 利用 Cross-Ratio 找出參考點之對應關係

在上一小節中，我們已經分別從 $U(S_{i-n})$ 與 $U(S_i)$ 中找出五顆棋子當作參考點，並且可以確認這五顆棋子為相同的五顆棋子，但我們還是不知道它們之間的對應關係。在本小節中，我們將利用 Cross-Ratio 的投影不變量（Projective Invariant）特性，找出此五個參考點的對應關係。以下，我們先對 Cross-Ratio 的理論做簡單的介紹。

3.3.2.1 Cross-Ratio 的定義

Cross-Ratio 的基本定義為：對於共線的四點 A, B, C, D ，所形成四線段的距離的比值。如式子(3.1)所描述的比例關係：

$$CR(A, B, C, D) = \frac{\overline{AC} / \overline{BC}}{\overline{BD} / \overline{AD}} = \frac{\overline{ACBD}}{\overline{BCAD}} \quad (3.1)$$

由於 A, B, C, D 四點的編排順序不同，所得的 Cross-Ratio 值也不同。共線四點可以組成 24 種排列，會有 24 個 Cross-Ratio 值，並可將其分為 6 類。其實各種形式的 Cross-Ratio 值均是相依的，在本篇論文中，我們使用(3.1)式的 Cross-Ratio 形式來做計算。

接下來我們欲定義二維空間中之 Cross-Ratio。假設在一個二維平面上有不共線的五點 O, A, B, C, D （如圖 3-19），如欲定義此五點的 Cross-Ratio 值，

我們可以在原點 O 與其他參考點 A, B, C, D 之間畫一直線 l_0 ，再將 A, B, C, D 投影到 l_0 上得到投影點 A', B', C', D' ，因此平面上五點 O, A, B, C, D 的 Cross-Ratio 值，可以用它們在直線上的投影點 A', B', C', D' 來定義。又由於點與線是對偶關係，以 O 點為原點到 A, B, C, D 的連線向量為 $\vec{A}, \vec{B}, \vec{C}, \vec{D}$ ，此四向量之間所形成的角度關係也會有相同的 Cross-Ratio 值，如式子(3.2)。

