

# 智慧型越獄路線規劃

國立臺北大學  
通訊工程研究所  
鄭意平 s79469204@  
webmail.ntpu.edu.tw

國立臺北大學  
電機工程研究所  
詹景裕  
gejan@mail.ntpu.edu.tw

國立臺北大學  
資訊工程學系  
莊東穎  
juang@mail.ntpu.edu.tw

## 摘要

機器人路徑規劃一直是個重要的研究方向。過去傳統的路徑規劃通常將機器人外型與複雜的現實環境作簡化的假設，以便求出由起始點至終點的最短路徑。在考慮了機器人的外在型態與環境的互動而有學者定義了鋼琴移動問題(Piano Mover's Problem)，機器人除了規劃最短路徑的同時還必須在經過狹窄道路時具備旋轉的能力。為了模擬真實的環境與減少簡化性的假設，我們以可移動與固定障礙物與權重區域還有探照燈建構出越獄場景，機器人必須在此場景中規劃出且能夠閃避探照燈、障礙物以及盡量避開高權重區域或是重力、風速、水流等自然因素而由起始點至終點的越獄路徑，我們將此問題定義為 Prison Breaker's Problem。本文的路徑演算法是由 High Geometry Maze Router 架構發展而來，在二維網格圖上的單一網格機器人能夠避開移動的探照燈光區並且考慮環境中的靜態與移動的權重區域與障礙物，必要時尋找掩蔽物與等待適當時機再繼續規劃通往終點的路徑。其時間複雜度為  $O(dN+qN)$ ，其中  $N$  為地圖的網格數目， $d$  為探照燈來回轉動週期， $q$  是為了達到動態規劃的目的由起始點至終點(假如可到達終點)含等待所需的時間，其值大於等於路徑總長度。

**關鍵詞:** 障礙物、權重區域、探照燈、High Geometry Maze Router, Piano Mover's Problem, Prison Breaker's Problem.

## 一、緒論

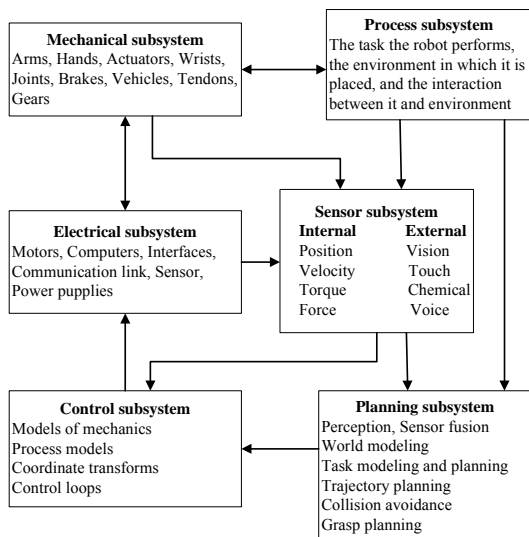
機器人的自動導航方面路徑規劃是一項重要的課題，當給予地圖資訊，機器人的導航系統必須快速的計算出由起始點到終點的路徑，無須人為的操縱與干涉。應用方面除了解決機器人避障路徑搜尋問題，也常應用於 PCB 與 VLSI 的佈線問題 [17][18]、電子海圖顯示資訊系統(Electronic Chart Display Information Systems, ECDIS)的路徑搜尋問題 [10]、道路地圖的繞行問題 [5]，電腦遊戲人工智慧 [16] 等。

日本機器人協會(Japanese Robot Association)將機器人依照其智慧程度分為下列六個等級：

1. 手動操作設備(Manual handling devices)：機器的功能由操作者控制的設備，使用於人類無法直接工作的惡劣環境下以及移動重物、危險物等。
2. 固定順序型機器人(Fixed sequence robots)：機器人依照事前定義的功能順序執行並且不容易重新改變其程序，經常使用於簡單且重覆性高的工作。
3. 可變程序型機器人(Variable sequence robots)：機器人依照事前定義的功能順序執行且易於重新定義功能。
4. 記憶再現型機器人(Playback robots)：操作者先是手動操作一次工作內容，機器人在經過一次記憶，就能夠開始重複操作者的動作。
5. 數值控制型機器人(Numerically controlled robots)：人類事先以程式語言下達作業步驟(位置、順序等)後，會遵照指令的內容進行作業的機器人。
6. 智慧型機器人(Intelligent robots)：依人工智能決定動行的機器人。

機器人的構成方面可分解為各種不同的子系統 [11]，如圖一所示。其中規劃系統(Planning subsystem)，也是本文所要探討的領域，由作業處理系統(Process subsystem)與感測系統(Sensor subsystem)接收資料後，能夠規劃出適當的方法來解決問題再將訊息通知控制系統(Control subsystem)來執行程式規劃的結果。本篇論文所討論越獄路線規劃中的路徑規劃機器人在規劃子系統中即屬於一種人工智慧型機器人，在面對各種不同的地形環境，例如：窪地、平坦路面、障礙物等，能夠先作觀察與判斷並決定出下一時間點所要移動的方向與位置。

而路徑規劃問題的研究在過去曾提出過各種不同的方法，常見的有 Cell decomposition [7][26] [12][15]、Road maps [24][13]和 Potential field [14] [2][25]。其中 Cell decomposition 是先將空間劃分成小區域(cell)，接著在彼此相鄰的小區域連結起來，組成類似 graph 的結構，然後在上面搜尋路徑。而 Road maps 的方式可分為建立連結的前處理階段(preprocessing phase)及計算路徑的查詢階段(query phase)兩個階段。Potential field 的方法是建立人工位能場，利用 robot 有從由高位能往低位能移動的趨勢，來找出路徑等方法。但是針對於機器人的外型以及複雜的環境各衍生出不同的問題。



圖一：機器人系統組成圖

在處理路徑規劃問題，有關地圖資訊的部份，主要可分為向量式資料結構(Vector Data Structure)及網格式資料結構(Raster Data Structure)兩大類。由於使用的資料結構不同，相對用來找尋最短路徑的演算法，也將會完全不同。由於網格式資料結構以有系統之固定間距的抽樣來表示地面資訊，適於空間複雜、變化性高之環境。地理資訊的取得有許多種方式，如加裝於機器人之上的感測器(Sensor)，所得到的資訊是屬於區域性範圍的資料，而全域性資料的擷取則是使用空拍圖的方式來處理，能夠通盤考量整個環境影響因素。機器人的型態又分為單一網格式機器人(Single-Cell Robot)與 Robot Domain (Multi-Cell Robot)，其中的分別在於網格式圖上所包含的範圍以及與環境的互動。在地圖上根據不同的特性，我們可以將網格式分為可通行與不可通行的區域。以各又可分為靜態、可移動、可轉動與可變形的區域，在自然環境影響的方面還包含重力(Gravity)、風速(Wind)、水流(Water flow)等。在外形方面單一網格式機器人則是將佔有區域，化簡至單一網格式，不需考慮外在形態的問題，而能專注於路徑的規劃。為了考慮到多樣化的環境，過去的路徑規劃提出過不同的研究[1][4][8]，其中包括可移動和靜態的障礙物與權重區[19][20][27]，本文將加入探照燈(Searchlight)利用兩點間通視(Light of Sight)的原理建構出光區(Illuminated Region) [3]，不斷來回照射，外型會隨著時間點的不同而有所改變，是一種大範圍的移動與阻擋。

