



龍華科技大學

電子工程系碩士班

碩士學位論文

時間導向斜線迷宮演算法之設計與實作
及其在電腦鼠比賽之應用

The Design and Implementation of a
Time-Based Diagonal Maze Solver and Its
Application in Micromouse Contest

研究生：蔡欣翰

指導教授：蘇景暉 博士

中華民國 106 年 7 月

龍華科技大學

碩士學位考試委員會審定書

本校 電子工程系 碩士班 蔡欣翰 君

所提論文時間導向斜線迷宮演算法之設計與實作及其在電腦鼠比賽之應用
經本委員會審定通過，特此證明。

學位考試委員會

委員：蘇景暉

謝財明

黃信雄

指導教授：蘇景暉

系主任（所長）：陳永裕



中華民國 106 年 6 月 20 日

摘要

論文名稱：時間導向斜線迷宮演算法之設計與實作及其在電腦鼠比賽之應用

頁數：107

校所別：龍華科技大學

研究所：電子工程系碩士班

畢業時間：105 學年度第 2 學期

學位：碩士

研究生：蔡欣翰

指導教授：蘇景暉 博士

關鍵詞：電腦鼠、迷宮演算法

由於時代進步快速，電腦鼠比賽的迷宮地圖越來越複雜化與多樣化，因此本論文主要發展一新式的時間導向斜線迷宮演算法，讓演算法以電腦鼠的運動性能來計算可以跑最快的最佳路徑。

本論文也實際製作半尺寸電腦鼠與具備吸地風扇之古典電腦鼠參與國際競賽，並根據各地比賽規則採取不同的比賽策略。

本論文的古典電腦鼠也因此取得好幾次國際競賽的冠軍，而半尺寸電腦鼠也在 2016 年的國際競賽中取得第四名的優異成績。

ABSTRACT

Thesis Title : The Design and Implementation of a Time-Based Diagonal Maze Solver
and Its Application in Micromouse Contest Pages : 107

University : Lunghwa University of Science and Technology

Graduate School : Department of Electronic Engineering

Date : July, 2017 Degree : Master

Graduate Student : Cai, Xin-Han Advisor : Su, Juing-Huei

Keywords : Micromouse, Maze Algorithm

The role of a maze solver in finding the best route from the start cell to the goal area in modern micromouse contests becomes more and more important for a contestant to win a contest. This is because the fast improvements in microcontrollers, infrared and microelectromechanical sensors (MEMS) have made those micromouse robots run faster and faster, and diagonally. The thesis is therefore devoted to the development of a time-base diagonal maze solver, which takes into account the motion capabilities of a micrmouse in a maze.

The time-based diagonal maze solver has also been successfully used in a half-size micromouse and a classic micromouse with a suction fan in international contests. Since there are slight differences between the rules in USA and Japan micromouse contests, different searching and fast run strategies are also devised in this thesis to maximize the chance of getting a prize.

The overall effort has made the classic micromouse designed in this thesis win several first prizes in both USA and Japan, and also made the half-size micromouse

win the fourth-place prize in Japan.



誌謝

本論文的完成，首先感謝指導教授蘇景暉老師耐心的指導，也感謝李齊雄老師、黃信雄老師與林煌凱先生的指導與幫助。

感謝研究室中的學長傳憲、師葦、凱筌、正彥、聖絃、玉誠、信茂、宗駿、凱翔、宗諭、軒聖、桓頡、兆惟、育志、逢峻、易展等學長在各項問題上的指導與協助；感謝同學文齊、昭毅、瑞鴻在各項問題上的協助；感謝系助理方慈、明玉在各項行政上的協助。

感謝這五年來在嵌入式控制系統研究室中遇到的一切，總覺得好像昨天才剛加入研究室，現在卻要畢業了。

最後感謝我的父母對我從小到大的栽培，且提供給我從小到大生活及就學中的各種開銷，讓我在學習上無後顧之憂。

目錄

摘要	i
ABSTRACT.....	ii
誌謝	iv
目錄	v
表目錄	vii
圖目錄	viii
第一章 緒論	1
1.1 研究動機與目的	1
1.2 論文之貢獻	2
第二章 時間導向斜線迷宮演算法	3
2.1 迷宮演算法介紹	3
2.2 基礎洪水演算法[3].....	3
2.3 尋找最短路徑	4
2.3.1 直走優先	5
2.3.2 轉彎優先	6
2.4 路徑選擇與探討	6
2.5 時間導向斜線迷宮演算法設計	10
2.5.1 加速洪水值設計	11
2.5.2 減速洪水補償值設計	13
2.6 資訊定義	14
2.6.1 迷宮牆壁定義	14
2.6.2 邊緣填洪水值設計	16
2.6.3 水流方向	16
2.6.4 重複方向次數	20
2.7 變數宣告	20
2.8 填洪水	23
第三章 迷宮地圖搜尋與最佳路徑計算	30
3.1 演算法花費時間	30
3.2 未知迷宮搜尋	32
3.3 已知地圖尋找最佳路徑	36
3.3.1 尋找絕對方向最佳路徑	37
3.3.2 化簡一般動作相對路徑	38
3.3.3 化簡斜線動作相對路徑	42
3.3.4 斜線動作相對路徑 2 次處理	46
3.4 洪水錯誤覆蓋問題與解決方案	51

第四章 系統架構與機構設計	58
4.1 半尺寸電腦鼠系統架構圖	58
4.1.1 微控制器	59
4.1.2 紅外線發射與接收	60
4.1.3 馬達驅動電路	65
4.1.4 編碼器	66
4.1.5 陀螺儀	67
4.2 半尺寸電腦鼠機構設計	69
4.3 古典電腦鼠風扇設計	71
第五章 比賽策略應用與競賽成果	74
5.1 電腦鼠比賽類型	74
5.2 採計迷宮時間與衝刺成績之比賽策略	74
5.2.1 單次搜尋設定	75
5.2.2 搜尋中直線路徑二次加速	76
5.2.3 搜尋中已知路徑轉換斜線動作	77
5.3 採計單次最佳成績之比賽策略	82
5.3.1 多次搜尋設定	82
5.3.2 多次搜尋結束方式	86
5.4 比賽實際應用實例	87
5.5 得獎紀錄	89
第六章 結論與未來展望	91
6.1 結論	91
6.2 未來展望	91
參考文獻	93
附錄	95
A 古典電腦鼠之競賽規則	96
B 半尺寸電腦鼠之競賽規則	103

表目錄

表 2.1 直線加速洪水值	11
表 2.2 斜線加速洪水值.....	12
表 2.3 直線減速洪水值	13
表 2.4 斜線減速洪水值	14
表 2.5 迷宮牆壁資訊定義表	15
表 3.1 各項動作編號.....	36
表 3.2 動作回推表 1.....	39
表 3.3 動作回推 2.....	39
表 3.4 直轉直與直轉斜動作關係表.....	44
表 3.5 斜轉斜與斜轉直動作關係表.....	44
表 3.6 起始動作處理表.....	46
表 3.7 半格動作處理條件表	46
表 3.8 結束動作處理表.....	47
表 4.1 吸力與功耗測試表	72
表 4.2 運動能力比較	73
表 5.1 資訊位置與對應位元表	78
表 5.2 2016 年 APEC 國際電腦鼠比賽前三名成績	88
表 5.3 2016 全日本半尺寸電腦鼠比賽 Mini Diu-Gow 比賽中各次成績	88

圖目錄

圖 1.1 本論文自行製作之半尺寸電腦鼠(左)與古典電腦鼠(右).....	2
圖 2.1 洪水演算法做洪水擴散與填洪水值圖	4
圖 2.2 路口相同洪水值.....	4
圖 2.3 以起點開口方向定義絕對方向	5
圖 2.4 直線優先挑選的路徑	5
圖 2.5 轉彎優先挑選的路徑	6
圖 2.6 未尋找到的迷宮路徑	6
圖 2.7 斜線行走路徑	7
圖 2.8 2011 年全日本古典電腦鼠比賽決賽路徑比較圖	8
圖 2.9 2014 年全日本半尺寸電腦鼠比賽決賽路徑比較	9
圖 2.10 斜線距離示意圖	12
圖 2.11 迷宮編號定義	15
圖 2.12 迷宮牆壁記錄資訊	15
圖 2.13 邊緣洪水位置及對應位置編號	16
圖 2.14 時間導向斜線迷宮演算法迷宮方向定義	16
圖 2.15 第 1 種水流擴散方式	17
圖 2.16 第 2 種水流擴散方式	18
圖 2.17 第 3 種水流擴散方式	19
圖 2.18 根據水流方向記錄重複方向次數.....	20
圖 2.19 變數宣告格式	21
圖 2.20 半尺寸電腦鼠中演算法變數宣告	22
圖 2.21 古典電腦鼠中演算法變數宣告	22
圖 2.22 直線時對應的洪水值	23

圖 2.23 斜線時對應的洪水值	23
圖 2.24 地圖各項資訊表示示意圖	24
圖 2.25 填洪水值步驟(a)~(f).....	25
圖 2.26 演算法執行步驟(g)~(l)	26
圖 2.27 演算法執行步驟(m)~(n)	27
圖 2.28 演算法執行步驟(o)~(r)	28
圖 2.29 演算法執行步驟(s)~(u)	29
圖 3.1 搜尋時最嚴苛的演算法運算距離示意圖	30
圖 3.2 演算法花費時間實測結果	31
圖 3.3 搜尋前各項資料初始化設定	32
圖 3.4 搜尋步驟(a)~(d)	33
圖 3.5 搜尋步驟(e)~(g)	34
圖 3.6 搜尋進入死巷示意圖	35
圖 3.7 使用預設全關式牆壁資料填洪水.....	37
圖 3.8 正確與錯誤找路徑方向示意圖	38
圖 3.9 一般動作示意圖	38
圖 3.10 絶對方向最佳路徑化簡一般動作相對路徑	40
圖 3.11 絶對方向最佳路徑化簡一般動作相對路徑特殊狀況.....	40
圖 3.12 絶對方向最佳路徑化簡一般相對動作初始示意圖	40
圖 3.13 絶對方向最佳路徑化簡一般動作相對路徑步驟	41
圖 3.14 化簡完成之一般動作相對路徑	41
圖 3.15 搜尋時絕對方向路徑化簡一般動作示意圖	42
圖 3.16 斜線動作與編號.....	43
圖 3.17 一般動作相對路徑化簡斜線動作相對路徑步驟	45
圖 3.18 化簡完成之斜線動作相對路徑	45

圖 3.19 斜線動作相對路徑 2 次處理步驟.....	48
圖 3.20 各次化簡結果比較圖	49
圖 3.21 斜線動作相對路徑 2 次處理步驟.....	50
圖 3.22 洪水錯誤覆蓋圖步驟 1	51
圖 3.23 洪水錯誤覆蓋圖步驟 2	52
圖 3.24 洪水錯誤覆蓋圖步驟 3	53
圖 3.25 洪水錯誤覆蓋圖步驟 4	54
圖 3.26 洪水錯誤覆蓋圖步驟 5	55
圖 3.27 洪水錯誤覆蓋之迷宮地圖與尋找絕對方向最佳路徑結果	56
圖 3.28 由重複方向次數決定洪水擴散順序示意圖	56
圖 3.29 順序 2 處狀況.....	57
圖 3.30 順序 3 處狀況.....	57
圖 4.1 半尺寸電腦鼠系統架構圖	58
圖 4.2 RX62T 系列晶片功能區塊圖	59
圖 4.3 RX62T 消耗電流.....	60
圖 4.4 OP265AD 最大額定值	61
圖 4.5 OP265AD 電子特性	61
圖 4.6 OP265AD 發射角度圖	62
圖 4.7 LTR-4206 最大額定值	62
圖 4.8 LTR-4206 電子特性	63
圖 4.9 LTR-4206 發射角度圖	63
圖 4.10 紅外線發射與接收電路圖	64
圖 4.11 2SC3325 規格特性曲線圖	65
圖 4.12 DRV8836 簡易原理圖	65
圖 4.13 控制輸出波形圖.....	66

圖 4.14 MTL MES-6 系列編碼器	66
圖 4.15 編碼器電路圖	67
圖 4.16 LY3200ALH 陀螺儀機械特性	68
圖 4.17 LY3200ALH 陀螺儀偵測角速度方向圖	68
圖 4.18 馬達架上齒輪與齒輪間距離示意圖	70
圖 4.19 半尺寸電腦鼠爆炸圖	70
圖 4.20 RE8 馬達規格表	71
圖 4.21 Solidworks 模擬馬達支撐架 3D 圖	71
圖 4.22 本論文設計之古典電腦鼠 Diu-Gow 4 成品圖	72
圖 5.1 2016 APEC Micromouse Contest Diu-Gow 4 單次搜尋路徑圖	75
圖 5.2 搜尋中執行二次加速流程圖	77
圖 5.3 搜尋中確認已知路徑示意圖	79
圖 5.4 搜尋中已知路徑轉換斜線動作	80
圖 5.5 搜尋中已知路徑轉換斜線動作二次處理	80
圖 5.6 已知路徑轉換斜線示意圖	81
圖 5.7 第 1 次搜尋路徑與最佳路徑計算路徑圖	83
圖 5.8 第 2 次搜尋路徑與最佳路徑計算路徑圖	84
圖 5.9 第 3 次搜尋路徑與最佳路徑計算路徑圖	85
圖 5.10 第 1 次搜尋示意圖	86
圖 5.11 第 2 次搜尋時結束圖	87
圖 5.12 得獎紀錄	90

第一章 緒論

1.1 研究動機與目的

本論文主要發展一個「時間導向斜線迷宮演算法、半尺寸電腦鼠與具備吸地風扇之古典電腦鼠」並藉以攻略國際間的兩種比賽「古典電腦鼠(Classic Micromouse)」與「半尺寸電腦鼠(Half Size Micromouse)」，並參加其在國際間主要的三大賽事，包括在臺灣已舉辦 12 屆的教育部人工智慧單晶片電腦鼠暨機器人國內及國際邀請賽[1]、在日本已舉辦 37 屆的全日本電腦鼠競賽(All Japan Micromouse Contest)[2]與在美國已舉辦 31 屆的 Applied Power Electronics Conference 電腦鼠競賽(APEC Micromouse Contest)。

古典電腦鼠比賽場地由 16×16 格的正方形迷宮組成，每一格邊長皆為 18 公分；半尺寸電腦鼠由古典電腦鼠演進而來，其場地由 32×32 格的正方形迷宮組成，每一格邊長皆為 9 公分。兩項比賽中選手皆要自行設計一臺可自行從迷宮起點搜尋到迷宮終點的機器人，並考驗其解迷宮的效率與運動能力，詳細比賽規則可參考附錄 A、附錄 B。

近年來由於科技進步快速，電腦鼠比賽甚至衍伸出半尺寸電腦鼠比賽項目，當迷宮地圖變的越來越大，越來越複雜時，單解出迷宮最短路徑已經不是首要考量，如何找出最適合電腦鼠運動性能的最佳路徑，才是比賽中的首要考量，因此本論文希望設計出一款新式演算法，讓電腦鼠在迷宮中可找到最好的路徑，並實際製作應用比賽演算法之電腦鼠參加競賽，如圖 1.1。

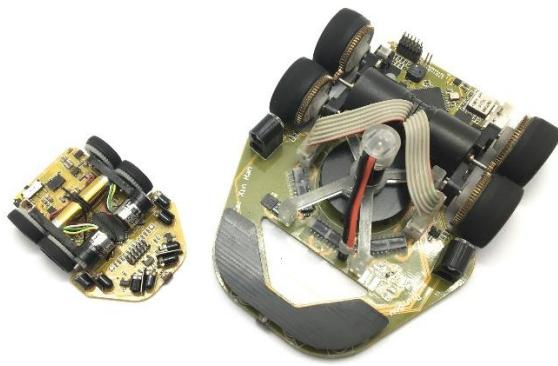


圖 1.1 本論文自行製作之半尺寸電腦鼠(左)與古典電腦鼠(右)

1.2 論文之貢獻

本論文「時間導向斜線迷宮演算法之設計與實作及其在電腦鼠比賽之應用」之貢獻如下：

1. 設計出一套最適合電腦鼠運動性能的時間導向斜線迷宮演算法。
2. 自行設計半尺寸電腦鼠之硬體電路與機構。
3. 針對不同競賽提出對應的策略。

第二章 時間導向斜線迷宮演算法

2.1 迷宮演算法介紹

在電腦鼠的比賽中，迷宮地圖是不會事先公布的，因此比賽選手必須撰寫一套迷宮演算法程式，讓電腦鼠第一次在迷宮中運行時先做搜尋的動作，同時將迷宮資訊紀錄到記憶體中，當第二次運行時，就可使用已記憶的迷宮資訊來計算路徑。

2.2 基礎洪水演算法[3]

洪水演算法為現今在電腦鼠比賽中較常見的演算法，相較於更古老的右手法則、左手法則、中右左法則與中左右法則，洪水演算法可以解出樹狀與迴圈的迷宮地圖，而前面提到的 4 個搜尋法則，則只能解出樹狀的迷宮地圖。

洪水演算法的概念是利用水流擴散的原理，先假設迷宮終點的洪水值為 0，洪水再開始往四個方向外的迷宮格子流，如果隔壁有牆就停止不流，如果隔壁沒牆就往隔壁流，並把洪水值(權重值)加 1 後填入下一格，如此反覆地填洪水值，直到洪水值填到起點時，就完成了洪水演算法填洪水值的動作，如圖 2.1，淺灰色格子(左下角)為起點，深灰色格子(右上角)為終點，以下迷宮起點及終點處皆以相同方式表示。

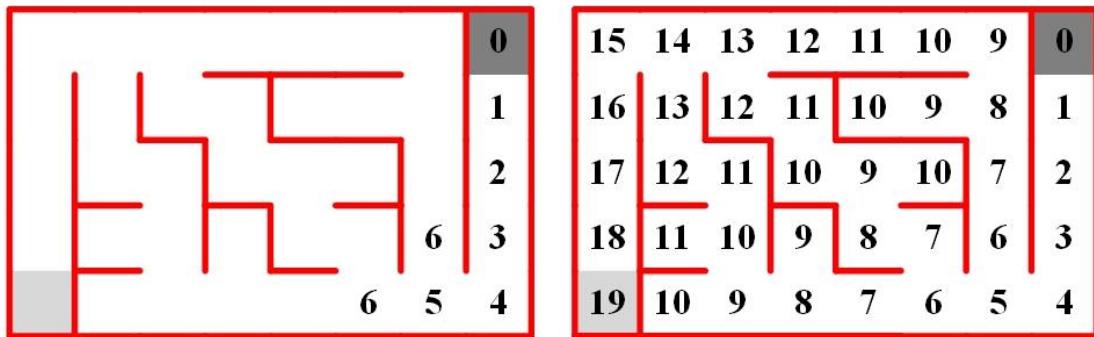


圖 2.1 洪水演算法做洪水擴散與填洪水值圖

2.3 尋找最短路徑

當完成填洪水值後就可以開始尋找最短路徑。其方式是從洪水值最大的地方(起點處)，往洪水值小的地方找，直到找到洪水值最小的地方(終點處)，就找完最短路徑了。但是在圖 2.2 中會發現，當洪水值從 14 往 13 找時，會遇到洪水值相同的路口，此時有 2 條不同的路徑可以走。因此在基礎洪水演算法中我們會定義絕對方向如圖 2.3，當發生洪水值相同時，就可依個人的設定方式選擇優先往相同方向尋找(直走優先)，或是往不同方向尋找(轉彎優先)。

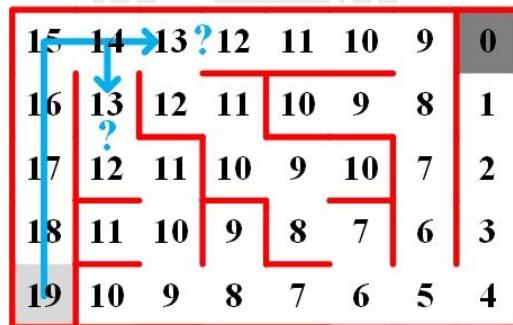


圖 2.2 路口相同洪水值

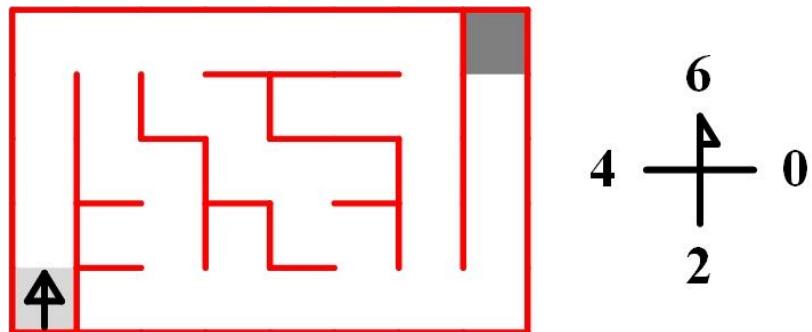


圖 2.3 以起點開口方向定義絕對方向

2.3.1 直走優先

若挑選路徑優先權為直走優先時，如圖 2.4，當洪水值從 15 往 14 走時，會在 15 的地方把絕對方向記錄為 0(往東邊)。14 往 13 走時，發現可走的方向有 2 個，因此需要和上一個紀錄的方向做比較，若往東邊走，則方向一樣為 0，若往南邊走，則方向會變成 2，但因為路線選擇設定為直走優先，因此會選擇把 14 的方向記錄為 0，也就是繼續往東邊走，如此反覆地尋找，就可以找到最短路徑，此地圖挑選到的路徑依絕對方向表示為 $6 \rightarrow 6 \rightarrow 6 \rightarrow 6 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 0 \rightarrow 6 \rightarrow 6 \rightarrow 6 \rightarrow 6$ 。

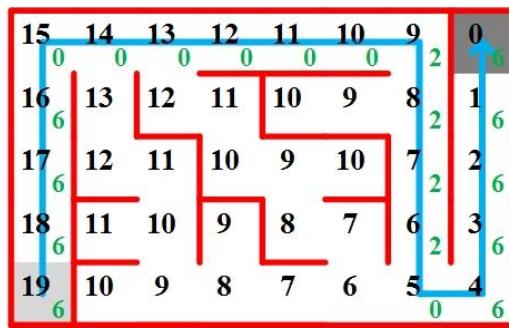


圖 2.4 直線優先挑選的路徑

2.3.2 轉彎優先

若挑選路徑優先權為轉彎優先時，如圖 2.5，15 往 14 時，記錄絕對方向為 0，14 往 13 時，因為轉彎優先，所以改走方向 2 的地方。轉彎優先也是以此方法反覆地尋找，就可以找到最短路徑，此地圖挑選到的路徑依絕對方向表示為 $6 \rightarrow 6 \rightarrow 6 \rightarrow 0 \rightarrow 2 \rightarrow 2 \rightarrow 0 \rightarrow 2 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 0 \rightarrow 6 \rightarrow 6 \rightarrow 6 \rightarrow 6$ 。

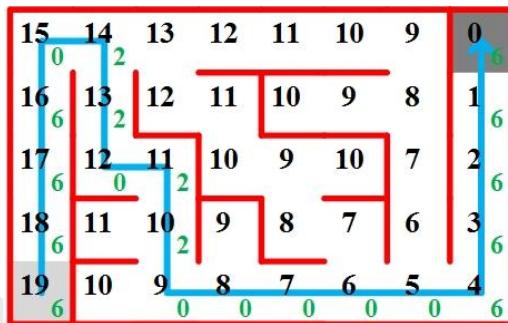


圖 2.5 転彎優先挑選的路徑

2.4 路徑選擇與探討

在 2.3.1 直線優先與 2.3.2 轉彎優先的章節中，會發現因為路口選擇權的不同，演算法就挑選到不同的路，但是若再仔細看這張地圖，會發現其實還有第三條未尋找到的路徑可以走，如圖 2.6。

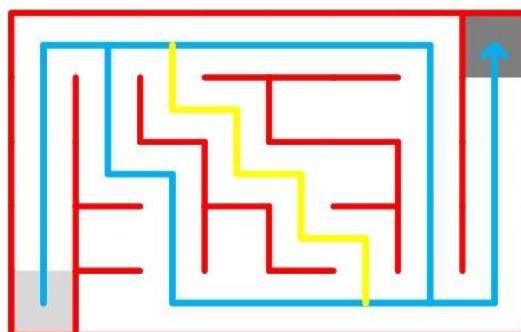


圖 2.6 未尋找到的迷宮路徑

在現今時代的競賽型電腦鼠，基本上皆具有斜線運動的能力，如圖 2.7，不像傳統的電腦鼠是走 90 度轉彎。斜線運動具有以下 2 個優點：

1. 減少行走距離
2. 長斜線可做加減速運動

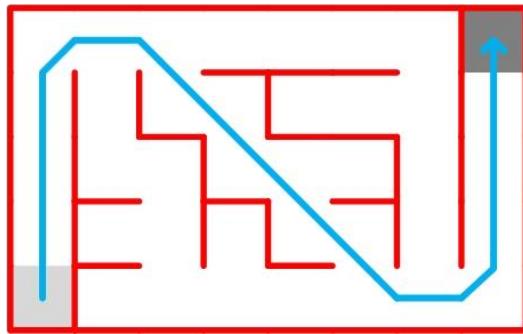
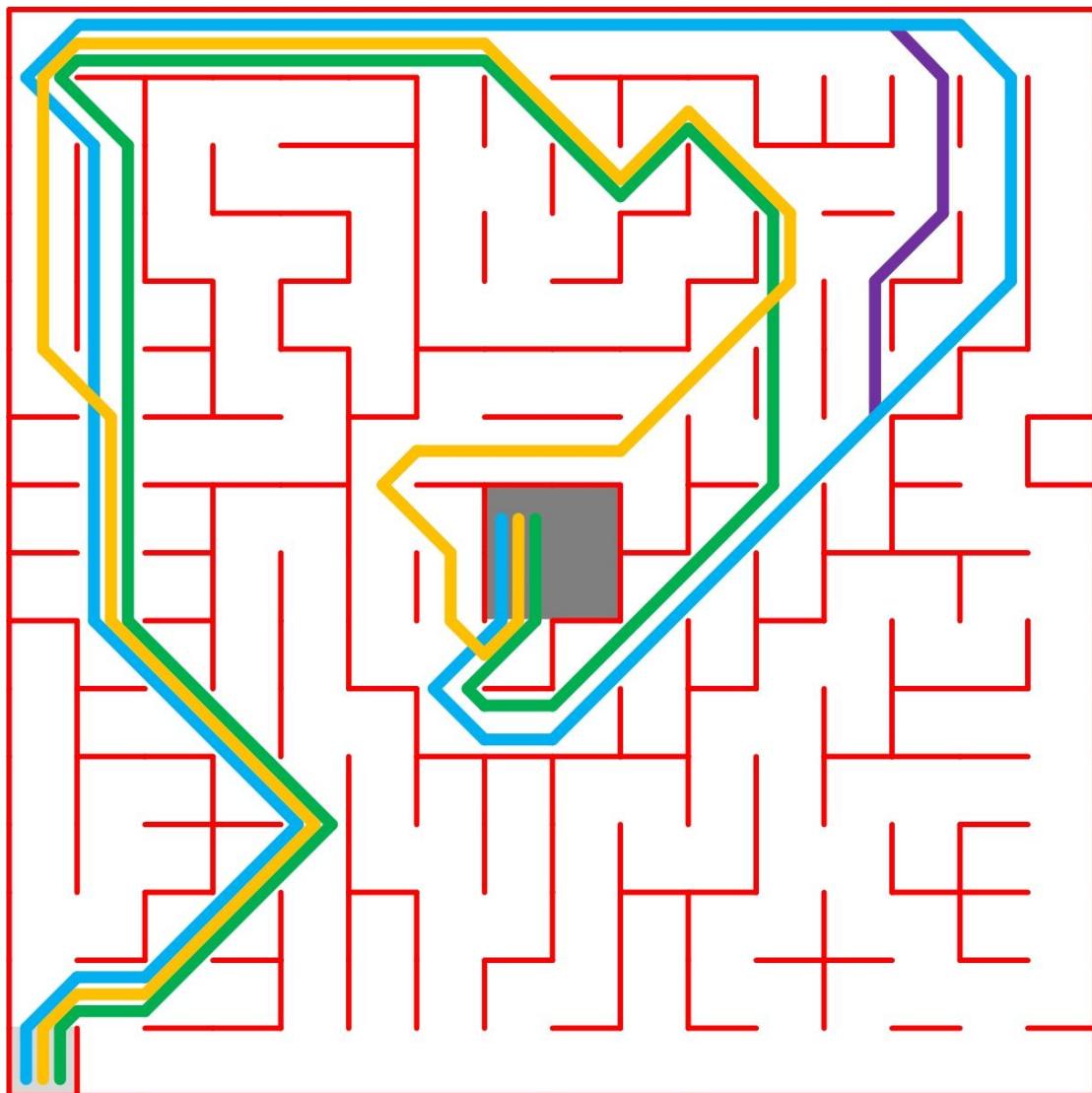


圖 2.7 斜線行走路徑

除了圖 2.6 使用基本洪水演算法找不到最短路徑之外，在參考文獻[4][5]中，皆有探討到關於 2011 年全日本電腦鼠決賽中，冠軍新加坡黃明吉老師與亞軍日本加藤雄資先生，2 人分別挑選了不同的路徑，若是以 2.2 章的基礎洪水演算法來尋找最短路徑，無論直線優先或斜線優先皆無法找到冠軍與亞軍挑選的路徑，如圖 2.8。在比賽中也證明了當電腦鼠在硬體機構與運動速度差不多，且皆具備斜線行走的能力下，選擇走最短路徑已經不適用在現今的電腦鼠上，因為有時路徑雖長，但是轉彎少，可做長距離衝刺，而有時路徑短，但是轉彎多，必須常常降低速度過彎。因此如何尋找到適合電腦鼠運動速度的「最佳」路徑，是現今比賽的重點之一。



- 黃明吉老師選的路徑
- 基礎洪水演算法(直走優先)
- 加藤雄資先生選的路徑
- 基礎洪水洪算法(轉彎優先)

圖 2.8 2011 年全日本古典電腦鼠比賽決賽路徑比較圖

在地圖更為複雜的半尺寸電腦鼠比賽中，路徑的挑選能力更是比古典電腦鼠重要。在 2014 年的半尺寸電腦鼠比賽中，前六名選手所挑選的路徑，如圖 2.9，其長度皆大於基礎洪水演算法所算出來的最短路徑，也是證實了以電腦鼠運動性能來挑選路徑的重要性。

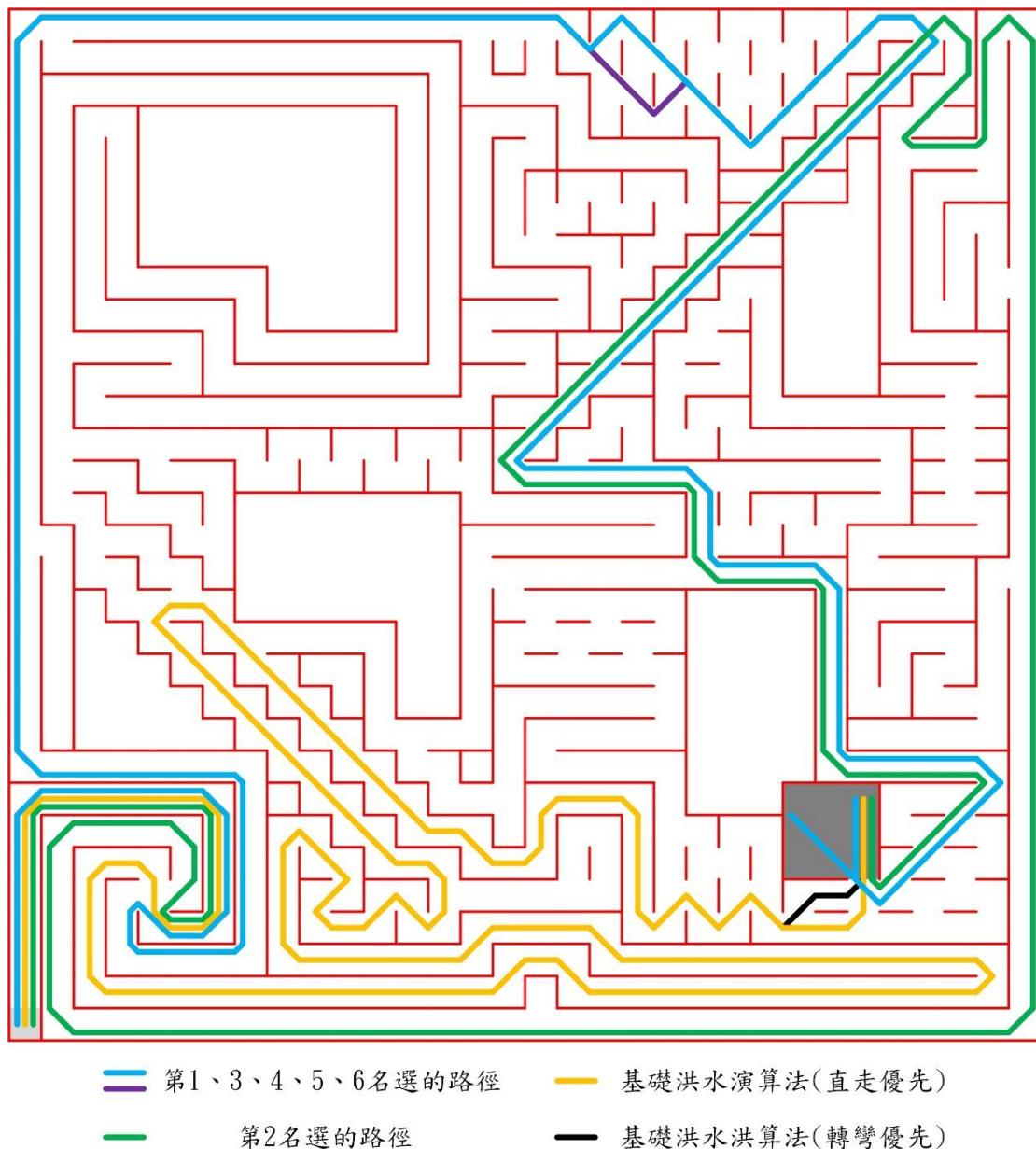


圖 2.9 2014 年全日本半尺寸電腦鼠比賽決賽路徑比較

2.5 時間導向斜線迷宮演算法設計

本論文關於時間導向斜線迷宮演算法的設計目的，是要讓演算法能依照電腦鼠的運動性能來計算最佳路徑，因人人設計的電腦鼠運動性能皆不同，以古典電腦鼠來說，有些人設計的電腦鼠直線最高速可超過 5m/s，但是轉彎平均速度小於 150cm/s，這樣的電腦鼠就很適合跑長直線與長斜線多的路，但也有人的電腦鼠直線速度只有 4m/s 左右，但是轉彎平均速度大於 150cm/s，就相對比較適合走轉彎多但直線與斜線距離短的路徑。

