使用 Dijkstra’s 演算法之以物件偵測爲基礎的最短避障路徑規畫

葉國良*
周利蔚**

摘 要

本文探討以 Dijkstra’s 演算法規劃一能夠避開環境中圓形障礙物，並到達指定目的地的最短路徑。我們利用圓形幾何性質中的圓周角特性，以及向量內積偵測法判斷影像圖片中是否存在圓形物體。在圓形物體外，我們同時考慮輪型機械人的最小旋轉半徑所需之安全邊界。透過 Dijkstra’s 最短路徑搜尋法，以外切六邊形頂點爲網路節點，最短路徑爲代價函數，完成避障路徑設計。爲要有效的避開障礙物，本文利用直線及凸邊形的幾何特性，提出「幾何禁行路徑偵測法」，找出座落於障礙範圍內的偵測點，以決定任意兩個頂點間連通性。模擬結果得知，不論從起始點至目標點之間的障礙物擺設情況爲何，本文所提方法皆可正確地避開障礙物，並規劃出最短路徑。

關鍵字：路徑規劃、向量內積圓偵測法、禁行路徑、Dijkstra’s 演算法。

---

* 康寧醫護暨管理專科學校 資訊管理科講師；台灣科技大學電子工程所博士候選人
** 台北科技大學機電科技所自動化組博士生
電子郵件：liwei@knjc.edu.tw
收稿日期：2007.11.12 修改日期：2008.5.14 接受日期：2008.5.14

An Object Detection Based Shortest Obstacle Avoidance Path Planning via the Dijkstra's Algorithm

Kuo-Liang Yeh*
Li-Wei Chou**

Abstract

In this paper, we propose an object detection based shortest obstacle avoidance path planning which connects the start point and the target point via the Dijkstra's algorithm. We apply the geometric characteristics of the circumferential angle and the inner product detection to determine whether there are round objects in the image. Outside round objects, the safety margins for robot rotation are also took into account. By using of the hexagon vertices of each obstacle object as nodes to create a network, and considering the shortest path as a cost, the Dijkstra's method is adopted to search the shortest path which avoids obstacles., The purpose is to plan a path to avoid obstacles. To determine the connectivity geometric detection are evolved respectively to determine the forbidden paths. The former applies geometric characteristics of straight lines and convex hull, to determine whether the detecting point is allocated within the range of obstacle area. Simulation results show that the proposed approach is applicable to different obstacle layout between the start and target point of running path.

Key Words：. Path planning、Inner Product Detection、Forbidden Paths、Dijkstra's Algorithm。

* Lecturer, Department of Information Management, Kang-Ning Junior College of Medical Care and Management; Candidate for Phd., Department of Electronic Engineer, National Taiwan University of Science and Technology

** Student,College of Mechanical & Electrical Engineering, National Taipei University of Technology

壹、前言

在這科技進步的時代，產業自動化需求逐漸的提昇，因此智慧型機器人的研究及相關技術的運用相當廣泛。小從機械手臂，無人搬運車，大至爆裂物拆除、海底探勘、太空探險、採礦、救災等等特殊任務，都是藉由自動化機械人來完成。然而在工作環境多變的情況下，機器人必須在許多外在因素干擾，隨著環境的不同，做出自我調整的功能，到達指定的地點來完成工作，所以機器人路徑規劃的問題是值得重視及探討的。如何讓機器人能夠避開環境中的障礙物，並且以最小的代價到達指定的目標是本文所要探討的主題。

障礙物偵測有許多方法，如紅外線感測器、超音波、雷達、雷射及電腦視覺等。紅外線感測器[1]及超音波感測器檢測[2]應用上成本低廉，但是偵測範圍較短，約小於 15 公尺。對於雷達而言，是利用無線電波原理[3]，把所偵測到的目標顯示出來，單價高，推廣困難。雷射則利用光反射回來的光程差推算測量距離，價格低廉，但會依環境的狀況有不同的精確度。本文採用電腦影像視覺技術，透過 CCD 攝影機擷取環境影像，經由邊點序向排列法[4]、向量內積偵測法，有效偵測出圓形障礙物位置及座標。

在形樣辨識中，圓形物體的偵測仍然佔有極重要的地位，早期的 Hough Transform (HT)以及 Probabilistic Hough Transform (PHT) 都是屬於一點取樣的圓形偵測法[9,10]，雖然可以獲得很好的偵測效果，但必需佔用大量的記憶體空間來儲存過去的取樣資料。然而，Coaxial Transform 演算法[11]是採用兩點取樣法，Random Sample Consensus(RANSAC)演算法[12]則是採用三點取樣來偵測圓形物體。Randomized Circle Detection(RCD)演算法[6,13]是從影像圖片中隨機取樣四個點，利用任意四點不容易共圓的特性來決定所謂候選圓(Possible Circle)，再進一步判斷這些候選圓是否爲真正圓(True Circle)。

許多資料結構或網路路由的搜尋法，應用於路徑的規畫，深度優先搜尋(Deep-First Search，DFS)是由 Lucas 與 Tarry 第一個提出[14]，它原本是用來設計走迷宮的演算法。廣度優先搜尋(Breadth-First Search，BFS)是較有組織有順序，是一層一層的往下走訪，皆屬於無資訊(uninformed)搜尋。在有資訊(informed)搜尋演算法中 Prim[15]首先提出最小成本生成樹演算法(Minimum spanning tree algorithm)，並廣泛應用於網路的路由(Routing)選擇。單一源頭最短路徑演算法是由 Dijkstra[16]所發展出來，乃利用 Best-First Search 處理最短路徑問題。

Johnson[17]首先使用堆疊實作 Dijkstra 的演算法。Floyd[18]將 Dijkstra 的演算法應用到網路中每個節點到其他各節點的最短路徑選擇。在這些有資訊搜尋法中，多以兩兩節點之間的直線距離作爲代價函數(Cost Function)，本文在考量各種演算法特性及適用性後，選擇具動態規劃的 Dijkstra's 演算法爲理論依據，並在計算兩節點代價函數時，加入本文所提出幾何禁行路徑偵測，當兩節點偵測爲一條禁行路徑時，修正其代價函數，並將直線距離值改變爲∞。