在路徑規劃的研究上對於機器人外型與複雜的環境因素的實際模擬為了便於研究，通常必須做出簡化性的假設。一般地圖架構在網格式圖上的機器人常用單一網格式機器人來簡化機器人的實際外型。在多變的環境下，地圖上存在的不仅是自由空間和靜止不動的障礙物。在真實環境下，還有可移動的障礙物或是不易通過的區域，以及各種自然影響因素，例如風速，水流與重力等。

因機器人外在形態在環境中佔有多個網格式的關係，為了具體模擬機器人與環境的互動關係，除了規劃最短路徑的同時還必須考慮在穿越狹窄道路時還必須具備旋轉的能力，此問題被定義為 Piano mover's Problem [21][22][23]。而在路徑規劃的場景環境，一般而言會作假設性的簡化，但是在真實的自然環境並非如此，如障礙物、風速、水流、重力等自然因素的影響。本文以可移動與不可移動的障礙物與權重區域再加入探照燈所建構的場景，減少簡化性的假設，機器人必須在此場景中規劃出能夠閃避探照燈、障礙物與權重區域由起始點至終點的越獄路徑。我們將此問題定義為 Prison Break's Problem。

本文所建構的 Prison Break's Problem 是以空拍圖的方式取得地理資訊並且以二維的網格式資料結構儲存環境資料，在機器人物體的設定方面，本文將以單一網格式機器人為例，規劃出越獄路線。環境方面除了有可通行的自由空間還包含了權重區域、障礙物、探照燈光區等影響因素，其特性以表一所示。在此問題中的單一網格式機器人具有觀察全局的能力，在出發之前蒐集探照燈的照射週期，分辨出可供躲避探照燈的掩護區。在規劃越獄路徑時遭受探照燈光區的全面攔阻時，必須將安全的掩護區當作中繼點，並且等待適當時機繼續規劃通往終點的路徑。其時間複雜度為  $O(dN+qN)$ ，其中  $N$  為地圖的網格式數目， $d$  為探照燈來回轉動週期， $q$  是為了達到動態規劃的目的由起始點至終點(假如可到達終點)含等待所需的時間，其值大於等於路徑總長度。若是存在一條越獄路徑，在我們的方法中，機器人能夠規劃出一條閃避探照燈光區、障礙物、或是高權重區域。

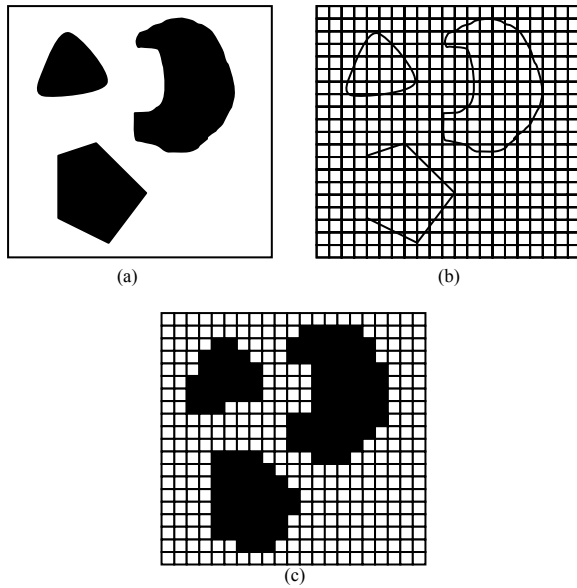
越獄路線規劃問題可分解為兩部份：建立環境模型與動態環境下的路徑規劃，將會在本文第二節來介紹，其中包括地圖基本架構、過去所提出過的各類型障礙物與探照燈光區及核心 HGMR (Higher Geometry Maze Router)演算法[9]的改進，包括演算法本身與其變數定義、時間與空間複雜度的分析。第三節將結合權重區域、障礙物，探照燈光區模擬越獄環境並且展示程式模擬範例的過程與結果；最後在第四節總結並提出未來研究方向。

## 二、動態環境中的各種影響因素

本節將介紹在動態環境下所包含的各種型態障礙物，其中包括權重區域、障礙物與光區，並且在場景內加入時間的觀念以 HGMR 為核心發展出

表一：地圖環境特性

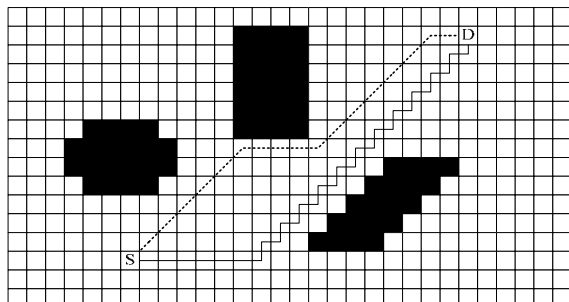
		靜態	移動	轉動	變形
可通行	自由空間				
	權重區域	v	v		
不可通行	障礙物	v	v		
	探照燈光區	v		v	v



圖二：歐式空間轉換成網格圖(a)場景之正上方所拍攝的空拍圖。(b)將空拍的歐式空間與網格圖作映對，把障礙物網格化。(c)轉換成網格圖的結果。

動態環境的路徑規劃。首先問題可分解為兩部份：建立環境模型與動態環境下的路徑選擇與規劃。建立環境模型的部份，首先，取得一個靜態或動態環境的場景之正上方空拍的歐式空間(Euclidean space)，並且將其自由空間與障礙物所涵蓋的範圍分別將其轉換成網格圖中不同屬性的網格，如圖二(a)所示，地圖中包含不同形狀的障礙物與可通行的自由區域，圖二(b)將(a)圖映對到網格圖上，將歐式空間作轉換，其中不滿一個網格的區域仍然將其當作障礙物空間，圖二(c)為轉換結果，黑色區域代表障礙物在網格圖所包含的區域，白色部份則是可安全通行的自由空間。如此一來，複雜又不易表示的幾何圖形透過此法轉換成簡單、結構化且易於運算的網格，有利於之後演算法計算最佳路徑的處理。