本論文所設計之時間導向斜線迷宮演算法改良了原有的基礎洪水演算法，保留了原有「洪水擴散」與「覆蓋」的特徵，亦改變了原有填洪水值的位置，還有新的洪水值設計，並新增了洪水方向紀錄與重複方向次數紀錄等資訊。

在基礎洪水演算法中，是以距離作為參考單位，當洪水往隔壁擴散時洪水值就加 1，可說是洪水值 1 等於 18 公分的意思(古典電腦鼠)，最後填完洪水值後再往回找距離最短的路。本論文提出的時間導向斜線迷宮演算法中，是以時間做為參考單位，為了實現出以時間為單位的洪水值，本論文假定電腦鼠性能設置了以下 4 個參數，並以這 4 個參數實現出新的洪水值設計：

1. 加速度(a)：10m/s/s
2. 直線最高速：3.5m/s
3. 斜線最高速：3.0m/s
4. 轉彎平均速度：100cm/s

考慮到運算演算法花費時間的問題，本論文在轉彎部分取平均速度做計算，而不使用 45、90、135、180 度個別轉彎速度計算。首先取最低速也就是轉彎平均速度 100cm/s，當作老鼠在迷宮中行走的基礎速度，以 100cm/s 的速度走過 1 格迷宮(18 公分)需耗時 0.18s，0.18s 即設定為洪水值 1，後續將以此做為時間的基礎，並探討不同衝刺距離時洪水值的設計與使用。

2.5.1 加速洪水值設計

本章節討論的是當電腦鼠做直線與斜線衝刺時，每加速 1 格所對應的洪水值關係。在直線衝刺部分，透過方程式(2.1)求出古典電腦鼠每走過 18 公分後的速度，再使用方程式(2.2)，用每 1 格的速度差求出經過時間，最後在經過正規化後即可求出跑直線時，每 1 格對應的洪水值，如表 2.1。當經過 3 次加速後若在第 4 次再加速則會超過設定的直線最高速 3.5m/s ，因此在第 3 次後的速度，都維持在 3.435m/s ，洪水值皆是 0.29。

$$V_{fn}^2 = V_0^2 + 2(10)n \times 0.18, n = 1, \dots, 3 \quad (2.1)$$

$$t_n = \frac{V_{fn} - V_0}{10}, \Delta t_n = t_n - t_{n-1} \quad (2.2)$$

表 2.1 直線加速洪水值

n 對應關係	1st	2nd	3rd	4th
$V_{fn}(\text{m/s})$	2.145	2.864	3.435	3.435
$\Delta t_n(\text{s})$	0.114	0.071	0.057	0.052
Δt_n 正規化洪水值	0.64	0.40	0.32	0.29

在斜線衝刺部分，它與直線略有不同，斜線的距離是 $18/\sqrt{2}=12.728$ 公分，如圖 2.10，因此以最低速 100cm/s 走過 1 格斜線花費時間約 0.127s ，正規化洪水值後得 0.707。

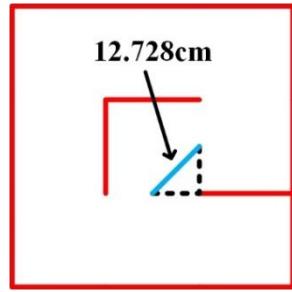


圖 2.10 斜線距離示意圖

透過方程式(2.3)求出每走過 12.728 公分後的速度，再使用方程式(2.4)，用每 1 格的速度差求出經過時間，最後在經過正規化後即可求出跑斜線時，每 1 格對應的洪水值，如表 2.2。當經過 3 次加速後若在第 4 次再加速則會超過設定的斜線最高速 3.0m/s，因此在第 3 次後的速度，都維持在 2.939m/s，洪水值皆是 0.24。

$$V_{fdm}^2 = V_0^2 + 2(10)m \times \frac{0.18}{\sqrt{2}}, m=1, \dots, 3 \quad (2.3)$$

$$t_m = \frac{V_{fdm} - V_0}{10}, \Delta t_m = t_m - t_{m-1} \quad (2.4)$$

表 2.2 斜線加速洪水值

n 對應關係	1st	2nd	3rd	4th
V_{fm} (m/s)	1.883	2.468	2.939	2.939
Δt_m (s)	0.088	0.059	0.047	0.043
Δt_m 正規化洪水值	0.49	0.33	0.26	0.24

2.5.2 減速洪水補償值設計

在 2.5.1 章中的加速洪水值是指電腦鼠每移動 1 格所加上去的對應洪水值，但是這個洪水值只有考慮到加速的部分，並沒有考慮到當遇到轉彎時的減速問題，因此本章節探討的是當洪水淹到轉彎處時，要補上一個現在速度與最低速度的時間差洪水補償值。

在直線減速部分，使用方程式(2.5)計算出直線每半格的最高速 V_{ft} ，再透過方程式(2.6)計算出每加速 n 個半格後再減速 n 個半格所花費的時間 $t_{n,t}$ ，最後再把 $t_{n,t}$ 減掉方程式(2.2)的 t_n ，即可求得有做完整加減速的時間與只做加速時間的時間差 Δt_{np} ，時間差 Δt_{np} 經過正規化後處理後即可求得減速洪水補償值，如表 2.3。在第 6 個半格後求得的速度皆已達到最高速，因此減速洪水補償值皆是 0.48。

$$V_{ft}^2 = V_0^2 + 2(10)\frac{n}{2} \times 0.18, n = 1, \dots, 6 \quad (2.5)$$

$$t_{n,t} = \frac{2(V_{ft} - V_0)}{10}, \Delta t_{np} = t_{n,t} - t_n \quad (2.6)$$

表 2.3 直線減速洪水值

n 對應關係	1st	2nd	3rd	4th	5th	6th
V_{ft} (m/s)	1.673	2.145	2.530	2.864	3.162	3.435
$t_{n,t}$ (s)	0.134	0.228	0.305	0.372	0.432	0.487
t_n (s)	0.114	0.186	0.243	0.295	0.348	0.400
Δt_{np} (s)	0.020	0.042	0.062	0.076	0.084	0.086
Δt_{np} 正規化洪水值	0.11	0.24	0.35	0.43	0.47	0.48

在斜線減速部分，使用方程式(2.7)計算出斜線每半格的最高速 V_{ftd} ，再透過方程式(2.8)計算出每加速 n 個半格後再減速 n 個半格所花費的時間 $t_{m,t}$ ，最後再把 $t_{m,t}$ 減掉方程式(2.4)的 t_m ，即可求得有做完整加減速的時間與只做加速時間的時間差 Δt_{mp} ，時間差 Δt_{mp} 經過正規化後處理後即可求得減速洪水補償值，如表 2.4。在第 6 個半格後求得的速度皆已達到最高速，因此減速洪水補償值皆是 0.36。

$$V_{ftd}^2 = V_0^2 + 2(10) \frac{m}{2} \times \frac{0.18}{\sqrt{2}}, m=1, \dots, 6 \quad (2.7)$$

$$t_{m,t} = \frac{2(V_{ftd} - V_0)}{10}, \Delta t_{mp} = t_{m,t} - t_m \quad (2.8)$$

表 2.4 斜線減速洪水值

n 對應關係	1st	2nd	3rd	4th	5th	6th
V_{ftd} (m/s)	1.508	1.883	2.195	2.468	2.714	2.939
$t_{m,t}$ (s)	0.101	0.176	0.239	0.293	0.342	0.387
t_m (s)	0.088	0.146	0.193	0.239	0.280	0.323
Δt_{mp}	0.013	0.029	0.045	0.056	0.062	0.063
Δt_{mp} 正規化洪水值	0.07	0.17	0.25	0.31	0.35	0.36

2.6 資訊定義

2.6.1 迷宮牆壁定義

本論文以迷宮格子為單位來定義每 1 格的編號，為了程式陣列撰寫上的方便，編號由 0 開始，並定義在起點位置，排序方式是先由西至東(由左至右)，再由南至北(由下而上)，如圖 2.11， 5×5 的迷宮地圖，迷宮編號的最大值為 $n^2 - 1$ ，本地圖的最大值為 $5^2 - 1 = 24$ ，其中 n 由迷宮的橫向或縱向排列的最大值決定。

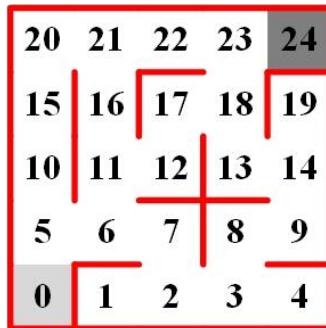


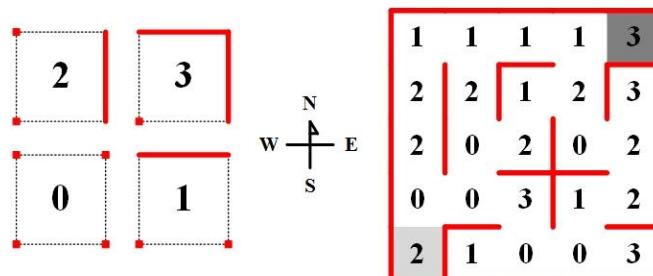
圖 2.11 迷宮編號定義

本論文在紀錄迷宮牆壁時，以格為單位並記錄北邊及東邊 2 面牆壁的資訊，如表 2.5，這種方式相較於東西南北 4 面牆壁都紀錄來說，具有節省記憶體的好處(只需 2 位元)。

表 2.5 迷宮牆壁資訊定義表

位元	2	1(最低位元)
紀錄方向	東	北

使用北邊及東邊 2 面牆紀錄資訊，在每 1 格中最多只有 4 種組合方式，如圖 2.12 (a)，每 1 格的南邊及西邊，都屬於其他格記錄的範圍，所以可以忽略不看，而最南邊及最西邊的迷宮外圍牆，由於已知是有牆狀態，所以不必在宣告變數去記錄，圖 2.12 (c)為 5×5 迷宮地圖牆壁資訊狀況。



(a) 牆壁組合種類 (b) 絶對方向 (c) 完整迷宮牆壁資訊

圖 2.12 迷宮牆壁記錄資訊

2.6.2 邊緣填洪水值設計

為了使演算法可以更容易區分出直線與斜線，本論文將演算法填洪水值的位置定義在每 1 格的邊緣上，如圖 2.13，每 1 個邊緣上皆帶有 3 種資訊，分別是洪水值、水流方向、重複方向次數，而最南邊及最西邊的迷宮外圍牆，一樣不必宣告變數紀錄。

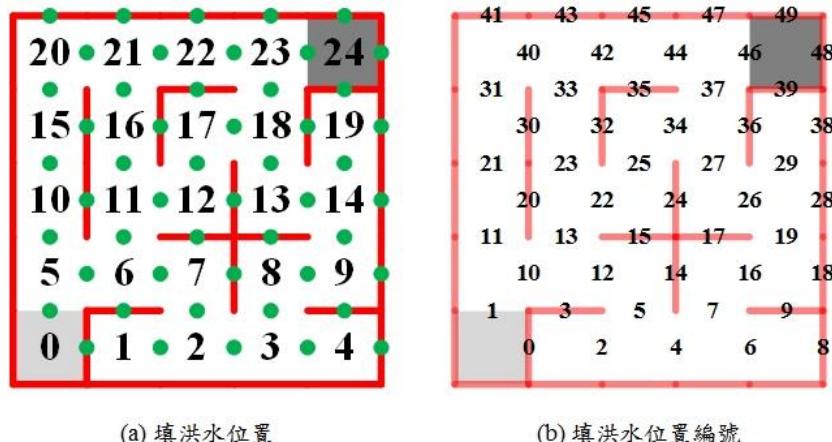


圖 2.13 邊緣洪水位置及對應位置編號

2.6.3 水流方向

在時間導向斜線迷宮演算法中使用 8 個方向來紀錄演算法水流方向，如圖 2.14，相較於早期常用的東、南、西、北 4 個方向來說，多新增的東南，西南、東北、西北 4 個方向，可以清楚記錄演算法走斜線時的狀況。

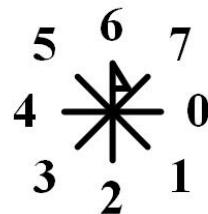


圖 2.14 時間導向斜線迷宮演算法迷宮方向定義

時間導向洪水演算法在迷宮中做洪水擴散時，水流方向必須依照設計的規則擴散出去，並且一樣要先確認完該處沒有牆壁後才可執行。

在 8 個方向中可分為 3 種擴散方法，第 1 種是方向為 0(東)、2(南)、4(西)、6(北)時往外擴散，如圖 2.15。

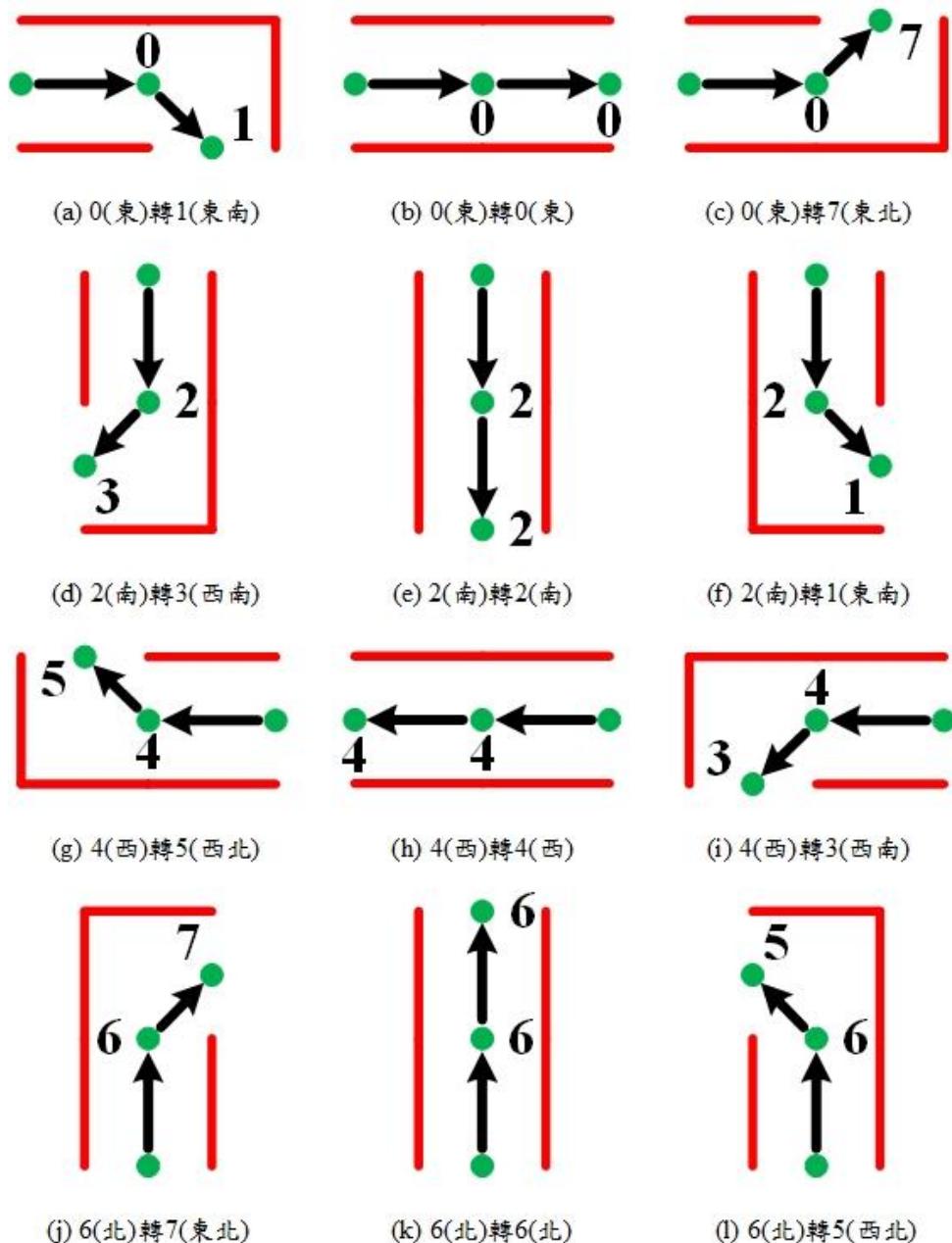


圖 2.15 第 1 種水流擴散方式

第 2 種是方向為 1 (東南)、3 (西南)、5 (西北)、7 (東北)，且當前填洪水位
置在奇數時(平行於柱子)往外擴散，如圖 2.16。

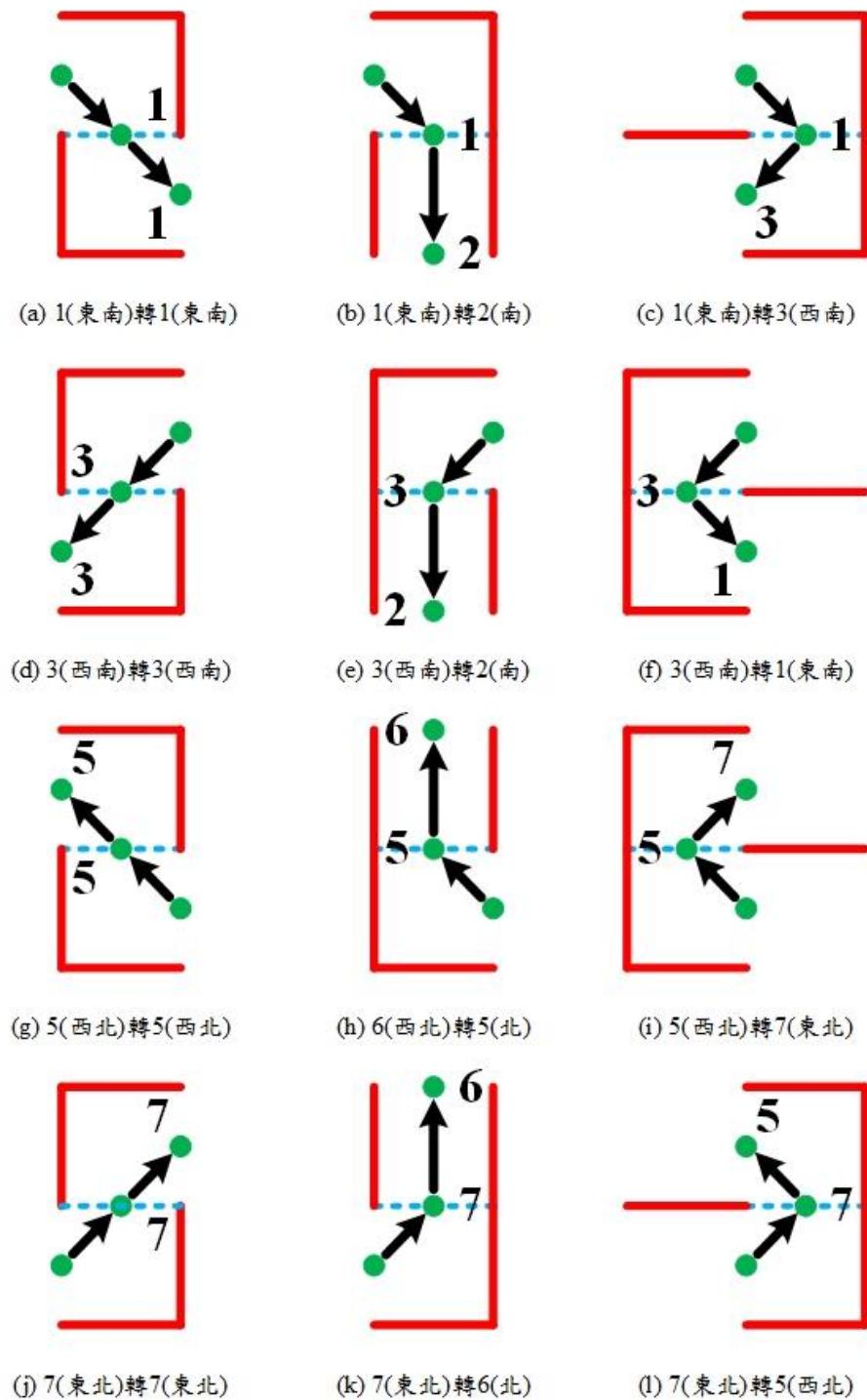


圖 2.16 第 2 種水流擴散方式

第3種一樣是方向為1(東南)、3(西南)、5(西北)、7(東北)，但當前填洪水位置在偶數時(垂直於柱子)往外擴散，如圖2.17。

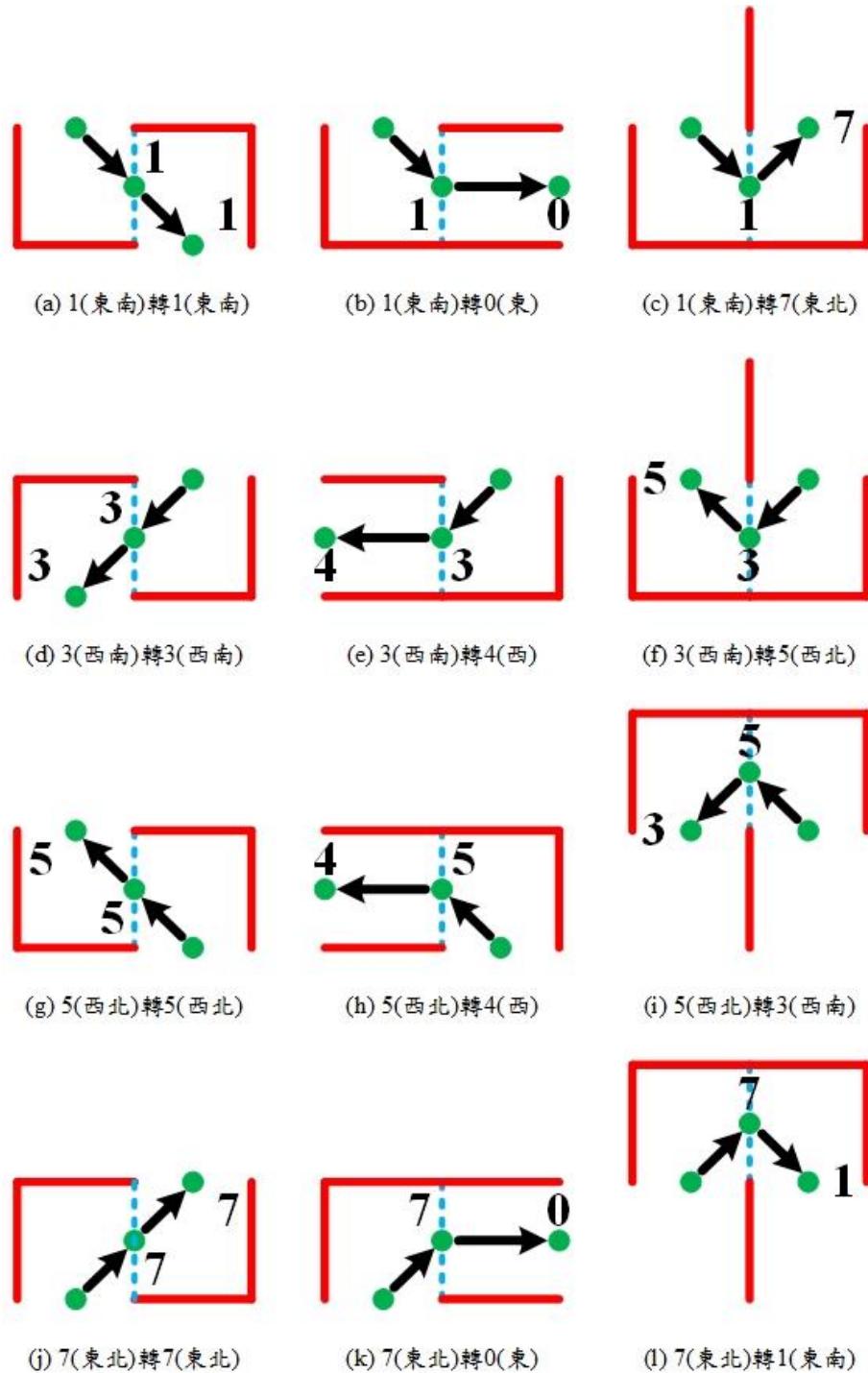


圖2.17 第3種水流擴散方式

2.6.4 重複方向次數

為了使時間導向斜線迷宮演算法可以實現出長直線或長斜線加速，因此本論文在填洪水時，會紀錄洪水在同一方向擴散了幾次，有了重複方向次數，就可以快速地查表並填入對應的洪水值，如圖 2.18，利用水流方向紀錄重複次數。

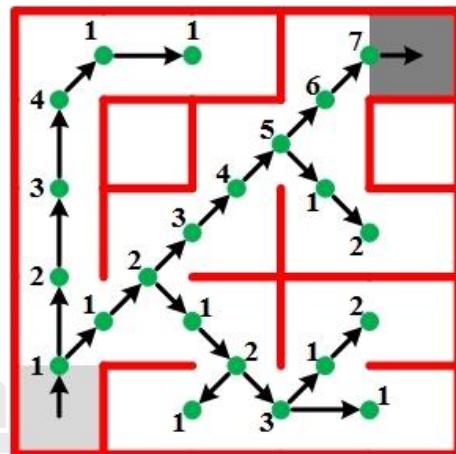


圖 2.18 根據水流方向記錄重複方向次數

2.7 變數宣告

為了使時間導向斜線迷宮演算法可相容在古典電腦鼠及半尺寸電腦鼠中，本論文在宣告變數型態時，以半尺寸電腦鼠的迷宮狀況做為參考，原因是半尺寸電腦鼠的迷宮是古典電腦鼠的 4 倍大，因此各項資訊的位元數需求都會比較多，使用上也可以直接向下相容古典電腦鼠迷宮的需求。

演算法在計算時需要大量的記憶體記憶各種資訊，以下是各種變數所需的位元數及資料量：

1. 預設全開式牆壁陣列：使用東邊及北邊牆壁記錄資訊，需要 2 位元，半尺寸電腦鼠需要 1024 筆，古典電腦鼠需要 256 筆。
2. 預設全關式牆壁陣列：使用東邊及北邊牆壁記錄資訊，需要 2 位元，半尺寸電腦鼠需要 1024 筆，古典電腦鼠需要 256 筆。

3. 拜訪記錄陣列：使用東邊及北邊牆壁記錄資訊，需要 2 位元，半尺寸電腦鼠需要 1024 筆，古典電腦鼠需要 256 筆。

4. 洪水值陣列：由於 C 語言中沒有浮點數陣列，因此在程式中實際使用時，會將 2.5.1 及 2.5.2 章求得的洪水值放大 100 倍，洪水值放大後即可設定洪水最大值，假設半尺寸電腦鼠以連續轉彎路徑走 $3/4$ 個場地，洪水值約是 $70 \times 768 = 54,528$ ，需要宣告 16 位元記錄，半尺寸電腦鼠需要 2048 筆，古典電腦鼠需要 512 筆。

5. 水流方向陣列：共 8 個方向需要 3 位元，半尺寸電腦鼠需要 2048 筆，古典電腦鼠需要 512 筆。

6. 重複方向次數陣列，半尺寸電腦鼠迷宮場地直線水流方向最多重複 31 次，斜線水流方向最多重複 61 次，因此需要 6 位元，半尺寸電腦鼠需要 2048 筆，古典電腦鼠需要 512 筆。

7. 最短路徑兼洪水順序暫存陣列，洪水順序暫存最大值為 2047，所以需要 11 位元，所需陣列數量則是假設最短路徑要跑 $3/4$ 個迷宮，因此半尺寸電腦鼠需要 768 筆，古典電腦鼠需要 192 筆。

根據 7 項資料需求，本論文為了節省資料量將預設全開式牆壁、預設全關式牆壁及拜訪記錄這 3 項宣告成結構變數使用，其他的變數都直接依所需的位元數宣告，半尺寸電腦鼠與古典電腦鼠都是使用一樣的資料型態格式，如圖 2.19。

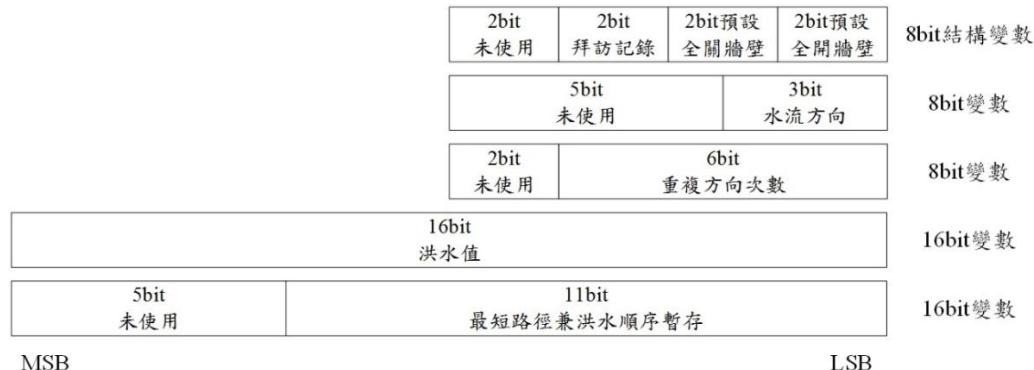


圖 2.19 變數宣告格式

在變數宣告的程式碼上，圖 2.20 為半尺寸電腦鼠的宣告，圖 2.21 為古典電腦鼠的宣告。本論文使用 Renesas RX62T 系列晶片，並在該公司的 High Perform Workspace 開發環境中撰寫程式，圖 2.20 與圖 2.21 的差異只有陣列的需求數因迷宮大小而有所不同。

```

//*****
//Time-Base Diagonal Maze Solver For Half Size Micromouse
//*****
union
{
    unsigned char imformation;
    struct
    {
        unsigned char open : 2;      //2bit預設牆壁全開
        unsigned char close : 2;     //2bit預設牆壁全關
        unsigned char visit : 2;     //2bit拜訪資訊
        unsigned char none : 2;      //2bit未使用
    };
}maze[1024];
unsigned char direction [2048];           //水流方向
unsigned char count[2048];                 //重複方向紀錄
unsigned short flood_value[2048];          //洪水值
unsigned short Action_Path[768];           //最短路徑兼洪水順序暫存陣列

```

圖 2.20 半尺寸電腦鼠中演算法變數宣告

```

//*****
//Time-Base Diagonal Maze Solver For Classic Micromouse
//*****
union
{
    unsigned char imformation;
    struct
    {
        unsigned char open : 2;      //2bit預設牆壁全開
        unsigned char close : 2;     //2bit預設牆壁全關
        unsigned char visit : 2;     //2bit拜訪資訊
        unsigned char none : 2;      //2bit未使用
    };
}maze[256];
unsigned char direction [512];            //水流方向
unsigned char count[512];                 //重複方向紀錄
unsigned short flood_value[512];          //洪水值
unsigned short Action_Path[192];           //最短路徑兼洪水順序暫存陣列

```

圖 2.21 古典電腦鼠中演算法變數宣告

2.8 填洪水

為了在填洪水時方便尋找對應的洪水值，本論文將表 2.1 的直線加速洪水值及表 2.3 的直線減速洪水補償值，整理成圖 2.22 的方式來表示；表 2.2 的斜線加速洪水值及表 2.4 的斜線減速洪水值，也一樣整理成圖 2.23 的方式來表示，並都已放大 100 倍，方便撰寫程式或除錯時使用。