在進行最短路徑規畫時，障礙物常常是主要影響規畫結果的因素之一，如何準確避開障礙物，規畫出一條最短路徑，成爲機器視覺的重要議題。我們利用向量內積偵測法找出影像圖片中的圓形物體。在圓形物體外，我們同時考慮輪型機械人的最小旋轉半徑所需之安全邊界。透過 Dijkstra's 最短路徑搜尋法，以外切六邊形頂點爲網路節點，最短路徑爲代價函數，完成避障路徑設計。本文同時利用直線及凸邊形的幾何特性，提出「幾何禁行路徑偵測法」，找出座落於障礙範圍內的偵測點，以決定任意兩個頂點間連通性。不論從起始點至目標點之間的障礙物擺設情況爲何，本文所提方法皆可正確地避開障礙物，並規劃出最短路徑。

本文的組織架構可區分如下: 貳、理論分析，說明路徑規劃所需的節點，即圓形障礙物外接六邊形頂點，如何從影像圖片中偵測，包括灰階轉換、邊緣檢測、邊點序向排列法、向量內積偵測法以及本文提出的幾何禁行路徑偵測法，詳細介紹如何將節點間的連通訊息轉換爲代價函數，並以 Dijkstra's 演算法，規劃出從起始點與目標點之避障路徑。參、模擬結果與討論中，以 Matlab 程式語言，針對本文提出的幾何禁行路徑偵測法及採用的 Dijkstra's 演算法進行模擬驗證。肆、爲結論以及未來的發展方向。

貳、理論分析

一、灰階轉換 (Gray Level Transformation)

由 CCD 擷取出的彩色影像，每一個像素都是由三個位元組 (Bytes) 所構成，分別代表紅(R)、綠(G)、藍(B)三種色彩資訊。首先，對輸入影像進行灰階轉換處理，將彩色像素(Pixel)轉成 0 - 255 灰階(Grey Level)值，0 代表黑色、255 則爲白色。轉換公式如下：

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \qquad (1)$$

其中 Y 爲轉換後的灰階值。

二、邊緣偵測 (Edge Detection)

邊緣包含影像中重要的資訊，可以用來測量影像中障礙物體大小，辨識物體的形狀或者進行物體的分類。本文障礙物體爲圓型物體，並採用 Sobel 法[6,7]，利用一階導數運算子對影像求邊緣梯度大小，並設定一二值化臨界值[7]，大於臨界值之像素設爲 1，表邊緣像素，反之爲 0，表背景像素。主要用於尋找影像中物體的邊緣，並且將相對應的像素資料 $f(i, j)$ 設爲 1，其餘設爲 0。

三、邊點序向排列 (Edge-Point Sorting)

邊緣偵測後，我們採用邊點序向排列法[4]，將所有物體的邊點資料進行有序排列，除了決定每個邊點之間的鏈結關係外，並確認分屬於同一物體或是影像雜訊的邊點，以得到屬於這張影像中所有物體的有序邊點座標。

四、向量內積偵測法 (Inner Product Detection Method)

影像圖片經由邊點序向排列法偵測後所得到的分群物體，都有可能爲圓形物體，因此我們將它視爲候選圓。每個候選圓是由不同序向排列邊點所組合而成的，圓形物體較大時，所需的邊點個數較多；反之，邊點個數較少。我們假設候選圓的總邊點個數爲 n，物體的個數爲 m，向量內積圓形偵測演算法如下所述：

[步驟一]：首先，針對每一個物體的邊點集合 $V_1, V_2, \cdots, V_m$，分別選取每一個集合內的第一個邊點 P_1^j 與最後一個邊點 $P_{n_j}^j$，其中 $j = 1, 2, \cdots, m$。倘若 $P_1^j = P_{n_j}^j$，表示該候選圓是一個封閉的曲線。相反的 $P_1^j \neq P_{n_j}^j$，則不爲封閉曲線，該物體的邊點不予考慮，重覆步驟一。

[步驟二]：避免數位化誤差影響偵測正確性，在第 j 個邊點集合中選擇非相鄰的 P_A^j、P_B^j、P_C^j 三點，判斷是否共線。在此，由 V_j 集合中選擇第一

個序向排列邊點當作 P_A^j 點，第 n_j ×20%個序向排列邊點爲 P_B^j 點，第 n_j ×80%個序向排列邊點爲 P_C^j 點，邊點座標分別爲 $P_A^j=(x_a^j,y_a^j)$ 、 $P_B^j=(x_b^j,y_b^j)$ 、 $P_C^j=(x_c^j,y_c^j)$ ，判斷 P_A^j 、 P_B^j 、 P_C^j 三點是否共線，不共線的三點才可以決定出一個圓形。亦即

$$(x_b^j-x_a^j)\times(y_c^j-y_a^j)-(x_c^j-x_a^j)\times(y_b^j-y_a^j)\neq 0 \qquad (2)$$

(2)若成立，則表示 P_A^j ， P_B^j ， P_C^j 三點可以構成圓形，並且作爲向量內積圓偵測法之三個基準點，步驟三進行像量內積圓偵測。

[步驟三]：爲了提高向量內積圓偵測法的準確度，我們將整個圓形分成兩個部分處理。首先，如圖 1、圖 2 之所示，偵測候選圓 30% 至 80% 的序向排列邊點，對應至 $\overline{P_A^jP_B^j}$ 所形成的夾角，判斷其是否符合圓周角 $\angle P_A^jP_C^jP_B^j$ 相等的特性，並計算符合的次數，定義爲 count。其次，選擇另外 50%的序向排列邊點作偵測，判斷其對應至 $\overline{P_{A'}^jP_{B'}^j}$ 所形成的夾角是否等於圓周角 $\angle P_{A'}^jP_{C'}^jP_{B'}^j$ 。利用幾何性質圓周角相等的特性，由圖 1 中可知

$$\angle P_A^jP_C^jP_B^j(\theta_C)=\angle P_B^jP_{B'}^{j'}P_A^j(\theta_{B'})=\angle P_B^jP_{A'}^jP_A^j(\theta_{A'})=\angle P_B^jP_{C'}^jP_A^j(\theta_{C'})$$

$$\cos\theta_C=\frac{\overrightarrow{P_C^jP_A^j}\cdot\overrightarrow{P_C^jP_B^j}}{\left|\overrightarrow{P_C^jP_A^j}\right|\left|\overrightarrow{P_C^jP_B^j}\right|}=\cos\theta_{B'}=\cos\theta_{A'}=\cos\theta_{C'} \qquad (3)$$