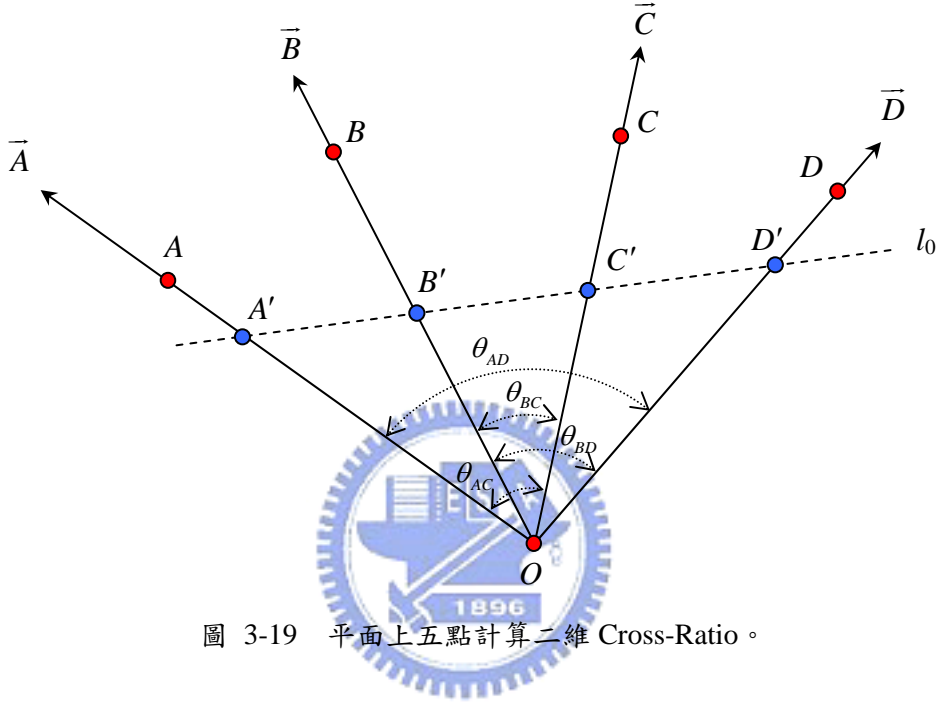


圖 3-19 平面上五點計算二維 Cross-Ratio。

$$CR_O(A, B, C, D) = \frac{\overline{A'C'B'D'}}{\overline{B'C'A'D'}} = \frac{\sin \theta_{AC} \sin \theta_{BD}}{\sin \theta_{BC} \sin \theta_{AD}} \quad (3.2)$$

我們將式子(3.2)以外積形式改寫成(3.3)式，

$$\begin{aligned} CR_O(A, B, C, D) &= \frac{\sin \theta_{AC} \sin \theta_{BD}}{\sin \theta_{BC} \sin \theta_{AD}} = \frac{|\vec{A}| |\vec{C}| \sin \theta_{AC} |\vec{B}| |\vec{D}| \sin \theta_{AD}}{|\vec{B}| |\vec{C}| \sin \theta_{BC} |\vec{A}| |\vec{D}| \sin \theta_{AD}} \\ &= \frac{|\vec{A} \times \vec{C}| |\vec{B} \times \vec{D}|}{|\vec{B} \times \vec{C}| |\vec{A} \times \vec{D}|} = \frac{\begin{vmatrix} A_x & C_x & B_x & D_x \\ A_y & C_y & B_y & D_y \end{vmatrix}}{\begin{vmatrix} B_x & C_x & A_x & D_x \\ B_y & C_y & A_y & D_y \end{vmatrix}} \quad (3.3) \end{aligned}$$

其中， $\vec{A} = (A_x, A_y)$ 、 $\vec{B} = (B_x, B_y)$ 、 $\vec{C} = (C_x, C_y)$ 、 $\vec{D} = (D_x, D_y)$ 。如此一來，

我們可以不必知道向量間的角度，也不牽涉到正弦函數的運算，只要知道 O, A, B, C, D 的座標位置，就可以得到向量 $\vec{A}, \vec{B}, \vec{C}, \vec{D}$ ，快速求得空間中五點 O, A, B, C, D 的 Cross-Ratio 值。

3.3.2.2 Cross-Ratio 投影不變量

Cross-Ratio 對於透視投影具有不變量的特性。對於三維空間物體的投影來說，Cross-Ratio 並不會因為物體平移、旋轉或大小等投影條件的改變而變化。也就是說，投影變換（Projective Transformation）後 Cross-Ratio 依然保持不變。

