

4. 攝影機的配置

本章描述如何在一停車場內配置攝影機。假設我們有一停車場的平面圖，首先我們將其掃描成為影像進入電腦，接下來在此影像上運用類神經網路規劃出一條能夠拜訪所有非障礙物空間的路徑，其好處在於將 2D 的覆蓋問題轉換成 1D 的問題，接著沿著這條路徑運用另一類神經網路來放置攝影機，使其達到監控整個停車場範圍的目的。我們因此將攝影機的配置分為兩階段，一、路徑的規劃，二、決定攝影機架設的位置。以下就此兩階段詳述其細節：

4.1 路徑的規劃

路徑規劃的方法有很多，我們已在第 1 章中做過介紹。本篇論文採用[Luo02]所提出的類神經網路方法來從事路徑規劃；首先我們將停車場平面圖掃描之後得到停車場平面圖影像，我們以一簡單例子說明，見圖 4.1，其中深色部分為障礙物或邊界，而白色部分為非障礙物；

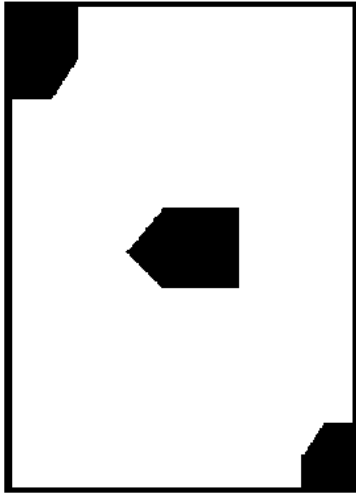


圖 4.1 停車場平面圖影像，黑色部分表示為障礙物及邊界，白色為非障礙物區域。

接著定義類神經網路的架構，我們的類神經網路為單一 layer，其上的神經元對應平面圖影像上的像素並且註記其是否為障礙物，而每個神經元與其鄰居的连接為 8-connected，如圖 4.2 所示。

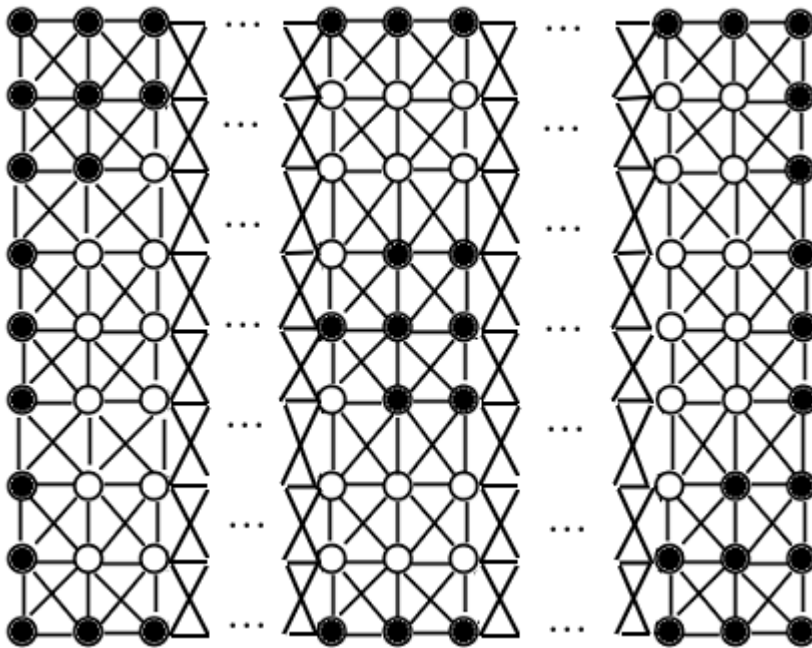


圖 4.2 對於圖 4.1 之類神經網路架構及神經元配置示意圖，每一圓點為一像素，其中空心圓點代表無障礙物，實心圓點代表障礙物，每個神經元與周圍鄰居為 8-connected。