其中以 $\cos\theta_C$ 爲基準值，依序計算從 30% 順時針方向至 80% 所有序向排列邊點 P_i^j 對應至 $\overline{P_A^jP_B^j}$ 所形成的夾角的餘弦值，定義以下向量

$$\overrightarrow{P_i^jP_A^j}=(x_i^j-x_a^j,y_i^j-y_a^j),\ \overrightarrow{P_i^jP_B^j}=(x_i^j-x_b^j,y_i^j-y_b^j)$$

利用 $\overrightarrow{P_i^jP_A^j}$ 與 $\overrightarrow{P_i^jP_B^j}$ 兩向量間夾角的餘弦值 $\cos\theta_i$ 定義出邊點餘弦值如下：

$$\cos\theta_i=\frac{\overrightarrow{P_i^jP_A^j}\cdot\overrightarrow{P_i^jP_B^j}}{\left|\overrightarrow{P_i^jP_A^j}\right|\left|\overrightarrow{P_i^jP_B^j}\right|};\quad \begin{cases}-1\le\cos\theta_i\le 1.\\ n_j\times 30\%<i\le n_j\times 80\%\ ,\ i\in N\end{cases} \qquad (4)$$

$\cos\theta_i$ 代表邊點座標 (x_i^j, y_i^j) 與兩個向量 $\overrightarrow{P_i^j P_A^j}$, $\overrightarrow{P_i^j P_B^j}$ 所形成夾角之餘弦函數值。若 $\left|\cos\theta_i - \cos\theta_C\right| \le T_a$ 相當小，小於我們用來判斷的門檻值 T_a 時，表示符合圓周角相等的圓的特性，認爲圓的可能存在，並將變數 count 值加 1。同樣的，我們偵測候選圓另外 50% 的序向排列邊點，計算符合圓周角相等的次數，繼續的累加變數 count。從圖 2 中可知，

$$\angle P_{A'}^j P_{C'}^j P_{B'}^j(\theta_{C'}) = \angle P_{A'}^j P_C^j P_{B'}^j(\theta_C) = \angle P_{A'}^j P_A^j P_{B'}^j(\theta_A) = \angle P_{A'}^j P_B^j P_{B'}^j(\theta_B)$$

$$\cos\theta_i = \frac{\overrightarrow{P_i^j P_{A'}^j} \cdot \overrightarrow{P_i^j P_{B'}^j}}{\left|\overrightarrow{P_i^j P_{A'}^j}\right| \left|\overrightarrow{P_i^j P_{B'}^j}\right|}; \quad \begin{cases} -1 \le \cos\theta_i \le 1. \\ 1 \le i \le n_j \times 30\% \text{ 且 } n_j \times 80\% < i \le n_j \text{ , } i \in N \end{cases} \quad (5)$$

當 $\left|\cos\theta_i - \cos\theta_{C'}\right| \le T_a$ ，count 值加 1，其中 T_a 爲浮動圓周角門檻值 [5] ，圓形物體的圓周是由邊點組合而成，邊點數愈多，數位化誤差的影響度較小， T_a 門檻值可以設定較大。

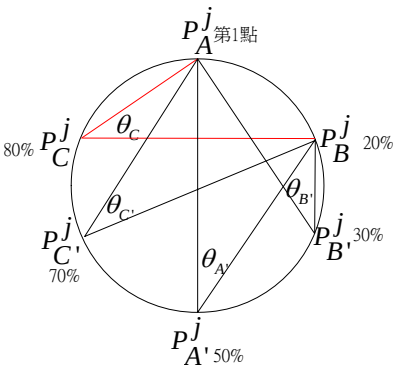

圖 1 下半圓偵測

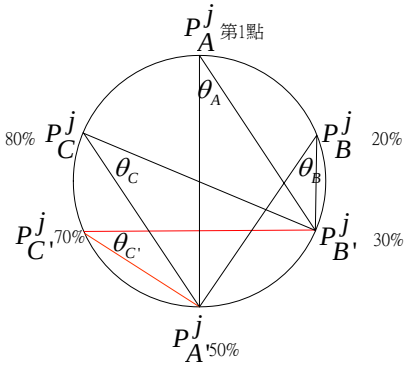

圖 2 上半圓偵測

[步驟四]：當變數 count 值與候選圓邊點總數比，達到某個比率時(本文設定 70%)，我們才認爲向量內積圓偵測法，真正偵測出一個圓形。反之，則爲非圓形物體。回到步驟一，繼續偵測下一個候選圓。