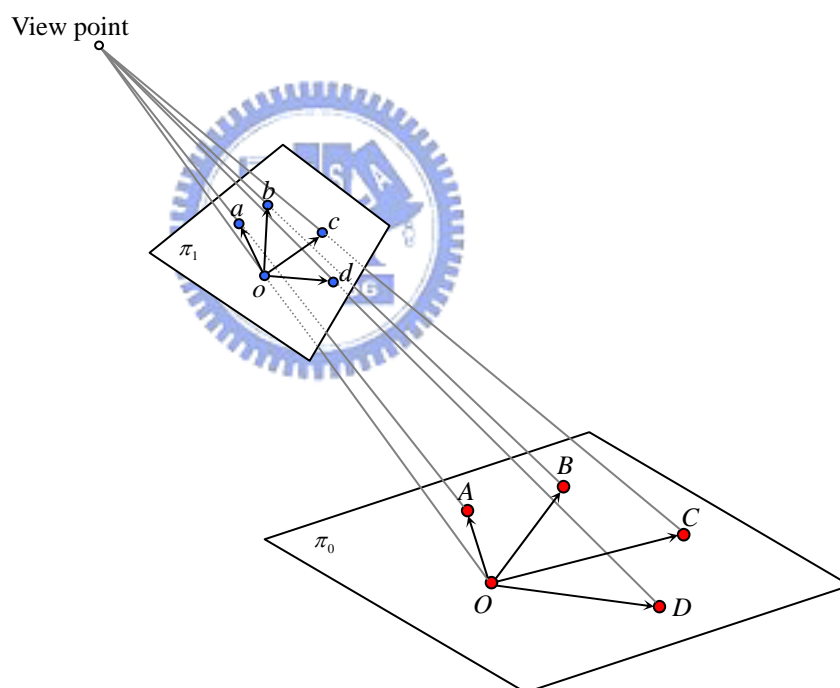


圖 3-20 透視投影模型。

圖 3-20 描述一個的透視投影模型。場景空間中有一平面 π_0 ，五個共平面的參考點 O, A, B, C, D 在 π_0 上，其中任意三點皆不共線。 O, A, B, C, D 投影在 π_1 上的投影點為 o, a, b, c, d 。由於 Cross-Ratio 於透視投影中是一個不變量，因此場景參考點所組成的 Cross-Ratio 值 $CR_O(A, B, C, D)$ 與其對應投影點所組成的

Cross-Ratio 值 $CR_o(a, b, c, d)$ 相等，亦即

$$CR_o(A, B, C, D) = \frac{|\vec{A} \times \vec{C}| |\vec{B} \times \vec{D}|}{|\vec{B} \times \vec{C}| |\vec{A} \times \vec{D}|} = \frac{|\vec{a} \times \vec{c}| |\vec{b} \times \vec{d}|}{|\vec{b} \times \vec{c}| |\vec{a} \times \vec{d}|} = CR_o(a, b, c, d) \quad (3.4)$$

其中 \vec{A}, \dots, \vec{D} 是平面 π_0 上 O 點到 A, \dots, D 的向量， \vec{a}, \dots, \vec{d} 是平面 π_1 上 o 點到 a, \dots, d 的向量。

3.3.2.3 找出參考點之對應關係

由以上介紹的 Cross-Ratio 數學運算，以及透視投影模型中 Cross-Ratio 的數值不變性質，我們知道若是棋盤移動前後的影像中，五個參考點的排序相同，則會得到相同的 Cross-Ratio，我們就利用此性質來找出參考點的對應關係。首先，我們在棋盤移動前的平面的參考點中，先任選出一個參考點當作的原點 O ，再依照掃描的順序排出 A, \dots, D ，算出此排序的 Cross-Ratio，而在棋盤移動後的平面中，我們分別把五個點當作原點 o ，再依照掃描的順序排出 a, \dots, d ，分別算出五個排序的 Cross-Ratio。若兩個平面中五個點的 Cross-Ratio 值一樣，則代表他們的原點 O 及 o 為同樣一顆棋子，而其餘的棋子則根據掃描順序依序對應，這樣我們就可以得到兩個平面中，五個參考點的對應關係了。

3.3.3 轉換移動前紀錄

得到了棋盤移動前後的影像平面中，五組相對應的參考點後，我們就可以計算出兩平面的投影轉換矩陣，並且把記錄器中，所有棋子之移動前的質心及輪廓資訊，都轉換到移動後的位置。如此一來，棋盤移動所造成的問題就解決了。圖 3-21 中的紅色輪廓，即為所有移動前棋子輪廓轉換至移動後的結果，而灰色區域為棋盤移動後的真實棋子區域，我們可以發現，紅色輪廓及灰色區域幾乎吻合。