定義好類神經網路的架構之後再來就是定義神經元的 activity 變

化如下式：

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i)S_i^e(t) - (D + x_i)S_i^i(t) \quad (4.1.1)$$

其中 x_i 表第 i 個神經元的 activity， A 、 B 及 D 為非負常數，分別代表被動衰退率(passive decay rate)、神經元 activity 的上界及下界，而 S_i^e 及 S_i^i 分別為神經元的激發和抑制源；接下來由於每個神經元與其鄰居的連接為 8-connected，所以每個神經元最多有八個刺激輸入，因此式(4.1.1)可以改寫如下式：

$$\frac{dx_i}{dt} = -Ax_i + (B - x_i) \left([I_i]^+ + \sum_{j=1}^k w_{ij}[x_j]^+ \right) - (D + x_i)[I_i]^- \quad (4.1.2)$$

其中 k 為鄰居神經元個數，而 $1 \leq k \leq 8$ ， I_i 為外部輸入定義如下：

$$I_i = \begin{cases} E & \text{if } i \text{ hasn't been visited and is not an obstacle} \\ -E & \text{if } i \text{ is an obstacle} \\ 0 & \text{if } i \text{ has been visited} \end{cases} \quad (4.1.3)$$

其中 E 遠大於 B ， $\left([I_i]^+ + \sum_{j=1}^k w_{ij}[x_j]^+ \right)$ 和 $[I_i]^-$ 分別對應式(4.1.1)中的激發及

抑制項， $[I_i]^+$ 和 $[I_i]^-$ 定義如下：

$$[I_i]^+ = \max\{I_i, 0\} \quad (4.1.4)$$

$$[I_i]^- = \max\{-I_i, 0\} \quad (4.1.5)$$

w_{ij} 為從第 i 個神經元連接到第 j 個神經元的 weight 定義如下：

$$w_{ij} = f(|q_i - q_j|) \quad (4.1.6)$$

其中 $|q_i - q_j|$ 為第 i 個神經元到第 j 個神經元之間的 Euclidean distance，

而函數 f 為一單調遞減函數定義如下：

$$f(a) = \begin{cases} u/a & 0 \leq a < r_0 \\ 0 & a \geq r_0 \end{cases} \quad (4.1.7)$$

其中 u 及 r_0 為正常數，並且 weight 的值為對稱的，也就是 $w_{ij} = w_{ji}$ 。

式(4.1.2)會使正的 neural activity 傳播到所有的神經元，而負的 activity

只能待在原神經元，這是因為在 $\left([I_i]^+ + \sum_{j=1}^k w_{ij}[x_j]^+\right)$ 中，僅有鄰居的 neural

activity 為正時， $[x_j]^+$ 才大於 0，否則為 0，因此非障礙物區域能吸引

產生路徑，而障礙物區域則排斥在其中選擇路徑。

在路徑規劃時，為了避免路徑產生太多的方向改變如圖 4.3 所示，我們參考上一路徑的方向，希望能夠盡量在方向上一致如圖 4.4 所示的路徑，數學上可表示如下：

$$x_{p_n} = \max\{x_j + C(1 - \Delta\theta), j = 1, 2, \dots, k\} \quad (4.1.8)$$