[步驟五]：計算圓心與半徑。取樣的三點分別爲 $P_A^j = (x_a^j, y_b^j)$ 、 $P_B^j = (x_b^j, y_b^j)$ 、

$P_C^j=(x_c^j,y_c^j)$，這三點所決定的圓心是(a_j,b_j)，半徑定義爲r_j，六邊形頂點座標爲$O_k^j=(x_k^j,y_k^j)$，$k=0\sim5$。圖 3 說明外切正六邊形邊長爲圓半徑$\frac{2\sqrt{3}}{3}$倍。

$$a_j=\frac{\begin{vmatrix}(x_b^j)^2+(y_b^j)^2-\left((x_a^j)^2+(y_a^j)^2\right) & 2(y_b^j-y_a^j)\\ (x_c^j)^2+(y_c^j)^2-\left((x_a^j)^2+(y_a^j)^2\right) & 2(y_c^j-y_a^j)\end{vmatrix}}{4\left((x_b^j-x_a^j)(y_c^j-y_a^j)-(x_c^j-x_a^j)(y_b^j-y_a^j)\right)} \tag{6}$$

$$b_j=\frac{\begin{vmatrix}2(x_b^j-x_a^j) & \left((x_b^j)^2+(y_b^j)^2\right)-\left((x_a^j)^2+(y_a^j)^2\right)\\ 2(x_c^j-x_a^j) & \left((x_c^j)^2+(y_c^j)^2\right)-\left((x_a^j)^2+(y_a^j)^2\right)\end{vmatrix}}{4((x_b^j-x_a^j)(y_c^j-y_a^j)-(x_c^j-x_a^j)(y_b^j-y_a^j))} \tag{7}$$

$$r_j=\sqrt{(x_a^j-a_j)^2+\left(y_a^j-b_j\right)^2}=\sqrt{(x_b^j-a_j)^2+\left(y_b^j-b_j\right)^2}=\sqrt{(x_c^j-a_j)^2+\left(y_c^j-b_j\right)^2} \tag{8}$$

$$O_k^j=\frac{2\sqrt{3}}{3}r_j\times\cos\left(\frac{2\pi k}{6}\right)+a_j\ ,\ \mathrm{y}_k^j=\frac{2\sqrt{3}}{3}r_j\times\sin\left(\frac{2\pi k}{6}\right)+b_j \tag{9}$$

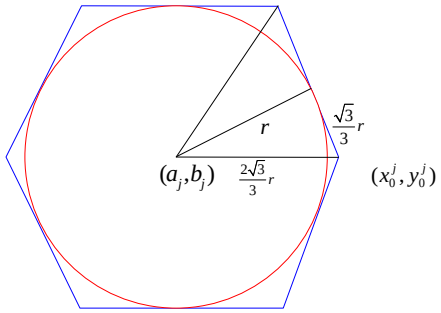

圖 3 圓外切正六邊形其半徑與邊長關係圖

另外，在處理路徑規畫時，必需引入安全邊界才是一條合理的避障路徑。亦即外切六邊形的每一個頂點必需增加輪型機械人的最小旋轉半徑$R_{\min}$[19]，輪型機械人才能夠順利轉彎，如圖 4 中所示。兩個前輪的驅動角分別爲 ϕ 以及 $\bar{\phi}$，並且滿足 $|\bar{\phi}|\le\bar{\phi}_{\max}$ 以及 $|\phi|\le\phi_{\max}$ 的條件，輪寬與輪距分別爲 2b 和 ρ，則最小的旋轉半徑可表示爲

$$R_{\min} = \rho \cot \phi_{\max} - 2b = \rho \cot \bar{\phi}_{\max} \quad (10)$$

圖 5 爲具安全邊界障礙物，其六邊形頂點座標 $O_k^j = (x_k^j, y_k^j)$ 被改寫爲(11)式。

$$O_k^j = (\frac{2\sqrt{3}}{3} r_j + R_{\min}) \times \cos\left(\frac{2\pi k}{6}\right) + a_j \ , \ y_k^j = (\frac{2\sqrt{3}}{3} r_j + R_{\min}) \times \sin\left(\frac{2\pi k}{6}\right) + b_j \quad (11)$$

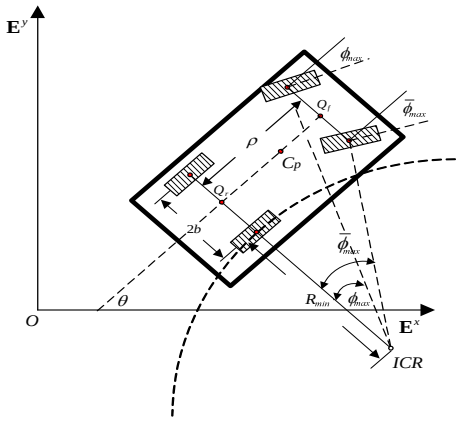

圖 4 輪型機械人最小旋轉半徑

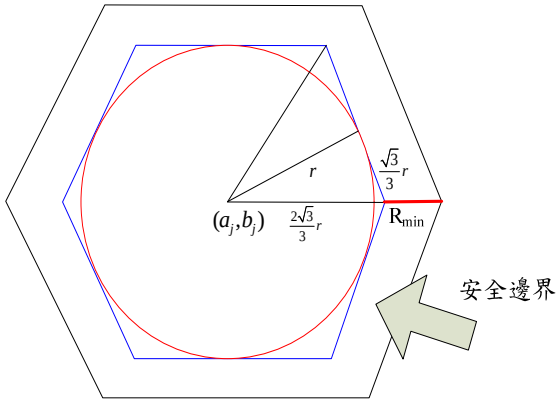

圖 5 具安全邊界圓外切正六邊形

五、幾何禁行路徑偵測 (Geometric Forbidden Paths Detection)

在找尋 Dijkstra's 最短路徑時，判斷兩節點之連通性扮演著關鍵性的角色。兩節點間無任何障礙物存在，則兩節點間具有連通性，是可以通行的。反之，兩節點間若有障礙物存在，則不具有連通性，稱爲『禁行路徑』。

5.1 環境資訊的定義

圖 6 是由 CCD 攝影機取得影像圖片，經由向量內積偵測法偵測圓形障礙物，並定義障礙物的分佈模型，將具安全邊界的外切六邊形頂點作爲網路節點，使用 Dijkstra's 最短路徑搜尋法，針對每一個節點評估進其代價函數，藉以搜尋一條最短路徑。倘若環境中存在 m 個多邊形障礙物，每一個障礙物包括 n_j (j=1,2,…,m)個頂點。給定起始點爲 $Q_1(x_1, y_1)$，障礙物的頂點依序定義爲 $Q_2(x_2, y_2)$ ~ $Q_{k-1}(x_{k-1}, y_{k-1})$，終點則定義爲 $Q_k(x_k, y_k)$。其中 $k = (\sum_{j=1}^{m} n_j) + 2$ ，等於所有頂點數目加 2。對於每一個節點間的距離，我們定義所謂的距離矩陣 M。

$$M = \begin{bmatrix} d_{11} & d_{12} & \cdots & \cdots & d_{1k} \\ d_{21} & d_{22} & \cdots & \cdots & d_{2k} \\ & & \cdots & \cdots & \\ & & \cdots & \cdots & \\ d_{k1} & d_{k2} & \cdots & \cdots & d_{kk} \end{bmatrix} \tag{12}$$