路徑選擇與規劃的部份，本文使用建立在八向連接網格的最短路徑演算法 Higher Geometry Maze Router 發展出適合動態環境的路徑規劃處理。利用八向路徑的結構相較於四向的環境能夠將



圖三：八向與四向路徑規劃比較。虛線為八向路徑規劃演算法所計算出的最短路徑，與實線的四向路徑規劃演算法相比較，轉彎次數少，總長度較短。

轉彎次數降低(圖三)，並且降低移動總距離，在實務的應用下有正面的幫助。

### (一)變數表

在處理隨時間不斷的變化的光區、權重區域與障礙物時，每一固定時間內必須取得環境內不同時間的網格圖，經由每一次計算出的最佳路徑來決定下一時間點的移動方向與位置直到抵達終點為止。以下，表二將介紹處理不同型態障礙物的路徑規劃演算法所使用到的變數與其說明。

### (二) 靜態、可移動權重區域與障礙物

在我們研究室過去所發表過有關於具有靜態、可移動權重區域與可移動障礙物環境下的路徑規劃。[19][20]在本文中將其副程式繼續沿用，其中包含有 *Acquire\_MaxARW & Update\_Max\_Cir()*、*Update\_AT with\_WR (C<sub>i,j</sub>(t), new\_AT<sub>i,j</sub>(t))*、*Update\_C<sub>i(h),j(h)</sub><sup>Path</sup>(t)(C<sub>i,j</sub>(t), C<sub>i,j</sub>(t))*、可移動權重區域初始化設定、可移動權重區域變數初始化、環境初始值設定與預估障礙物移動領域擴張等。整體耗費的時間複雜度由起點至終點所時間  $q$  乘上每一次計算最短路徑為  $O(N)$ ，所以為  $O(qN)$ ，而空間複雜度仍為  $O(N)$ 。

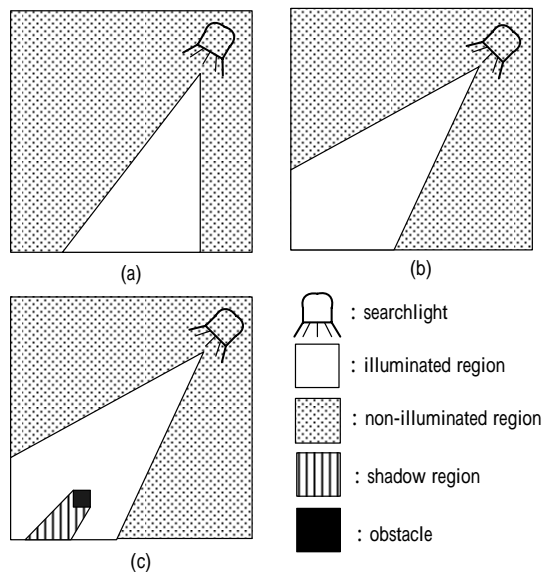
### (三) 探照燈光區

本節將介紹單一網格機器人在一個包含探照燈光區的路徑規劃，並且在場景內加入時間的觀念，而能夠處理動態環境下不斷移動光區位置的障礙物避碰與路徑規劃。本文所討論探照燈所產生的移動光區也是如此隨著時間移動而改變位置，不僅如此，移動光區還因為照射位置的不同，所涵蓋的區域範圍也有所改變如圖四所示，圖四(a)由光源所發射出的探照燈光區，經過一段時間光源以固定的光區展開角度轉向至圖四(b)的位置，由此(a)、(b)兩圖可知，其光區涵蓋範圍不盡相同；若照射到一般障礙物時，產生的形變如圖四(c)所示。

表二：路徑規劃演算法變數表

變數	型態	說明
$AT_{i,j}(t)$	浮點數	<i>Arriving Time</i> ，為起點至 $C_{i,j}(t)$ 所需的時間。
$C_{i,j}(t)$	二維陣列	<i>Cell</i> ，二維網格圖分割成 $I \times J$ 個網格， $(i,j)$ 為空間座標，是為二維陣列的索引值，範圍： $0 \leq i \leq I-1, 0 \leq j \leq J-1, t$ 表時間。
$d$	整數	探照燈來回轉動週期。
$D$	二維陣列	<i>Destination</i> ，表示網格中機器人移動之

		終點位置。
$h$	整數	Single-Cell Robot 行進路徑的移動起始點。範圍： $0 \leq h \leq H-1$
$H$	整數	Single-Cell Robot 由路徑起點到終點網格數。
$index$	整數	$LL$ 串列指標陣列的索引值， $0 < index < Max\_Cir(t)$ 。
$ISO_{ij}(t)$	整數	<i>Illuminated / Shadow / Other Region</i> ，觀察探照燈光區移動後標示出每個時間點 $t$ ，網格的三種狀態。 <i>Illuminated</i> 代表光區所在位置， <i>Shadow</i> 表示遮蔽區，其他非光區所在位置或遮蔽區則以 <i>Other</i> 表示。
$LL^{BacktrackingPath}$	鏈結串列	紀錄回溯後所得路徑上網格的串列。
$LL_{index}(t)$	鏈結串列	表歸類排序後的相鄰網格串列。
$Max\_Cir(t)$	整數	Circulate， $LL$ 串列指標陣列的 <i>Bucket</i> 數目， $t$ 表時間點。
$N$	整數	地圖的網格數目。
$q$	整數	由起始點至終點(假如可到達終點)含等待所需的時間。
$RW_{i,j}^l(t)$	浮點數	<i>Regional Weight</i> ， $RW_{i,j}^l(t) = \infty$ ，表示 $C_{i,j}(t)$ 為障礙物， $RW_{i,j}^l(t) = 1$ ，表可通行且平滑的自由空間區域，若 $RW_{i,j}^l(t)$ 大於 1 的正值，表各種不同權重的自由空間區域， $t$ 表時間點，考慮動態時適用， $l$ 變數用於多組權重區域的分組。
$S$	整數	<i>Source</i> ，表示網格中 Single-Cell Robot 之起始位置。
$TL$	鏈結串列	暫存串列。



圖四：探照燈光區的模擬。(a)由光源處發射出三角形的光區。(b)光區在移動時因照射位置的不同，其涵蓋範圍也不相同。(c)當光區照射到一般障礙物時，會和現實狀況一樣產生遮蔽的效果。

在地圖上探照燈光區的模擬，是採用通視演算法(Light of sight)，假設平面空間有光源  $R(x_r, y_r)$  與照射標的  $M(x_m, y_m)$  兩點，以  $R$  點及  $M$  點作一線段，稱此線段為  $L_{RM}$ ，位於在  $L_{RM}$  之上任何點  $(x, y)$  都能以直線參數式方程式來決定，其表示式如方程式一所示。

$$\begin{aligned} x &= x_r + t(x_m - x_r) \\ y &= y_r + t(y_m - y_r) \end{aligned} \quad (1)$$