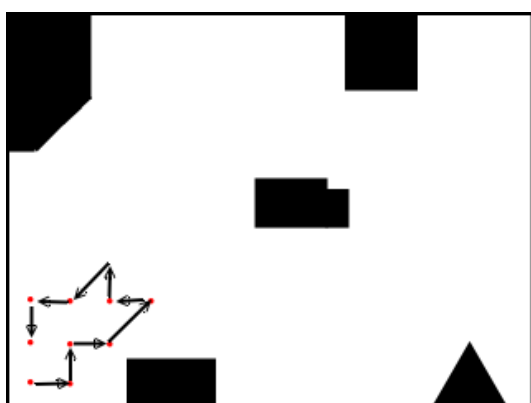


圖 4.3 路徑的方向轉變過多之示意圖

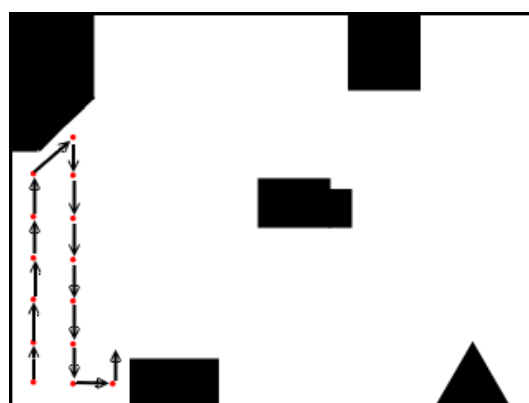


圖 4.4 路徑的方向轉變較少之示意圖

其中 p_n 指的是下一個路徑要到達的位置， x_{p_n} 為該位置上神經元的 activity， C 為一正常數， k 為目前神經元所連接的神經元個數，這個式子的意思就是說假定目前神經元連接了 k 個鄰居，所以下一步的選擇就有 k 個，我們運用式子 $x_j + C(1 - \Delta\theta_j)$ 計算出值最大的鄰居來當作下一路徑位置。上述方法之所以能避免路徑規劃時有太多方向的改變其關鍵就在於 $1 - \Delta\theta_j$ ， $\Delta\theta_j$ 定義如下：

$$\Delta\theta_j = |\theta_j - \theta_c| = \left| \tan^{-1}\left(\frac{j_y - c_y}{j_x - c_x}\right) - \tan^{-1}\left(\frac{c_y - p_y}{c_x - p_x}\right) \right| \quad (4.1.9)$$

其中 (j_x, j_y) 為下一可能位置的座標， (c_x, c_y) 為目前的座標， (p_x, p_y) 為前一位置的座標，因此從前一位置到目前位置與目前位置到下一可能位置的方向一樣時 $1 - \Delta\theta_j$ 的值會最大，所以我們利用式 (4.1.9) 就可以避免路徑規劃時有太多的方向改變。

有時候由於地形的關係，於路徑規劃時會走到死角，如圖 4.5(a)，此時我們會檢查該路徑位置的鄰居，若都為障礙物或是已拜訪過，我們將從原路返回，直到有一路徑位置的鄰居中有至少一個未拜訪過，如圖 4.5(b)所示。

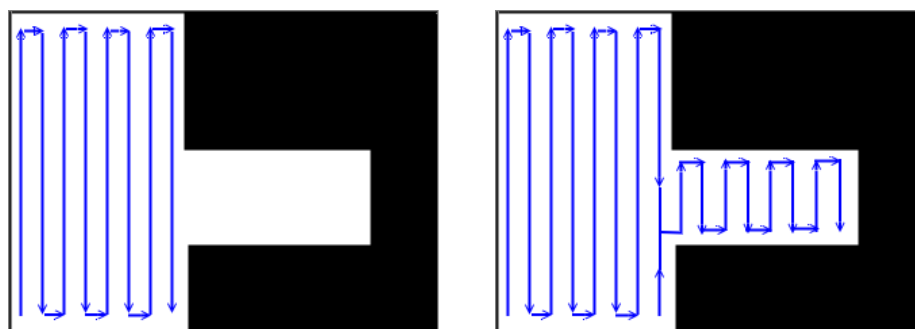


圖 4.5(a)

圖 4.5(b)