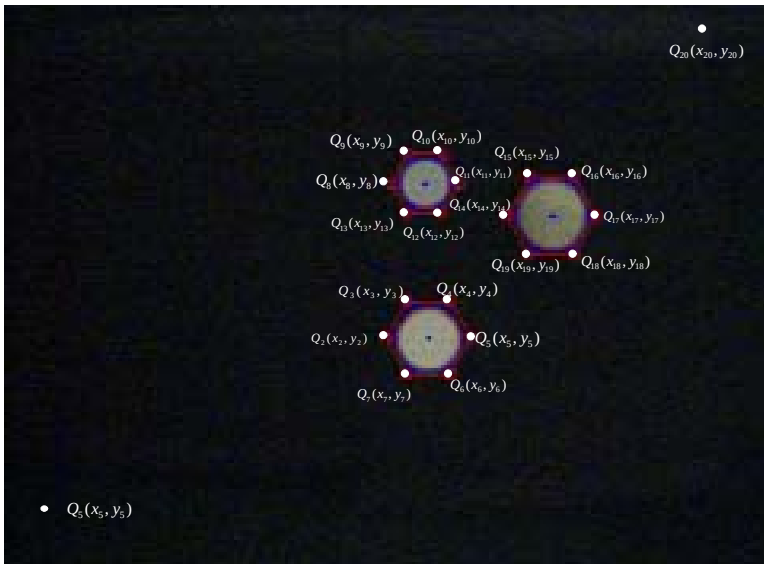

圖 6 圓形障礙物分佈模型

其中 d_{ab} 代表節點 Q_a 與 Q_b 之間的距離。同一個節點間的距離定義爲零，故 $d_{11} = d_{22} = d_{33} = \cdots = d_{kk} = 0$。此外，倘若 Q_a 與 Q_b 兩點不相連通，代表兩節點間有障礙物存在，我們定義爲 $d_{ab} = \infty$。程式模擬時，我們給定 inf 表示其間距離爲 ∞。

5.2 凸形障礙物模型

在二維平面上，一條直線的方程式可以表示成

$$L:\{\mathbf{a}\,|\,\mathbf{n}^T(\mathbf{a}-\mathbf{a}_0)=0\} \tag{13}$$

其中 $\mathbf{n}=[n_x \;\; n_y]^T$ Ų P2 爲這條直線 L 的法線向量。$\mathbf{a}_0=[x_0 \;\; y_0]^T$ 爲這條直線上的任一點，且 $\mathbf{a}=[x \;\; y]^T$ ，因此 $\mathbf{a}-\mathbf{a}_0$ 與 n 爲兩正交向量。(13)可以進一步化簡成直線的一般式

$$n_x x+n_y y+(-n_x x_0-n_y y_0)=0 \tag{14}$$

(14)所代表的直線可將 P2 平面分爲兩個部份。$\mathbf{n}^T\mathbf{a}>b$ 表示與 n 向量同一方向的半平面(如陰影部份)，而 $\mathbf{n}^T\mathbf{a}<b$ 表示與 n 向量相反方向的半平面，其中 $b=\mathbf{n}^T\mathbf{a}_0\in R$ ，如圖 7 所示。

一多邊形的障礙物 Ω 可藉由上述方法加以描述，圖 8 中障礙物是由 n 個半平面所組合而成。如(14)，我們可利用 n 條直線方程式建構此一模型。倘若我們找出每一條直線方程式的法線向量 $\mathbf{n}_1, \mathbf{n}_2,\cdots,\mathbf{n}_n$ ， 其方向均向外，此時凸邊形障礙物的模型可表示如下：

$$\Omega=\{\mathbf{a}\,|\,\mathbf{n}_i^T\mathbf{a}<b_i,\; b_i=\mathbf{n}_i^T\mathbf{a}_0,\; i=1,2,\cdots,n\} \tag{15}$$

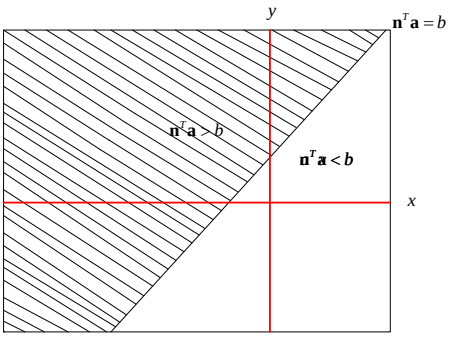

圖 7 法線式直線表示法

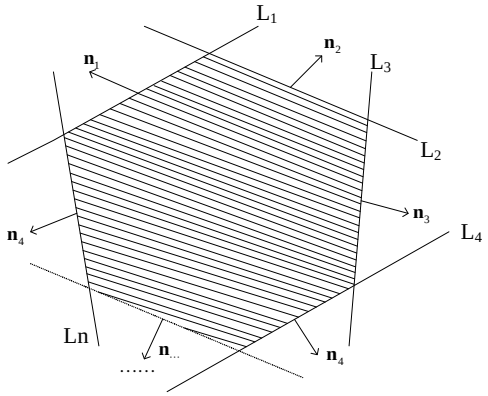

圖 8 凸邊形障礙物模型

5.3 幾何禁行路徑偵測步驟

我們將針對障礙物的所有頂點 $Q_1, Q_2, Q_3,\ldots,Q_k$ ，判定兩兩節點間是否連通。考慮任意兩個不同節點 Q_a 以及 Q_b，將 Q_a 定義爲起始頂點，而 Q_b 爲目標頂點。介於 Q_a 以及 Q_b 之間，亦即在 $\overline{Q_a\,Q_b}$ 線段上，所有點的集合定義爲『禁行偵測點』(Forbidden Detection Point)。幾何禁行偵測點的決定方法可