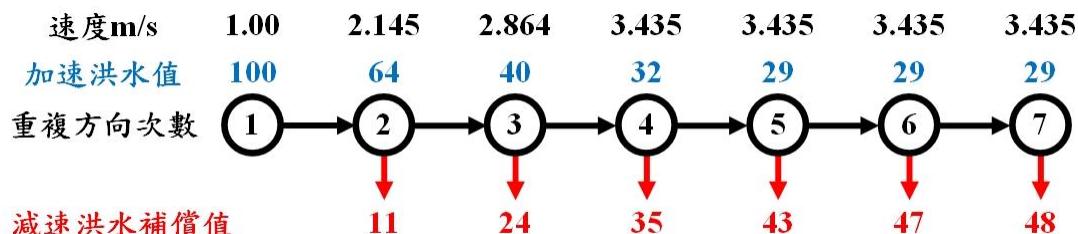


圖 2.22 直線時對應的洪水值

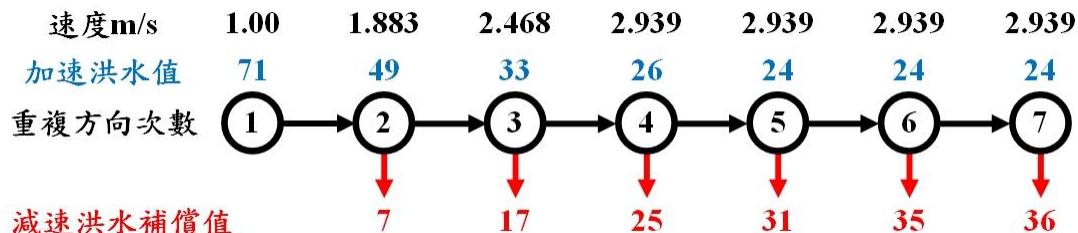


圖 2.23 斜線時對應的洪水值

在開始執行洪水演算法前，需先瞭解演算法在執行時的各種資訊，如圖 2.24。演算法以車頭方向當作洪水起始點，並且預設起始位置洪水值為 50，其他點的洪水值皆預設為最大值 65535(圖 2.24 中不顯示)，設定完後便可開始做洪水擴散。

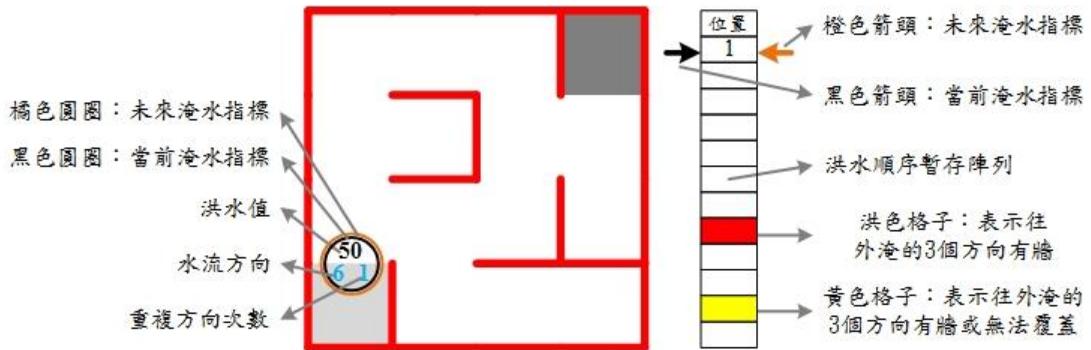


圖 2.24 地圖各項資訊表示示意圖

當洪水從 A 點要往 B 點擴散時，若 A 點的洪水值加上欲到 B 點的加速值與減速補償值後，小於 B 點的原有洪水值，洪水就可以擴散過去，以圖 2.25(a)為例，位置 1 的洪水要擴散到位置 9 時，洪水值 $50+64(\text{加速值})+0(\text{減速補償值})=114$ ，小於原本位置 9 的洪水值 65535，因此可以覆蓋過去，圖 2.25 至圖 2.29為執行洪水擴散時的完整示意圖。

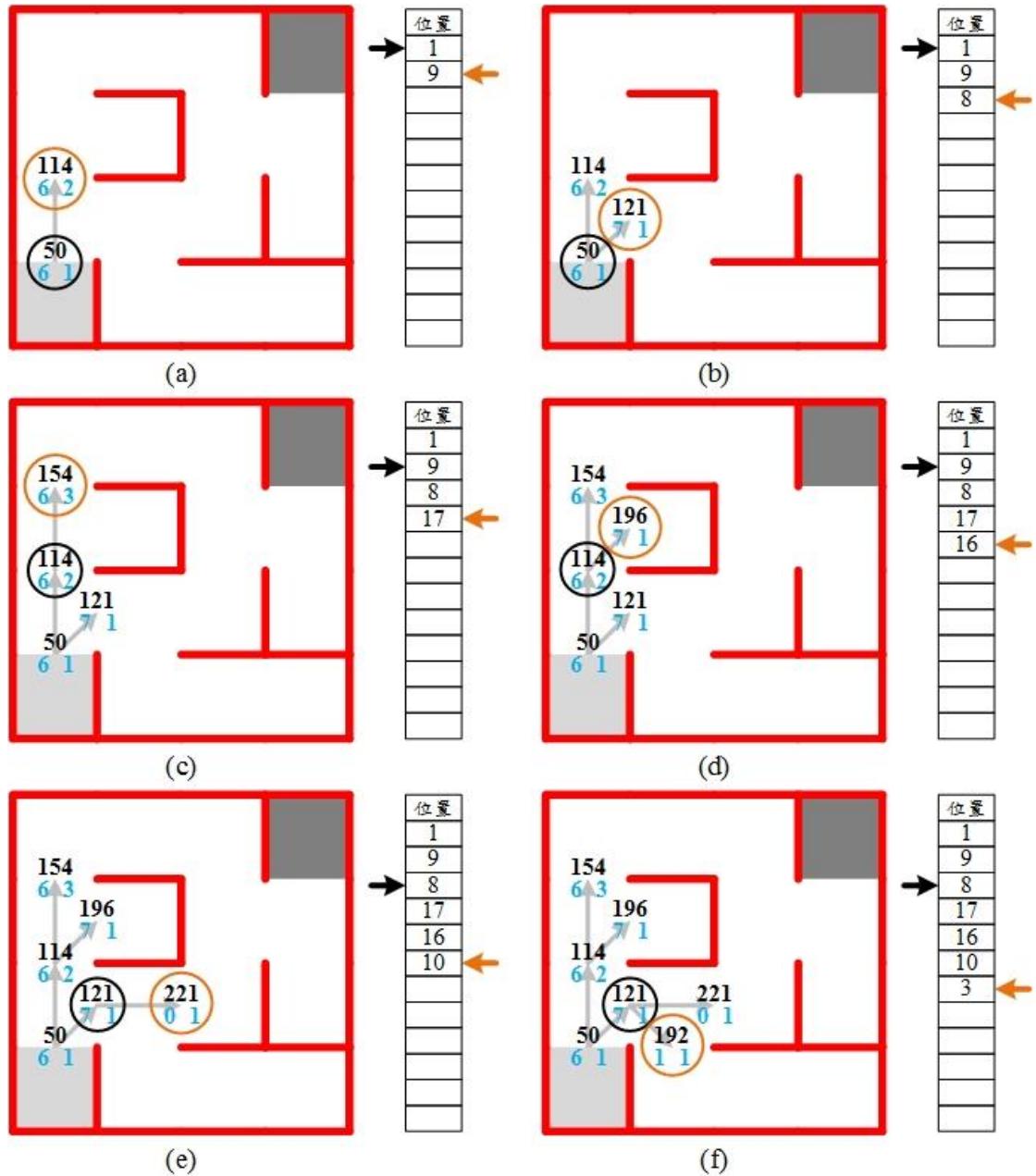


圖 2.25 填洪水值步驟(a)~(f)

圖 2.26(h)當當前指標在位置 16，發現三面有牆洪水無法往外擴散時，當前指標直接加 1 繼續執行即可。

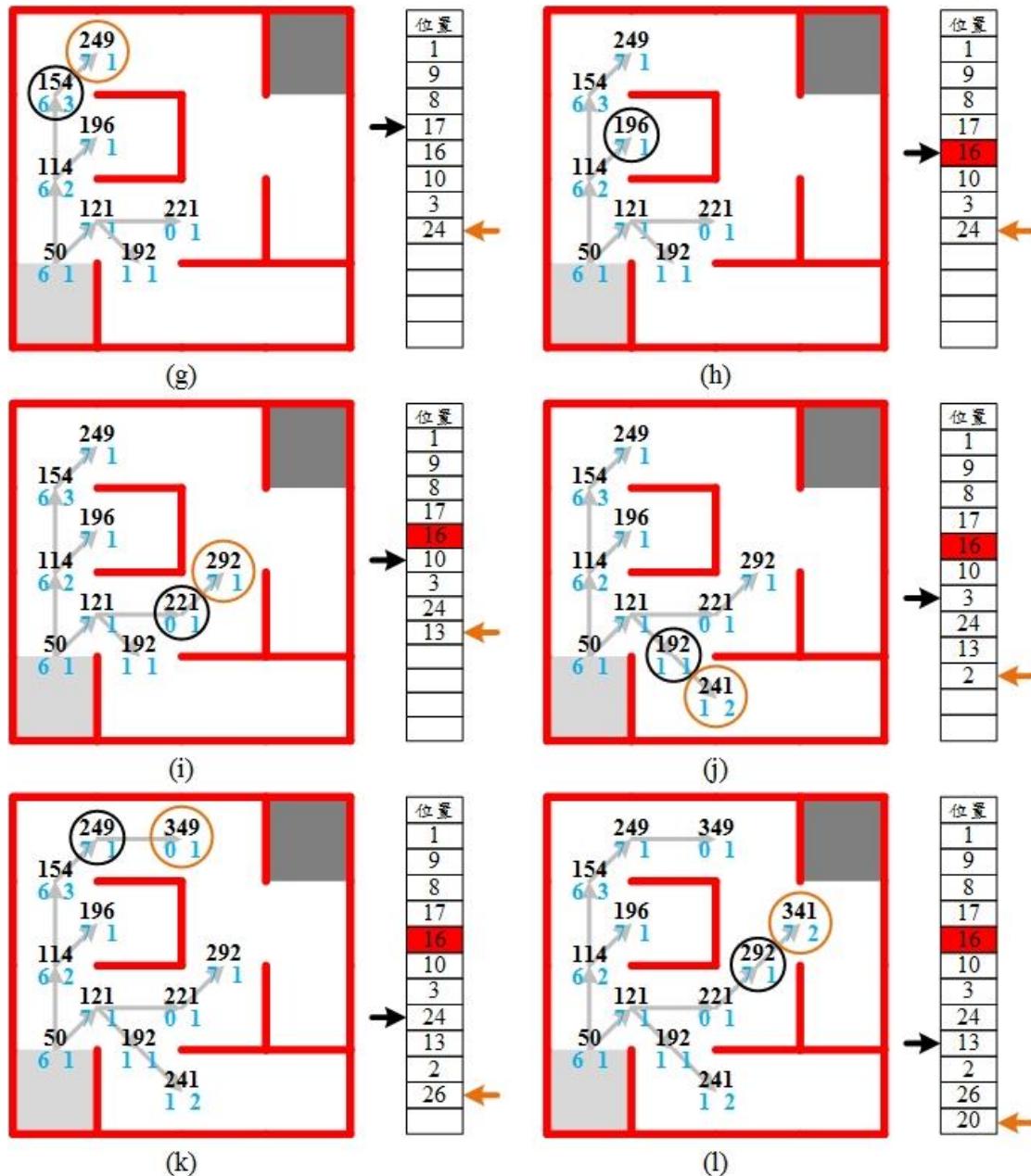


圖 2.26 演算法執行步驟(g)~(l)

圖 2.27(m)當未來指標到達陣列最後一格時，下次的未來指標直接回到陣列 0 的地方覆蓋原有的位置資料。

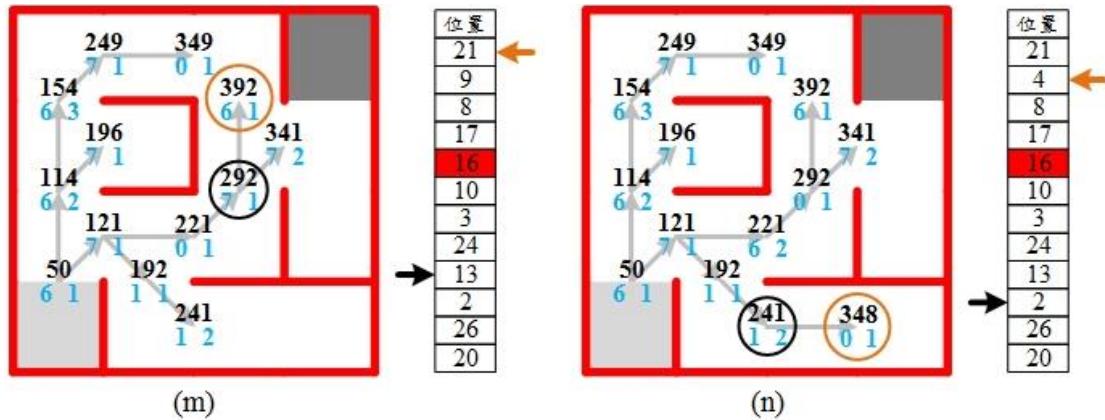


圖 2.27 演算法執行步驟(m)~(n)

圖 2.28(o)當位置 26 的洪水要擴散到位置 21 時，洪水值 $349+71$ (加速值)+0(減速補償值)=420，大於位置 21 原本的洪水值 392，因此無法覆蓋。

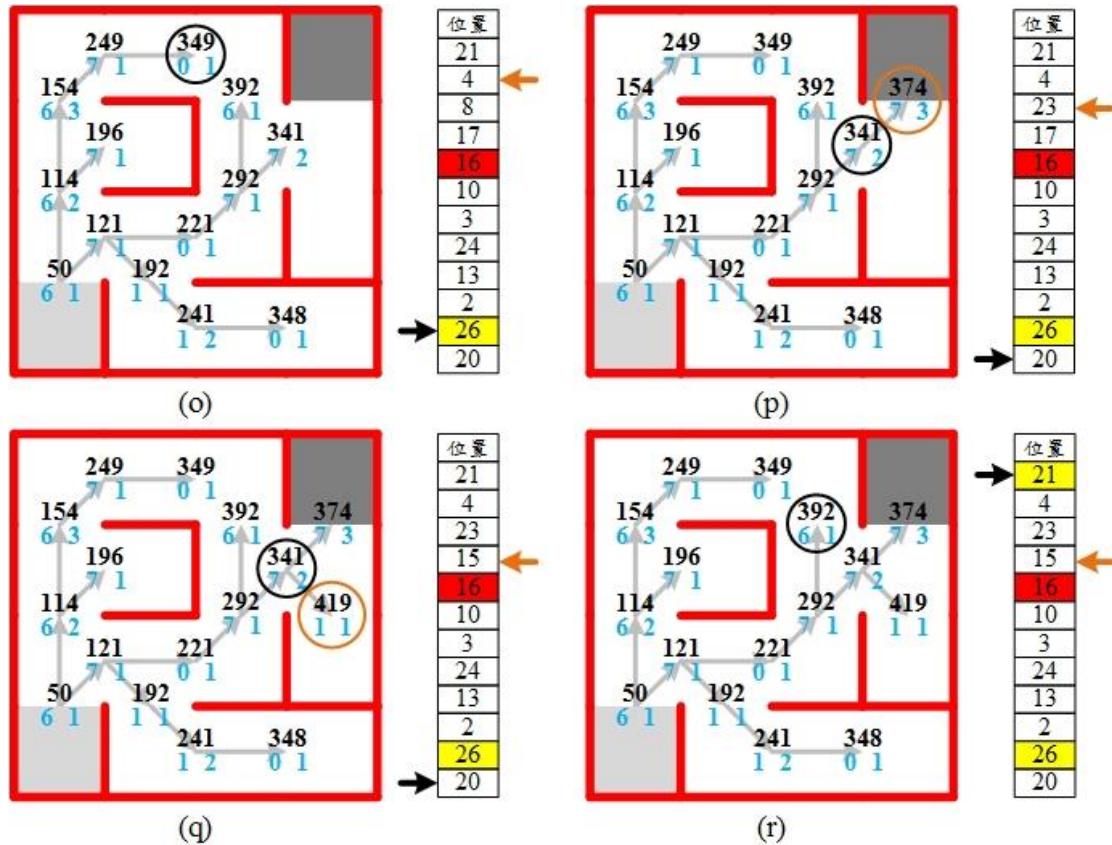


圖 2.28 演算法執行步驟(o)~(r)

圖 2.29(u)當現在指標與未來指標重疊時，表示洪水已無法再擴散，演算法執行洪水擴散的步驟到此結束。

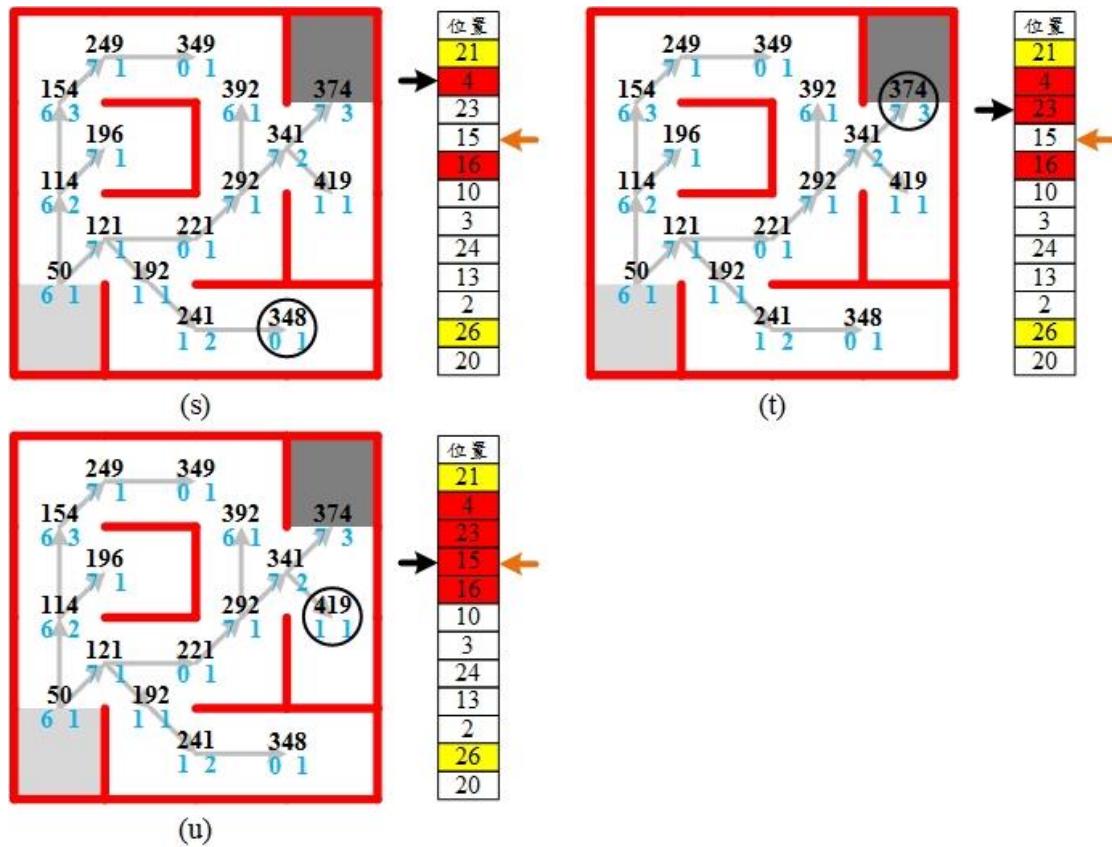


圖 2.29 演算法執行步驟(s)~(u)

第三章 迷宮地圖搜尋與最佳路徑計算

3.1 演算法花費時間

演算法在電腦鼠中的功能就是做路徑計算，但其實可再細分成 2 種不同的狀況，第 1 種是搜尋時的路徑計算，第 2 種是已經搜尋完成後的最佳路徑計算。兩者最大的差異在搜尋時的計算是動態的，需要邊走邊算。搜尋完後的最佳路徑計算則是靜態的，在衝刺前計算即可。因此在搜尋時我們必須注意演算法計算時所花費的時間，否則若是搜尋時運算時間不足，導致迷宮地圖搜尋失敗，後續的最佳路徑計算可能也將無法完成。

在搜尋時電腦鼠的動作有三種，直走、迴轉與左右轉。以本論文設計的半尺寸電腦鼠為例，在直線道中行進到路口前 4 公分處時，即可看到下一格的牆壁資訊，因此在走直走的動作時，可提前 4 公分就探索完牆壁，並計算下次的路徑，但是在左右轉時就無法提前 4cm 看下一格牆壁，因為電腦鼠在轉彎時分成三個步驟(1)直走(2)轉彎(3)直走，本論文設計的半尺寸電腦鼠在第 1 與第 3 步驟的直走都是 1.5 公分，也就是當電腦鼠探索完下格牆壁後，演算法只剩 1.5 公分的距離可計算路徑，如圖 3.1 搜尋時最嚴苛的演算法運算距離示意圖。

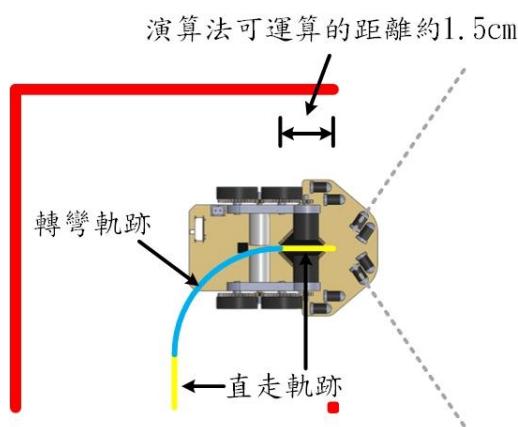


圖 3.1 搜尋時最嚴苛的演算法運算距離示意圖

已知搜尋時最嚴苛的運算距離為 1.5cm，而本論文設計之半尺寸電腦鼠的搜尋速度為 25cm/s，透過公式可求出演算法運算時的限制時間：

$$t = \frac{D}{V} = \frac{1.5(cm)}{25(\frac{cm}{s})} = 0.06(s) = 60(ms) \quad (3.1)$$

雖然以 25cm/s 速度走過 1.5cm 有 60ms 的時間，但實際上電腦鼠在搜尋時，本論文設定每 1ms Timer 計時中斷執行 1 次讀取編碼器、紅外線 A/D 轉換與運動控制，所以演算法實際可計算的時間會小於 60ms，因此在將演算法套用至電腦鼠上並實際使用時需特別注意，否則電腦鼠可能在走到路口時，還不知道下一步該怎麼走。為測試本論文設計之半尺寸電腦鼠是否符合搜尋時間限制，本論文設定 1 個 1ms 中斷的 Timer，並宣告 1 個變數紀錄演算法開始至結束所花時間，再利用 UART 傳輸至電腦上的 AccessPort 軟體觀察，如圖 3.2，以第 1 個路口前演算法運算時間為例，電腦鼠總共計時中斷 18 次，其表示花費時間為 $18 \times 1\text{ms} = 18\text{ms}$ ，其中 18ms 中電腦鼠做的事已包含執行演算法、讀取編碼器、紅外線 A/D 轉換與運動控制，因此低於 60ms 也完全符合限制時間規定。

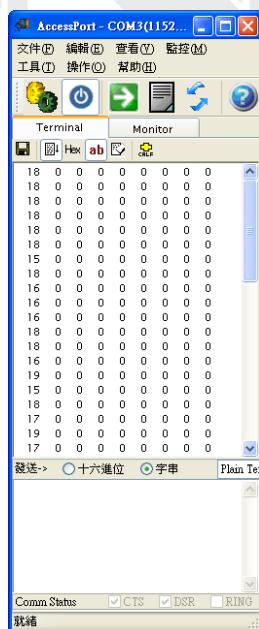


圖 3.2 演算法花費時間實測結果

3.2 未知迷宮搜尋

在 2.7 節的變數宣告中，有 1 項結構變數陣列，裡面包括了 3 種資訊，預設全開式牆壁、預設全關式牆壁及拜訪記錄。在搜尋時主要是依靠預設全開式牆壁資訊來尋找路徑至目標區域，也就是終點或起點，並且依靠拜訪記錄來判斷牆壁是否已記錄，預設全關式牆壁資訊則是用來計算衝刺路徑時使用。3 項變數都各有不同的功能，只是有不同的使用時機，因此在搜尋時這些資訊都要同步更新，資料格式如表 2.5，紀錄預設全開式牆壁與預設全關式牆壁資料時，有牆即在對應方位處填入 1，無牆即為 0，拜訪記錄資料則是只要該處探索過就填入 1，若無探索過就保持為 0。

以 4×4 地圖為例，如圖 3.3(a)，由於已知地圖的大小為 4×4 ，因此在搜尋前可先將預設全開式牆壁與預設全關式牆壁的第 0 格與邊緣資料先做初始化，如圖 3.3(b)與圖 3.3(c)。

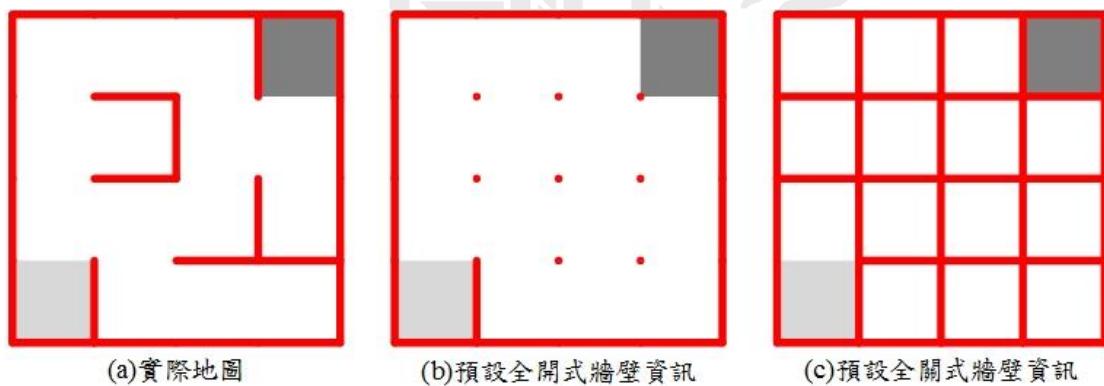


圖 3.3 搜尋前各項資料初始化設定

電腦鼠的搜尋其實也是最短路徑規劃的其中一種，它的步驟就是從電腦鼠的當下位置進行下一格的牆壁探索後，接著使用預設全開式牆壁資訊與 2.8 章的方式填洪水，最後在計算最短路徑到目標區域，每行走一步便執行一次演算法計算，直到電腦鼠到達目標區域，如圖 3.4 至圖 3.5。

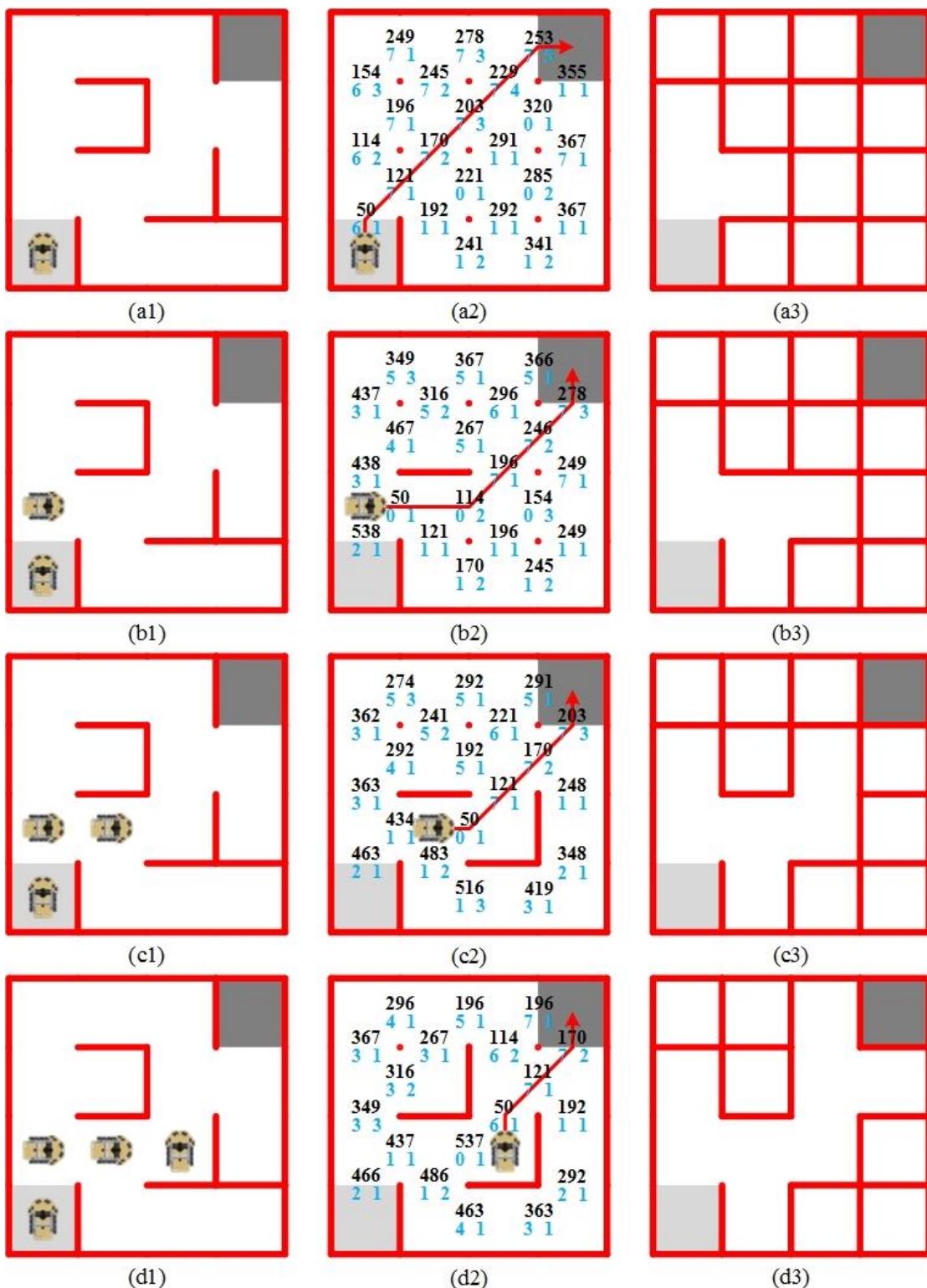


圖 3.4 搜尋步驟(a)~(d)

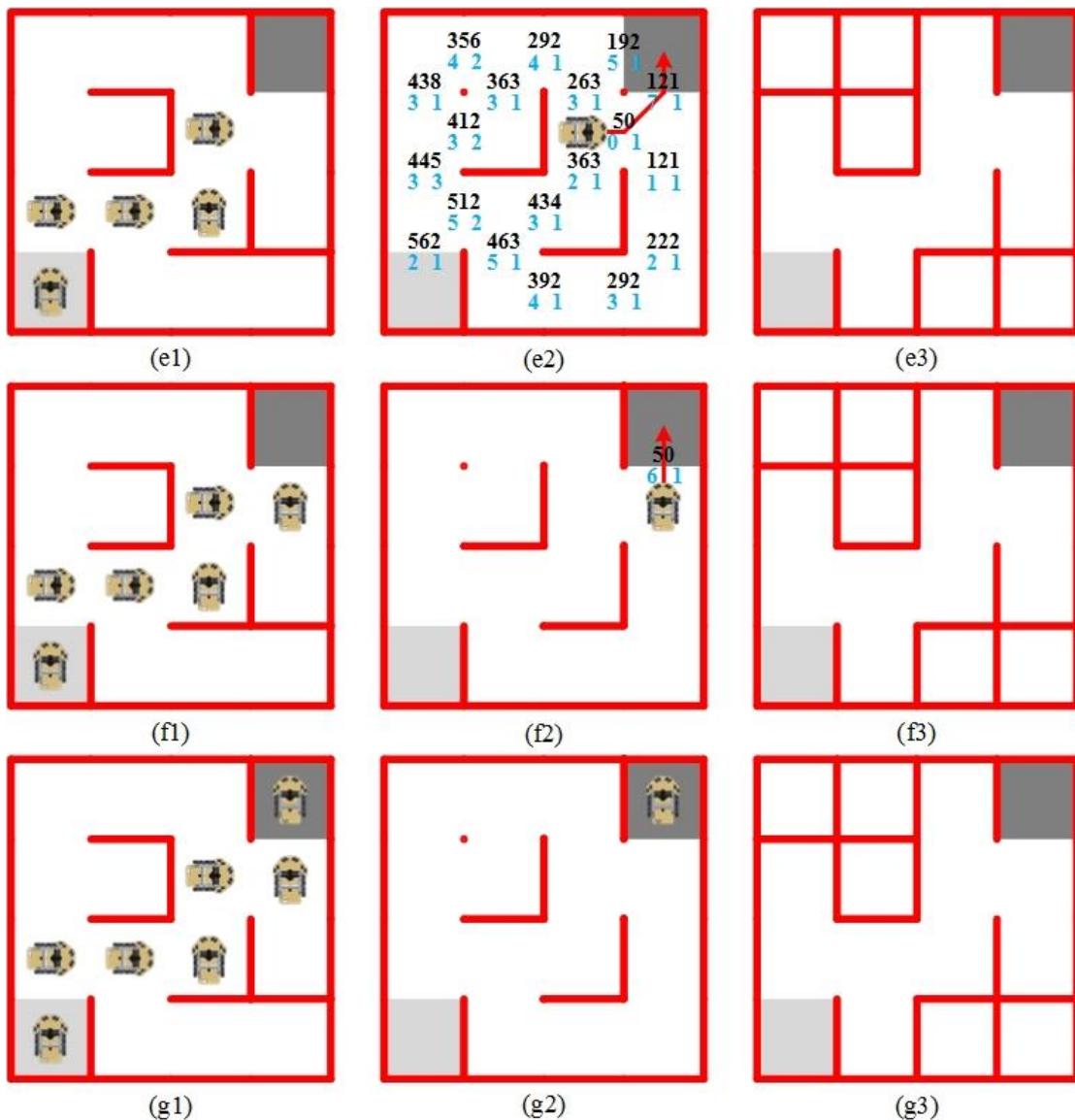


圖 3.5 搜尋步驟(e)~(g)