圖 4.5. (a)路徑規劃時遇到死角的例子，(b) 從原路返回，直到有一路徑位置的鄰居中有至少一個未拜訪過。

接下來我們要證明神經網路為穩定的(stable)，亦即網路中所有神經元的 activity 均會介於 $[B, -D]$ 。由式(4.1.1)中我們知道 x_i 以 $(B-x_i)S_i^e$ 項來增加，若 x_i 越接近 B ，那麼 x_i 增加的速度就越緩慢，當 x_i 小於 B 時，那麼刺激項就為正值，使得 neural activity 增加；若 x_i 等於 B 時，那麼刺激項就為 0；而當 x_i 大於 B 時， $(B-x_i)$ 為負值，此時 x_i 值會被拉近 B ，所以 x_i 的值會被強迫低於 B ，也就是 neural activity 的上界，同理抑制項 $(D+x_i)S_i^i$ 也會強迫 neural activity 大於 $-D$ ，所以神經網路模型會保持穩定。

圖 4.6 為路徑規劃的一範例，出發點為左上角，黑色部分為障礙物及邊界，黑色線條為路徑規劃後之結果，在這個例子我們可以看到

接著從第 i 個神經元連接到第 j 個神經元的 weight w_{ij} 定義如下：

$$w_{ij} = \begin{cases} 1 & \text{if Euclidean distance}[i, j] \leq R_i \\ 0 & \text{otherwise} \end{cases} \quad (4.2.1)$$

然後我們定義神經元的 activity 變化如下式：

$$\frac{dx_i}{dt} = -ex_i + (U - x_i)[V_i([f_i^o]^+ + f_i^m) + P_i] - (L + x_i)[f_i^o]^- \quad (4.2.2)$$

其中 x_i 表第 i 個神經元的 activity， e 、 U 及 L 為非負常數，分別代表被動衰退率(passive decay rate)、神經元 activity 的上界及下界，而 $[V_i([f_i^o]^+ + f_i^m) + P_i]$ 及 $[f_i^o]^-$ 分別為神經元的激發和抑制源， V_i 及 P_i 為外部輸入定義如下：

$$V_i = \begin{cases} 1 & \text{if } i \text{ has not been visited} \\ 0 & \text{if } i \text{ has been visited} \end{cases} \quad (4.2.3)$$

$$P_i = \begin{cases} E & \text{if a camera has been placed on } i \\ 0 & \text{otherwise} \end{cases} \quad (4.2.4)$$

其中 E 遠大於 U ，接著我們定義激發和抑制項中的 f_i^o 及 f_i^m 如下：

$$f_i^o = \begin{cases} C & \text{if } \max\{[P_k]^* O(i, k), j = 1, 2, \dots, r, k = 1, 2, \dots, n\} \leq \text{Threshold} \\ -C & \text{otherwise} \end{cases} \quad (4.2.5)$$

$$[P_k]^* = \begin{cases} 1 & \text{if } P_k = E \\ 0 & \text{otherwise} \end{cases} \quad (4.2.6)$$

其中 C 為一正值常數， r 為 i 所連接的鄰近神經元 j 的個數，而 n 為神經元 i 某一鄰居 j 所連接的鄰居 k 的個數， $Threshold$ 為於位置 i 放置攝影機時與其他攝影機視場重疊程度可接受的門檻值，這個式子的目的在判斷於神經元 i 的位置放置攝影機時，與其他攝影機視場重疊的程度 O 可否接受，若可，則 f_i^o 的值為 C ，若否，則 f_i^o 的值為 $-C$ 。而 O 的定義如下：

$$O=2(R_i^2 \theta - L_1 L_2) / \pi R_i^2 \quad (4.2.7)$$

這個式子的目的為求出兩圓所重疊的面積與單一圓面積的比例，說明如下：

我們先將兩圓圓心連接，然後於此線段上畫出其中垂線，因為兩圓的大小相同，所以此中垂線必通過兩圓交點，且將兩圓重疊之部分分割成兩相等部分，而兩圓圓心連接之線段亦是；在此我們為了描述方便，我們令一圓心 (k_x, k_y) 為 e ，兩圓圓心連接之線段與中垂線交點為 g ，而兩圓相交其中一交點為 f ，線段 \overline{eg} 的長度為 L_1 ，線段 \overline{fg} 的長度為 L_2 ，而線段 \overline{ef} 的長度為半徑長度 R_i ，如圖 4.8 所示， θ 為 \overline{eg} 與線段 \overline{ef} 之夾角，則 L_1 為：

$$L_1 = \sqrt{\left(k_x - \frac{i_x + k_x}{2}\right)^2 + \left(k_y - \frac{i_y + k_y}{2}\right)^2} \quad (4.2.8)$$