由起始頂點開始，沿著 x 軸的方向遞增或遞減(視Q_a 、Q_b之間的方向決定)，且間隔爲 1 Pixel，直到目標頂點 Q_b 爲止。透過 $\overline{Q_a\ Q_b}$ 的斜率，找出其所對應 y 座標，這些點我們稱作禁行偵測點，並定義爲 $Q_t(x_t, y_t)$，參考圖 9。

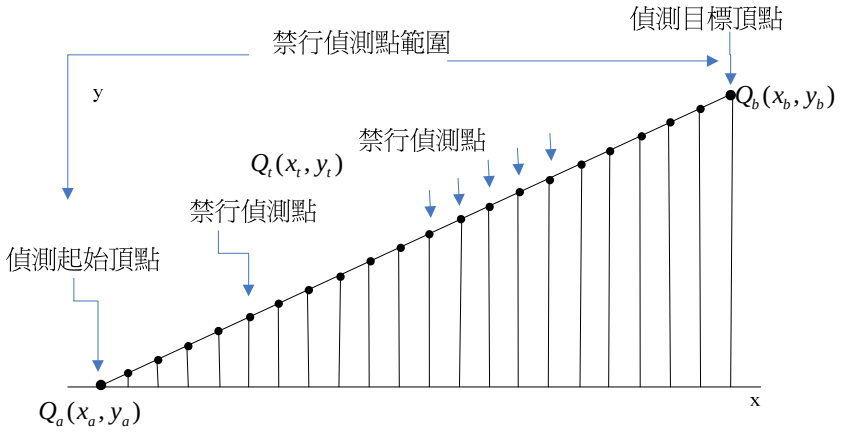

圖 9 兩頂點間偵測點範圍-幾何禁行路徑偵測法

幾何禁行路徑偵測步驟如下：

[步驟一]：爲了完成避障路徑的規劃，我們必需決定起始點及終點的位置。倘若，我們選定像素座標 Q_1 爲起點位置，像素座標 Q_k 爲目標位置，影像圖片中所偵測的所有障礙物Ω_j，$j=1\cdots m$，其頂點座標將依序定義爲$Q_2, Q_3 \ldots Q_{k-1}$，其中$k=(\sum_{j=1}^{m} n_j)+2$。

[步驟二]：計算 $Q_1, Q_2, Q_3 \ldots Q_k$ 所有頂點與頂點間的距離、斜率，並將頂點與頂點間的距離記錄於 M 距離矩陣中。

[步驟三]：找出任意兩個頂點間所有禁行路徑的偵測點。

[步驟四]：經由步驟三找出每一個禁行路徑偵測點，透過(15)式判定是否座落在Ω_j凸邊形障礙物範圍內，只要有一個禁行路徑偵測點包含在障礙物範圍之內，這兩個頂點即爲不可通行的路徑，也就是所謂的禁行路徑。

[步驟五]：兩頂點間若爲禁行路徑，修正其距離矩陣 M 之記錄，將矩陣內對應之兩頂點距離元素 d 設定 inf，表示兩頂點間爲不連通。重覆步驟三，至所有頂點皆完成偵測。

5.4 Dijkstra's algorithm 最短路徑搜尋法

影像圖片中每一個障礙物的邊，都是兩個頂點所形成，Dijkstra's 最短路徑搜尋法的精神就是邊的拓展，其作法是以某一節點爲起點，計算從該節點出發到其他節點的最短路徑。對於每一個節點來說，它不只考慮相鄰節點的代價函數，而是評估由起始點至該節點所有可能路徑的代價函數，代價函數愈小表示所走的路徑愈短，愈符合最佳化之設計準則。

爲了幫助 Dijkstra's 演算法的說明，我們先定義演算法中幾個重要變數如下：

- 首先以第一個節點當作起始點，重新定義爲 Q_s。
- $d(u,v)$ 表示從節點 Q_u 到節點 Q_v 的距離，倘若任意兩節點 Q_u 到 Q_v 沒有路徑相連，則令 $d(u,v)=\infty$，本文以兩個節點之間的距離作爲權重函數值(Weight Function)。
- 起始點 Q_s 至其他節點 Q_i 之代價函數定義爲 $cost[i]$, $i=2,3,\cdots,k$，起始點 Q_s 本身的代價函數定義爲 $cost[s]$=0。
- $flag[u]$ 爲相對應的節點處理完成旗標， $flag[u]$=1 表示第 u 個節點已完成最短路徑之搜尋。
- $index[v]=u$ 記錄了 Q_v 節點所對應的來源節點 Q_u，亦即最短路徑規劃依序由節點 $Q_u \rightarrow Q_v$。

Dijkstra's 最短路徑搜尋法步驟如下：

[步驟一]：起始節點 Q_s 的代價函數初始值爲 0 $(cost[s]=0)$，計算所有節點與節點之間的代價函數爲 $cost[i]=cost[s]+d(s,i),\ i=2,3,\cdots,k$。

[步驟二]：尋找所有節點代價函數的最小值，並將此節點定義爲 Q_u 節點，代表著 Q_s 到 Q_u 間距離最短。

[步驟三]：依序計算由 Q_s 經 Q_u 節點到達其它節點 Q_v, $v=2,\cdots,k$ 之代價函數 $CV=cost[u]+d(u,v)$。CV 值與原先由 Q_s 直接到達 Q_v 節點之代價函數 $cost[v]$ 作一比較，當 CV < $cost[v]$ 表示從起始點 Q_s 經過 Q_u 節點到達 Q_v 節點的距離會比直接從 Q_s 到達 Q_v 節點的距離來得短。在這種情況下，我們可以將節點 Q_u 視爲 Q_s