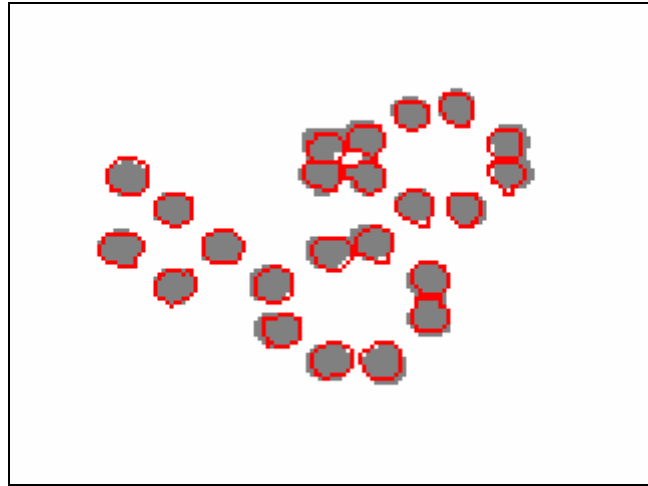


圖 3-21 所有移動前棋子輪廓轉換至移動後的結果。

3.4 總結

在本章中，我們介紹了「基於棋子相對位置之定位」的自動棋譜產生系統，包含了如何建立棋子間的相對位置，到建立起相對於棋盤格線的座標，以及棋盤移動的處理。我們拍攝了數段影片當作系統輸入，發現此方法更能夠容許棋子放置位置的偏斜，系統判斷每一手新置放棋子座標的正確率非常高。另外，此方法最大的好處是完全不需要棋盤的資訊，所以可以免除偵測棋盤格線的麻煩及誤差，是一個創新且更具彈性的棋子座標定位方法。

第四章 實驗結果與討論

4.1 棋譜格式與呈現

我們的系統記錄了每一手棋子的座標位置、顏色以及思考時間，並且根據SGF(Smart Game Format)標準輸出為txt檔。SGF是一種記錄雙人棋盤遊戲的通用格式，為樹狀結構文字檔案，檔案小方便流通，通常被用來記錄圍棋、六子棋等棋譜。詳細的檔案格式可參考[7]。

本系統所產生之SGF棋譜檔案，交由交通大學資訊工程系吳毅成教授所發展的「六子棋棋譜呈現程式」呈現。吳教授所發展的「六子棋棋譜呈現程式」為一個Web base的程式，可以讀取SGF棋譜檔案，並把棋譜以圖形方式呈現，且具有逐步播放棋譜、即時標示棋形（如迫著、活三、死三、活二）等功能，提供棋友一個非常方便且實用的使用者介面。此棋譜呈現介面如圖 4-1 所示，關於此介面的詳細使用方式可參考[8]。

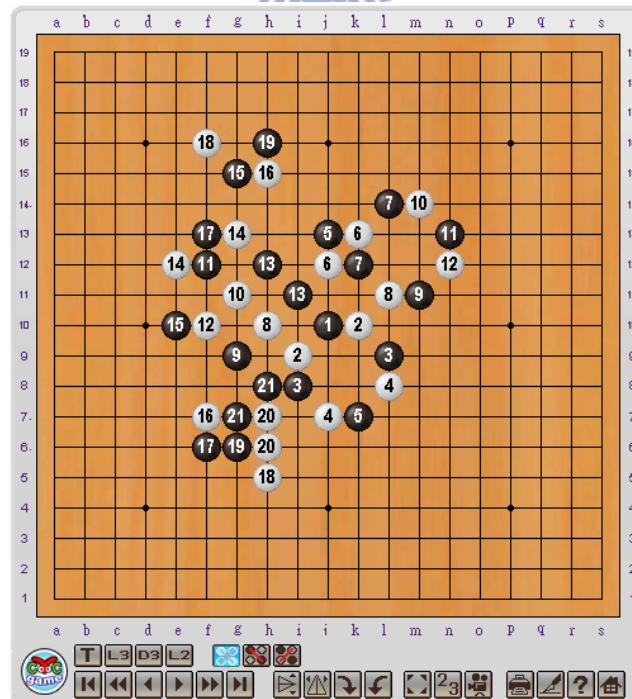


圖 4-1 六子棋棋譜呈現介面。

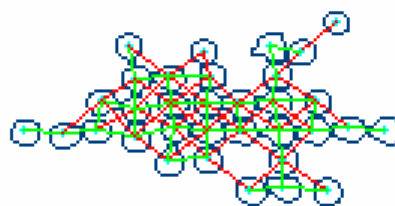
4.2 棋譜分析結果與討論

除了在論文中作為說明範例的「2007 交大六子棋公開賽」決賽的影片之外，我們也以不同的角度拍攝更多六子棋賽局，當作系統的輸入影片，其分析結果將在以下一一呈現及討論。每一個例子以三個畫面呈現，分別是原始影片的最後畫面、所有棋子之關係連線結果，以及系統產生棋譜之圖形呈現。