L_2 為：

$$L_2 = \sqrt{R_i^2 - L_1^2} \quad (4.2.9)$$

θ 為：

$$\theta = \cos^{-1}\left(\frac{R_i}{L_1}\right) \quad (4.2.10)$$

要求出兩圓所重疊的面積只要將圖 4.8 中灰色的扇形面積減掉三角形

$\triangle fge$ 的面積再乘 4 倍即可，三角形面積如下式：

$$\Delta fge = \frac{L_1 L_2}{2} \quad (4.2.11)$$

$$\begin{aligned} \text{而扇形 } \widehat{edf} \text{ 的面積則為 } & \pi R_i^2 \times \frac{\theta}{2\pi} \\ & = \frac{R_i^2 \theta}{2} \end{aligned} \quad (4.2.12)$$

將式(4.2.12)及式(4.2.11)相減的結果乘以 4 倍即為兩圓重疊的面積，

再除以圓面積即可得到式(4.2.7)的結果。

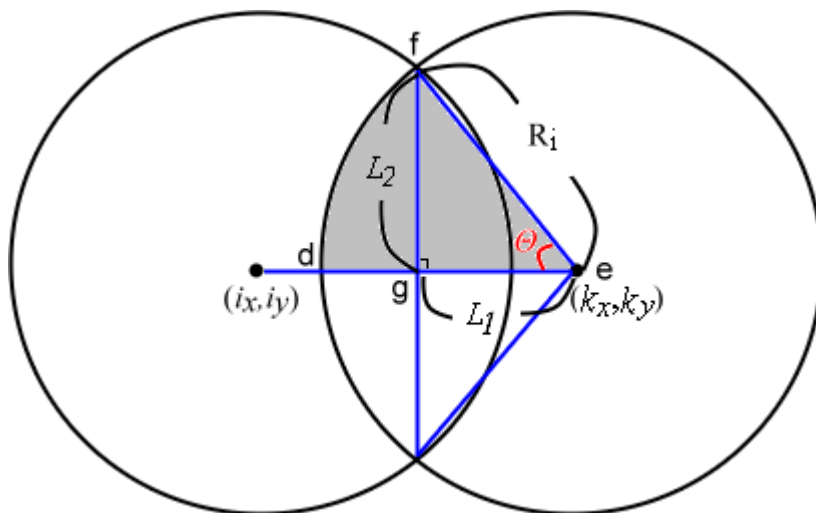


圖 4.8. 求出兩圓所重疊的面積之示意圖。

f_i^m 的定義如下：

$$f_i^m = \begin{cases} E & \text{if } \min\{\sum_{k=1}^n (P_k + V_k), j = 1, \dots, r\} = 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.2.13)$$

其中 r 為 i 所連接的鄰近神經元 j 的個數，而 n 為神經元 i 某一鄰近神經元 j 所連接的鄰居 k 之個數， V_k 為式(4.2.3)中表示 k 是否拜訪過，式(4.2.13)簡單的說就是檢查若不在 i 上置放攝影機時，會不會因此造成其鄰居 j 將不會被任何攝影機所覆蓋到，做法為一個個測試神經元 i 的鄰居 j ，若 j 的鄰居中有一個神經元的外部輸入 P_k 為 E ，則代表在 k 上已放置有攝影機，其視場能夠覆蓋到 j ；若 j 的鄰居的外部輸入 P_k 皆為 0 ，這時考慮到其鄰居是否都被拜訪過了，若否，則遺漏覆蓋的可能尚未發生，這是因為假設若決定不在 i 上放置攝影機，此時如果 j 沒有被其他攝影機覆蓋到，因為 j 的鄰居尚未全部拜訪過，所以仍有可能在拜訪 j 的鄰居時決定於其上放置攝影機，所以此時不管於 i 上放不放置攝影機都不會發生其鄰居遺漏覆蓋的狀況；若 j 的鄰居都已被拜訪過了，則對於 j 而言，此時 i 是 j 的鄰居中最後一個未被拜訪的，如果不在 i 上放置攝影機，那麼 j 將永遠不會被任何一個攝影機所覆蓋到，因此遺漏覆蓋 j 的可能發生了；使用式(4.2.13)時，在一個個測試 i 的鄰居 j 時，只要有一個最小值為 0 時，即代表有遺漏覆蓋的可能發生。