若在搜尋中發生洪水已擴散完，但是目標區域的洪水值卻還是保持在預設值 65535，則表示電腦鼠走進了死巷子，需要迴轉並繼續搜尋，如圖 3.6。

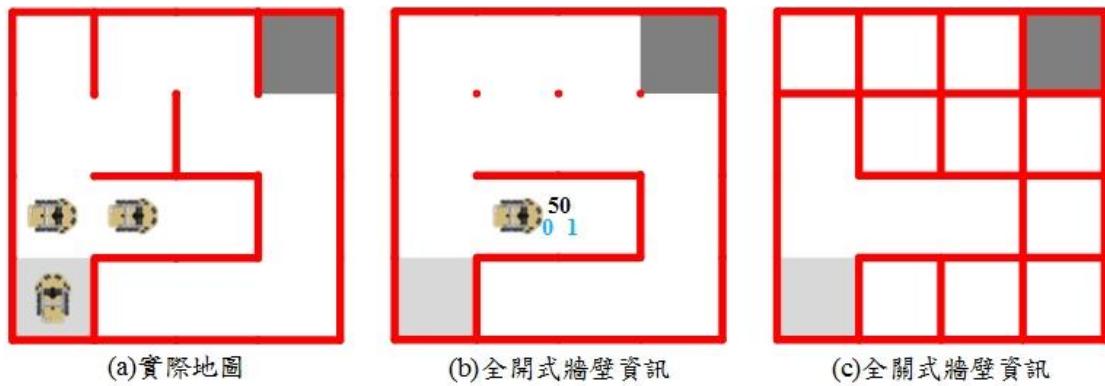


圖 3.6 搜尋進入死巷示意圖

3.3 已知地圖尋找最佳路徑

本論文的搜尋與衝刺的動作編號與動作行為皆使用參考文獻[6]的動作規劃，如表 3.1，差異在於使用的演算法不一樣，因此化簡的過程也不一樣。

表 3.1 各項動作編號

相對動作	編號	
	右轉	左轉
45 度直線接斜線轉彎	0x08	0x06
45 度斜線接直線轉彎	0x18	0x16
90 度直線接直線轉彎(L Turn)	0x21	0x25
90 度斜線接斜線轉彎(V Turn)	0x11	0x15
135 度直線接斜線轉彎(J Turn)	0x02	0x04
135 度斜線接直線轉彎(J Turn)	0x12	0x14
180 度直線接直線轉彎(U Turn)	0x0B	0x0C
90 度弧形轉(搜尋動作)	0x01	0x05
迴轉	0x03	
直走 9cm(古典電腦鼠)	0x1A	
直走 4.5cm(半尺寸電腦鼠)		
直走 18cm(古典電腦鼠)	0x07	
直走 9cm(半尺寸電腦鼠)		
斜線直走 12.7cm(古典電腦鼠)	0x17	
斜線直走 12.7cm(半尺寸電腦鼠)		
從起點出發至柱子處	0x0A	
從柱子處至終點停止	0x00	

需要尋找最佳路徑的狀況可分為 2 種，第 1 種是搜尋的時候，要用預設全開式牆壁資料運算，第 2 種是衝刺的時候，要用預設全關式牆壁資料運算。而最佳路徑計算分為 4 個步驟：

1. 選擇預設全開式牆壁或預設全關式牆壁資料並填洪水值

2. 根據洪水資料尋找絕對方向路徑
3. 將絕對方向路徑化簡成一般動作相對路徑
4. 將一般動作相對路徑化簡成斜線動作相對路徑

3.3.1 尋找絕對方向最佳路徑

以尋找衝刺最佳路徑為例，使用圖 3.5(g)已搜尋到終點的地圖，尋找從起點到終點的最佳路徑，先使用預設全關式牆壁資料從起點處填洪水，如圖 3.7 所示。

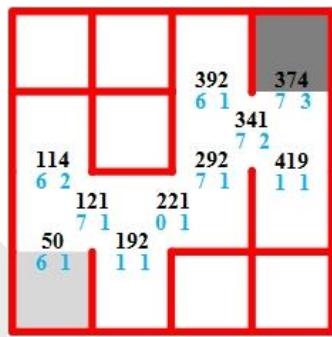


圖 3.7 使用預設全關式牆壁資料填洪水

當填完洪水後，接著就要尋找絕對方向最佳路徑，此步驟要注意尋找路徑時必須從目標區域找回原始區域。以衝刺最佳路徑為例，從終點處使用先前填洪水時的絕對方向反推找回起點，如圖 3.8(b)所示找到的絕對方向最佳路徑順序為 $7 \rightarrow 7 \rightarrow 7 \rightarrow 0 \rightarrow 7 \rightarrow 6$ 。假設從起點處找，當遇到路口時，則會不知道要往哪個方向前進，而無法找到最佳路徑，如圖 3.8(a)。

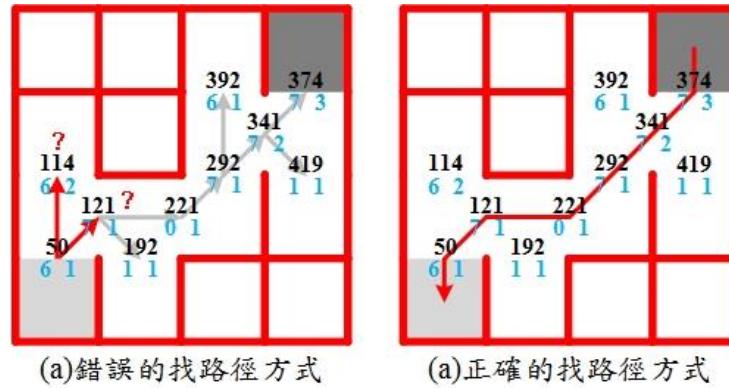


圖 3.8 正確與錯誤找路徑方向示意圖

3.3.2 化簡一般動作相對路徑

在 2.6.3 的水流方向章節中，規範了填洪水時可流的方向，因此可以從這些規範中找出對應的組合，將 2 筆絕對方向組合化簡成 1 個一般動作，一般動作就是指搜尋時用的動作，直走、左右轉與迴轉，如圖 3.9，沒有 45、135、180 度等轉彎動作。

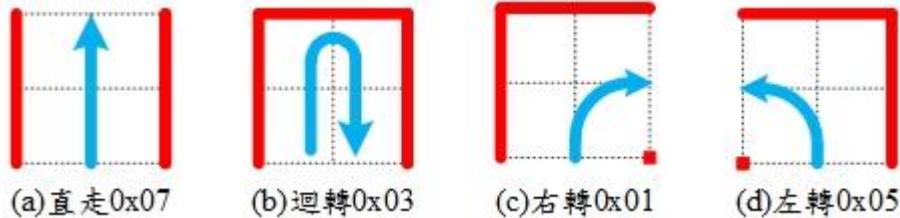


圖 3.9 一般動作示意圖

本論文依 2.6.3 章節中的規則整理了表 3.2 與表 3.3 動作回推表，如圖 3.10 所示，若絕對方向最佳路徑為 $0 \rightarrow 7$ ，查表後可化簡為一般動作相對路徑 5(左轉)。在化簡中有 4 個特殊狀況為絕對方向最佳路徑為 $1 \rightarrow 1$ 、 $3 \rightarrow 3$ 、 $5 \rightarrow 5$ 、 $7 \rightarrow 7$ 時，當化簡時若無考慮上一個動作方向，則當下的動作化簡就有可能發生錯誤，如圖 3.11。因此在表 3.3 中註記的特殊符號(\ast)，其表示動作的判斷無法從當前方

向與下個方向推算出來，必須要從已推算出的上個動作當作依據，才能推出下個動作。如果上個動作是 1(右轉)，則下個動作則是 5(左轉)，反之，上個動作是 5(左轉)，下個動作就是 1(右轉)。

表 3.2 動作回推表 1

當前方向	下個方向	動作	當前方向	下個方向	動作
0(東)	1(東南)	1(右轉)	2(南)	3(西南)	1(右轉)
0(東)	0(東)	7(直走)	2(南)	2(南)	7(直走)
0(東)	7(東北)	5(左轉)	2(南)	1(東南)	5(左轉)
4(西)	5(西北)	1(右轉)	6(北)	7(東北)	1(右轉)
4(西)	4(西)	7(直走)	6(北)	6(北)	7(直走)
4(西)	3(西南)	5(左轉)	6(北)	5(西北)	5(左轉)

表 3.3 動作回推 2

當前方向	下個方向	動作	當前方向	下個方向	動作
1(東南)	1	※	3(西南)	3	※
1(東南)	2	7(直走)	3(西南)	2	7(直走)
1(東南)	3	1(右轉)	3(西南)	1	5(左轉)
1(東南)	7	5(左轉)	3(西南)	5	1(右轉)
1(東南)	0	7(直走)	3(西南)	4	7(直走)
5(西北)	5	※	7(東北)	7(東北)	※
5(西北)	6	7(直走)	7(東北)	6(北)	7(直走)
5(西北)	7	1(右轉)	7(東北)	5(西北)	5(左轉)
5(西北)	3	5(左轉)	7(東北)	1(東南)	1(右轉)
5(西北)	4	7(直走)	7(東北)	0(東)	7(直走)

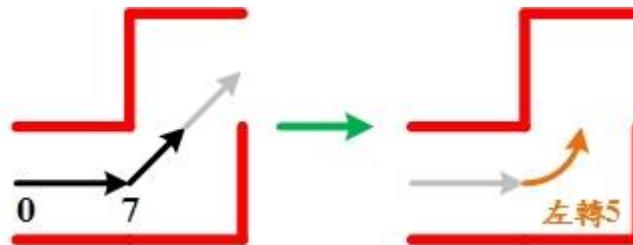


圖 3.10 絕對方向最佳路徑化簡一般動作相對路徑

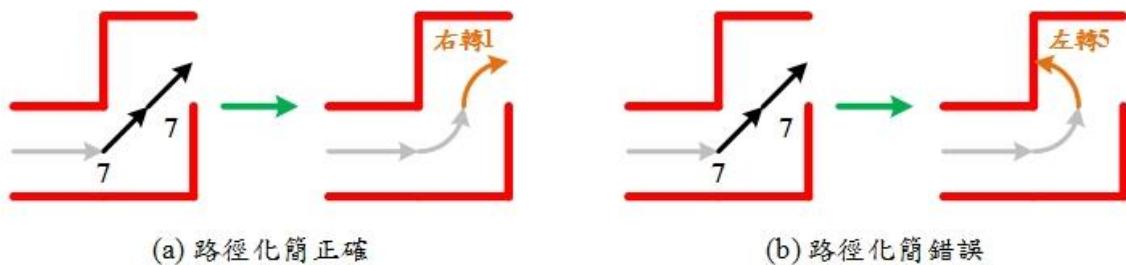


圖 3.11 絕對方向最佳路徑化簡一般動作相對路徑特殊狀況

以圖 3.8(b)為例，尋找衝刺的絕對方向最佳路徑，並將路徑填入陣列中，如圖 3.12，接下來就可開始化簡一般動作相對路徑。

陣列編號	0	1	2	3	4	5	6	7	8
最佳路徑方 向陣列反序	7	7	7	0	7	6			
最佳路徑動 作陣列正序									

圖 3.12 終點方向最佳路徑化簡一般相對動作初始示意圖

首先先將一般動作相對路徑陣列 0 直接填入 7(直走)，如圖 3.13(a)，接下來根據表 3.2 及表 3.3 的資料進行動作化簡，如圖 3.13(b)至圖 3.13(d)，當遇到當前方向與下個方向相同時，需依靠上次化簡完的動作做參考，如圖 3.13(e)與圖 3.13(f)，最後指標到了最後一格，只有當前方向沒有下個方向時化簡結束，在一

般動作相對路徑陣列直接補上 7 與 0 後即結束，如圖 3.13(g)，化簡完成的路徑如圖 3.14。

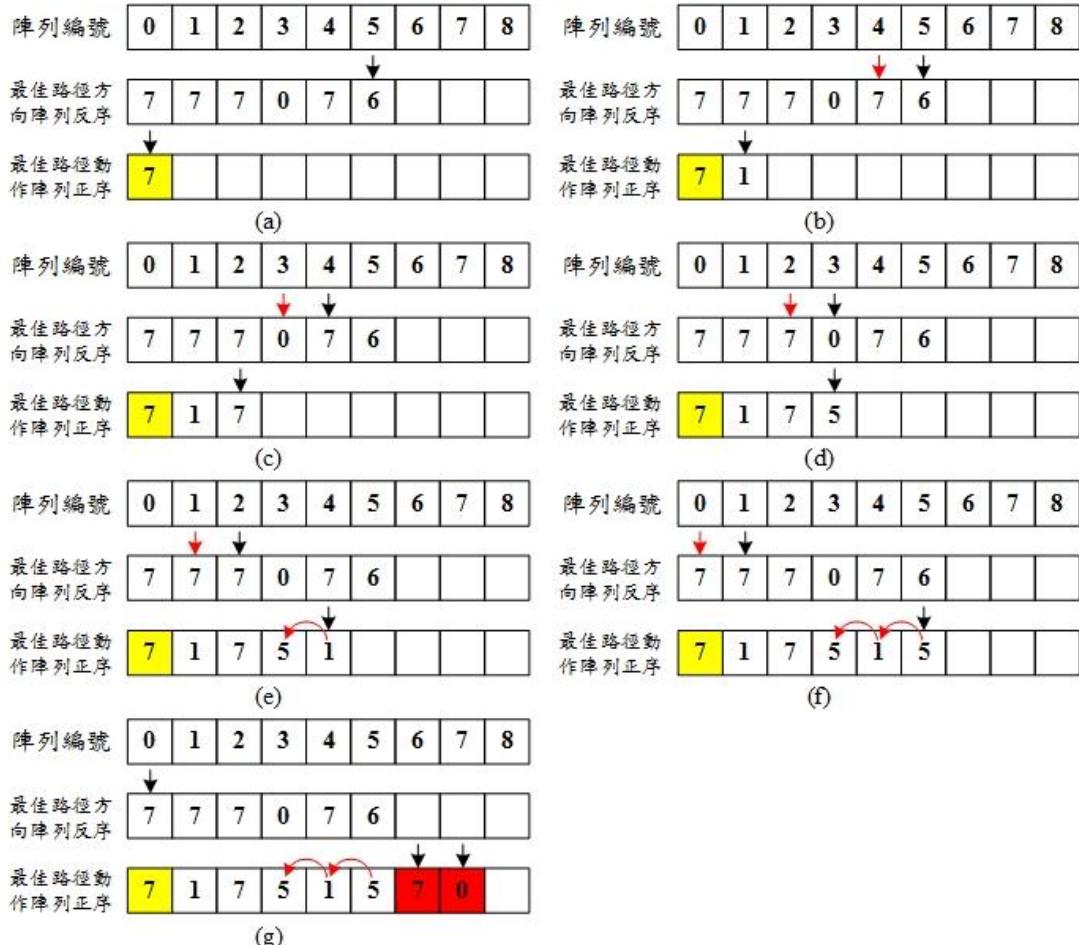


圖 3.13 絕對方向最佳路徑化簡一般動作相對路徑步驟

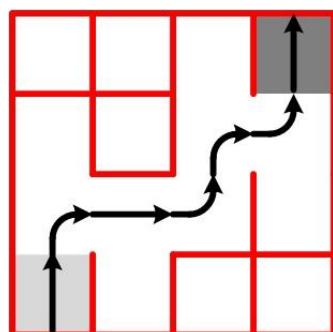


圖 3.14 化簡完成之一般動作相對路徑

若是搜尋時尋找最佳路徑，以圖 3.4(a2)的例子為例，在化簡一般動作時，不用預先填入 7(直走)，且只需化簡一次動作即可知道下一步要走什麼動作，如圖 3.15。

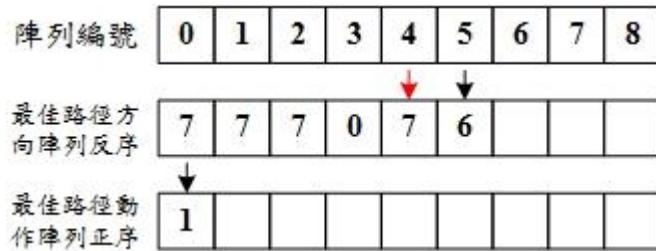


圖 3.15 搜尋時絕對方向路徑化簡一般動作示意圖

3.3.3 化簡斜線動作相對路徑

在得到一般動作相對路徑後接著就可進行斜線動作相對路徑化簡，如圖 3.16 所示為各種斜線的動作及對應編號。每個斜線動作的起始位置與結束位置皆具有規範，起始或結束時的角度為 90 的倍數時，其位置訂在半格處，而起始或結束時的角度為 45 的倍數時，其位置則是訂在柱子處。

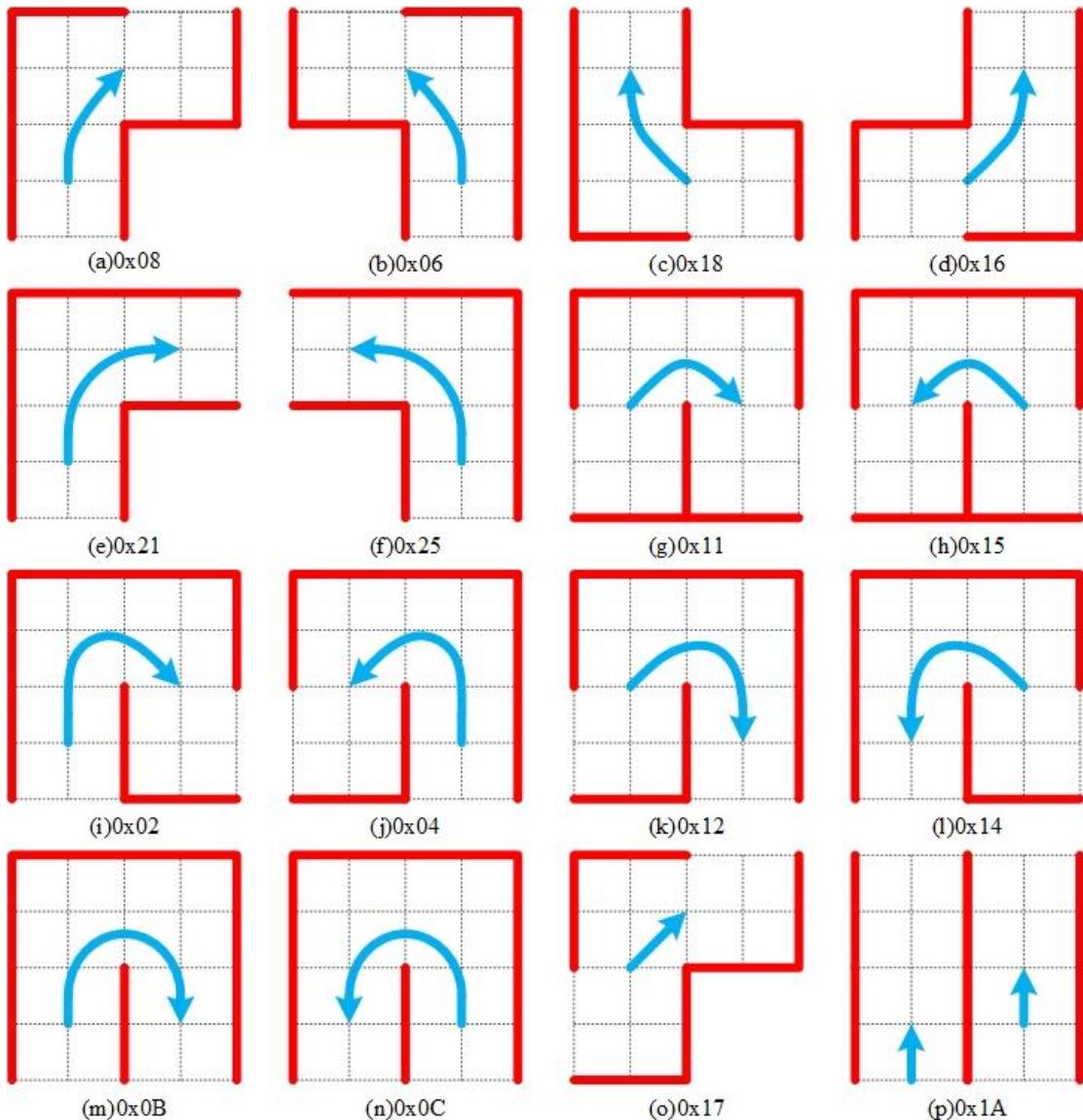


圖 3.16 斜線動作與編號

在化簡斜線動作時，可從圖 3.16 中找到各個動作的對應關係，例如一般動作相對路徑為 7(直走) \rightarrow 1(右轉) \rightarrow 1(右轉) \rightarrow 7(直走)，可化簡成斜線相對動作 0x0B(180 度右彎)。每個斜線動作皆由不同的一般動作組成，不會有一樣的狀況發生，表 3.4 為直線轉斜線與直線轉直線的對應關係，表 3.5 為斜線轉直線與斜線轉斜線的對應關係。

表 3.4 直轉直與直轉斜動作關係表

組合				化簡動作	指標增加
陣列 n	n+1	n+2	n+3		
7	0			0x00	0
7	7			0x07	1
7	1	5		0x08	2
7	5	1		0x06	2
7	1	7		0x21	2
7	5	7		0x25	2
7	1	1	5	0x02	3
7	5	5	1	0x04	3
7	1	1	7	0x0B	3
7	5	5	7	0x0C	3

表 3.5 斜轉斜與斜轉直動作關係表

組合			化簡動作	指標增加
陣列 n	n+1	n+2		
1	5		0x17	1
1	7		0x18	1
1	1	7	0x12	2
1	1	5	0x11	2
5	1		0x17	1
5	7		0x16	1
5	5	7	0x14	2
5	5	1	0x15	2

在化簡動作時從陣列 n 開始，當 n 與 n+1 條件成立時即可化簡動作，若無法成立，則繼續加入條件 n+2 或甚至到 n+3；在動作化簡後，指標必須增加到該組合的最後端，而舊的組合最後端同時也是新的組合的最前端。

以圖 3.14 為例，一般相對路徑 $7 \rightarrow 1 \rightarrow 7 \rightarrow 5 \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 0$ ，經過表 3.4 與表 3.5 的查表與化簡後，得到的斜線相對動作路徑為 $0x21 \rightarrow 0x06 \rightarrow 0x17 \rightarrow 0x16 \rightarrow 0x00$ ，化簡步驟如圖 3.17，化簡後的路徑如圖 3.18。



圖 3.17 一般動作相對路徑化簡斜線動作相對路徑步驟

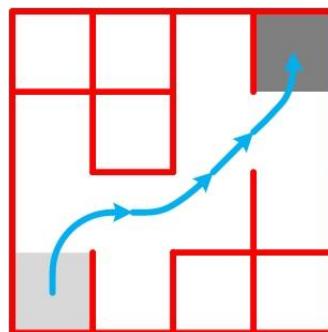


圖 3.18 化簡完成之斜線動作相對路徑

3.3.4 斜線動作相對路徑 2 次處理

在 3.3.3 節找完斜線動作相對路徑後，必須在做一些細節處理，本章節需處理的路徑資料有以下 3 種，在程式中可依序執行：

1. 起始動作：若剛出發時馬上就要執行轉彎，則需轉換成不同的動作編號，因其速度是從 0cm/s 開始，屬於特殊狀況，如表 3.6。若第 1 個動作不需馬上轉彎則是直接填入 0x0A 即可。

表 3.6 起始動作處理表

原始動作	替代動作	新增動作
0x08	0x38	
0x21	0x31	
0x02	0x32	
0x07		0x0A

2. 半格動作：在 3.3.3 節中完成斜線相對動作轉換後，走 1 格直線的起始與結束位置會變成是 A 格的中間到 B 格的中間，這種方式會讓電腦鼠在衝刺時的牆壁校正變得複雜，因此新增半格的目的就是要讓走 1 格直線的起始與結束點，變成是從柱子到柱子，化簡規則如表 3.7。

表 3.7 半格動作處理條件表

組合		新增動作	指標增加
陣列 n	n+1		
7	$\neq 7 \& \neq 0$	0x1A	1
$\neq 7$	7		2

3. 結束動作：此步驟與起始動作一樣，若在轉彎後必須馬上停下來，也是屬於特殊情況，需使用不同動作編號取代，如表 3.8。

表 3.8 結束動作處理表

右轉		左轉	
原始動作	替代動作	原始動作	替代動作
0x18	0x48	0x16	0x46
0x21	0x41	0x25	0x45
0x0B	0x4B	0x0C	0x4C
0x12	0x42	0x14	0x44



以圖 3.18 迷宮地圖為例，斜線相對路徑(HEX)21→06→17→16→0，經過起始動作、半格動作及結束動作處理後，得到完整的路徑為 31→06→17→46→0，處理步驟如圖 3.19，經過處理後，替換了起始與結束動作。



圖 3.19 斜線動作相對路徑 2 次處理步驟

以另一地圖為例，圖 3.20(a)一般動作相對路徑化簡後得到圖 3.20(b)斜線動作相對路徑，在圖 3.20(b)可發現直走是從格子中心走到格子中心，再經過 2 次處理後可得完整的路徑，如圖 3.20(c)，其 2 次處理步驟如圖 3.21。

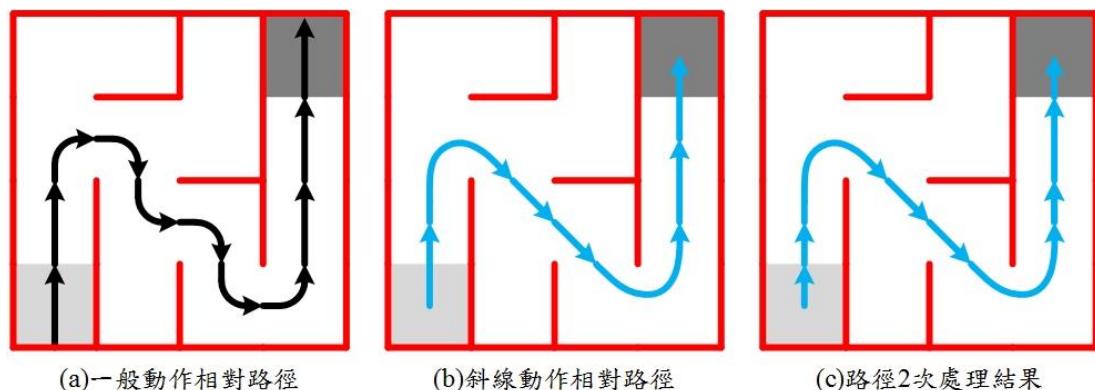


圖 3.20 各次化簡結果比較圖



圖 3.21 斜線動作相對路徑 2 次處理步驟

3.4 洪水錯誤覆蓋問題與解決方案

本論文設計之時間導向斜線迷宮演算法在經過多張地圖測試後，發現邊緣洪水點的設計在執行洪水擴散與覆蓋時，有機會因為覆蓋問題導致後續尋找最佳路徑時發生錯誤。以一實際在測試時發生問題的地圖為例，如圖 3.22 所示，示意圖將從洪水問題欲發生處繼續做洪水擴散。

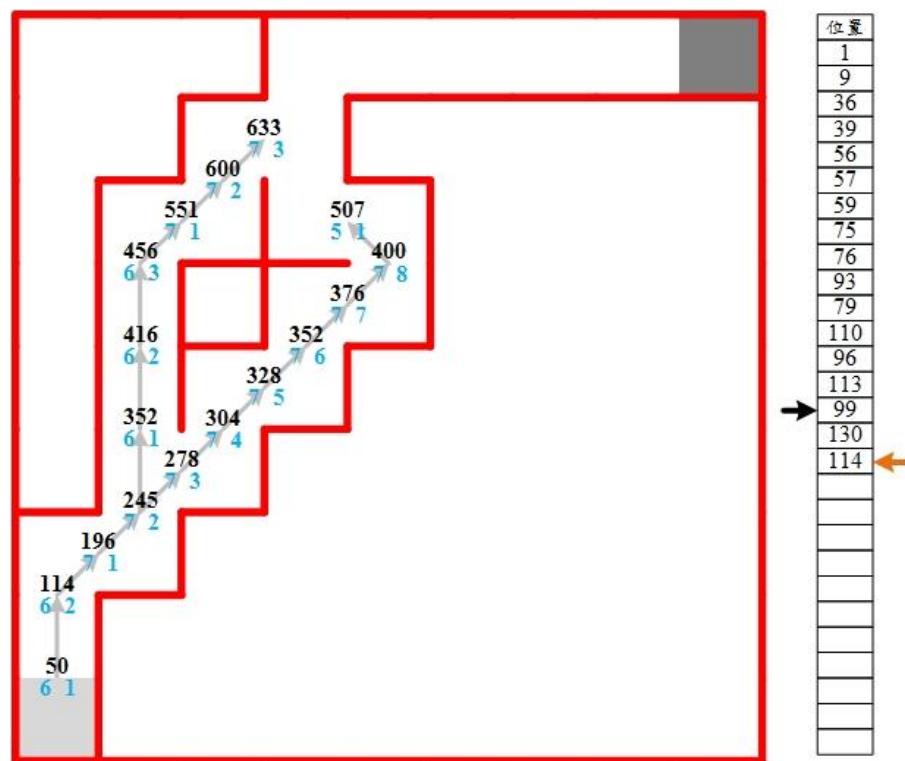


圖 3.22 洪水錯誤覆蓋圖步驟 1

圖 3.23 洪水從位置 130 擴散到位置 133 與 115，兩處洪水值都是預設值，執行覆蓋。

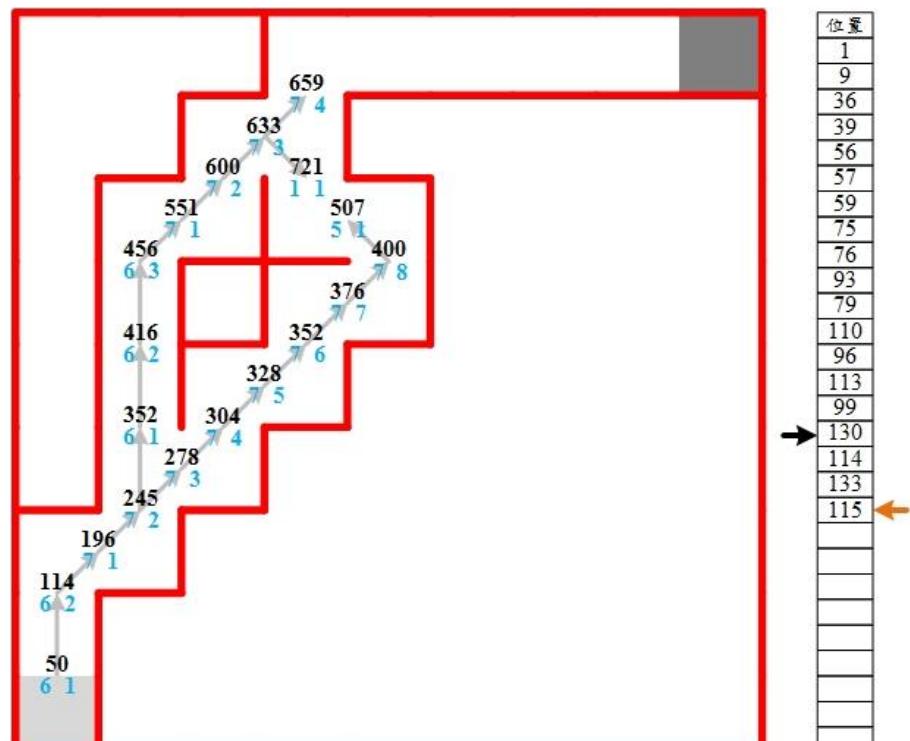


圖 3.23 洪水錯誤覆蓋圖步驟 2

圖 3.24 洪水從位置 114 擴散，當欲擴散至位置 115 時，洪水值 $507+49$ (加速值)+0(減速補償值)=556，小於原本位置的洪水值 721，執行覆蓋。