到達 Q_v 節點之中繼點，並更新代價函數 $\cos t[v] = CV$ 及記錄 Q_v 對應的來源節點爲 Q_u，即 $index[v] = u$。 假如 CV > $cost[v]$ 時，表示由起始點到節點 Q_v 的距離較短，故 $cost[v]$ 保持不變。最後設定 Q_u 所對應處理完成旗標值 $flag[u]$=1 並將 $\cos t[u]$ 設定爲 inf，代表節點 Q_u 已處理完畢，不用再與其它節點比較距離。

[步驟四]：重複步驟二，直到每個節點之對應處理完成旗標 $flag[i]$ 值全部爲 1。

[步驟五]：由目標點開始藉由 $index$ 找尋其對應之來源節點，直到來源節點爲起始點爲止。將這些頂點依照搜尋順序反向排列後，便可找到從起始頂點 Q_s 到目標頂點 Q_k 的最短路徑。

參、模擬結果與討論

我們利用 Matlab7.0 工具箱的圖形使用者介面(Graphical User Interface；GUI)撰寫最短避障路徑模擬軟體，於 Pentium 4 (2.8GHz) CPU 個人電腦下執行。整合本文提出向量內積圓形偵測法、禁行路徑偵測法、Dijkstra's 最短路徑搜尋法，完成最短避障路徑搜尋之模擬驗證。經由我們建立的圖形使用者介面，可以任意於執行階段重新設定安全邊界、起始點、目標點參數的初始值，無需對程式進行任何改變，可以方便地觀察和分析參數變化對最短路徑搜尋的影響。

我們假設，以本文所提出向量內積圓形偵測法偵測每個影像圖片中圓形物體，並且計算出圓心、半徑及其外切正六邊形頂點座標，並且繪出所偵測的圓形物體及其外切正六邊形(以藍色線條標示)。接著考量輪型機械人最小旋轉半徑之問題，紅色線條繪出加上安全邊界的障礙物位置，即最外層的外切正六邊形，以其頂點爲 Dijkstra's 的節點。

程式執行時，我們先載入我們先前以 CCD 拍攝的照片，設定安全邊界的參數值、起始點及目標點，模擬程式會記錄所搜尋的最短路徑通過點於螢幕右上方，右下方記錄障礙物頂點(版面關係並未能全部記錄)。由本文所提出的幾何禁行路徑偵測法來決定兩兩頂點間的連通性，透過 Dijkstra's 演算法搜尋最短避障路徑。由模擬結果得知，本文所提出的『幾何禁行路徑偵測』法，能有效的偵測圖片中頂點間連通性，而運用 Dijkstra's 搜尋法可以完善的規畫出一條連接起始點與目標點之最短避障路徑。由實驗結果推論，本文所提出方法具有

不錯的實用價值。其中圖 10~15 說明最短避障路徑偵測處理的結果。

在圖 10 中，我們處理一個障礙物問題，並設定障礙物的安全邊界設爲 20 個單位，起始點設定爲 (10、450)，目標設定爲(500、75)，經由程式模擬，以藍線繪出搜尋後得到的圓型障礙物體之外切六邊型及其六個頂點，紅線繪出加入輪型機械人所需最小旋轉半徑安全邊界之外切六邊形及其六個頂點，直線爲 Dijkstra's 演算法搜尋到的最短避障路徑。在圖 11、12 中，我們增加障礙物的數目，設定不同的起始、目標點及安全邊界。直得一提的是，在圖 12 中，我們將障礙物及中擺設在起始點與目標點之間，且設定了較大的安全範圍，本文所提方法依舊可準確地找出避障路徑。圖 13 中，說明了當起始、目標點間沒有障礙物時，模擬的結果。在圖 14 ~ 15 中，我們設計了較多的障礙物，甚至在圖 15 中，設定較大的旋轉半徑安全邊界(相較於圖 12)，本文所提出「幾何禁行路徑偵測法」，皆可準確設計出最短避障路徑。

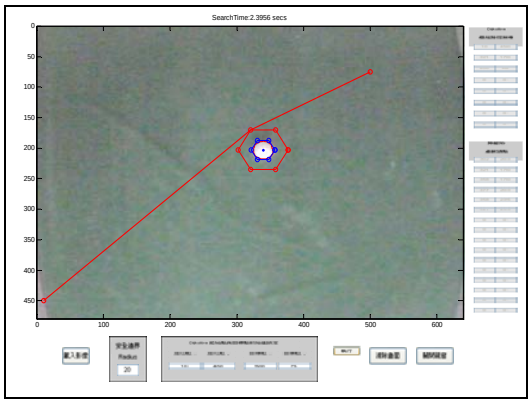

圖 10 起始點：(10、450)，目標：(500、75)
安全邊界 Rmin：20

圖 11 起始點：(200、225)，目標：(400、200)
安全邊界 Rmin：10

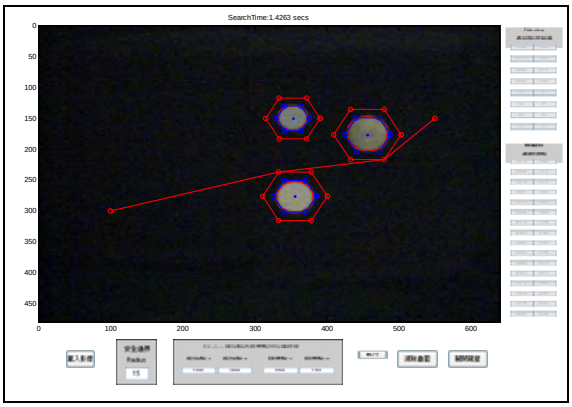

圖 12　起始點：(10、300)，目標：(550、150)