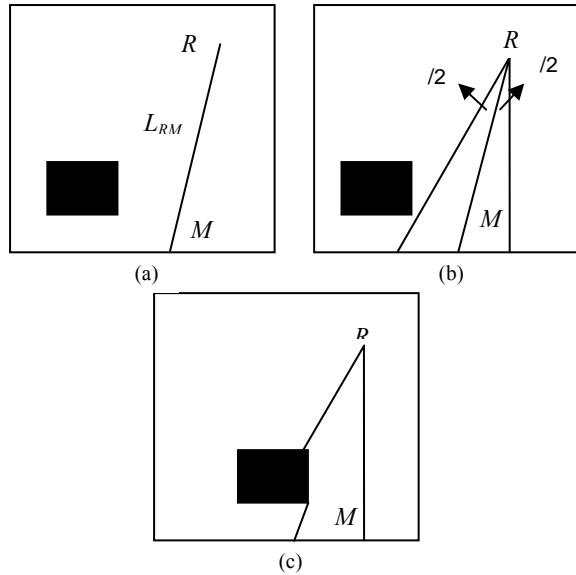
再利用座標轉換，光區標的點為  $M$ ，繞著光源  $R$  點移動任意角度  $\theta$ ，利用下列座標轉換方程式二求得光區照射標的點  $M_1(x_1, y_1)$ ，由旋轉過後求得的  $M_1$  點與光源  $R$  點兩點可得出一直線方程式，此直

$$\begin{aligned} x_1 &= (x_m - x_r) \cos \theta - (y_m - y_r) \sin \theta + x_r \\ y_1 &= (x_m - x_r) \sin \theta + (y_m - y_r) \cos \theta + y_r \end{aligned} \quad (2)$$

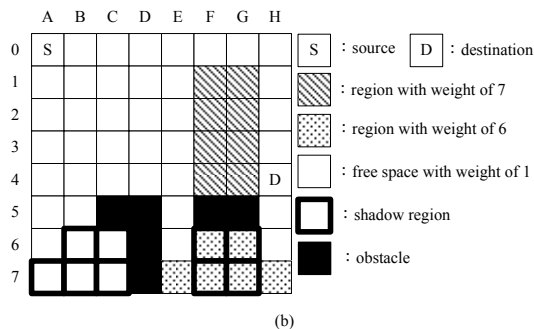
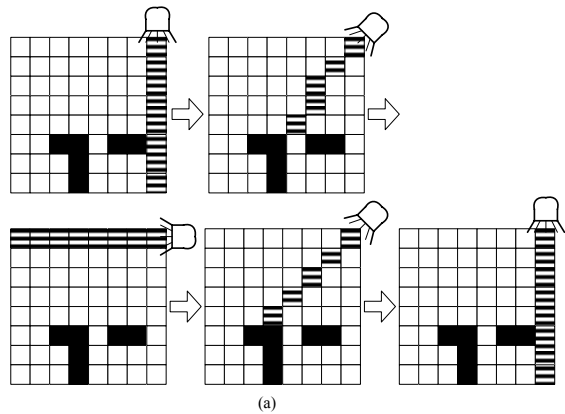
線方程式與  $L_{RM}$  和地圖邊界所涵蓋的範圍，再利用通視演算法，偵測障礙物，將障礙物後的非光區照射範圍還原回自由空間，即為光區真正涵蓋的範圍，如圖五所示。

演算法的一開始，機器人必須先觀察光區障礙物的移動走向與位置以及蒐集可供躲避光區的安全區域，如圖六所示。接下來呼叫由 HGMR 所修改而來的”障礙物避碰路徑規劃”來計算出最短的路徑(STEP 3)。由”障礙物避碰路徑規”所回傳的資訊有二：到達終點與無法到達終點。若是傳回到達終點的訊息時代表在往終點的路上可以順利躲

避光區；如果傳回無法到達終點的訊息時，表示往終點的路徑已被光區所阻斷，如圖七所示。這時必須規劃出一條能夠躲避探照燈光區的次佳路線。在遭遇移動的探照燈光區的情況下策略有兩步驟：一、增加中繼點，前往可順利到達的安全區域。二、

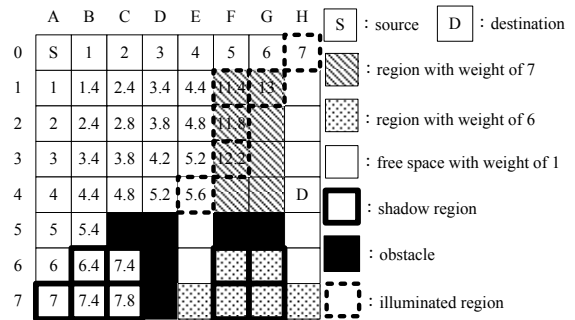


圖五：探照燈光區模擬過程。(a)用通視演算法，光源  $R$  點及照射標的  $M$  點作一線段  $L_{RM}$ 。(b)以  $\theta$  角度展開光區。(c)障礙物後遮蔽區還原回自由空間。

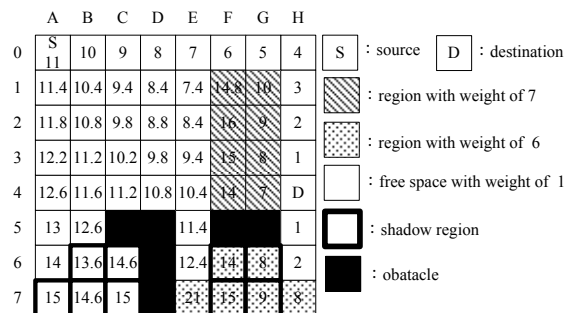


圖六：越獄路線規劃初始動作。(a)機器人必須先觀察探照燈光區的移動走向與位置，光區由左上圖開始，隨著時間點的不同逐漸移動，再返回起始位置，作週期性的移動。(b)由觀察探照燈光區移動的資訊，蒐集可供躲的掩護區域。