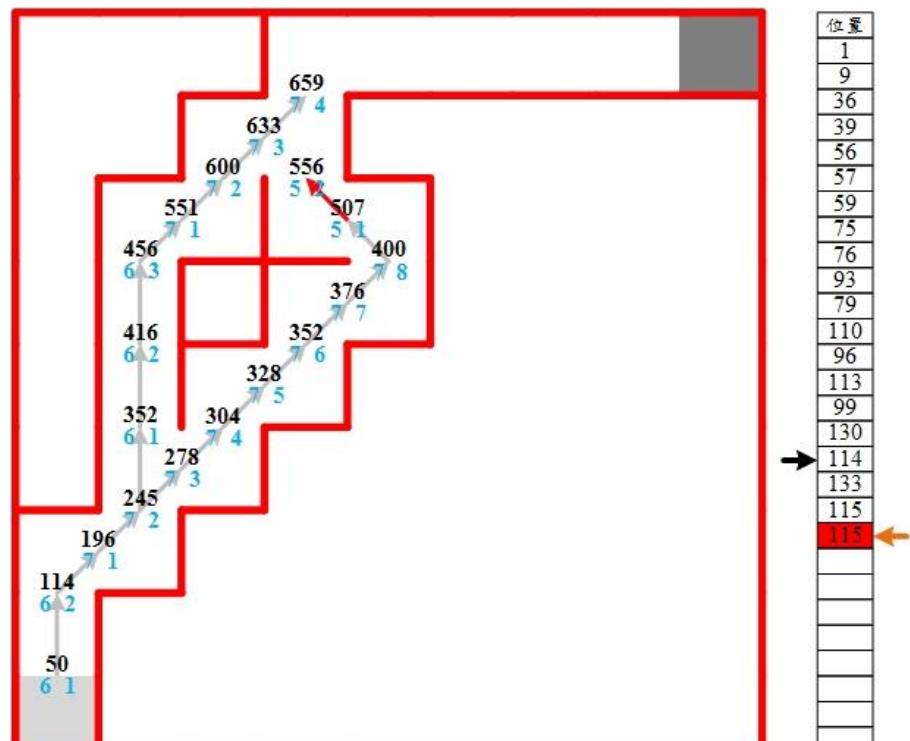


圖 3.24 洪水錯誤覆蓋圖步驟 3

圖 3.25 洪水從位置 133 擴散到位置 150，該處洪水值為預設值，執行覆蓋。

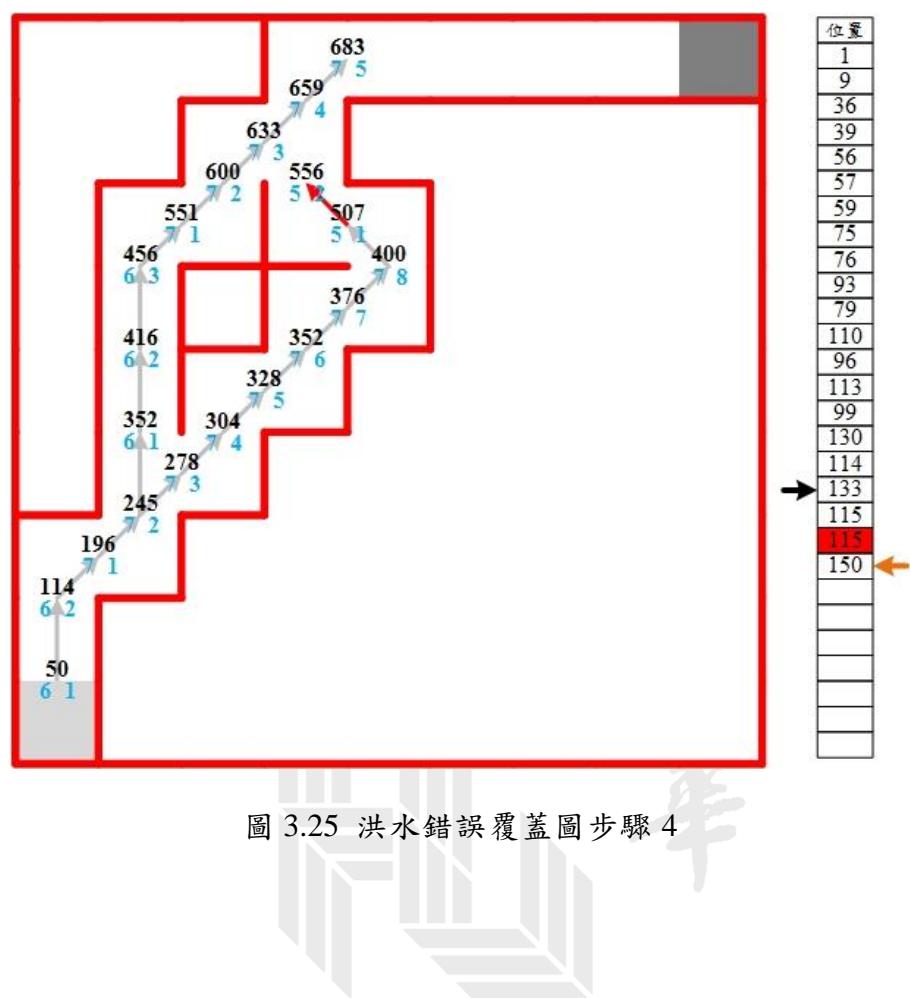


圖 3.25 洪水錯誤覆蓋圖步驟 4

圖 3.26 洪水從位置 115 擴散，當欲擴散至位置 130 時，洪水值 $556+33$ (加速值)+0(減速補償值)=589，小於原本位置的洪水值 633，執行覆蓋。當另一方向欲擴散至位置 133 時，洪水值 $556+100$ (加速值)+7(減速補償值)=663，大於原本位置的洪水值 659，不可覆蓋。此步驟做完洪水擴散後正是問題的發生點。

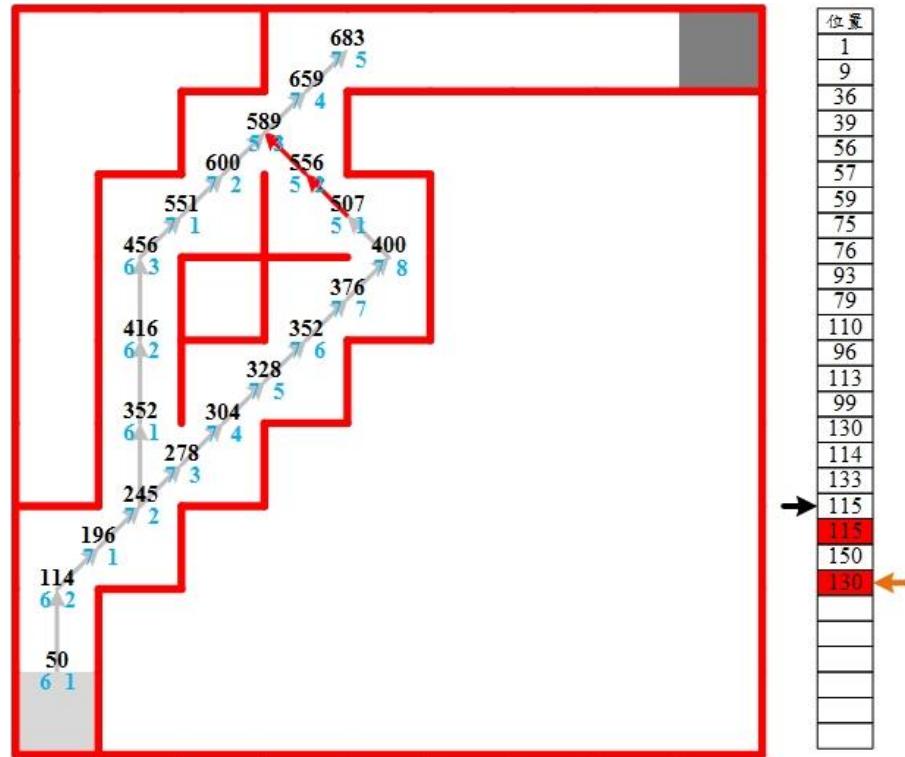


圖 3.26 洪水錯誤覆蓋步驟 5

圖 3.27 當整張地圖做完完整的洪水擴散後，接著進行絕對方向最佳路徑化簡時，會發現圖 3.27 圓圈處中洪水覆蓋時中斷了其他路徑，導致路徑尋找時改變了方向，且重複方向次數也連接不起來，從 5 找回 4 但還沒找回 3 時就轉彎了，雖然後續成功尋找完絕對方向最佳路徑，但在化簡成一般相對路徑時，其路徑為 $7 \rightarrow 7 \rightarrow 1 \rightarrow 5 \rightarrow 1 \rightarrow 5 \rightarrow 1 \rightarrow 5 \rightarrow 5 \rightarrow 1 \rightarrow 5 \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 0$ ，化簡結果是錯誤的。

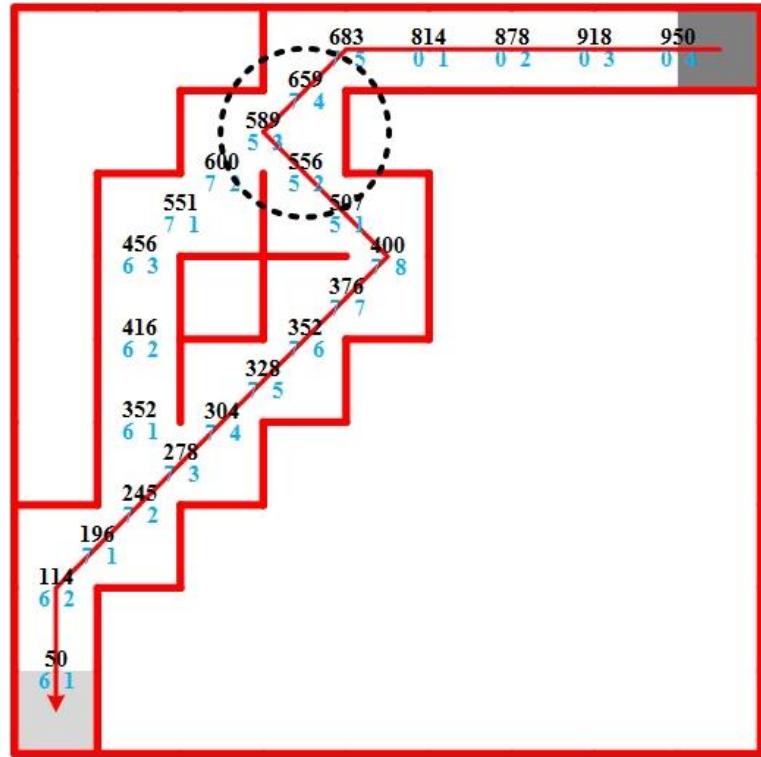


圖 3.27 洪水錯誤覆蓋之迷宮地圖與尋找絕對方向最佳路徑結果

為解決錯誤覆蓋的問題，本論文提出一項解決方案，共 2 道步驟。第 1 步，在洪水要進行擴散前，首先先查看欲擴散之鄰格中的重複方向次數，並以數值大的為優先擴散洪水，如圖 3.28。



圖 3.28 由重複方向次數決定洪水擴散順序示意圖

第 2 步，依順序做洪水擴散，其中順序 1 可直接做洪水擴散，但順序 2 與順序 3 需加入特殊條件。

當要往順序 2 處擴散洪水時，若順序 2 處與順序 1 處的方向相同，強制不可往順序 2 處做洪水擴散；當要往順序 3 擴散洪水時，若順序 3 與順序 2 方向相同或順序 3 與順序 1 方向相同時，強制不可往順序 3 做洪水擴散，如圖 3.29 條件成立不可做洪水擴散，圖 3.30 條件不成立可做洪水擴散。在每次做洪水擴散前，只要以這 2 道步驟先行檢驗，就可避免錯誤覆蓋的問題發生。

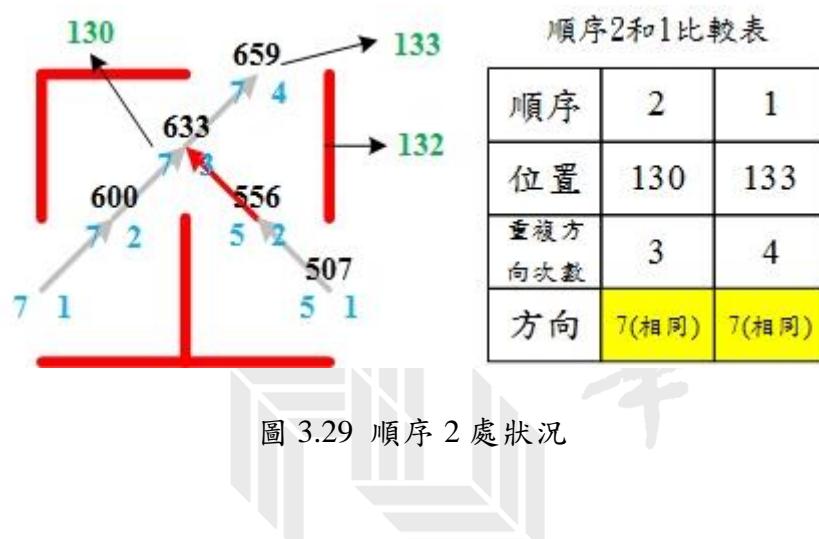


圖 3.29 順序 2 處狀況



圖 3.30 順序 3 處狀況

第四章 系統架構與機構設計

4.1 半尺寸電腦鼠系統架構圖

圖 4.1 為本論文設計之系統架構圖，整台半尺寸電腦鼠使用(1)1 顆 ZIPPY 70mAh 的鋰聚合物電池供應，其電壓範圍從 3.7V 至 4.2V，部份元件經由穩壓器降壓至 3.3V 使用，部份元件則從鋰電池直接供應之；(2)瑞薩(Renesas)公司的 RX62T 晶片做為整臺車的微控制器；(3)4 個 OP265AD 紅外線發射器，並與 4 個 LTR-4206 紅外線接收器搭配使用，做牆壁探索與車身偏移修正；(4)LY3200ALH 陀螺儀感測轉彎角速度，減少轉彎時輪胎打滑造成的誤差；(5)MTL 公司的編碼器讀取 AB 相訊號偵測輪子轉動圈數並推算移動距離；(6)DRV8836 驅動 2 個直流空心背馬達，並使用脈波寬度調變(Pulse Width Modulation,PWM)控制馬達；(7)2 顆 RGB 燈做為人機介面，進行選單選擇與簡易除錯；(8)瑞薩(Renesas)公司開發的 E1 燒錄器，進行程式燒錄與線上 Debug 功能。

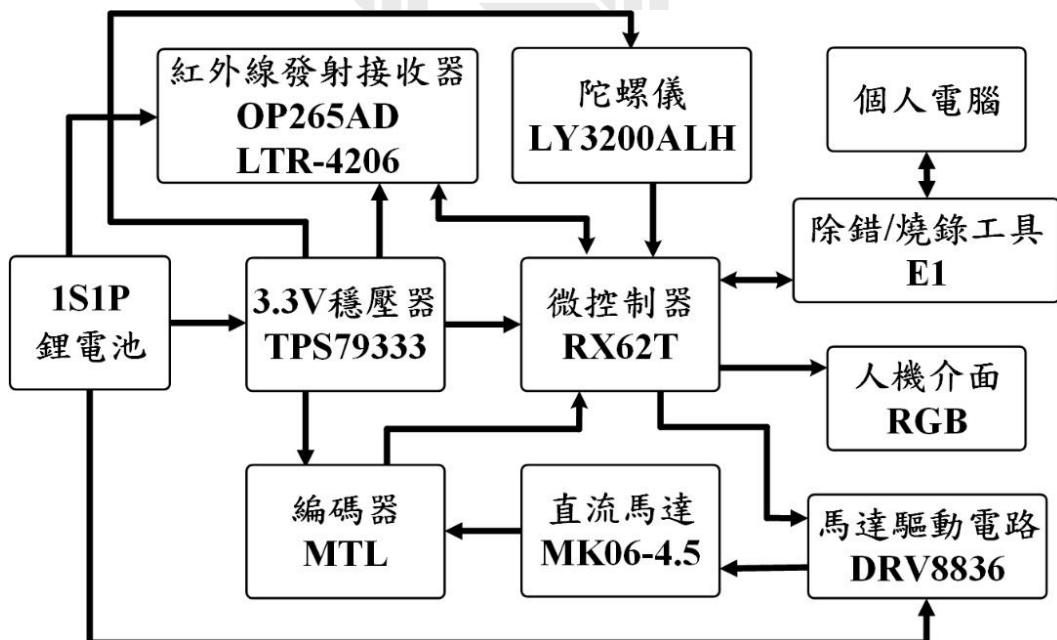


圖 4.1 半尺寸電腦鼠系統架構圖

4.1.1 微控制器

本論文使用瑞薩(Renesas)公司 32 位元 RX62T[7]晶片做為整台電腦鼠的微處理器，它具有最高工作頻率 100Mhz、單精度浮點運算單元、程式快閃記憶體(Program Flash)最大 256KB、SRAM 最大 16KB、Data Flash 最大 32KB、12bit A/D 轉換器 8 通道，16bit PWM 計時器 8 通道等等工作能力，符合本論文用於電腦鼠之需求，如圖 4.2 所示。

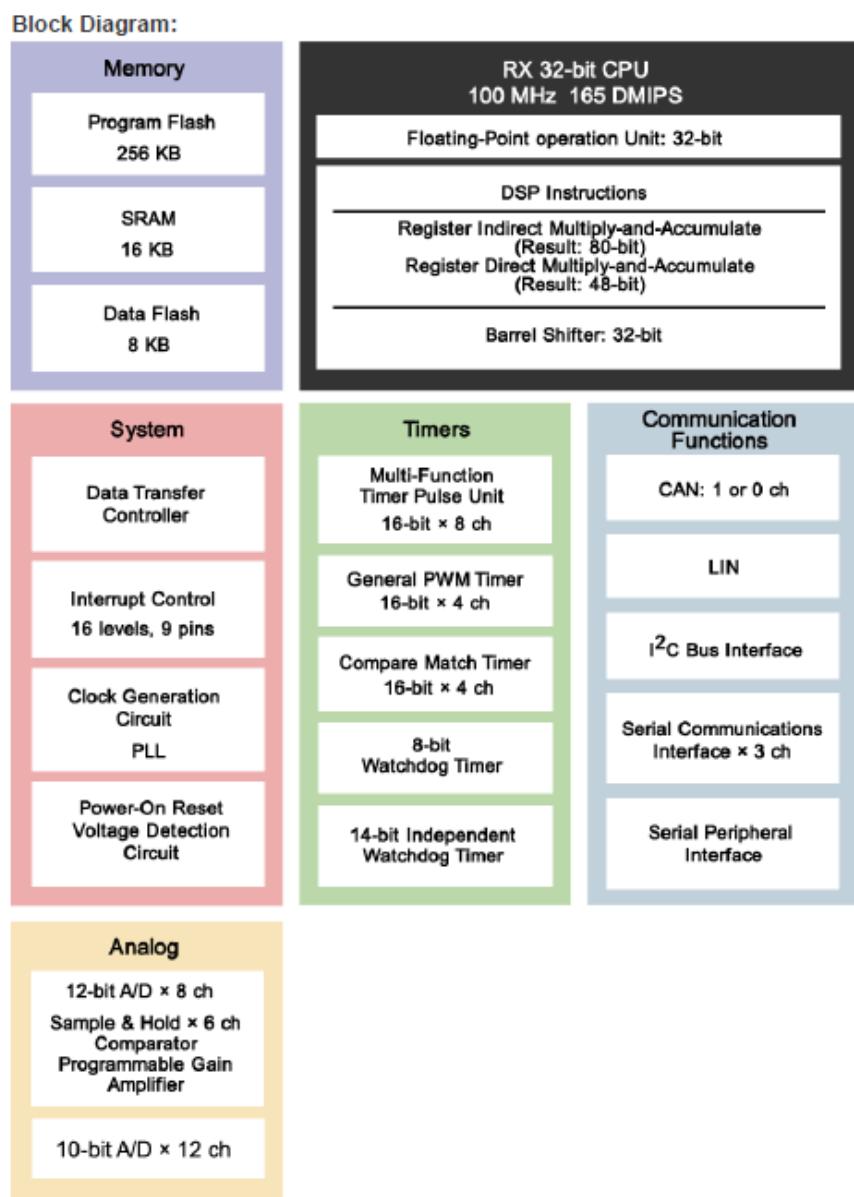


圖 4.2 RX62T 系列晶片功能區塊圖

在電源消耗方面，RX62T 系列晶片在全速運轉情況下，最高耗費電流為 70mA，如圖 4.3 所示。本論文估計若微控制器平均耗費 70mA，紅外線 4 組平均耗費 15mA，2 組編碼器平均耗費 60mA，2 組馬達平均耗費 200mA，總共 345mA，若持續放電情況下，可使用約 12 分鐘，超過比賽限制時間 10 分鐘，符合本論文之需求。

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Supply current ^{*1}	In operation	Max. ^{*2}	I_{CC}^{*3}	-	-	70		mA	ICLK = 100 MHz PCLK = 50 MHz
		Normal ^{*4}		-	35	-			
		Increased by BGO operation ^{*5}		-	15	-			
	Sleep				22	60			
	All-module-clock-stop mode ^{*6}				14	28			
	Standby mode	Software standby mode		-	0.10	3	mA		
		Deep software standby mode		-	20	60	μA		

圖 4.3 RX62T 消耗電流

4.1.2 紅外線發射與接收

本論文使用紅外線當做電腦鼠的眼睛，先由紅外線發射器發射紅外線後，再由紅外線接收器接收反射回來紅外光，最後使用微控制器中的 A/D 轉換器轉換數值並使用。紅外線發射與接收器在電腦鼠中的功能包括搜尋時判斷有無牆壁、判斷離牆壁多遠並修正車身偏移，判斷牆壁邊緣特徵並修正錯誤之行走距離等功能。

本論文選用的紅外線發射器型號為 OP265AD[8]，其順向偏壓 1.8V，波長 850nm，發射角度±18°，功耗 100mW，請參考圖 4.4、圖 4.5 與圖 4.6。

Absolute Maximum Ratings ($T_A=25^\circ\text{C}$ unless otherwise noted)

Storage and Operating Temperature Range	-40° C to +100° C
Reverse Voltage	2.0 V
Continuous Forward Current	50 mA
Peak Forward Current (1 μs pulse width, 300 pps)	3.0 A
Lead Soldering Temperature [1/16 inch (1.6 mm) from case for 5 seconds with soldering iron]	260° C ⁽¹⁾
Power Dissipation	100 mW ⁽²⁾

圖 4.4 OP265AD 最大額定值

Electrical Characteristics ($T_A = 25^\circ\text{C}$ unless otherwise noted)

SYMBOL	PARAMETER	MIN	TYP	MAX	UNITS	TEST CONDITIONS
Input Diode						
E_E (APT)	Apertured Radiant Incidence OP265FAA OP265FAB OP265FAC OP265FAD	5.50 7.50 11.50 15.50	- - - -	12.5 16.5 - -	mW/cm ²	$I_F = 20 \text{ mA}$ Aperture = 0.081" diameter Distance = 0.590" from seating surface to aperture surface
V_F	Forward Voltage	-	-	1.80	V	$I_F = 20 \text{ mA}$
I_R	Reverse Current	-	10	-	μA	$V_R = 10 \text{ V}$
λ_P	Wavelength at Peak Emission	-	850	-	nm	$I_F = 10 \text{ mA}$
$\Delta\lambda_P/\Delta T$	Spectral Shift with Temperature	-	± 0.18	-	nm/°C	$I_F = \text{Constant}$
θ_{HP}	Emission Angle at Half Power Points	-	18	-	Degree	$I_F = 20 \text{ mA}$
t_r	Output Rise Time	-	10	-	ns	$I_{F(PK)} = 100 \text{ mA}, PW = 10 \mu\text{s}, D.C. = 10.0\%$
t_f	Output Fall Time	-	10	-	ns	

圖 4.5 OP265AD 電子特性

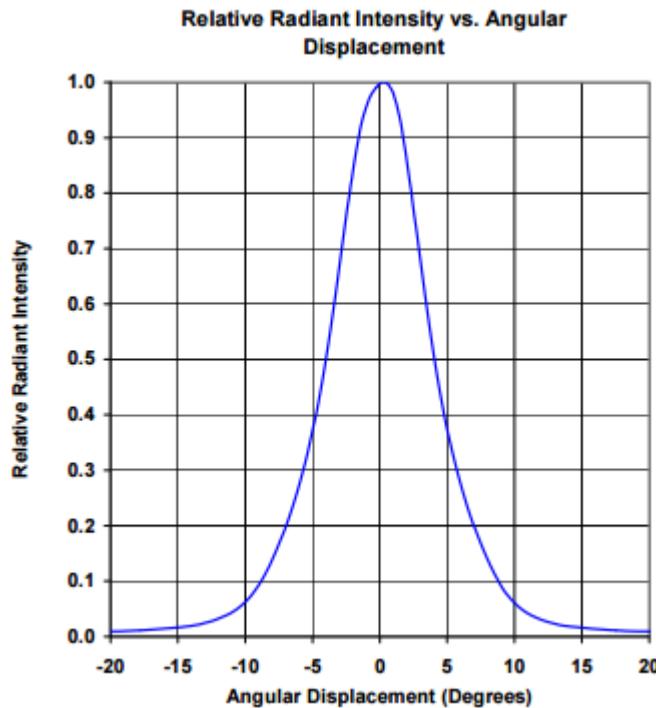


圖 4.6 OP265AD 發射角度圖

本論文選用的紅外線接收器型號為 LTR-4206[9]，最敏感波長 940nm，上升時間 10us，下降時間 10us，接收角度 $\pm 10^\circ$ 、功耗 100mW，請參考圖 4.7、圖 4.8 與圖 4.9。

ABSOLUTE MAXIMUM RATINGS AT $T_A=25^\circ\text{C}$

PARAMETER	MAXIMUM RATING	UNIT
Power Dissipation	100	mW
Collector-Emitter Voltage	30	V
Emitter-Collector Voltage	5	V
Operating Temperature Range	-40°C to $+85^\circ\text{C}$	
Storage Temperature Range	-55°C to $+100^\circ\text{C}$	
Lead Soldering Temperature [1.6mm(.063") From Body]	260°C for 5 Seconds	

圖 4.7 LTR-4206 最大額定值

ELECTRICAL / OPTICAL CHARACTERISTICS AT $T_A=25^\circ\text{C}$

PARAMETER	SYMBOL	MIN.	TYP.	MAX.	UNIT	TEST CONDITION	BIN NO.
Collector-Emitter Breakdown Voltage	$V_{(\text{BR})\text{CEO}}$	30			V	$I_C = 1\text{mA}$ $E_e = 0\text{mW/cm}^2$	
Emitter-Collector Breakdown Voltage	$V_{(\text{BR})\text{ECO}}$	5			V	$I_E = 100 \mu\text{A}$ $E_e = 0\text{mW/cm}^2$	
Collector Emitter Saturation Voltage	$V_{\text{CE}(\text{SAT})}$			0.4	V	$I_C = 0.5\text{mA}$ $E_e = 1\text{mW/cm}^2$	
Rise Time	Tr		10		μs	$V_{\text{CC}} = 5\text{V}$ $I_C = 1\text{mA}$	
Fall Time	Tf		10		μs	$R_L = 1\text{K}\Omega$	
Collector Dark Current	I_{CEO}			100	nA	$V_{\text{CE}} = 10\text{V}$ $E_e = 0\text{mW/cm}^2$	
On State Collector Current	$I_{\text{C}(\text{ON})}$	0.8		2.4	mA	$V_{\text{CE}} = 5\text{V}$ $E_e = 1\text{mW/cm}^2$ $\lambda = 940\text{nm}$	BIN C
		1.6		4.8			BIN D
		3.2		9.6			BIN E
		6.4					BIN F

圖 4.8 LTR-4206 電子特性

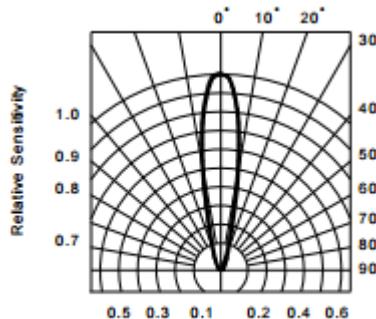


FIG.5 SENSITIVITY DIAGRAM

圖 4.9 LTR-4206 發射角度圖

本論文使用 2SC3325[10]電晶體當做開關控制紅外線發射，並藉由 R1 電阻調整 I_B 電流(4.1)，電路如圖 4.10 所示。

$$I_B = \frac{V_B - V_{BE}}{R_1} = \frac{3.3 - 0.8}{4.3K\Omega} = 0.581mA \quad (4.1)$$

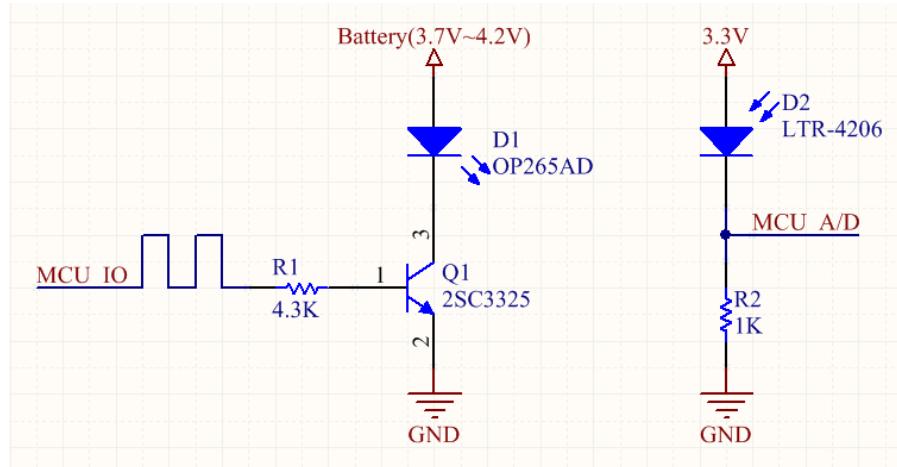


圖 4.10 紅外線發射與接收電路圖

假設鋰電池輸出為 3.7V，紅外線發射器 OP265AD 順向偏壓 1.8V，可計算出 V_{CE} 電壓。

$$V_{CE} = 3.7 - 1.8 = 1.9V \quad (4.2)$$

由圖 4.11 可用 I_B 電流與 V_{CE} 查出 I_C 位於飽和區，操作電流約 100mA。

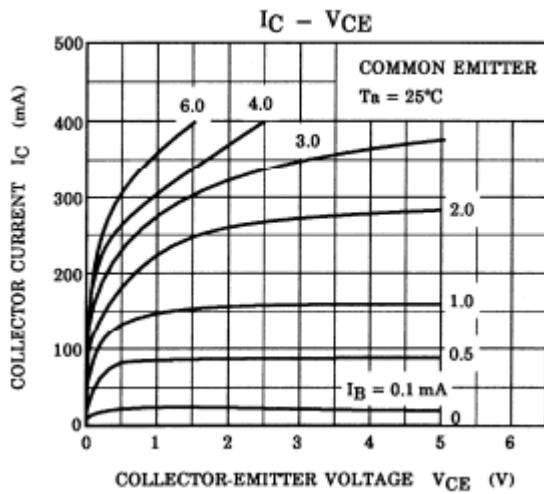


圖 4.11 2SC3325 規格特性曲線圖

4.1.3 馬達驅動電路

本論文使用 DRV8836[11]雙 H 橋驅動晶片驅動馬達，它具有單 H 橋最大 1.5A 的驅動能力，雙 H 橋最大共 3A，操作電壓 2V~7V，如圖 4.12。

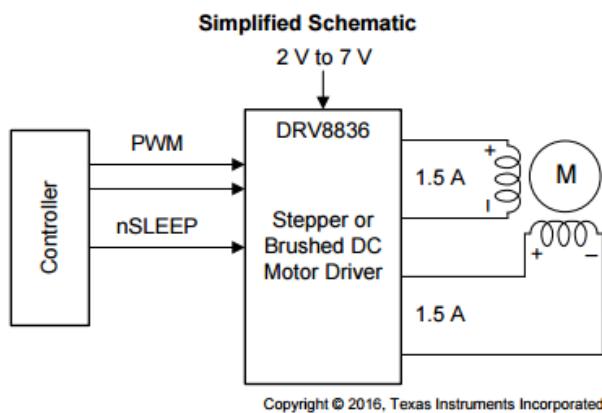


圖 4.12 DRV8836 簡易原理圖

使用上需要用到 1 組微控制器的 I/O 腳控制相位，與 1 組單極性 PWM 控制即可，如圖 4.13。

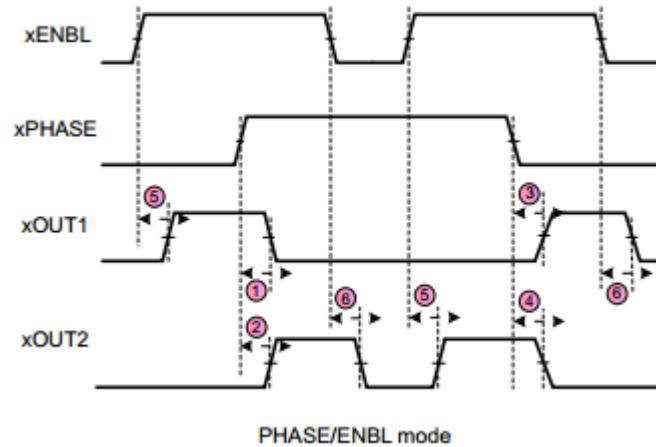


圖 4.13 控制輸出波形圖

4.1.4 編碼器

在半尺寸電腦鼠中為了偵測電腦鼠移動了多少距離，本論文使用日本 MTL 公司的 MES-6 系列編碼器[12]，回授電腦鼠的移動脈波數，該編碼器轉 1 圈可得 2000 脍波數。該編碼器工作電壓 5V，消耗電流 30mA，重量 5g，直徑 7.5mm，長 10.5mm，適合用在半尺寸電腦鼠上，如圖 4.14。

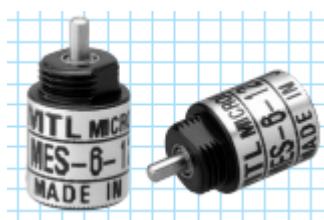


圖 4.14 MTL MES-6 系列編碼器

本論文實測 MTL 編碼器發現工作電壓在 3.3V 下亦可正常使用，依照使用手冊上的建議在 AB 相加上上拉電阻，即可正常使用，如圖 4.15。本論文之半尺寸電腦鼠輪子轉 1 圈可得 24000 脈波數(4.3)。

$$2000(\text{編碼器脈波數}) \times 4(\text{AB相解析}) \times 3(\text{齒輪比}) = 24000(\text{脈波數/圈}) \quad (4.3)$$

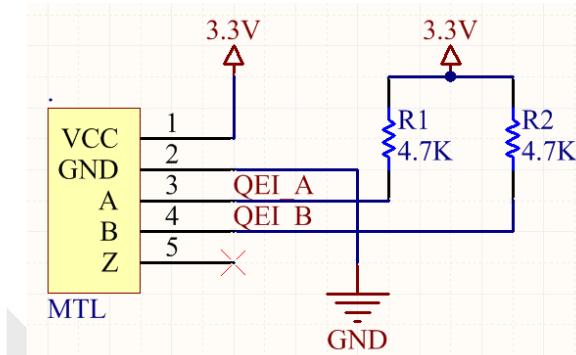


圖 4.15 編碼器電路圖

4.1.5 陀螺儀

電腦鼠在高速行走時，若是輪胎上沾了太多灰塵，在高速轉彎時會因為離心力過大導致摩擦力不足造成側滑現象，此時若是使用編碼器回授轉彎資訊，將有機會得到錯誤的回授資料。為解決此問題，本論文使用意法半導體 (STMicroelectronics) 製造的 LY3200ALH[13] 陀螺儀，其工作電壓 2.7V~3.6V，偵測範圍±2000°，靈敏度為 0.67mV/dps(degree per second)，請參考圖 4.16 與圖 4.17。

Table 3. Mechanical characteristics

Symbol	Parameter	Test condition	Min.	Typ. ⁽¹⁾	Max.	Unit
FS	Measurement range			±2000		dps
So	Sensitivity ⁽²⁾			0.67		mV/dps
SoDr	Sensitivity change vs. temperature	Delta from 25°C		0.01		%/°C
Voff	Zero-rate level ⁽³⁾			1.5		V
OffDr	Zero-rate level change vs. temperature	Delta from 25°C		0.09		dps/°C
NL	Non linearity	Best fit straight line		±1		% FS
BW	Bandwidth ⁽³⁾			140		Hz
Vst	Self-test output change			400 ⁽⁴⁾		mV
Rn	Rate noise density			0.074		dps/√Hz
Top	Operating temperature range		-40		+85	°C

1. Typical specifications are not guaranteed.
2. Sensitivity and zero-rate offset are not ratiometric to supply voltage.
3. The product is capable of measuring angular rates extending from DC to the selected BW.
4. Self test typical absolute value.