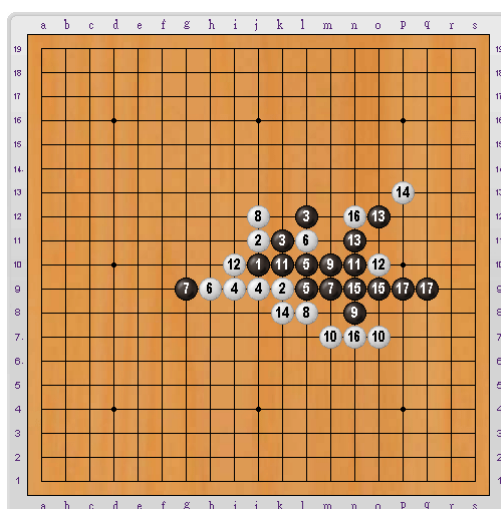
首先我們把棋盤擺正，只調整攝影機的高度。若以較大的俯角拍攝棋盤，結果如圖 4-2，分析的結果完全正確，可見攝影機在如此的拍攝角度下，系統都可以正確的分析出棋譜。接著我們把攝影的俯角降低，結果如圖 4-3，雖然棋子都有被正確偵測，但由於棋子變形太多，棋子間距的參考度降低，導致部分的關係連線錯誤，本實驗約有 30% 的棋子無法被正確的定位。



(a) 原始影片之最後畫面



(b) 棋子關係連線之結果

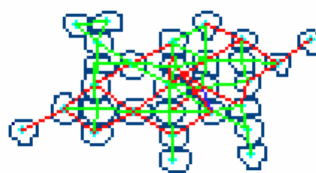


(c) 所產生棋譜之圖形呈現

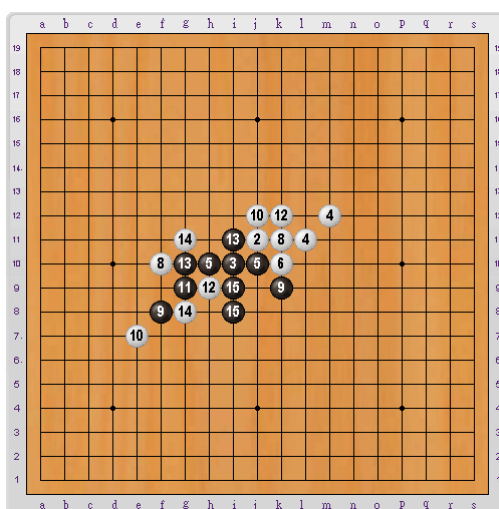
圖 4-2 較大俯角拍攝棋盤之實驗結果。



(a) 原始影片之最後畫面



(b) 棋子關係連線之結果



(c) 所產生棋譜之圖形呈現

圖 4-3 較小俯角拍攝棋盤之實驗結果。

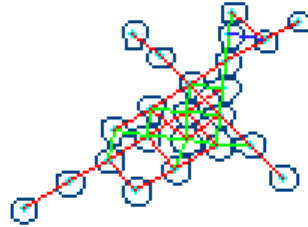
由以上實驗看來，為了避免棋子變形太多，本系統不允許太小的拍攝俯角，於是我們改變攝影機拍攝棋盤的方位角度，看看是否能正確的分析棋譜。圖 4-4 為攝影機偏斜小方位角之結果，系統可以正確的分析棋譜，接著我們再加大偏斜的角度，如圖 4-5 所示，發現系統仍能正確分析棋譜，另外由綠色的連線，我們還可以得知棋盤所擺放的角度。

由實驗結果可以發現，只要攝影機架設在一定的俯角之上，本系統對於拍攝方位角度有相當大的彈性，也有非常高的正確率。這個特性也是本系統與相關研究最大的不同之處，攝影機不需要有固定的架設位置，大幅的提升了本系統在使用上的方便性。並且由於本系統不需要任何棋盤的資訊，只處理特徵較明顯的黑