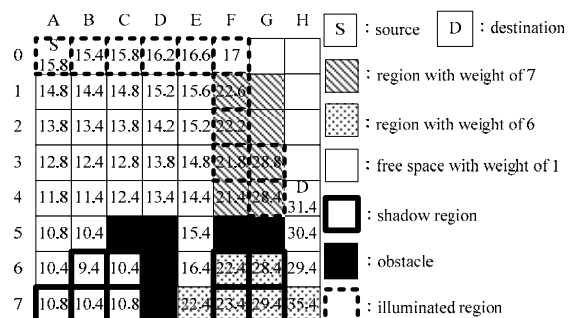
在安全區域等待適當的一段時間，再繼續往原訂的終點出發。在圖七中所顯示出路徑的規劃已被移動光區所攔阻，而無法抵達終點。此時根據一開始所蒐集到的安全區域範圍加入路徑規劃的經過之地，再挑選何處當作躲避之處。我們利用 HGMR 來找出離終點最近且不與光區碰撞的網格(圖八)，再將其點作為必經之地。抵達安全區域後繼續呼叫障礙物避碰路徑規劃(STEP 5)，還未能規劃出到終點的路徑則返回 STEP 4.2，若原地等待不動



圖七：躲避光區的計算結果。依照障礙物避碰路徑規劃所傳回的結果是光區會在網格中的  $AT$  值於相同時間抵達產生碰撞而無法規劃出由起點到終點的路徑。



圖八：利用 HGMR 來找出離終點最近且不與光區照射的網格。由  $D$  點網  $S$  點洪氾，計算出  $AT$  值，並從中挑選出能不與光區碰撞且離終點最近且不與光區碰撞的網格，由圖七得知， $B6$  將選為安全躲避區域。



圖九：抵達安全區域計算結果。 $B6$  相較於圖七，等待了 3 個時間單位，才規劃出通往終點  $D$  的路徑，其路徑即為能夠躲避光區規劃結果。

的情況其  $AT$  值必須增加一個單位。本範例在圖九表示在 B6 位置等待了三個單位時間再執行障礙物避碰路徑規劃而能夠到達終點，此路徑即為能躲避光區與固定障礙物的路徑並且考慮權重區域的不同的結果。在針對移動的權重區域障礙物方面，只要經過每一次路徑的重新計算，依照規劃的結果能成功躲避探照燈光區以及可移動障礙物。

演算法 1：探照燈光區迴避路線規劃

BEGIN

STEP 1：將預定的起始點與終點、障礙物、定點式可旋轉的探照燈光區起始位置和邊界的二維地圖等參數輸入至“障礙物避碰路徑規劃”。

STEP 2：將光區一個週期內所行經的 cell 依時間與不同屬性存入  $ISO_{ij}(t)$ 。

STEP 3：呼叫 HGMR。

STEP 4：判斷是否傳回“已到達終點”訊息。

STEP 4.1：若傳回“已到達終點”訊息，由最短路徑  $LL^{Backtracking Path}$  繼續往終點前進。未達終點前再返回 STEP 3 繼續規劃路徑。

STEP 4.2：若傳回“未到達終點”訊息，則尋找“中繼點”。

STEP 4.3：找出離終點最近的掩護區，而且是 STEP 3 執行後，由起點所能到達的範圍。

STEP 4.4：執行 HGMR 找出由起點到 STEP 4.3 所找出的掩護區，並儲存由起點往中繼點的路徑。

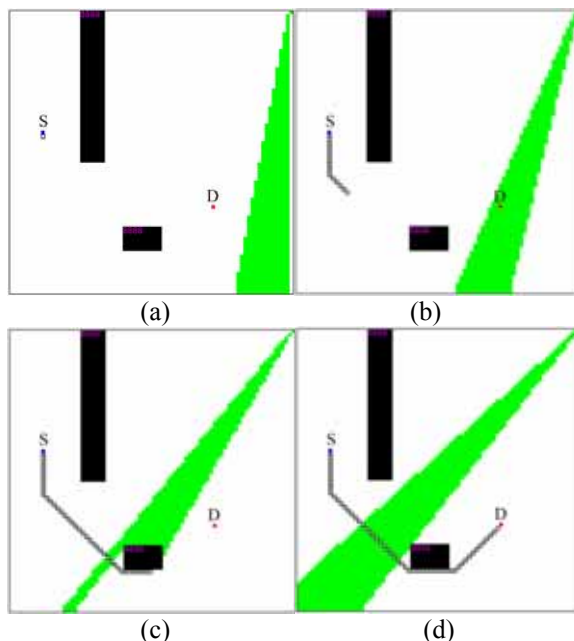
STEP 5：到達中繼點後繼續執行 HGMR 並且將起點改為剛抵達的中繼點，終點變更回預定的終點。

STEP 5.1：若傳回“未到達終點”訊息則返回 STEP 5.2，更進一步如果找到的中繼點與現階段的起點是相同位置的話，則在原地等待一個單位時間， $AT_{ij}(t)$  值加 1 再返回 STEP 4.2，直到抵達先前預定的終點，傳回規劃的路徑。

END{探照燈光區迴避路線規劃}

關於探照燈光區避碰路線規劃是架構於 HGMR，有關空間複雜度，基本上使用了 1 個  $I \times J$  的二維陣列  $C_{ij}(t)$  來儲存網格中  $N$  個節點的狀態。所佔的空間複雜度為  $O(N)$ 。本文所提出的移動光區，使用了  $AT_{ij}(t)$ 、 $Vis_{ij}(t)$ 、 $ISO_{ij}(t)$ 、 $RW_{ij}^l(t)$  等參數來表示  $N$  個網格狀態，所佔的空間複雜度為  $O(N)$ 。

時間複雜度方面，探照燈光區避碰路線規劃中，STEP 2 將儲存移動光區的資訊到  $ISO_{ij}(t)$ ，其時間複雜度為一個週期內每一次移動所涵蓋的網格數，若一個週期內所觀察到探照燈光區移動的時間點  $d$  次，則為  $O(dN)$ 。STEP 3 啟動 HGMR，其時間複雜度為  $O(N)$ 。STEP 4 及 STEP 5 皆隨著地圖更新  $q$  次，而呼叫障礙物避碰路徑規劃  $q$  次直到 Single-Cell Robot 抵達終點，時間複雜度為  $O(qN)$ 。



圖十：固定障礙物與探照燈構成模擬場景。(a)起始場景  $S$  為起始點， $D$  為終點，探照燈光區由右上角照射，其餘皆為固定式障礙物。(b)往固定障礙物後的安全區域躲避。(c)抵達躲避位置。(d)成功躲避光區，抵達終點。

總時間複雜度為  $O(dN+qN)$ 。

程式模擬結果首先以固定障礙物與探照燈構成模擬場景如圖十(a)所示，其中  $S$  為起始點， $D$  為終點，探照燈由右上角照射出來，其餘物體為固定是障礙物。在圖十(b)中其路徑的規劃並不是直接的往終點  $D$  移動，而是往固定障礙物背後探照燈無法照射到的遮蔽區前進，避免被光區照射到。圖十(c)，抵達安全區域避免由障礙物上方探照燈光區經過的區域行走。在圖十(d)光區離開後路徑的規劃便以最短路徑抵達終點，完成了本次的模擬。由路徑歷史軌跡可以看出，此次的規劃深受探照燈光區所影響，考慮的不僅是固定障礙物的避碰，還有光區的因素才能規劃出一條能夠順利抵達終點的路徑。