式(4.2.2)會使得已經拜訪過的神經元之 activity 在其上有放置攝影機時為正，而未放置時為負，這是因為當神經元已拜訪過後其激發源中僅剩 P_i ，所以未在神經元上放置攝影機時便沒有激發源，因此其 activity 為負值，而在神經元上若放置了攝影機，此時 P_i 為 E ，亦即激發項的值，因為 E 遠大於 activity 的上界 U ，所以其 activity 為正值；對於尚未拜訪到的神經元，若於其上放置攝影機與其他攝影機視場重疊程度可接受時， f_i^o 為激發項，不可接受時則為抑制項；而當 f_i^m 為 0 時(沒有遺漏覆蓋發生)，若 f_i^o 為激發時(要放置攝影機)，此時抑制項為 0，神經元的 activity 為正，若 f_i^o 為抑制時，此時激發項為 0，神經元的 activity 就為負，但當 f_i^m 為 E 時，也就是有遺漏覆蓋的可能發生時，不管重疊程度是否可接受，我們仍要置放攝影機，此時不管 f_i^o 為激發或抑制，因為 E 遠大於 U ，神經元的 activity 依然為正。

接著我們根據前一節所規劃的路徑，沿著路徑中的順序，一一拜訪此節討論的神經網路中的神經元，由於激發項與抑制項中考量到放置攝影機時與其他攝影機的視場重疊程度，所以我們必須先決定第一個攝影機的架設位置，這將在下一段落詳細描述，決定好之後便從架設第一個攝影機的神經元開始出發，在此我們設定每拜訪一個神經元花費一個時間單位，而在拜訪下一個神經元前便根據其 neural activity 來決定是否於其上放置攝影機，如此依序將每一個神經元拜訪完畢。

接下來我們討論如何決定第一個攝影機的放置位置，由於我們是一一循序拜訪神經元，而且於決定放置攝影機時必須考慮與其他攝影機覆蓋範圍重疊的程度，所以第一個攝影機的放置位置決定是必要的；由於我們在規劃路徑時起始點通常是角落，而且一開始的路徑通常沿著邊界，若此時放置攝影機往往會造成大部分的監控範圍是在邊界外，如圖 4.9 所示，圖 4.9(a)為直接在起始路徑上放置攝影機的結果，其中我們可以觀察到放置這兩個攝影機所覆蓋的範圍，如果適當使用一個攝影機即可完整覆蓋，如圖 4.9(b)所示，所以在挑選第一個攝影機的架設位置時我們採用跳躍的方法，也就是由出發的神經元位置上找出其所連接的鄰居中於其上放置攝影機能覆蓋最多像素點的位置，數學式子如下：

$$I_p = \max \left\{ \sum_{j=1}^r \sum_{l=1}^n w_{jk} \right\} \quad (4.2.14)$$

其中 I_p 為放置第一個攝影機的位置， r 為出發神經元所連接的鄰居神經元 j 的個數， k 為神經元 i 某一鄰近神經元 j 所連接的鄰居， n 為 k 所連接的鄰居神經元數，如此便可達成如圖 4.9(b)的結果。