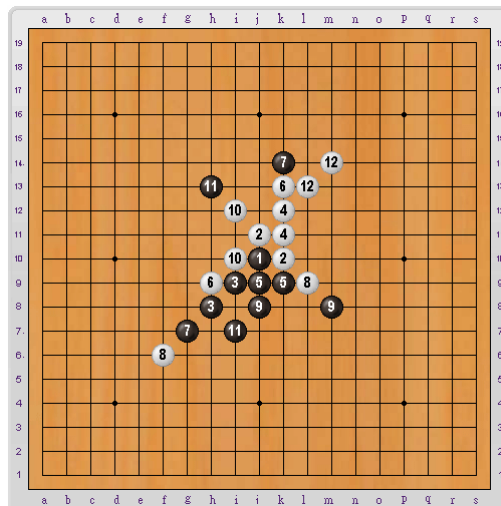
白棋子，因此本系統在棋盤格線拍攝模糊的較差條件下，也可以正確的定位棋子。



(a) 原始影片之最後畫面



(b) 棋子關係連線之結果

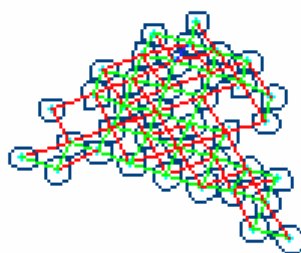


(c) 所產生棋譜之圖形呈現

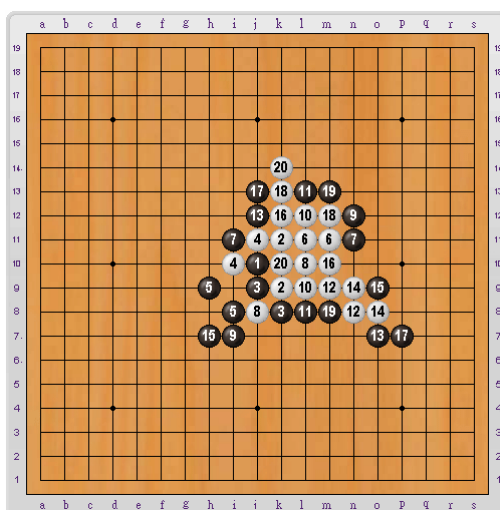
圖 4-4 偏斜小方位角拍攝棋盤之實驗結果。



(a) 原始影片之最後畫面



(b) 棋子關係連線之結果



(c) 所產生棋譜之圖形呈現

圖 4-5 偏斜大方位角拍攝棋盤之實驗結果。

第五章 結論與未來展望

在本篇論文中，我們介紹了一個使用一般攝影機影像，且拍攝角度較為自由的自動棋譜產生系統，並且提出了兩種棋子定位方式，一種是「相對於棋盤之棋子定位」，另一種是「基於棋子相對位置之定位」。本系統的分析結果會輸出為SGF 棋譜檔案，該檔案格式可以透過吳教授所發展的「六子棋棋譜呈現程式」呈現。

「相對於棋盤之棋子定位」方法，以棋盤格線的四個角落當作參考點，利用投影轉換矩陣（Homography）把棋子的影像座標轉換至棋盤的座標，簡單且判斷的正確率高，唯一的缺點在於需要在棋盤格線的四個角落貼上標示點。而「基於棋子相對位置之定位」方法，先建立棋盤上所有棋子的相對關係，再藉此資訊重建棋盤的座標系，免除了在棋盤貼標記的麻煩，方法創新且同樣可以正確建立棋子之相對座標。

此自動棋譜產生系統不只可以記錄六子棋的棋譜，圍棋或是任何使用黑白棋子的棋盤遊戲皆適用。期望未來可將本系統推廣至各大小圍棋及六子棋競賽，以智慧型的方式記錄比賽棋譜，可節省大量人力；也希望推廣至圍棋教室，即時記錄各棋局的棋譜，讓檢討及學習更有效率。