圖 4.16 LY3200ALH 陀螺儀機械特性



圖 4.17 LY3200ALH 陀螺儀偵測角速度方向圖

本論文使用微控制器中的 12 位元 A/D 轉換器讀取陀螺儀的輸出變化，微控制器工作電壓為 3.3V，可將偵測到的 0V~3.3V 電壓值轉換為 0~4095 的數值，其兩者間的關係為：

$$\frac{3.3V}{4096} = 0.805 \left(\frac{mV}{ADC} \right) \quad (4.4)$$

將(4.4)除上陀螺儀的靈敏度 0.67mV/dps ，可求出 ADC 與角速度變化關係(4.5)。

$$0.805\left(\frac{\text{mV}}{\text{ADC}}\right) \times \frac{1}{0.67}\left(\frac{\text{dps}}{\text{mV}}\right) = 1.201\left(\frac{\text{dps}}{\text{ADC}}\right) \quad (4.5)$$

4.2 半尺寸電腦鼠機構設計

在機構設計上，本論文自行設計馬達與編碼器的支撐架。由於半尺寸電腦鼠的車身空間有限，因此本論文將馬達與編碼器設計為前後擺放的方式，並使用不同齒數的正齒輪互相帶動馬達、輪胎與編碼器。小齒輪中心至大齒輪中心的距離可由方程式(4.6)計算出來。

$$\left(\frac{(\text{大齒輪} + \text{小齒輪})}{2} \times \text{齒輪模數}\right) + \text{間隙} = \text{齒輪中心距離} \quad (4.6)$$

本論文選用馬達小齒輪 8 齒、輪胎大齒輪 36 齒，模數 0.3，間隙 0.08mm，經方程式(4.7)計算後得到距離為 6.68mm；編碼器小齒輪 12 齒、輪胎大齒輪 36 齒，模數 0.3，間隙 0.08mm，經方程式(4.8)計算後得到距離為 7.28mm。本論文使用 Solidworks 軟體繪製機構並加工，如圖 4.18 圖 4.19。

$$\left(\frac{(36+8)}{2} \times 0.3\right) + 0.08 = 6.68\text{mm} \quad (4.7)$$

$$\left(\frac{(36+12)}{2} \times 0.3\right) + 0.08 = 7.28\text{mm} \quad (4.8)$$

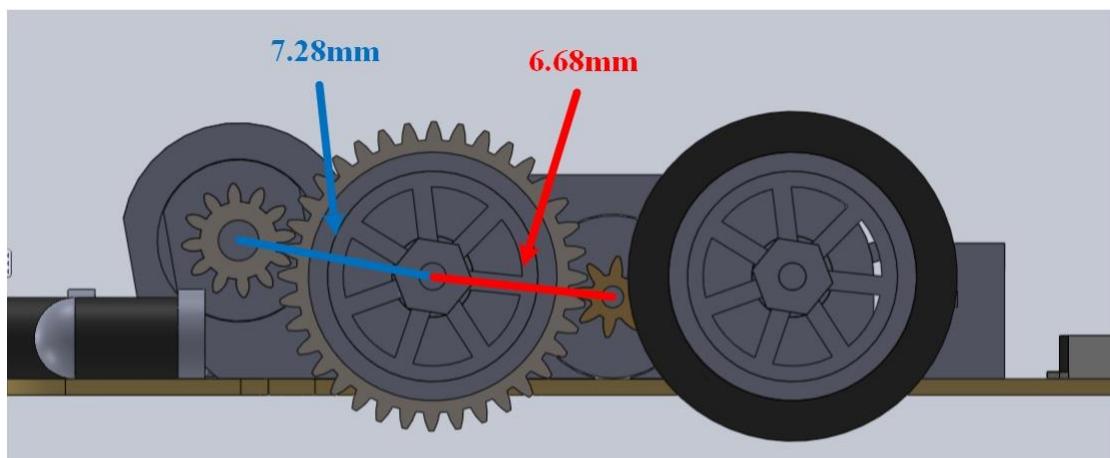


圖 4.18 馬達架上齒輪與齒輪間距離示意圖

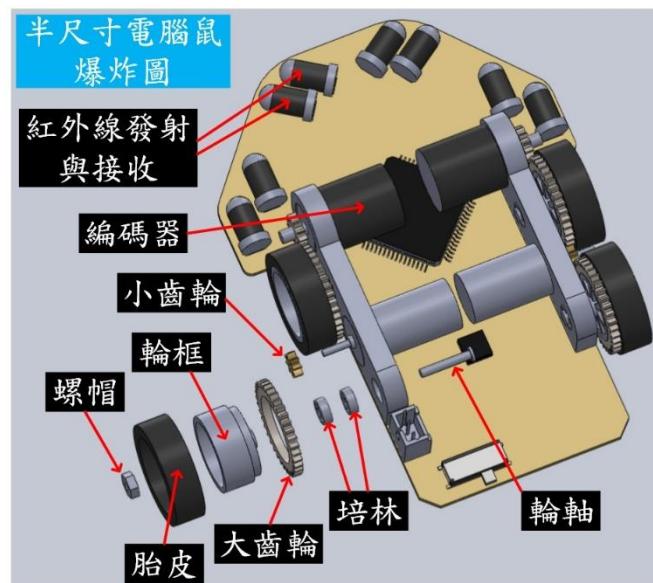


圖 4.19 半尺寸電腦鼠爆炸圖

4.3 古典電腦鼠風扇設計

以往之古典電腦鼠，在迷宮中高速轉彎時會因離心力過大導致摩擦力不足，進而影響電腦鼠過彎之穩定性，而無法發揮馬達的最大效能。本論文為解決此問題，設計了類似抽真空的風扇機構，其目的是將電腦鼠底盤的空氣抽離，讓車底產生負壓並吸在地板上，並預期電腦鼠在轉彎時可提升正向力，並增加摩擦力，讓電腦鼠可以克服離心力使用更快的速度來過彎。

本論文使用 Maxon 公司的 RE8 馬達[14]做為風扇的驅動馬達，其額定工作電壓為 2.4V，空載轉速 13900rpm，堵轉扭力 0.925mNm，參考圖 4.20。本論文使用 Solidworks 軟體繪製機構並加工製作馬達支撐架與風扇，如圖 4.21 與圖 4.22。



額定電壓值	
額定電壓	2.4 V
空轉轉速	13900 rpm
空載電流	19.2 mA
額定轉速	4320 rpm
額定扭力 (最大連續扭力)	0.63 mNm
額定電流 (最大連續電流)	0.412 A
堵轉扭力	0.925 mNm
啟動電流	0.581 A
最大效率	67 %

圖 4.20 RE8 馬達規格表

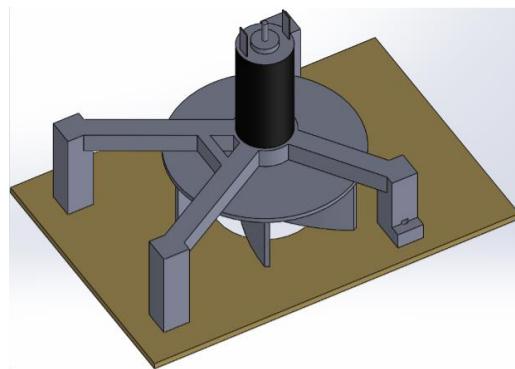


圖 4.21 Solidworks 模擬馬達支撐架 3D 圖

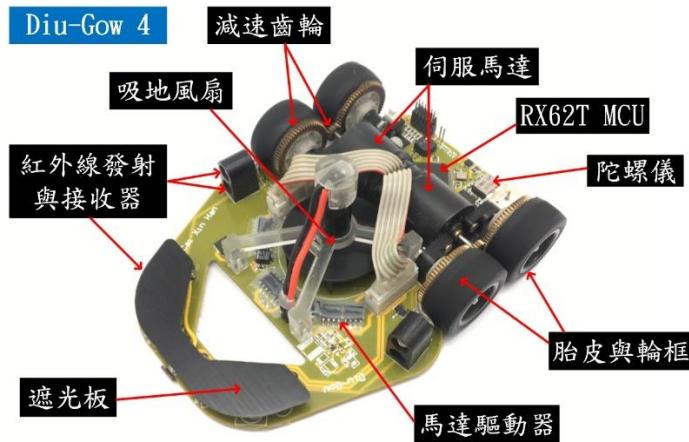


圖 4.22 本論文設計之古典電腦鼠 Diu-Gow 4 成品圖

本論文使用 1 個盒子與砝碼做實驗，以 5V 電源供電給吸地風扇馬達，並使用微控制器中的脈波寬度調變(PWM)功能控制馬達電源，以脈波寬度 10% 的變化量從 0% 至 100%，測試各階段吸地風扇將重物吸取至空中並掉落的最大重量，並紀錄各項數據，如表 4.1 所示。

表 4.1 吸力與功耗測試表

PWM(%)	電壓(V)	電流(mA)	功率(W)	吸力(g)
10	0.5	150	0.02	0(×)
20	1.0	200	0.09	1.5
30	1.5	240	0.195	4.5
40	2.0	280	0.34	27.75
50	2.5	310	0.5	44.5
60	3.0	320	0.63	70.5
70	3.5	330	0.77	89
80	4.0	330	0.88	113.56
90	4.5	330	0.99	134.7
100	5.0	360	1.25	151.8

本論文完成吸地風扇電腦鼠並實際在迷宮場地中測試後，發現具備吸地風扇的電腦鼠，在高速轉彎時吸地風扇能有效地抵消離心力，讓電腦鼠的穩定性與過彎速度皆有明顯地提升。平均整體的轉彎性能提升了約 43.3%，詳細狀況如表 4.2。

表 4.2 運動能力比較

比較情形 轉彎速度	無吸地速度	有吸地速度	提升率(%)
45 Turn(cm/s)	150	220	46.6%
90 Turn(cm/s)	160	220	37.5%
135 Turn(cm/s)	135	180	33.3%
180 Turn(cm/s)	140	210	50%
V Turn(cm/s)	120	180	50%

第五章 比賽策略應用與競賽成果

5.1 電腦鼠比賽類型

目前在世界各地的電腦鼠比賽中，比賽成績計算的方式大致上可分為兩種，以下以本論文常參加的比賽舉例：

1. 依加權成績排名：在臺灣的 TMIRC 古典電腦鼠比賽，與美國 APEC Micromouse contest 比賽中，皆使用加權成績排名。以臺灣的 TMIRC 古典電腦鼠比賽為例，比賽成績算法為，單次成績=運動時間+(迷宮時間/30) - 3 秒(沒有觸碰)，成績計算時不只考慮到電腦鼠搜尋的效率，且若無觸碰還會給予獎勵。
2. 依單趟最佳成績排名：在臺灣的半尺寸電腦鼠、日本的半尺寸與古典電腦鼠，皆使用最快單趟最佳成績排名。此種方式只採計電腦鼠從起點跑至終點的最快成績，後續有無自動走回起點都不影響比賽成績。

5.2 採計迷宮時間與衝刺成績之比賽策略

在此種有採計迷宮時間的比賽中，電腦鼠的穩定性絕對是首要考量，因為在比賽中如果沒有觸碰電腦鼠，就可以讓加權成績減 3 秒(APEC 電腦鼠比賽為 2 秒)；在其次為迷宮時間，在比賽中若是搜尋可以減少 15 秒，經過加權成績計算可以減少 $15/30=0.5$ 秒，減少 0.5 秒意味著衝刺時間可以慢別人 0.5 秒，在現在競爭激烈的電腦鼠比賽中，很多時候選手間電腦鼠衝刺的成績差異可能只差 0.01 秒，甚至是 0.001 秒都發生過，因此盡量提升搜尋速度或提升演算法的搜尋效率等，都是比賽中策略的應用。而本論文針對此種競賽提出三項策略，單次搜尋、搜尋中已知路徑轉換斜線動作與搜尋中直線路徑二次加速，希望盡量減少搜尋時間來增加獲獎的機會。

5.2.1 單次搜尋設定

比賽第一趟採用最基礎的策略，演算法設定終點區域為迷宮終點，並開始搜尋，當電腦鼠到終點時，演算法終點區域改為迷宮起點，並搜尋回起點，如圖 5.1。完成搜尋後第二趟即可做衝刺。

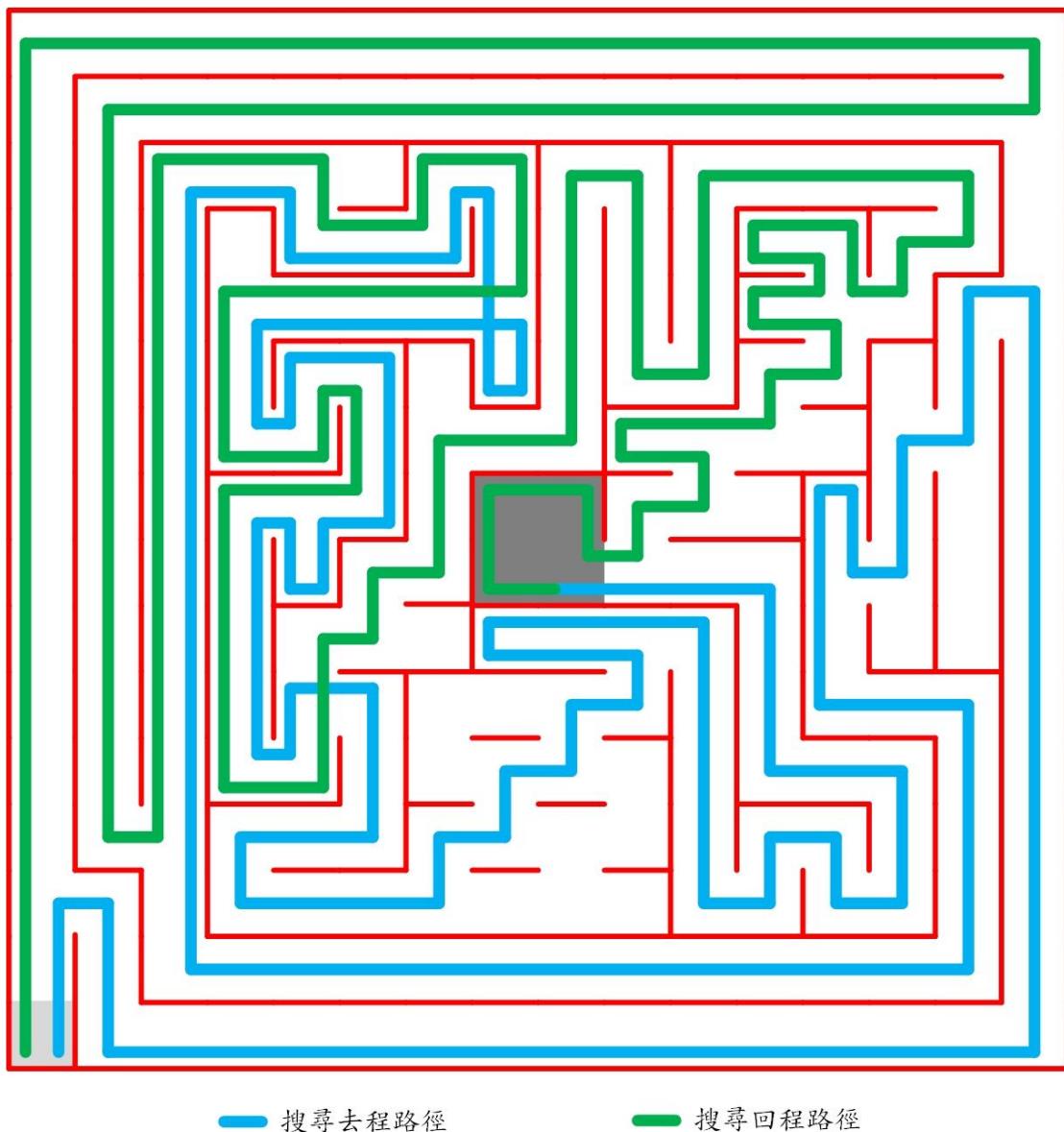


圖 5.1 2016 APEC Micromouse Contest Diu-Gow 4 單次搜尋路徑圖

5.2.2 搜尋中直線路徑二次加速

在古典電腦鼠國際比賽中，選手常使用 60cm/s 至 80cm/s 的速度進行搜尋，本論文考慮到電腦鼠搜尋時的穩定性，將搜尋速度設定為 70cm/s，70cm/s 的速度雖然穩定性佳，但相對的是增加了電腦鼠的搜尋時間。

為提升搜尋速度與保持原有的穩定性，本論文設計當古典電腦鼠搜尋走直線動作時，執行二次加速至最高速 100cm/s；當遇到下格動作為左右轉或迴轉時，在減速至底速 70cm/s 過彎，加速度與減速度皆設定為 15m/s/s，利用方程式(5.1)可求出加速與減速的需求距離：

$$D = \frac{V_1^2 - V_0^2}{2a} = \frac{100^2(\frac{cm}{s})^2 - 70^2(\frac{cm}{s})^2}{2 \times 1500(\frac{cm}{s^2})} = 1.7\text{ cm} \quad (5.1)$$

為使電腦鼠可以正常地銜接下個動作，在直線動作的末端，電腦鼠得提早探索下格迷宮的牆壁，並做演算法計算，若下格為直走動作，可保持最高速繼續行走，若下格為左轉、右轉或迴轉動作，則需提早 1.7 公分執行減速。透過已知的直線最高速 100cm/s，與演算法計算需求時間 0.02 秒，使用方程式(5.2)反推計算演算法需求距離：

$$S = vt = 100(\frac{cm}{s}) \times 0.02(s) = 2\text{ cm} \quad (5.2)$$

由方程式(5.1)與方程式(5.2)，可推算出電腦鼠執行直走動作時，最少得提早 $1.7+2=3.7$ 公分探看牆壁與執行演算法，本論文則是選擇提早 4 公分。完整的搜尋中直線路徑二次加速流程圖如圖 5.2 所示。

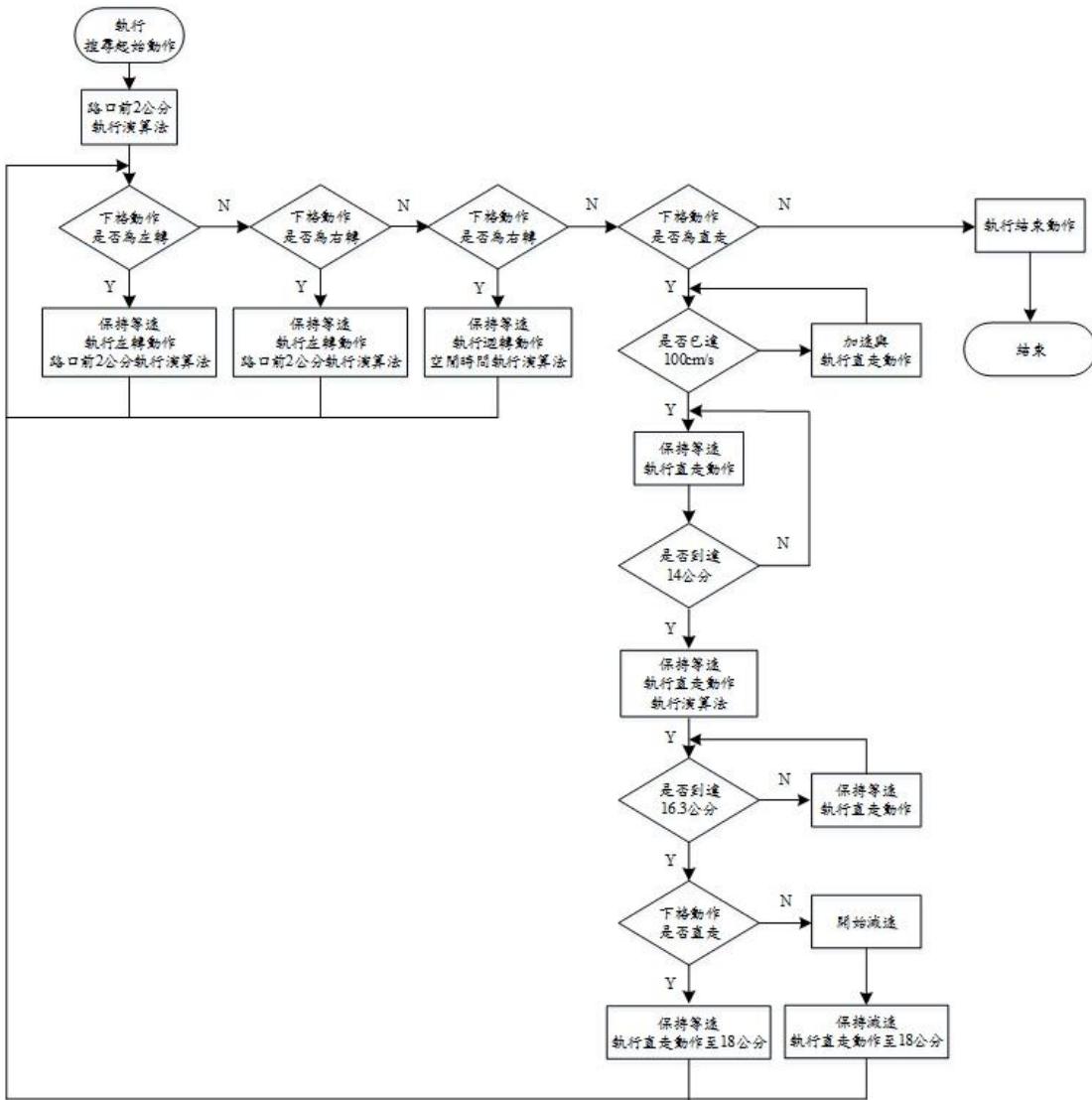


圖 5.2 搜尋中執行二次加速流程圖

5.2.3 搜尋中已知路徑轉換斜線動作

當電腦鼠在迷宮中搜尋時，有時候因為進入死巷內，就得迴轉走回路口處再繼續搜尋，此時相同路徑就會重複走第二次，為了減少搜尋時間，對於這些已走過的路徑，即可直接轉換成斜線路徑行走，不只可提升搜尋速度，若在遇到連續轉彎路徑時，還可減少發生碰撞導致搜尋失敗的機會。

以圖 5.1 迷宮地圖為例，電腦鼠使用 5.2.1 搜尋設定，並從終點區域搜尋回起點，當搜尋至位置 100 時，使用預設全開式牆壁資訊規劃最短路徑至起點，

路徑依序為 $7 \rightarrow 1 \rightarrow 1 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 1 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 5 \rightarrow 1 \rightarrow 1 \rightarrow 7 \rightarrow 1 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 5$
 $\rightarrow 7 \rightarrow 5 \rightarrow 5 \rightarrow 1 \rightarrow 7 \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 1 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 7 \rightarrow 0$ 。在找出路徑後，需要符合以下 2 個條件才能執行轉斜線的動作：

1. 使用拜訪紀錄資訊確認最短路徑上的迷宮組合已探索過
2. 路徑上的頭端與尾端都需要有直線

使用拜訪紀錄資料確認已探索牆壁時，需搭配電腦鼠的當前位置與當前方向資訊推算。由於迷宮單格中採用東邊與北邊牆壁紀錄，因此當電腦鼠面對的方向不同時，其查看已探索牆壁的條件式也會不同。假設電腦鼠在位置 n ，欲查看其視野的前方牆、左方牆與右方牆的探索狀況，資訊位置與對應位元如表 5.1，其中 size 為迷宮地圖橫軸或縱軸的最大數， $\langle x \rangle$ 中的 x 值為對應位元。當欲查看的拜訪紀錄對應位元為 1 時，代表該面牆壁已探索過，若為 0 則反之。

表 5.1 資訊位置與對應位元表

面對方向對應牆面	前方牆	左方牆	右方牆
北	$n\langle 1 \rangle$	$(n-1)\langle 2 \rangle$	$n\langle 2 \rangle$
東	$n\langle 2 \rangle$	$n\langle 1 \rangle$	$(n-size)\langle 1 \rangle$
南	$(n-size)\langle 1 \rangle$	$(n-size)\langle 2 \rangle$	$(n-size-1)\langle 2 \rangle$
西	$(n-1)\langle 2 \rangle$	$(n-size)\langle 1 \rangle$	$n\langle 1 \rangle$

經由拜訪紀錄資料發現當路徑行進至位置 49 時，該格的前方牆與右方牆還未探索過，因此最短路徑只能行進至位置 50，後面的路徑 $7 \rightarrow 5 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 0$ 是不可走的，必須在行進至此處時繼續搜尋，如圖 5.3 所示，在經由已確認後的路徑中 $7 \rightarrow 1 \rightarrow 1 \rightarrow 7 \rightarrow 7 \rightarrow 1 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 5 \rightarrow 1 \rightarrow 1 \rightarrow 7 \rightarrow 1 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 1 \rightarrow 7 \rightarrow 1 \rightarrow 5 \rightarrow 7 \rightarrow 5 \rightarrow 7 \rightarrow 1$ ，可發現頭端的第 2 個路徑與尾端倒數第 2 個路徑皆是直走，因此此路徑符合轉換斜線的條件。

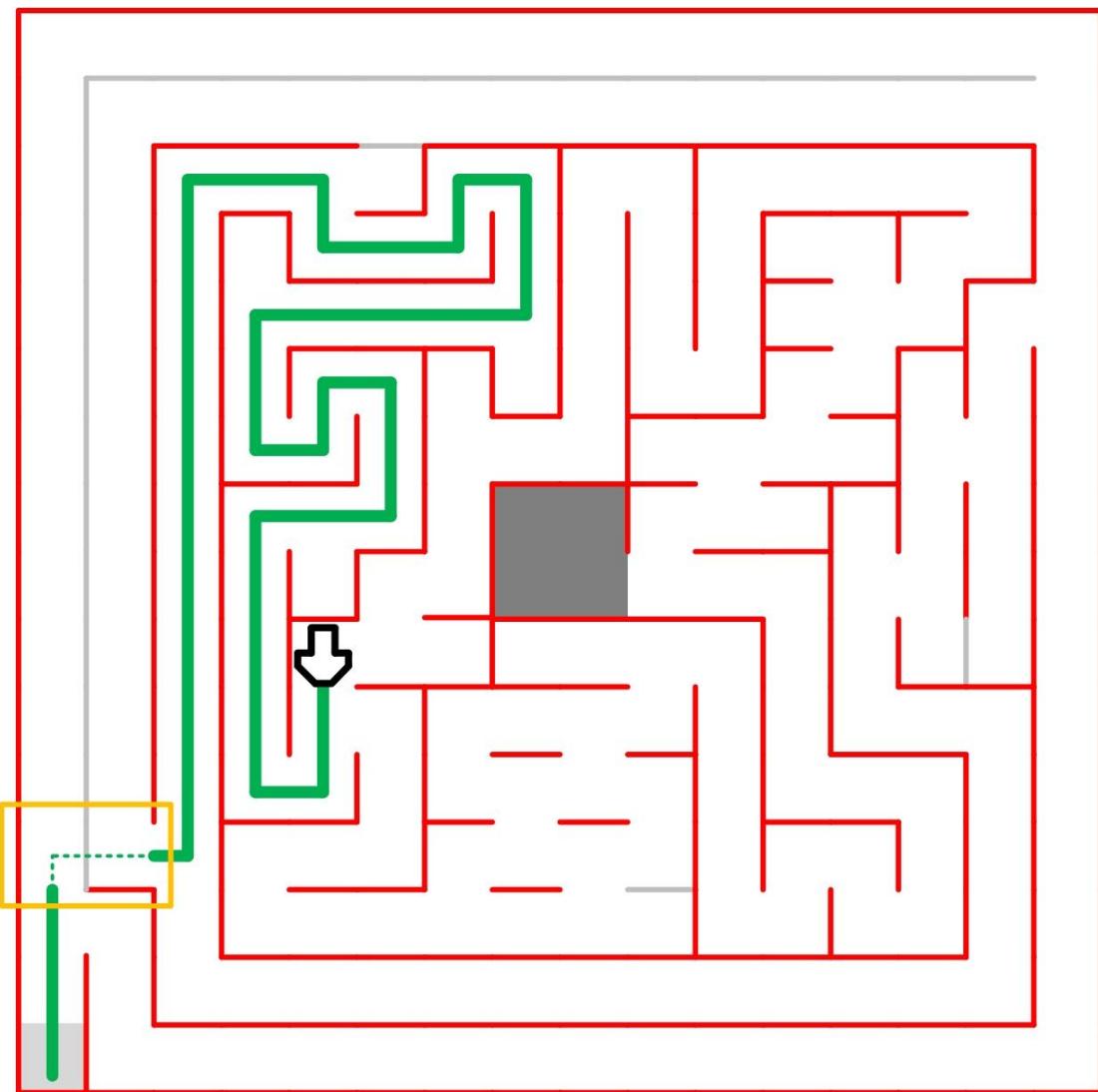


圖 5.3 搜尋中確認已知路徑示意圖

搜尋中已知路徑轉換斜線的方法，與 3.3.3 節中的化簡斜線相對路徑大致相同，差別在於此路徑遇到最後一個直走後，後面的路徑就不可在化簡成斜線路徑，需以一般相對動作繼續行走，其意義表示後面路徑的牆壁還未探索過，必須再繼續探索，如圖 5.4 所示。