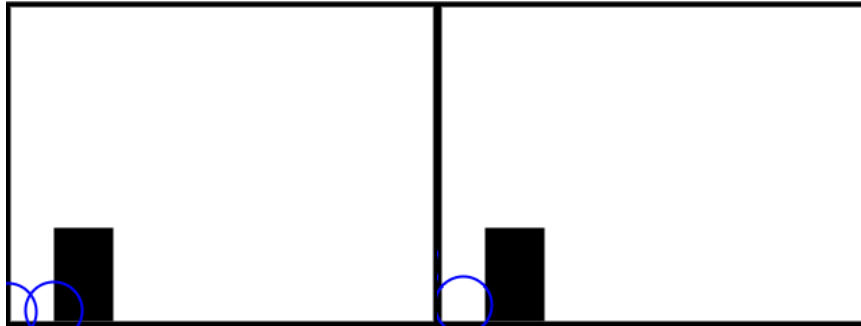


圖 4.9.(a)

圖 4.9.(b)

圖 4.9.決定架設第一個攝影機之示意圖 (a)為從第一個路徑位置架設攝影機(b)採用跳躍法決定第一個攝影機架設位置。

由於我們決定第一個攝影機的架設位置為跳躍決定，因此起始神經元的位置不再為前一節中所規劃出路徑的出發位置，而是 I_p ，所以我們拜訪的順序為從 I_p 出發，一一循序拜訪直到先前路徑規劃的最後一個神經元，然後再從 I_p 反方向將未拜訪的神經元一一拜訪，如此，便完成攝影機架設的決定，圖 4.10(a)及圖 4.10(b)分別為路徑規劃及攝影機架設的結果。

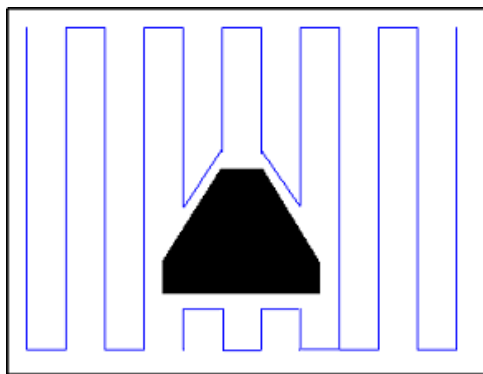


圖 4.10.(a)

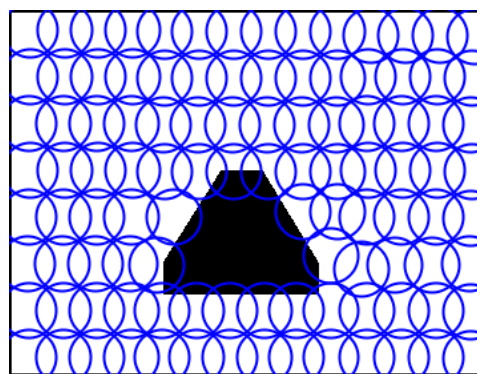


圖 4.10.(b)

圖 4.10. (a)為路徑規劃的示意圖，(b)為沿著(a)中路徑架設攝影機的结果。

接下來我們要證明神經網路為穩定的(stable)，亦即網路中所有神經元的 activity 均會介於 $[U,-L]$ 。由式(4.2.2)中我們知道 x_i 以 $(U-x_i)$

$[V_i([f_i^o]^+ + f_i^m) + P_i]$ 項來增加，若 x_i 越接近 U ，那麼 x_i 增加的就越緩慢，當 x_i 小於 U 時，那麼刺激項就為正，使得 neural activity 增加；若 x_i 等於 U 時，那麼刺激項就為 0；而當 x_i 大於 U 時， $(U-x_i)$ 為負，此時 x_i 值會被拉近 U ，所以 x_i 的值會被強迫低於 U ，也就是 neural activity 的上界，同理抑制項 $(L+x_i) [f_i^o]^-$ 也會強迫 neural activity 大於 $-L$ ，所以神經網路模型會保持穩定。