參考文獻

- [1] K. Yanai and T. Hayashiyama, "Automatic "Go" record generation from a TV program," *International Multi-Media Modelling Conference*, pp. 4, 2006.
- [2] D. C. Kang, H. J. Kim, and K. H. Jung, "Automatic extraction of game record from TV Baduk program," *International Conference on Advanced Communication Technology*, vol. 2, pp. 1185 – 1188, 2005.
- [3] K. Shiba and K. Mori, "Detection of Go-board contour in real image using genetic algorithm," *SICE Annual Conference*, vol. 3, pp. 2754 – 2759, 2004.
- [4] D. Kakiuchi, T. Fukuyama, A. Domaie, K. Koromo ,and K. Okazaki, "Automatic recording of Shogi-record by image processing," *IAPR Workshop on Machine Vision Applications*, pp.52 – 55, 2002.
- [5] 劉振益, "圍棋影像計時、記譜系統及方法," 中華民國專利 No. 00415849, 2000.
- [6] 曹成宰, "自動計時棋盤及其計時裝置," 中華民國專利 No. 00305032, 1997.
- [7] 《SGF User Guide》主頁, http://www.red-bean.com/sgf/user_guide/index.html
- [8] 《六子棋網路呈現介面》主頁, <http://www.connect6.org/board/produce.php>

附錄 棋子間距臨界值的決定

為了避免棋子的陰影干擾棋子範圍的偵測，本系統所偵測到的棋子區域會略小於棋子的實際大小，即偵測到棋子的間距會略大於實際間距，因此我們所訂下的棋子間距臨界值略大於理論上的期望值。為了找出最適合的臨界值，我們實際統計出 3.2 節例子中所有相鄰棋子的間距。圖 1 為所有相鄰棋子之間距的直方圖，藍色部分表示水平或垂直相鄰棋子，紅色部分表示斜角相鄰棋子，我們可以發現斜角相鄰棋子之間距幾乎都會在 0.3 之上，而水平或垂直相鄰棋子之間距都在 0.45 之下，而距離在這兩個數值間的棋子則難以判定關係，此外距離超過 0.8 則棋子並不相鄰，因此我們的間距臨界值定 TH'_1 為 0.3， TH'_2 為 0.45， TH'_3 為 0.8。

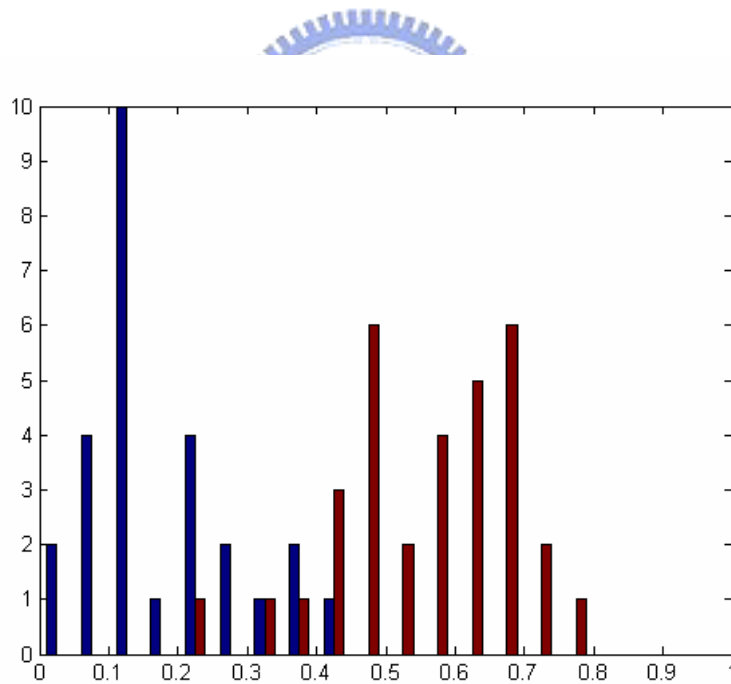


圖 1 所有相鄰棋子之間距的直方圖。