圖 5.4 搜尋中已知路徑轉換斜線動作

在轉換完斜線相對路徑後，如同 3.3.4 節一樣要執行二次處理，但一樣與原章節略有差異。首先第 1 筆若馬上為轉彎動作，則要在加入動作 0x1A 做為轉彎前的直線銜接，後續只需做 3.3.4 節中的半格處理至最後一筆直走動作為止，如圖 5.5 所示。

圖 5.5 搜尋中已知路徑轉換斜線動作二次處理

圖 5.6 為經由斜線動作相對路徑轉換與二次處理後的路徑，圖中綠線為斜線動作相對路徑，藍線為一般動作相對路徑。

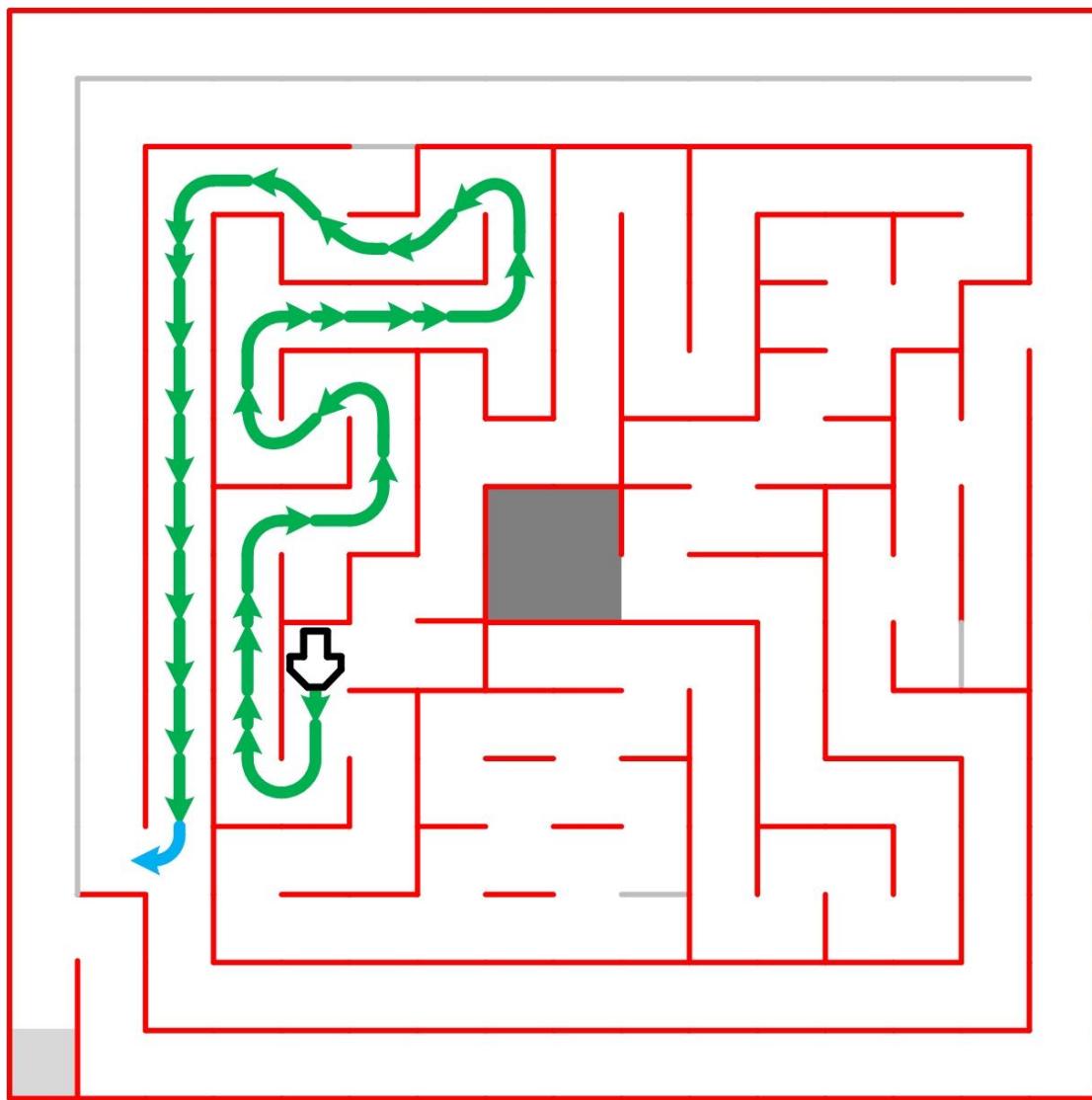


圖 5.6 已知路徑轉換斜線示意圖

5.3 採計單次最佳成績之比賽策略

在此種只採計最佳成績的比賽中，本論文採用策略是儘量在第一趟搜尋時，就盡可能地找到最佳路徑，原因是在第 2 至第 5 趟衝刺時，可以一直在最佳路徑上嘗試更快的衝刺，爭取更好的成績。

本論文設計之電腦鼠在搜尋時，電腦鼠將目標區域設定在終點處並搜尋後，再將目標區域設定在起點處並搜尋，這樣的行為本論文稱之為一次搜尋；若再一次將目標設定至終點後，再將目標設定至起點，則是在迷宮中的第二次搜尋，本論文將超過一次後的搜尋泛稱為多次搜尋。

5.3.1 多次搜尋設定

當電腦鼠設定為多次搜尋時，以本論文設計之半尺寸電腦鼠，參予 2016 全日本電腦鼠比賽半尺寸電腦鼠項目，決賽迷宮地圖之實例為例，當第一次搜尋回起點時，在半尺寸電腦鼠運行至位置 1 時，由拜訪紀錄資訊發現回起點的後續路徑皆已走過，此時演算法即將目標轉設為終點處，開始第二次搜尋，如圖 5.7 所示，而第一次搜尋花費時間約 2 分 5 秒。

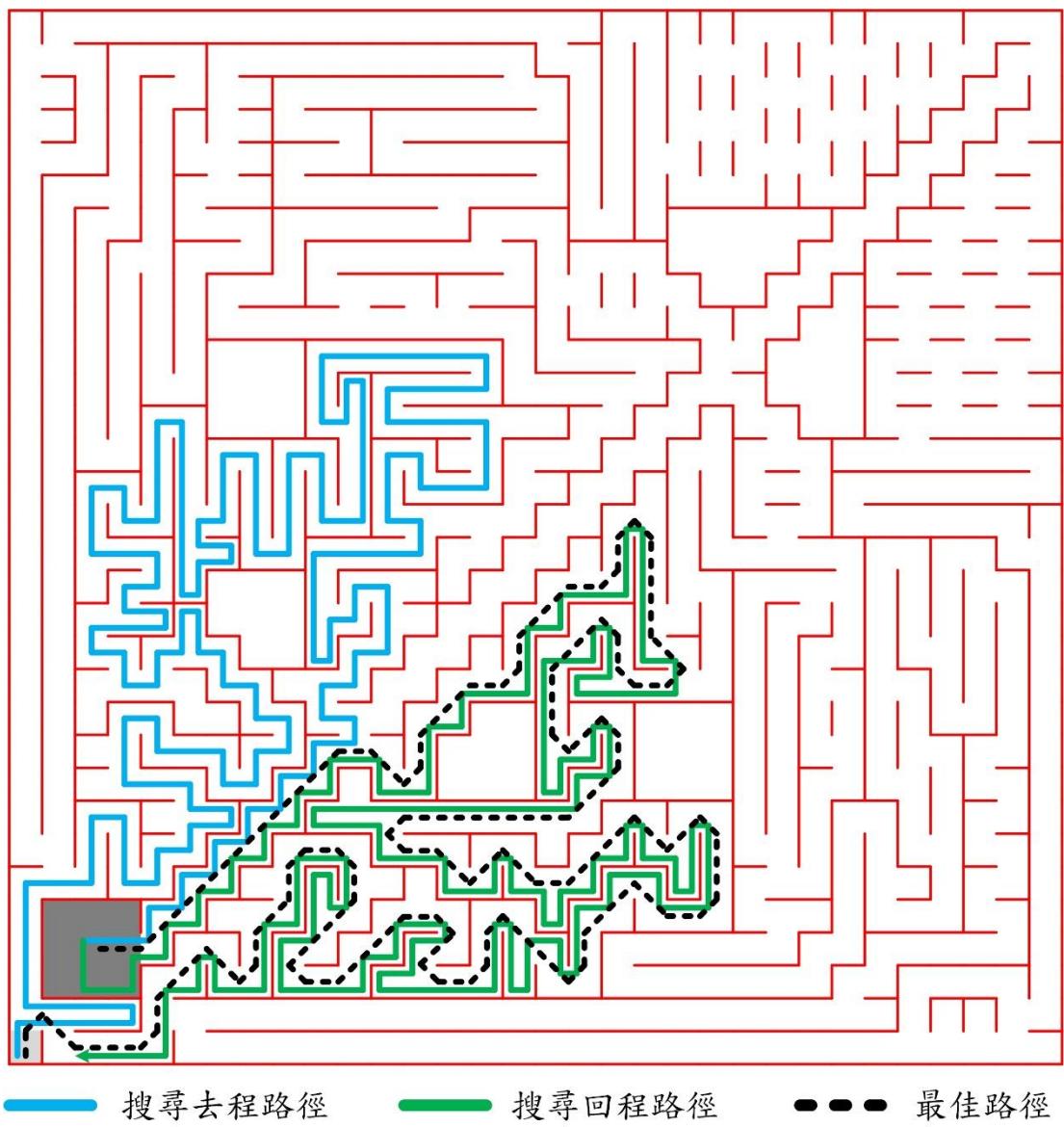


圖 5.7 第 1 次搜尋路徑與最佳路徑計算路徑圖

電腦鼠在開始多次搜尋後，搜尋順序一樣是先將目標區域設定在終點處，之後在將目標區域設定至起點處。但是在多次搜尋且目標區域為終點時，電腦鼠在搜尋時每走一格都需搭配拜訪紀錄資訊，確認現在位置欲搜尋至終點區域的路徑是否已走過，若未走過則可執行搜尋動作，若欲搜尋路徑皆已走過，則搜尋目標區域改設定至起點處，不需等到進終點後才改搜尋目標區域，因為在先前搜尋時已走進去過。在轉換目標區域至起點後，電腦鼠一樣每走一格都需搭

配拜訪紀錄資訊，確認現在位置欲搜尋至起點處的路徑是否已走過，未走過則搜尋，已走過就在轉換目標區域至終點區域，繼續下一次的搜尋。

圖 5.8 為本論文之半尺寸電腦鼠在比賽中第 2 次搜尋路徑，第 2 次搜尋花費時間約 1 分 53 秒，加上先前搜尋時間，共花費 3 分 58 秒，而演算法計算出的最短路徑與第一次搜尋後計算出的最佳路徑相同。

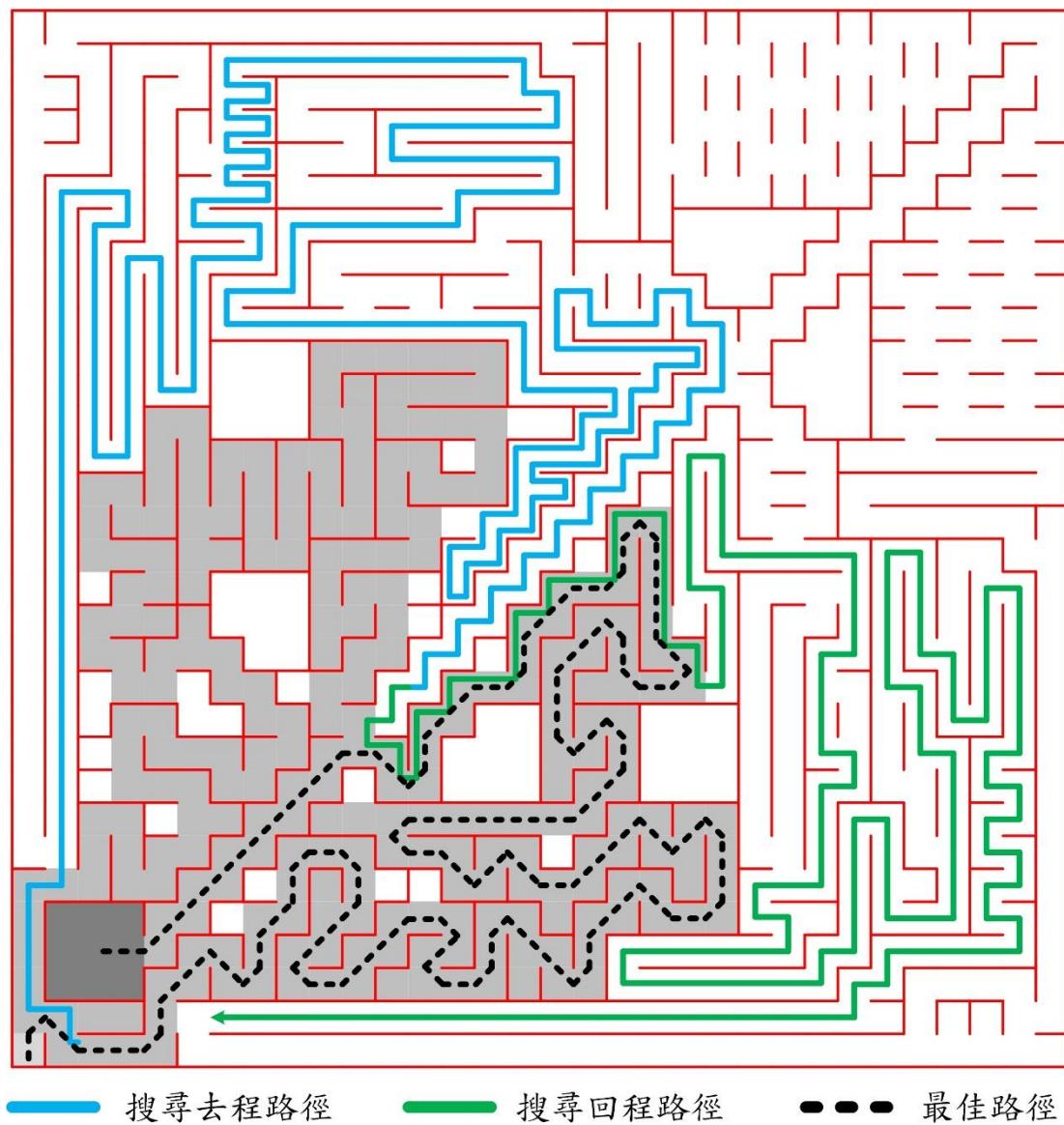
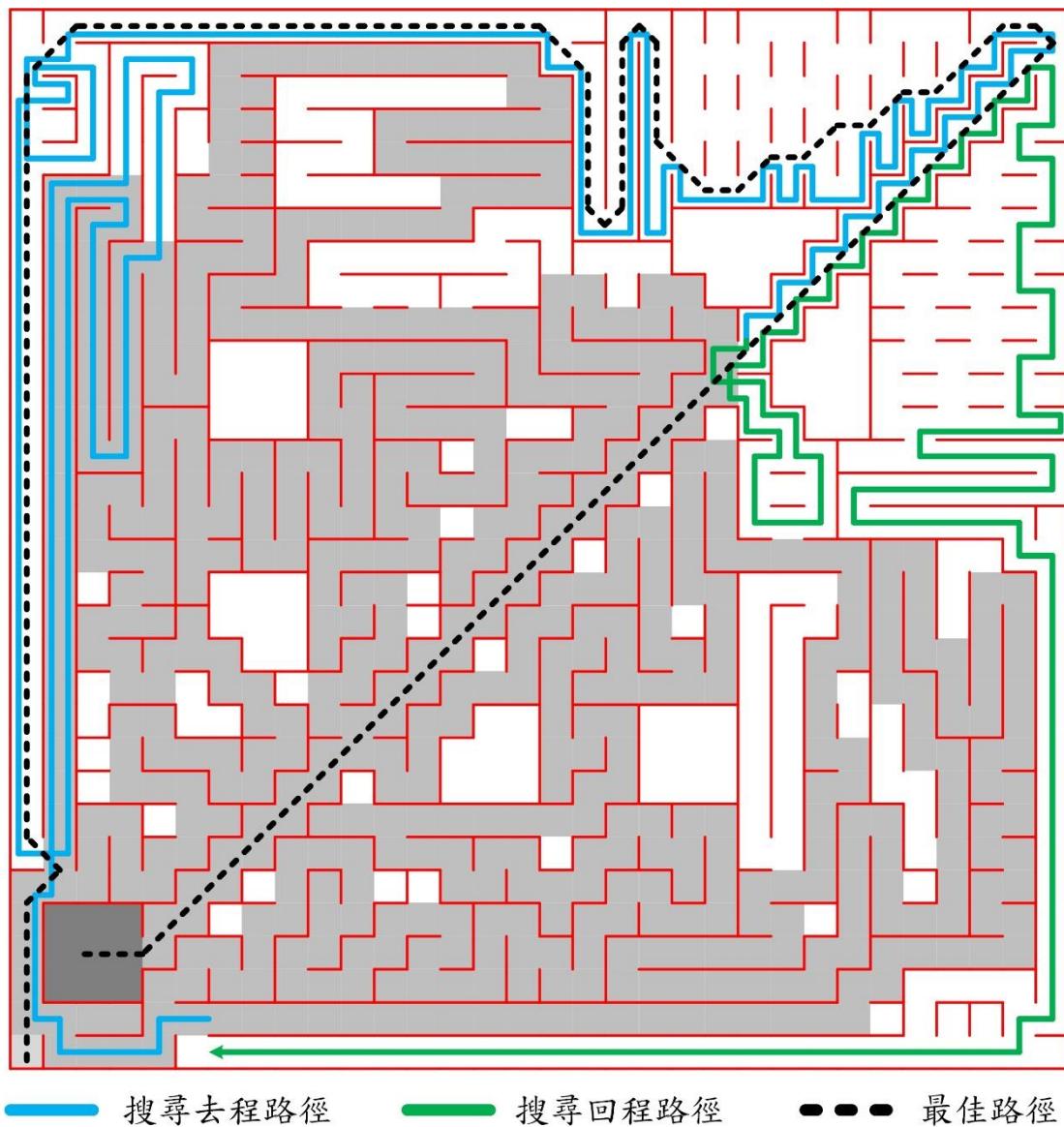


圖 5.8 第 2 次搜尋路徑與最佳路徑計算路徑圖

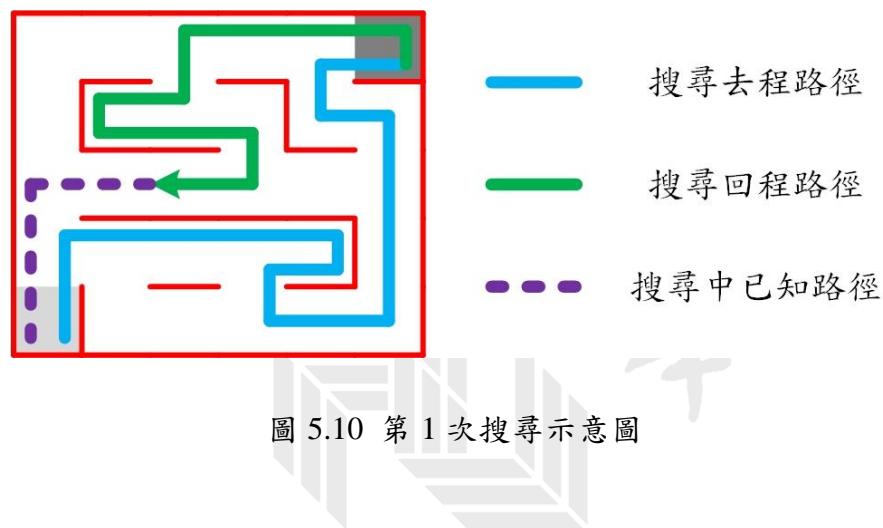
圖 5.9 為本論文之半尺寸電腦鼠在比賽中第 3 次搜尋路徑，第 3 次搜尋花費時間約 1 分 37 秒，加上先前搜尋時間，共花費 5 分 35 秒，而演算法計算出了不同於前次搜尋找到的最佳路徑。



5.3.2 多次搜尋結束方式

本論文設計之多次搜尋有 2 種結束方式，第 1 種為演算法認為當前欲搜尋路徑皆已找過而結束搜尋，第 2 種為計算搜尋次數，當到達設定次數時，強制結束搜尋。

在第 1 種方式中，當演算法切換搜尋方向後，緊接著下次欲搜尋路徑皆已完全走過時則結束搜尋。以圖 5.10 為例，第 1 次搜尋至終點且回起點的路上發現路徑皆已走過時(紫色路徑)，開始第 2 次搜尋。



第 2 次搜尋往終點區域，當發現欲前往終點的路徑已走過時，如圖 5.11 紫色路徑，因此搜尋目標區域改至起點，但此時才剛轉換目標區域時，又發現至起點的路徑皆已走過，因此符合結束搜尋條件，電腦鼠不再切換目標區域，直接走回起點處，完成本次搜尋。

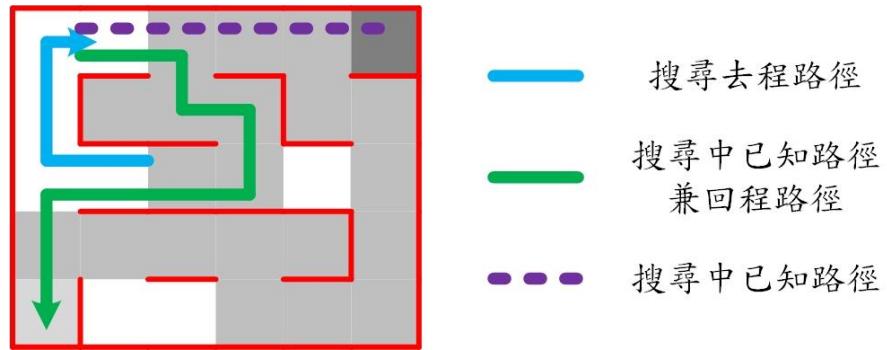


圖 5.11 第 2 次搜尋時結束圖

在第 2 種方式中，本論文使用來輔助第 1 種結束方式，在半尺寸電腦鼠每完成一次搜尋時便計數一次，本論文設定當搜尋次數達到第 5 次時，無論電腦鼠認為是否還有未搜尋之路徑可走，皆強制電腦鼠必須搜尋回起點處，不得再進行第 6 次搜尋。此結束方式是為了防止電腦鼠因迷宮地圖過於複雜時，可能會因為想搜尋到更好的路，反而耗盡了比賽時間，而無法進行衝刺，導致錯失爭取更好成績的機會。

5.4 比賽實際應用實例

在 2016 年美國 APEC 國際電腦鼠的比賽中，成績計算有採計迷宮時間，因此本論文製作之古典電腦鼠 Diu-Gow 4 採用 5.2 章的三項策略，單次搜尋、搜尋中已知路徑轉換斜線動作與搜尋中直線路徑二次加速。根據參考文獻[15]紀錄的數據，本論文將此比賽中，前三名的前兩次各項運行數據整理如表 5.2 所示，本論文設計之古典電腦鼠 Diu-Gow 4，雖然在第二次運行時比其他隊伍慢了 0.11 秒與 0.21 秒，但是靠著提升搜尋時間的方式，迷宮時間除以 30 的秒數卻比其他隊伍少了 0.69 秒與 0.86 秒，使最後計算出的加權成績低於其他隊伍，因此比賽中該策略的應用可以說是效果顯著。

表 5.2 2016 年 APEC 國際電腦鼠比賽前三名成績

隊名	迷宮時間		單次衝刺成績			加權成績		
	1	2	1	2	排名	1	2	排名
Hippo C	0.0	81.57	48.12	9.34	1	48.12	12.06	2
Green Giant 5.1V	0.0	86.78	53.39	9.44	2	53.39	12.33	3
Diu-Gow 4	0.0	60.83	29.40	9.55	3	29.40	11.58	1

在 2016 年全日本電腦鼠比賽的半尺寸電腦鼠比賽中，成績計算只採計單趟最佳時間，本論文製作之半尺寸電腦鼠 Mini Diu-Gow 採用 5.3 章的多次搜尋策略，並將搜尋次數上限訂為最多 5 次。電腦鼠在比賽的第 1 趟搜尋中共搜尋了 3 次，路徑如圖 5.7、圖 5.8 與圖 5.9，在進行第 4 次搜尋時，雖然電腦鼠因姿態控制錯誤導致撞牆而中止搜尋，但在前 3 次的搜尋中已成功找到該地圖的最佳路徑，因此有了 4 次可跑最佳路徑的機會。根據比賽單位[16]紀錄的比賽結果，本論文製作之半尺寸電腦鼠 Mini Diu-Gow 在嘗試衝刺中，雖只成功完成一次衝刺，但卻靠著優異的路徑，獲得了第四名的成績。

表 5.3 2016 全日本半尺寸電腦鼠比賽 Mini Diu-Gow 比賽中各次成績

次數	1	2	3	4	5
成績	01:02.278	R	00:09.98	R	R

5.5 得獎紀錄

本論文在設計與撰寫時間導向洪水演算法並實際使用後，在參予國內及國外的比賽中皆有獲得不錯的成績，如圖 5.12。

1. 2015 美國 APEC 電腦鼠競賽 獲得第二名、最佳學生獎。
2. 2015 臺灣 TMIRC 古典電腦鼠大專組 獲得第一名。
3. 2015 臺灣 TMIRC 古典電腦鼠國際組 獲得第一名。
4. 2015 全日本電腦鼠競賽古典電腦鼠項目第一名。
5. 2016 美國 APEC 電腦鼠競賽 獲得第一名、最佳學生獎、最快速度獎。
6. 2016 臺灣 TMIRC 古典電腦鼠大專組 獲得第一名。
7. 2016 臺灣 TMIRC 古典電腦鼠國際組 獲得第二名。
8. 2016 臺灣 TMIRC 半尺寸電腦鼠國際組 獲得第五名。
9. 2016 全日本電腦鼠競賽古典電腦鼠項目第一名。
10. 2016 全日本電腦鼠競賽半尺寸電腦鼠項目第四名。
11. 2017 葡萄牙電腦鼠競賽 獲得第一名。



圖 5.12 得獎紀錄

第六章 結論與未來展望

6.1 結論

本論文在完成時間導向斜線迷宮演算法後，在參予各項電腦鼠競賽中，其路徑選擇皆有讓電腦鼠完整發揮其應有的運動能力；本演算法在撰寫程式的記憶體需求用量上也偏低，半尺寸電腦鼠約需 10.5Kbyte，古典電腦鼠約需 2.625Kbyte；在 32×32 的迷宮地圖上執行速度 20ms 左右屬於中等，對電腦鼠而言不會有算不完需要停下來算的問題；由於轉彎速度採取固定速度的策略，因此在程式撰寫上難易度也屬於中等。

本論文在硬體部分完成新的半尺寸電腦鼠的設計，完成移植演算法與成功在比賽中找出演算法認為的最佳路徑並衝刺；在古典電腦鼠中上，以新的吸地風扇設計，成功提升古典電腦鼠的運動能力，突破以往速度提升不起來的窘境。

6.2 未來展望

本論文在未來發展還可更加精進：

1. 以不同轉彎速度計算最佳路徑：本論文目前設計之演算法，其在計算路徑時皆以相同轉彎速度計算，雖然在大部分的情況下皆如願找到最佳路徑，但難保在未來電腦鼠比賽的發展中，會出現較極端的地圖組合，例如大量的連續 V 型轉彎組合，此種彎道非常不利於電腦鼠高速行走，因此該轉彎速度低於平均轉彎速度，此時若用平均轉彎速度計算最佳路徑，就有可能因此挑選到比較不好的路徑，因此如何使用不同轉彎速度計算最佳路徑，可為日後研究的項目之一。

2. 半尺寸電腦鼠吸地設計：本論文設計的吸地風扇古典電腦鼠在實作上速度成功提升了 46.6%，未來當半尺寸電腦鼠做到瓶頸時，可將吸地風扇的經驗從

古典電腦鼠轉移過去，藉由吸力抵抗離心力，讓半尺寸電腦鼠可以運行的更快與更穩定。



參考文獻

- [1] 人工智慧單晶片電腦鼠暨機器人國內及國際邀請賽, <http://robot.lhu.edu.tw/>, Accessed on June 15, 2017.
- [2] The 37th All Japan Micromouse Contest, http://www.ntf.or.jp/mouse/micromouse2016/index_EN.html, Accessed on June 15, 2017.
- [3] 林國恩, 電腦鼠的設計與實作, 碩士論文, 龍華科技大學, 桃園(2010)。
- [4] 張晉嘉, 佳者優先搜尋策略之迷宮最佳路徑演算法之研究, 碩士論文, 南台科技大學, 台南(2013)。
- [5] 黃正宇, 高速電腦鼠機器人之設計與實作, 碩士論文, 龍華科技大學, 桃園(2012)。
- [6] 趙師葦, 智慧輪型機器人設計與實作, 碩士論文, 龍華科技大學, 桃園(2013)。
- [7] RX62T, <https://www.renesas.com/zh-tw/products/microcontrollers-microprocessors/rx/rx600/rx62t.html>, Accessed on June 15, 2017.
- [8] OPTEK Technology Inc., "Plastic Infrared Emitting Diode," datasheet of OP265AD, 2017.
- [9] LITE-ON Technology Inc., " IR Emitter and Detector," datasheet of LTR-4206, 2002.
- [10] TOSHIBA Inc., "TOSHIBA Transistor Silicon NPN Epitaxial Type (PCT process)," datasheet of 2SC3325, 2014.
- [11] Texas Instruments Inc., "DRV8836 Dual Low-Voltage H-Bridge IC," datasheet of DRV8836, 2016.

[12] MTL MES-6-P,

<https://www.mtl.co.jp/product/index.php/item?cell003=%E3%82%A4%E3%83%B3%E3%82%AF%E3%83%AA%E3%83%A1%E3%83%B3%E3%82%BF%E3%83%AB%E5%BC%8F&cell004=ME%E3%82%B7%E3%83%AA%E3%83%BC%E3%82%BA&id=3&label=1>,

Accessed on June 15, 2017.

[13] STMicroelectronics Inc., "MEMS motion sensor:

high performance ± 2000 dps analog yaw-rate gyroscope," datasheet of LY3200ALH, 2009.

[14] RE8 DC motor,

<http://www.maxonmotor.com.tw/maxon/view/product/motor/dcmotor/re/re08/347723>, Accessed on June 15, 2017.

[15] Micromouse USA Blog, <http://micromouseusa.com/?p=1876>, Accessed on June 15, 2017.

[16] マイクロマウス 2016 記録集,

http://www.ntf.or.jp/mouse/micromouse2016/MM2016recode_v101.pdf, Accessed on June 15, 2017.