### 三、越獄路線規劃

越獄路線規劃中包含探照燈所造成的探照燈光區以及各種可移動和靜態的障礙物。為了使本文的路徑規劃能針對光區與各種不同地形做處理，以 HGMR 演算法為核心，將其擴充發展。在處理隨時間不斷的變化的光區、權重區域與障礙物時，每一固定時間間距內取得環境的網格圖，經由每一次計算出的最佳路徑來決定下一時間點的移動方向與位置直到抵達終點為止。

(一)演算法

演算法的一開始以 2.3 節探照燈光區迴避路線規演算法為主軸，配合呼叫由 HGMR 修改而來

的越獄路線規劃副程式，達成越獄路徑規劃的目的。在越獄路線規劃副程式中為了處理各種型態的障礙物分別呼叫 2.2 節所提出的副程式。演算法如下所示：

演算法 2：越獄路線規劃

Input：包含起始點、終點、障礙物、權重區域、探照燈光區與二維網格圖。

Output：由起始點至終點的最短路徑。

Begin

STEP 1：初始化設定：

Call 環境初始值設定副程式

Call 可移動權重區域初始化設定副程式

Call 可移動權重區域變數初始化副程式

Call 預估障礙物移動領域擴張副程式

將光區一個週期內所行經的 cell 依時間與不同屬性存入  $ISO_{ij}(t)$ 。

STEP 2：搜尋網格途中所有網格之最大  $RW$  值與計算  $Bucket$  的數目

Call  $Acquire\_MaxARW\&Update\_Max\_Cir()$

STEP 3：計算  $AT$  值

STEP 3.1：從  $LL_{index}(t)$  取出座標： $(i, j)$ ，設定  $(i, j)$  的  $Vis_{ij}(t)=TRUE$ 。

STEP 3.2：對  $(i, j)$  計算所有相鄰網格座標： $(i', j')$ ，依照其  $New\_AT_{i',j'}$  值檢查是否存在  $ISO_{ij}(t)$  所紀錄的光區中，是的話，如一般障礙物處理。

STEP 3.2.1：

CASE 1：當  $|i'-i|^2+|j'-j|^2=1$ ，則

Call  $Update\_AT\&Vis\_With\_WR(C_{i',j'}(t), AT_{i',j'}(t)+(1 \times RW_{i',j'}^l(t)))$

CASE 2：當  $|i'-i|^2+|j'-j|^2=2$ ，則

Call  $Update\_AT\&Vis\_With\_WR(C_{i',j'}(t), AT_{i',j'}(t)+(\sqrt{2} \times RW_{i',j'}^l(t)))$

STEP 3.3：重複 STEP 3.1 直到  $LL_{index}(t)$  裡無任何網格的座標。

STEP 4：歸類排序

STEP 4.1：從  $TL$  取出座標： $(i, j)$ ，將  $(i, j)$  存入  $LL_{[AT_{i,j}] \bmod Max\_Cir(t)}$ 。

STEP 4.2：重複 STEP 4.1 直到  $TL$  裡無任何網格的座標。

STEP 5：處理下一層之網格， $index = (index + 1) \bmod Max\_Cir(t)$ 。重複 STEP 3 直到  $LL_{index}(t)$  中都沒有任何網格的座標。

STEP 6：回溯

STEP 6.1：搜尋  $LL^{Backtracking\ Path}$  尾端的網格為  $C_{ij}(t)$  之所有相鄰網格從  $C_{ij}(t)$  所有相鄰網格中，取出  $AT_{ij}(t)$  最小的網格  $C_{i',j'}(t)$  的座標。

STEP 6.2：將  $C_{i',j'}(t)$  的座標存入  $LL^{Backtracking\ Path}$  串列中。

STEP 6.3：若未到達終點，則傳回是否到達終點訊息。

STEP 7：判斷是否傳回”已到達終點”訊息。

STEP 7.1：若傳回”已到達終點”訊息，由最短路徑  $LL^{Backtracking\ Path}$  繼續往終點前進。未達終點前再返回 STEP 2 繼續規劃路徑。

STEP 7.2：若傳回”未到達終點”訊息，則尋找”中繼點”。

STEP 7.3：找出離終點最近，並且是由起點所能到達的範圍的掩護點作為中繼點。

STEP 7.4：找出由起點到中繼點的路徑，並儲存。

STEP 8：到達中繼點後返回 STEP 2 繼續規劃路徑並且將起點改為剛抵達的中繼點，終點變更回預定的終點。

STEP 8.1：若傳回”未到達終點”訊息則返回 STEP 8.2，更進一步如果找到的中繼點與現階段的起點是相同位置的話，則在原地等待一個單位時間， $AT_{ij}(t)$  值加 1 再返回 STEP 7.2，直到抵達先前預定的終點，傳回規劃的路徑。

END {越獄路線規劃}

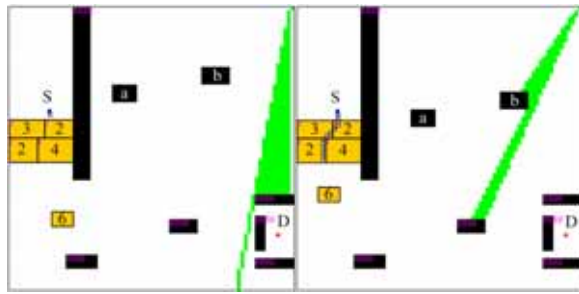
(二)效能分析

關於越獄路線規劃演算法是架構於 HGMR，有關空間複雜度，基本上使用了 1 個  $I \times J$  的二維陣列  $C_{ij}(t)$  來儲存網格中  $N$  個節點的狀態。所佔的空間複雜度為  $O(N)$ 。另外為了處理可移動障礙物與權重區域在我們先前的研究說明出所花費的時間複雜度為  $O(N)$  及本文所提出的移動光區，並且使用了數種參數來表示  $N$  個網格狀態，所佔的空間複雜度為  $O(N)$ ，由所有空間複雜度相加可得出總空間複雜度為  $O(N)$ 。