安全邊界 Rmin：15

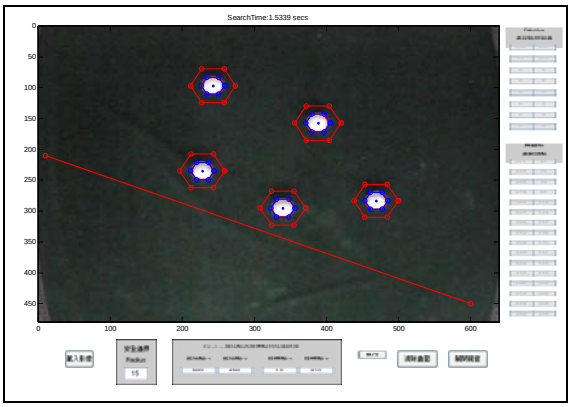

圖 13　起始點：(600、450)，目標：(10、210)

安全邊界 Rmin：15

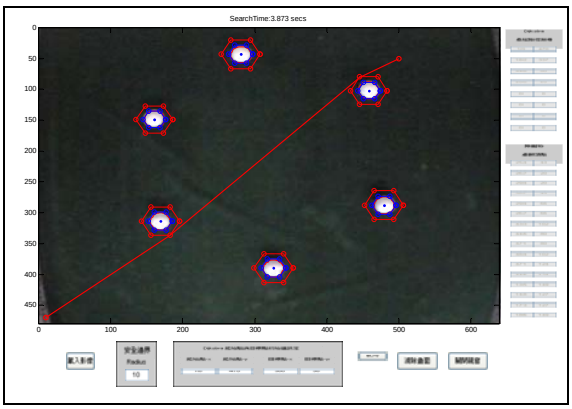

圖 14　起始點：(10、470)，目標：(500、50)

安全邊界 Rmin：10

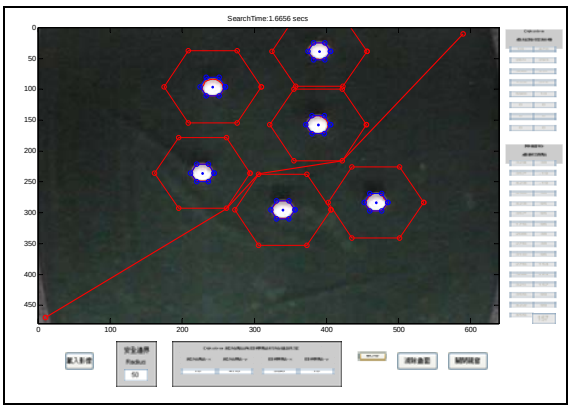

圖 15　起始點：(10、470)，目標：(590、10)

安全邊界 Rmin：50

肆、結　　論

本文利用影像處理的技術，以物件偵測爲基礎，提出「幾何禁行路徑偵測法」，並透過 Dijkstra's 搜尋法尋找一條連接起始點與目標之避障路徑，詳細記錄了這條避障路徑必須經過那些節點，可得到付出最少代價的最短路徑，做爲智慧型機器人路徑決策的參考。不論從起始點至目標點之間的障礙物擺設情況爲何，本文所提方法皆可正確地避開障礙物，並規劃出最短路徑。然而，本文僅限於平面上障礙物的考量，三維立體空間之研究、動態障礙物偵測、或者從圖片中截取部份像素特徵，進行圖形的比對及辨識搜索，也是相當值得探討的。

參考文獻

中文部分

簡卓建，應用模糊控制於自走車路徑導引避障之整合設計，國立成功大學碩士論文, 2003。[1]

林泰宏，自走車避障與導航性能之分析與設計，逢甲大學碩士論文, 2004。[2]

鍾翼能，雷達追蹤控制系統設計，大葉大學碩士論文, 2003。[3]

周利蔚，蔡樸生，陳珍源。轉折點偵測演算法在路徑規劃上的應用，九十五年中華技術學院校慶論文發表研討會, pp 11-22。[4]

張淙華，周利蔚，蔡樸生。向量內積偵測法在圓形識別上的應用，中華民國第十四屆模糊理論及其應用會議。[5]

陳閩雄，一個應用托勒密定理的隨機圓形偵測演算法，台北科技大學碩士論文, 2002。[6]

鍾國亮，影像處理與電腦視覺，東華書局。[7]

繆紹綱，數位影像處理 活用 Matlab，全華科技圖書。[8]

英文部分

J. Illingworth, and J. Kittler, A survey of the Hough Transform, Computer Vision, Graphics, and Image Processing, vol. 44, pp. 87-116, 1988. [9]

N. Kiryati, Y. Eldar, and A. M. Bruckstein, A Probabilistic Hough Transform, Pattern Recognition, vol. 52, pp. 57-77, 1990. [10]

Y. C. Cheng, and Y. S. Liu, A new Polling Method and Coaxal Transform for Robust Circle Detection , Proc. of the 4th Asia Conf. on Computer Vision, pp. 336-340, 2000. [11]

M. A. Fischler, and R. C. Bolles, Random Sample Consensus: A paradig fitting with application to image analysis and automated cartography , Communications of ACM, vol. 24, no, 6, pp. 381-396, 1981. [12]

T. C. Chen, K. L. Chung, An Efficient Randomized Algorithm for Detecting Circles , Computer Vision and Image Understanding, vol. 63, no. 83, pp. 172-191, 2001 [13]

E. Lucas, Recreations Mathematiques , 4 tomes, Gauthier Villars,. Paris, 1891, 1896, 1893, 1894. [14]

R. C. Prim, Shortest connection networks and some generalizations , Bell System Technical Journal. vol. 36, pp. 1389-1401, 1957. [15]

E. W. Dijkstra, A note on two problems in connexion with graphs , Numer. Math. vol. 1, pp. 269-71, 1959. [16]

D. B. Johnson, Efficient algorithms for shortest paths in sparse networks , Journal of the ACM, pp, 1-13, 1977. [17]

R. W. Floyd, Algorithm 97: Shortest Path, Communications of Assoc. Comput. Machinery. pp, 345, 1962. [18]

P. S. Tsai, Ph. D. Modeling and Control for Wheeled Mobile Robots with Nonholonomic Constraints , thesis, Univ. of NTU, 2006. [19]