附錄

附錄 A 古典電腦鼠之競賽規則	96
附錄 B 半尺寸電腦鼠之競賽規則	103



A 古典電腦鼠之競賽規則

目的

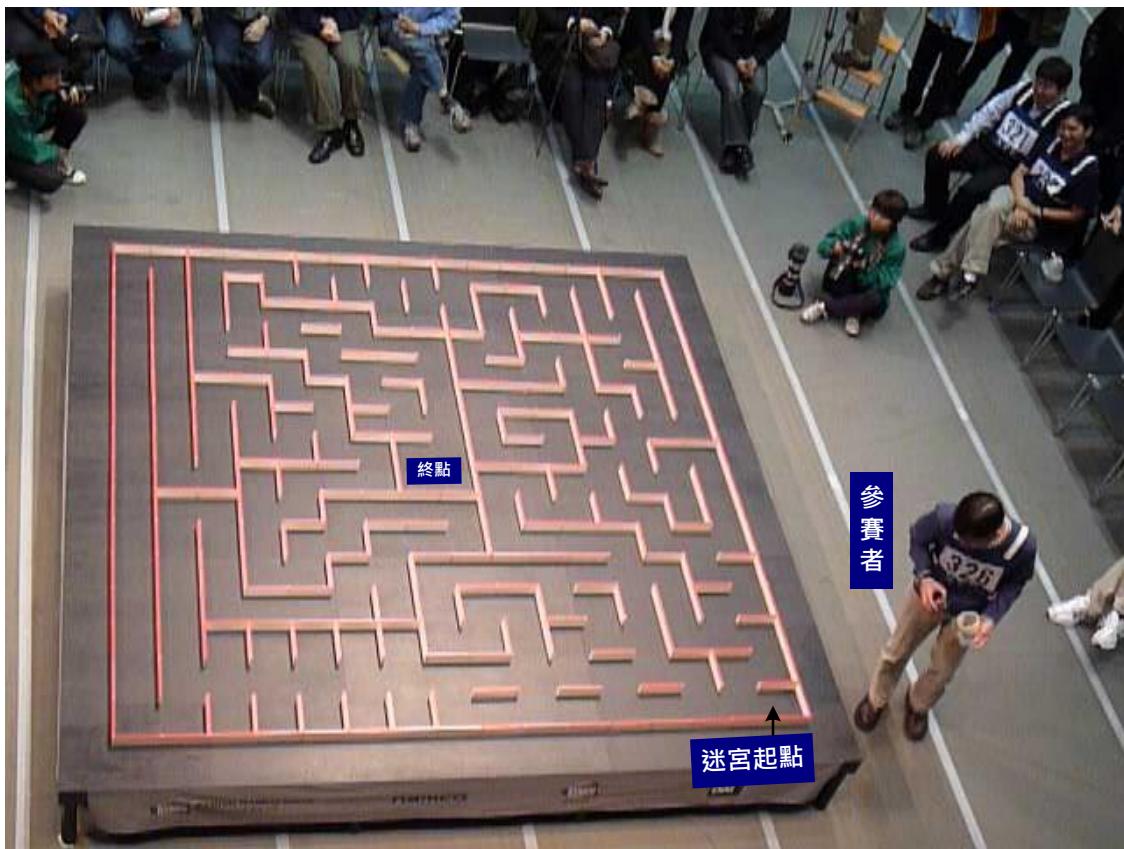
本競賽的主要目的在設計並製作一個可以在最短時間內找到迷宮終點的機器人。所有參加本競賽的機器人都稱為「電腦鼠」。負責將該「電腦鼠」放置於場地，並讓「電腦鼠」開始動作的人稱為「操作員」。



迷宮的規定

1. 電腦鼠迷宮由 16×16 個迷宮方塊組成，每個單位方塊為 $18\text{cm} \times 18\text{cm}$ 。
2. 迷宮牆壁的高度為 5cm ，厚度為 1.2cm ，因此迷宮中電腦鼠的走道會有 16.8cm 寬。迷宮的外圍全部以迷宮牆相連接起來。
3. 迷宮牆壁壁面的顏色為白色，頂部為紅色。迷宮中電腦鼠行進的地面是以表面塗有黑色去光澤塗料的木材製成。迷宮牆壁壁面與頂部的塗料必須能反射紅外線的投射，而行進地面的塗料則必須吸收紅外線的投射。

4. 迷宮的「起點」位於四個迷宮角落方塊的其中一個。迷宮的「起點」應有三面迷宮牆，其中如果出發的方向為北方，那麼迷宮外圍的牆壁應該在西方與南方。迷宮的「終點」由迷宮中心的四個方塊組成，這四個迷宮方塊間不會有迷宮牆壁。
5. 在每一個迷宮方塊四個角落上 $1.2\text{ cm} \times 1.2\text{ cm} \times 5\text{ cm}$ 的小方柱，稱為「格點」。除了迷宮終點的四個方塊外，每一個迷宮方塊的格點必須至少接觸一面迷宮牆壁。
6. 迷宮大小的精確度應該在 5% 或 2cm 內。迷宮地面的銜接處不能有超過 0.5mm 的高度差，並且迷宮地面銜接處的斜率也不能超過 4 度。相鄰迷宮方塊間牆壁的間隙不能超過 1mm。



電腦鼠的規定

1. 電腦鼠必須要能獨立自行控制，並且不能用燃燒的方式取得能源。
2. 電腦鼠的長寬不應超過 $25\text{cm} \times 25\text{cm}$ ，高度沒有限制。如果電腦鼠在行進間會改變幾何結構的話，也必須符合上述的規定。
3. 電腦鼠在迷宮中行進時，不得在後方遺留任何的東西。

4. 電腦鼠不得躍過、攀登、損傷或破壞迷宮壁面。

競賽規則

1. 電腦鼠的基本功能是要能由迷宮起點行進至終點，這稱為「一趟」，使用的時間稱為「運動時間」，但是電腦鼠由迷宮終點回到起點的時間不能算成是「一趟」的「運動時間」。
2. 電腦鼠由啟動後到某一趟開始行進前的時間稱為「迷宮時間」。電腦鼠在行進期間如果需要操作員的人為協助，稱為「發生觸碰」。**但若電腦鼠自行回到起點後，在下一次啟動前操作員清潔電腦鼠輪胎的動作，不視為觸碰。**
3. 電腦鼠的成績就根據「運動時間」、「迷宮時間」以及「發生觸碰」等三個參數來評定它在速度、解迷宮的效率以及自主性上的表現。
4. 電腦鼠在迷宮場地的競賽中，最多只能擁有**7**分鐘的時間。在這個時間限制下，電腦鼠可以嘗試至多**7**次由迷宮起點出發找出迷宮終點以及最短的運動路徑（比賽隊伍數太多時，主辦單位保留調整時間與嘗試次數的權利）。
5. 電腦鼠到達迷宮終點後，可以人為的方式將電腦鼠移至起點，或是讓它自行回到起點。如果以人為的方式將電腦鼠移至起點，代表「發生觸碰」，後續的成績計算就無法享受到沒有「發生觸碰」**3**秒鐘的時間回饋。
6. 每一趟「運動時間」的計算，是量測電腦鼠離開起點方塊後直到進入終點的時間間隔，操作員準備完成後必須向裁判示意開始計時。而「迷宮時間」的計算則是量測電腦鼠第一次啟動後，直到某一趟運動的起始期間的時間間隔。
7. 電腦鼠成績計算的公式以及最佳成績的登錄方式如下：

單次成績 = 「運動時間」 + (「迷宮時間」除以 30) - **3**秒(沒有「發生觸碰」)

最佳成績 = 各單次成績中時間最短者。

範例 1：假設某一操作員欲開始比賽，準備完成後向裁判示意開始計時，電腦鼠開始啟動後，經過 80 秒找到迷宮終點，這一隻電腦鼠接著離開迷宮終點，繼續搜尋迷宮的牆壁排列情形，又花了 55 秒回到起點。那麼第一次的運動時間 T1 就是 80 秒。由於第一次的啟動並沒有「迷宮時間」，並且電腦鼠自行返回終點，並無「發生觸碰」，因此第一次的成績是

$$80 - 3 = 77 \text{ 秒}。$$

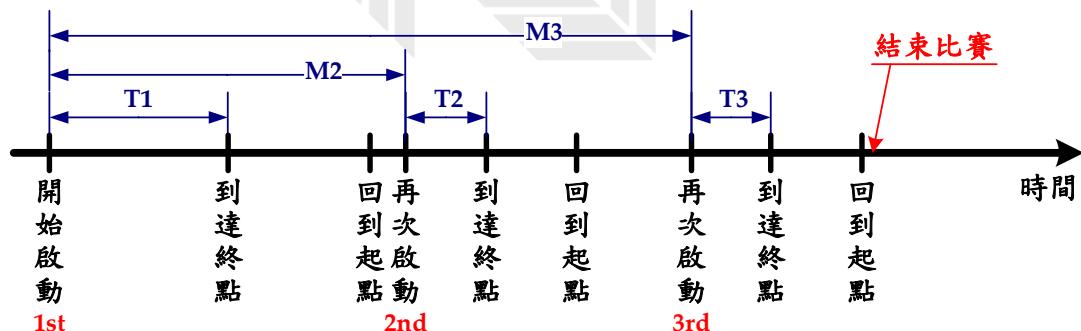
此時操作員將它拿起來清除輪胎上的灰塵，再次放回迷宮場地的起點，清除灰塵至再次啟動共花費 22 秒。接著以搜尋到的最短路徑，用了 10.23 秒衝刺到終點。接著電腦鼠花費 12.02 秒再次回到起點。此時第二次的運動時間 T2 就是 10.23 秒，而之前所使用的搜尋時間 80 秒加上 55 秒再加上 22 秒，總共 157 秒的時間就是在計算第二次成績的迷宮時間 M2。由於沒有「發生觸碰」，因此第二次的成績是

$$10.23 + 157/30 - 3 = 12.46 \text{ 秒}。$$

當這一隻電腦鼠再次回到起點，操作員將它拿起來清除輪子上的灰塵後，再次放回迷宮場地的起點進行第二次由迷宮起點到終點的嘗試，清除灰塵至再次啟動共花費 20.75 秒。這一次以 10.05 秒衝刺到終點。此時第三次的運動時間 T3 就是 10.05 秒，而之前所使用的搜尋與衝刺時間一共是 80 秒加上 55 秒加上 22 秒加上 10.23 秒還有 12.02 秒以及 20.75 秒，總共 200 秒的時間就是在計算第三次成績的迷宮時間 M3。接著電腦鼠花費 11.55 秒再次回到起點。由於沒有「發生觸碰」，因此第三次的成績是

$$10.05 + 200/30 - 3 = 13.72 \text{ 秒}。$$

最後，操作員向裁判示意不再嘗試後結束競賽。這一隻電腦鼠的最佳成績，則以三次成績的最佳值 12.46 秒登錄為比賽成績。下圖是上述過程的圖例說明。



範例 2：假設另一隻電腦鼠開始啟動後，經過 70 秒找到迷宮終點，那麼第一次的運動時間 T1 就是 70 秒。這一隻電腦鼠接著離開迷宮終點，繼續搜尋迷宮的牆壁排列情形，很不幸地還未回到起點時就因為撞牆而發生失誤，因此，操作員向裁判示意後，由場地中取出該隻電腦鼠，因此沒有「發生觸碰」3 秒鐘的時間回饋。由於第一次的啟動並沒有「迷宮時間」，因此第一次的成績是

$$70 = 70 \text{ 秒}。$$

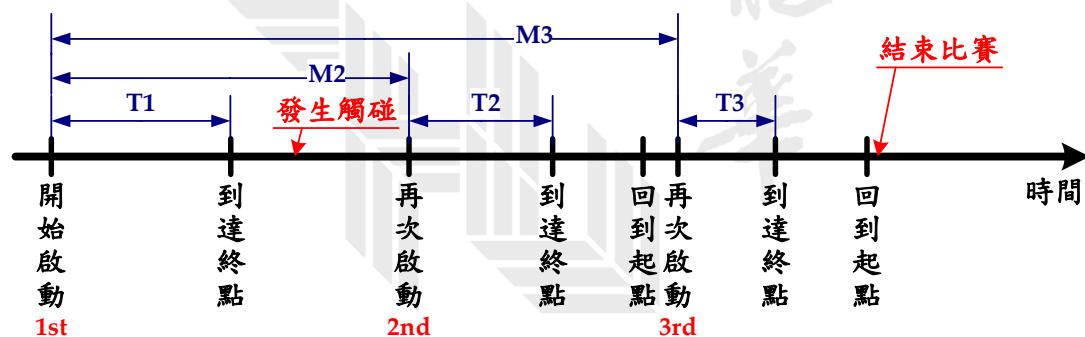
操作員稍做整理後，重新開始第二次的搜尋，此時距離第一次電腦鼠開始啟動，總共是 150.35 秒。當它使用 42.68 秒找到迷宮終點後，又花了 22.50 秒回到起點，並停止 3 秒，自動 180 度回轉離開迷宮起點。接著以搜尋到的最短路徑搭配最快的速度，用了 16.23 秒衝刺到終點。這時，操作員向裁判示意不再嘗試後結束競賽。

此時第二次的運動時間 T_2 就是 42.68 秒，第三次的運動時間 T_2 就是 16.23 秒。而與第二次運動時間相關的迷宮時間 M_2 是 150.35 秒，與第三次運動時間相關的迷宮時間 M_3 是 $150.35+42.68+22.50+3=218.53$ 秒。由於「發生觸碰」因此沒有 3 秒鐘的時間回饋，第二與第三次的成績分別是

$$42.68 + 150.35/30 = 47.69 \text{ 秒}。$$

$$16.23 + 218.53/30 = 23.51 \text{ 秒}。$$

這一隻電腦鼠的最佳成績，則以三次成績的最佳值 23.51 秒登錄為比賽成績。
下圖是上述過程的圖例說明。

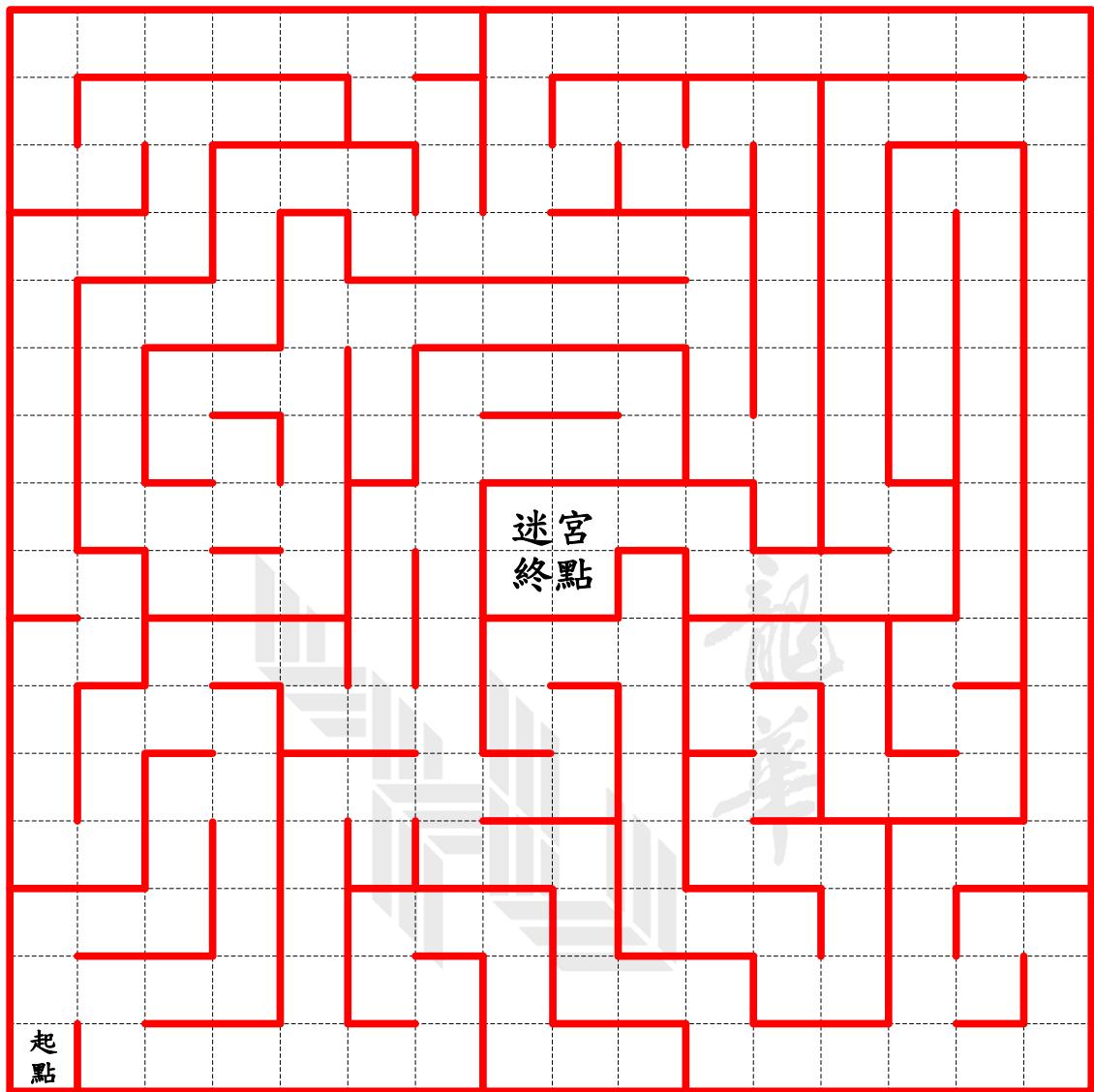


8. 「運動時間」、「迷宮時間」的計算，是由裁判人員手動的方式或由紅外線感測器自動地在電腦鼠迷宮的起點方塊與迷宮的終點方塊間量測而得。當使用紅外線感測器時，迷宮起點的感測器會裝設在起點方塊與下一個迷宮方塊的邊界上，而迷宮終點的感測器則會裝設在迷宮終點方塊的入口上。紅外線感測器會保持水平的發射與接收角度，並且離迷宮地面大約 1cm 高的位置。
9. 電腦鼠的啟動程式中，不應包含提供操作員針對迷宮特性，選擇不同的行動策略。
10. 迷宮的組成架構一但揭露給所有的參賽者知道時，操作員就不能再對電腦鼠輸入任何的資訊。

11. 迷宮所在位置的亮度、溫度與溼度與一般的室內環境是相同的，如果參賽者要求調整場地的亮度，必須得到大會的同意才可以。
12. 當電腦鼠失常時，操作員可以要求裁判允許放棄該次電腦鼠行進到迷宮終點的嘗試，並將電腦鼠重新移到迷宮的起點方塊中。但若只是轉錯彎等非電腦鼠功能失常的因素，則不在同意之列。
13. 比賽進行中，如果電腦鼠更換任何一個零件(如電池或可抹除式唯讀記憶體)或執行顯著的調整(諸如速度控制與行動策略的選擇等)時，在重新啟動前，電腦鼠內部關於迷宮組成結構的記憶內容必需消除掉才行。但在大會裁判根據臨場的判斷，認定是簡單的調整(例如光感測器的調整)時，可以不用消除掉電腦鼠內部關於迷宮組成結構的記憶。
14. 除了電池可能被大會裁判允許外，電腦鼠上的任何其他零件不得在比賽進行中移轉給其他不同參賽隊伍的電腦鼠。舉例而言，如果同一個電腦鼠基座，換上第二個不同的控制器，應視為同一個參賽隊伍的電腦鼠，必須在 7 分鐘的時限內完成比賽。當更換不同的控制器時，電腦鼠內部關於迷宮組成結構的記憶內容必需消除掉才可以繼續進行比賽。
15. 當大會裁判判斷比賽用的迷宮場地，如果在參賽隊伍的電腦鼠持續進行時可能遭到破壞，那麼大會有權要求參賽隊伍的電腦鼠停止繼續進行比賽，或是裁定失去比賽的權利。
16. 當參賽隊伍的電腦鼠完成一次由迷宮起點到終點的運動後，在下一次離開迷宮起點前，必須停留在迷宮起點方格中至少 3 秒鐘。如果迷宮場地使用的是紅外線感測器來計時，那麼當電腦鼠停留在迷宮起點方格時不可以干擾到紅外線感測器的運作。
17. 當完成賽事的電腦鼠隊伍少於比賽獎項時，必要時可從缺。

迷宮場地範例

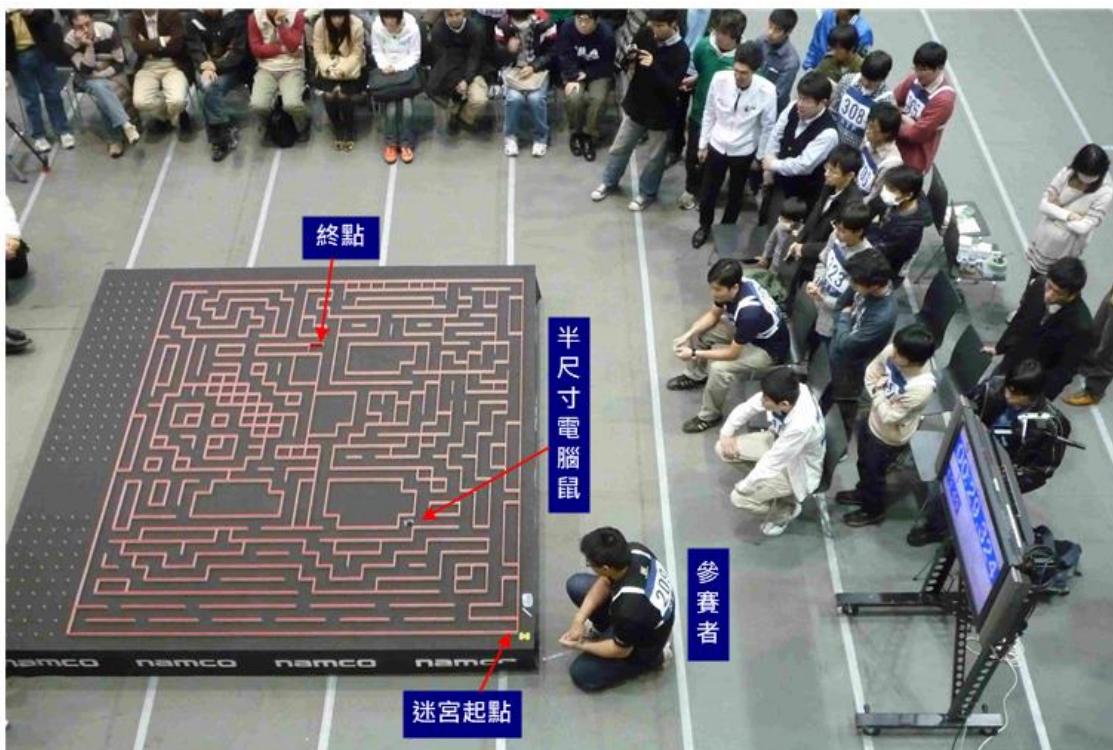
下圖是一個電腦鼠的迷宮場地範例。



B 半尺寸電腦鼠之競賽規則

目的

本競賽的主要目的在要求參賽選手設計並製作一個可以在迷宮方格與牆面規格皆為古典電腦鼠迷宮一半的場地中，以最短時間內找到任意指定之迷宮終點的機器人。所有參加本競賽的機器人都稱為「半尺寸電腦鼠」。負責將該「半尺寸電腦鼠」放置於場地，並讓「半尺寸電腦鼠」開始動作的人稱為「操作員」。



迷宮的規定

1. 迷宮牆壁牆面的顏色為白色，頂部為紅色。迷宮中電腦鼠行進的地面上是以表面塗有黑色去光澤塗料的木材製成。迷宮牆壁牆面與頂部的塗料必須能反射紅外線的投射，而行進地面的塗料則必須吸收紅外線的投射。
2. 電腦鼠迷宮是由多個 $9\text{cm} \times 9\text{cm}$ 的單位正方格所組成。迷宮中可以包含最多的單位正方格數為 32×32 格。沿著 $9\text{cm} \times 9\text{cm}$ 的單位正方格四周所構建的迷宮牆壁，高度是 2.5cm ，厚度則是 0.6cm (參考圖 1)。

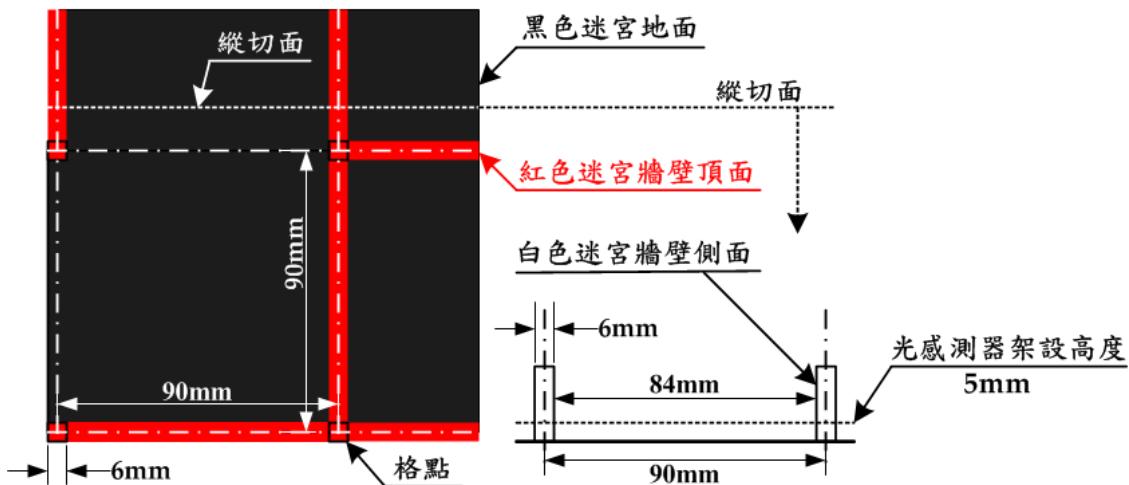


圖 1 半尺寸迷宮的結構圖

3. 迷宮的「起點」位於四個迷宮角落方格的其中一個。電腦鼠會以順時針的方向出發。迷宮「終點」的大小與位置會在每一次比賽前決定。迷宮「終點」的入口會以一組 x-y 座標來表示(參考圖 2 關於迷宮終點入口處的說明)。
4. 在每一個迷宮方格四個角落上 $0.6 \times 0.6 \times 2.5\text{ cm}$ 的小方柱，稱為「格點」。除了迷宮終點區域內的格點外，每一個迷宮方格的格點都必須至少接觸一面迷宮牆壁。迷宮的外圍全部以迷宮牆連接起來。

電腦鼠的規定

1. 電腦鼠必須要能獨立自行控制，並且不能用燃燒的方式取得能源。
2. 電腦鼠在迷宮地面上的投影不應超過 12.5cm 的正方形，高度沒有限制。如果電腦鼠在行進間會改變幾何結構的話，也必須符合上述的規定。
3. 電腦鼠在迷宮中行進時，不得在後方遺留任何的東西。
4. 電腦鼠不得躍過、攀登、損傷或破壞迷宮牆面。

競賽規則

1. 電腦鼠從迷宮的起點運動到終點的最短時間會被紀錄為正式的比賽時間。每一次的競賽都是根據電腦鼠的運動時間、獨立性，以及如何完成最短的運動時間來評量參賽者的成績。評量的標準會在每壹次比賽前決定。

2. 迷宮的組成架構一但揭露給所有的參賽者知道時，操作員就不能再對電腦鼠輸入任何的資訊。也不可以在比賽進行中，藉由開關等裝置修正、或消去部分的迷宮相關資訊。

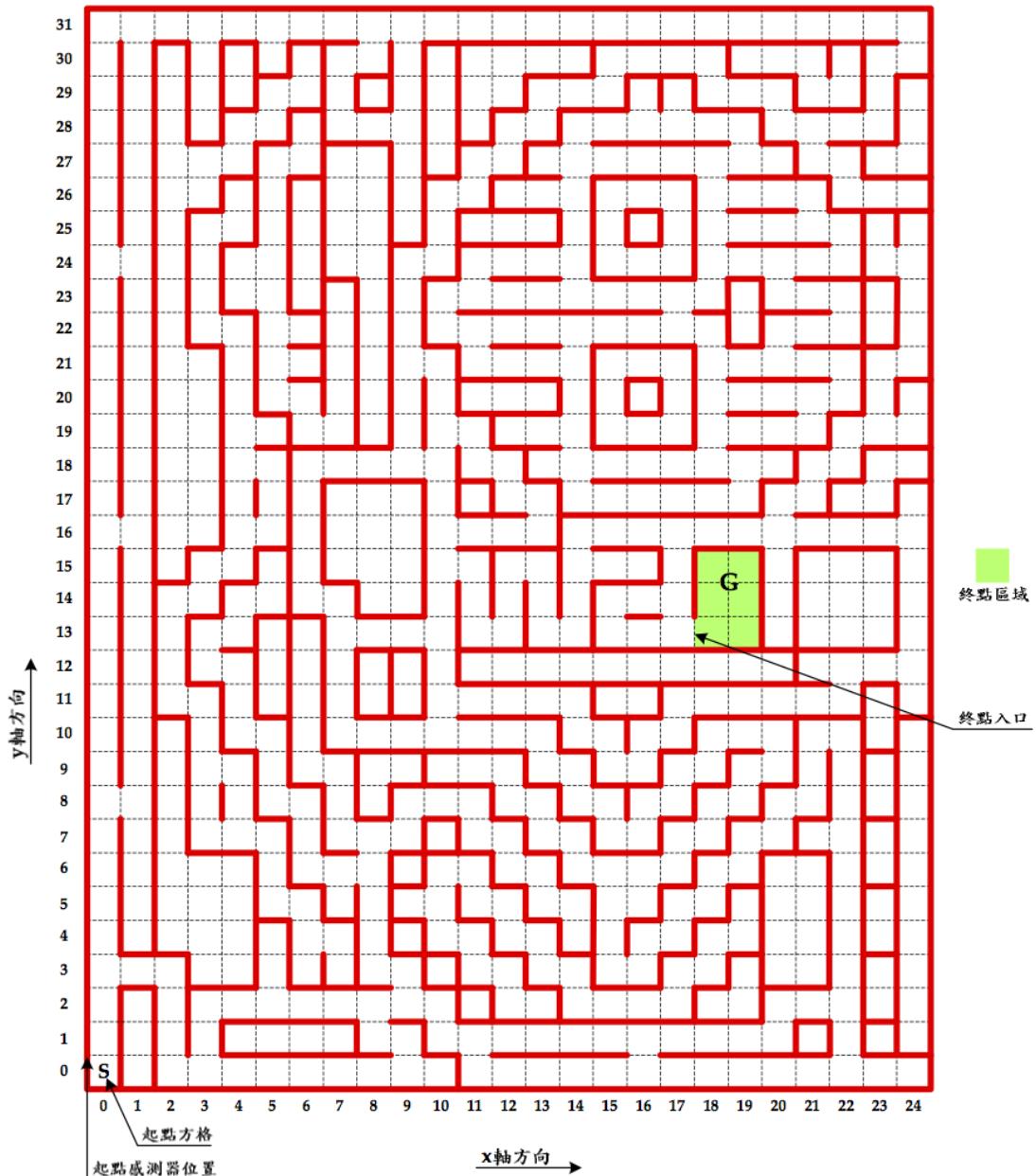


圖 2 半尺寸電腦鼠迷宮競賽範例。半尺寸電腦鼠離開迷宮起點後的前方為 y 軸的方向，右手邊則為 x 軸的方向。因此迷宮終點的入口處在 $(x18, y13)$ 。

3. 電腦鼠在迷宮中每一次的探索都是由起點開始，再回到起點時停止至少兩秒鐘後結束，或者是經由裁判的允許結束該次的探索。

4. 如果電腦鼠返回到起點後，要自動地再開始進行下一次的探索，必須在起點的位置停止 2 秒以上。
5. 電腦鼠在迷宮的探索過程中，參賽者不得碰觸電腦鼠，除非有大會裁判長的指示，獲得到現場裁判的允許。如果電腦鼠有明顯的故障，大會裁判長得接受參賽者的請求中止電腦鼠該次在迷宮中的探索。但參賽者如果是其他的理由提出中止電腦鼠探索的請求時，必須是在消去全部記憶的迷宮資訊條件下，才能獲得許可。
6. 每一次的競賽中，電腦鼠探索迷宮的時間限制，都會在比賽前決定，最多是 15 分鐘。在指定的時間內，電腦鼠最多只能探索迷宮 5 次。
7. 電腦鼠只有在離地板 2.5cm 以內的部分全部進入終點區域時，才能被認可完成迷宮的探索。然而電腦鼠的運動時間則是由起點位置的感測器感應到電腦鼠起算，直到迷宮中點區域入口的感測器感應到同一隻電腦鼠為止的這一段時間。
8. 迷宮所在位置的亮度、溫度與溼度與一般的室內環境是相同的，參賽者不得要求調整場地的亮度。
9. 比賽的大會裁判長，如果認為必要，有權要求參賽者針對他(她)的電腦鼠提出說明。大會的裁判長同樣有權要求中止電腦鼠在迷宮中的探索、宣告參賽者失去資格，或是給予參賽者其他必要的指示。
10. 比賽進行中，參賽者不得載入程式至電腦鼠中，或是更換唯讀記憶體。參賽者也不可以利用連接到電腦鼠上的與電腦鼠本體無關的發展工具或是控制盒給予電腦鼠有關程式執行的指示。
11. 參賽者可以在比賽進行中，利用膠帶等工具去除粘著在輪胎上的灰塵或碎片，但不得使用溶劑等物品來增加摩擦力。
12. 電腦鼠在每一次的迷宮探索過程中，都可以持續搜索迷宮的結構，即使前幾次的探索已經到達過終點。在這一個情形下，當次探索運動時間的計算是紀錄由起點出發後電腦鼠第一次到達迷宮終點的時間。
13. 假使電腦鼠回到起點後，在兩秒鐘內就開始進行下一次的探索，雖然被認定已經開始另一次的探索，但卻是一次無效的探索。

14. 參賽者不得因調整或其他目的將電腦鼠放置在迷宮起點以外的方格中，除非是電腦鼠自行在探索過程中到達。
15. 迷宮是以一般的工作精度製作，因此會有某種程度尺寸上的誤差。同時，為了讓迷宮在組裝與收納上更有彈性，除了在迷宮牆及地板的銜接處會有大約 1mm 左右的間隙外，也可能會有褪色、變色與污漬等不一致性存在。
16. 水平照射的紅外線感測器是架設在離迷宮地面大約 0.5cm 的高度上。迷宮起點的感測器是裝設在迷宮起點與下一個迷宮方格的交界上。
迷宮終點區域內迷宮牆壁的內壁面為紅色。而且迷宮終點的計時器是裝設在迷宮終點區域的入口處。