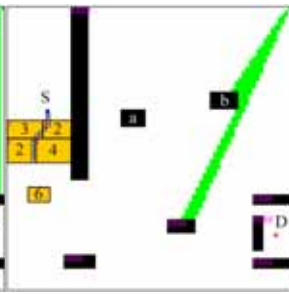
時間複雜度方面，越獄路線規劃中，STEP 1 為對於網格圖中之所有不同屬性的網格作初始化設定所花費時間複雜度為  $O(N)$ ，觀察探照燈光區花費為  $O(dN)$ ， $d$  為探照燈來回轉動週期，STEP 2 為了計算  $Bucket$  的數目所花費的時間複雜度為  $O(N)$ ，STEPs 3 至 8 為 HGMR 的主體部份其時間複雜度為  $O(N)$ ，完成寵路線規劃呼叫越獄路線規劃演算法  $q$  次，時間複雜度為  $O(qN)$ ， $q$  是為了達到動態規劃的目的由起始點至終點(假如可到達終點)含等待所需的時間，其值大於等於路徑總長度。最後可推出總時間複雜度為  $O(dN+qN)$ 。

(三)範例模擬

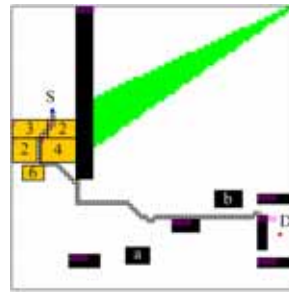
本節以程式模擬監獄場景，整合過去所提出並解決的權重區域與移動式障礙物加上探照燈的移動光區建構成越獄路線規劃的場景如圖十一所示。圖十一(a)中，起始點為  $S$ ，終點為  $D$ ，兩可移動障礙物  $a$  與  $b$ ，四塊權重區域，其權重值以數字標示，其中權重為六的區域可移動，探照燈的光區由右上角照射，其餘為固定障礙物。圖十一(b)，在經過權重區域時必須選出一條花費最少單位時間成本的路徑，所以在進入權重為二的地區後轉而經過權重為三的地區再由權重為二的地區通過，其



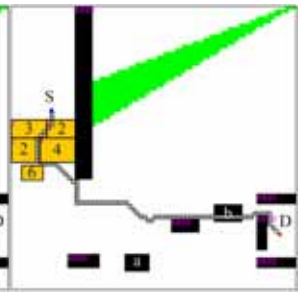
(a)



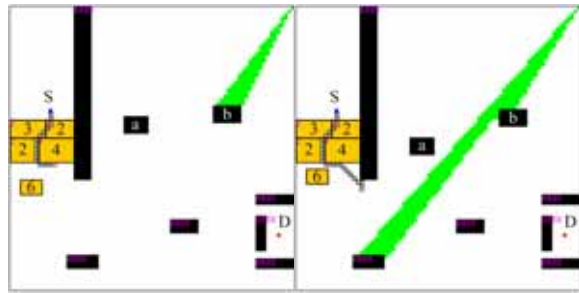
(b)



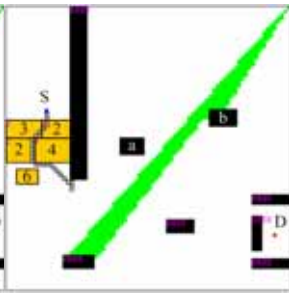
(k)



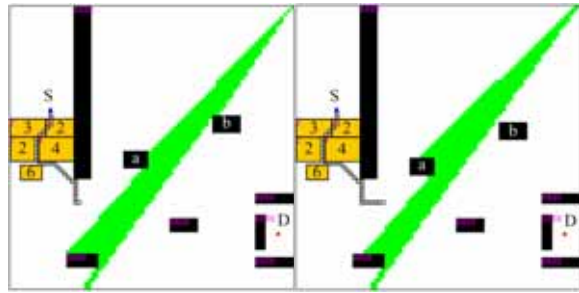
(l)



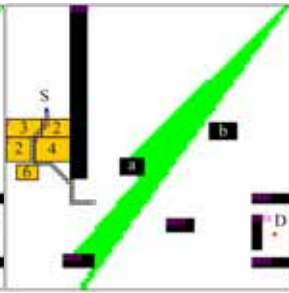
(c)



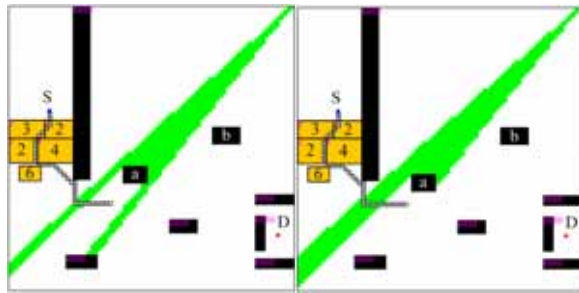
(d)



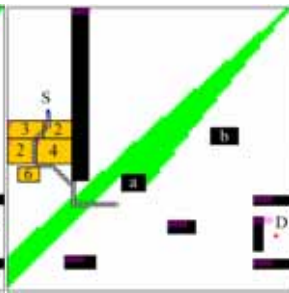
(e)



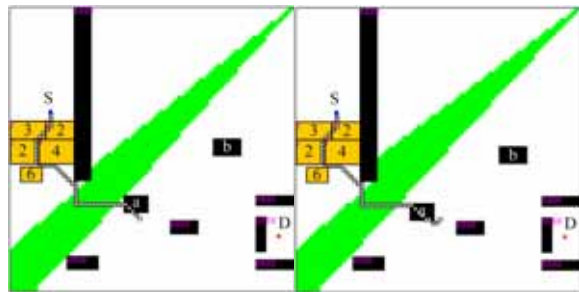
(f)



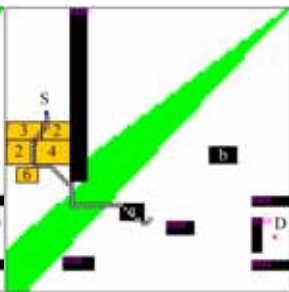
(g)



(h)



(i)



(j)

圖十一：越獄路線規劃的場景。(a)起始點為  $S$ ，終點為  $D$ ，兩可移動障礙物  $a$  與  $b$ ，四塊權重區域，其權重值以數字標示，其中權重為六的區域可移動，探照燈的光區由右上角照射，其餘為固定障礙物。(b)經過權重區域的路徑。(c)閃避可移動權重區域。(d)與障礙物  $a$  同步移動躲避光區。(e)~(h)閃避光區的經過。(i),(j)閃避障礙物  $a$ 。(k),(l)路徑規劃順利到達終點。

原因在行走於權重為五的區域過於耗費單位時間成本，而繞過此區。解決權重區域的問題後，在圖十一(c)離開權重區域繼續往終點前進。在下一階段為了躲避光區，此時計算的結果是往下方障礙物後的掩護區移動如圖十一(d)所示。此時圖十一(e)至圖十一(h)，可移動障礙物  $a$  的經過，將光區遮蔽，形成出了一塊可通行的區域，此時利用此區域往終點前進。障礙物  $a$  移動的過程單一網格機器人必須閃避，如圖十一(i)、(j)，通過障礙物  $a$  後繼續往終點前進，完成本次的路徑規劃，如圖十一(k)、(l)。

#### 四、結論與未來研究

本文所探討的越獄路線規劃，主要是利用不同類型的障礙物，與探照燈光區來模擬情境。其中處理的不只是靜態環境下的路徑規劃，在有包含探照燈光區的動態環境之下，機器人不只是單純的依照單一次路徑規劃的結果來行走，而是必須根據障礙物的移動，在每一次的時間點重新規劃路徑，藉此躲避各種移動的障礙物。在效能方面，對於處理不同的地形環境。為了達到動態規劃的目的，與躲避探照燈光區，所花費時間複雜度為  $O(dN+qN)$ 。

在未來的研究方向，空間環境方面將可朝向三度空間的發展。在三維空間將會帶來許多額外的因素影響其動作行為，例如重力、障礙物的攀爬。在機器人的擬態，可發展成飛行器的行為模式。不同於在地面上行走的機器人，在三度空間中飛行器可飛越障礙物與更多不受地面因素影響的各種行為。探照燈方面可以由固定光源的探照燈改變成移動式，模擬警衛巡邏的環境，更接近真實情況。



## 參考文獻

- [1] N. Aggarwal and K. Fujimura, "Motion Planning Amidst Planar Moving Obstacles," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2153–2158, May 1994.
- [2] J. Chuang and N. Ahuja, "An analytically tractable potential field model of free space and its application in obstacle avoidance," *IEEE Trans. on System, Man, and Cybernetics, part: B*, vol. 28, no. 5, pp. 729–736, Oct. 1998.
- [3] K.Y. Chang, G.E. Jan, W.Q. Luo and C.P. Yin, "The optimal path-planning for mobile vehicles with GPS on 3-D electronic maps," *Journal of Aeronautics, Astronautics and Aviation*, Series B, Vol. 38, No. 1, pp. 63-70, 2006.
- [4] K. Fujimura, "Motion Planning Using Transient Pixel Representations," *Proceedings of the 1993 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 34–39, May 1993.
- [5] J. Fawcett and P. Robinson, "Adaptive Routing for Road Traffic," *IEEE Computer Graph*, vol. 20, no. 3, pp. 46-53, June 2000.
- [6] Y.K. Hwang and N. Ahuja, "Gross Motion Planning - A Survey," *ACM Computing Surveys*, pp. 219-291, 1992.
- [7] P.E. Hart, N.J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.
- [8] Z. Huiliang and H.S. Ying, "Dynamic map for obstacle avoidance," *Proceedings of the 2003 IEEE Intelligent Transportation Systems*, vol. 2, pp. 12-15, Oct. 2003.
- [9] G.E. Jan, K.Y. Chang and I. Parberry, "A New Maze Routing Approach for Path Planning of A Mobile Robot," *Proceedings of The IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Kobe, Japan, vol. 1, pp. 552-557, July 2003.
- [10] G.E. Jan, M.B. Lin and Y.Y. Chen, "Computerized Shortest Path Searching for Vessels," *Journal of Marine Science and Technology*, vol. 5, no. 1, pp. 95-99, June 1997.
- [11] P. John, "Introduction to robotics," *McKerrow*, pp. 123-124, 1991.
- [12] C.Y. Lee, "An algorithm for path connections and its applications," *IRE Trans. Electron. Comput.* EC-10, pp. 346–365, 1961.
- [13] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars, "Probabilistic Roadmaps for Fast Path Planning in High-Dimensional Configuration Spaces," *IEEE Transactions on Robotics and Automation*, 12:566-580, 1996.
- [14] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Rob. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
- [15] D. Leven and M. Sharir, "An efficient and simple motion planning algorithm for a ladder moving in two-dimensional space amidst polygonal barriers," in *Proc. 1st ACM Symp. Computational Geometry*, Nice, France, pp. 1208–1213, 1997.
- [16] M.V. Lent, "Game Smarts," *IEEE Computer Society*, vol. 40, Issue 4, pp. 99-101, Apr. 2007.
- [17] Y.L. Lin, Y.C. Hsu and F.S. Tsai, "Hybrid Routing," *IEEE Trans. on CAD*, pp. 151-157, Feb. 1990.
- [18] K. Mikami and K. Tabuchi, "A Computer Program for Optimal Routing of Printed Circuit Connections," *IFIPS Proc.*, vol. H-47, pp. 1475-1478, 1968.
- [19] J.L. Shih, G.E. Jan, C.M. Su and F.Y. Lin, "Path Planning on Raster in Environments PART I: Static and Movable Weighted Regions," *Conference of Intelligent Living Technology*, pp.727-734, Taiping, Taichung County, Taiwan, June 2006.
- [20] J.L. Shih, G.E. Jan, C.M. Su and F.Y. Lin, "Path Planning on Raster in Environments PART II: Movable Obstacle," *Conference of Intelligent Living Technology*, pp.1174-1181, Taiping, Taichung County, Taiwan, June 2006.
- [21] J.T. Schwartz and M. Sharir, "On The Piano Movers' Problem: I. The Case of A Two-Dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers," *Comm. on Pure and Applied Math.*, Vol. 36, pp. 345-398, 1983.
- [22] J.T. Schwartz and M. Sharir, "On the Piano Movers' Problem: II. General Techniques for Computing Topological Properties of Real Algebraic Manifolds," *Advances in Applied Math.*, Vol. 4, pp. 298-351, 1983.
- [23] J.T. Schwartz and M. Sharir, "On The Piano Movers' Problem: V. The Case of A Rod Moving in Three Dimensional Space Amidst Polyhedral Obstacles," *Technical Report 083 R13*, New York University, Courant Institute, Dept. of Computer Sciences, 1983.
- [24] O. Takahashi and R.J. Schilling, "Motion planning in a plane using generalized Voronoi diagrams," *IEEE Trans. Robot. Autom.*, vol. 5, no. 2, pp. 143–150, Apr. 1989.
- [25] K.P. Valavanis, T. Hebert, R. Kolluru, and N. Tsourveloudis, "Mobile robot navigation in 2-d dynamic environments using an electrostatic potential field," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 30, no. 2, pp. 187–196, Mar. 2000.
- [26] C.W. Warren, "Fast path planning using modified a\* method," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, pp. 662–667, 1993.
- [27] B. Xu, D.Z. Chen and R.J. Szczerba, "A new algorithm and simulation for computing optimal paths in a dynamic and weighted 2-D environment," *IEEE Int. Conf. Systems, Man, and Cybernetics*, pp. 313 – 318, Oct. 2000.