

第二章 敞開式多面體之路徑規劃

在敞開式多面體搜尋最短路徑可分為規則性平面與不規則性曲面兩種，規則性平面以平面或平面與斜面所組成之多面體為代表，其搜尋之路徑僅沿著平面或斜面之表面，不規則性曲面則以自由曲面之多面體為代表，其搜尋之路徑則非僅局限在某些平面而已。本章將著重在敞開式多面體上探討避碰後之最短路徑，搜尋並登錄所有可通行之路徑與距離，並以資料結構之統計方法由所有可通行路徑中計算最短路徑。先前的學者曾針對上述問題提出解決方法，但大多局限於某個特定的範疇，因此，沿著物件表面以網格規劃法來計算最短路徑，此法最大優點是不論其領域有多複雜皆能獲得最短路徑，而其最大缺點是在複雜領域規劃最短路徑時其計算量亦很大。在敞開式多面體之路徑規劃流程係 1.利用網格規劃法以求得任意可通行點至其他可通行點之距離以產生鄰接矩陣。2.再以迪吉斯托(Dijkstra)來求得軌跡矩陣。3.從軌跡矩陣中自動搜尋最短路徑。

2.1 網格規劃法

在實際使用上網格規劃法約可分成三種連接方式：第一種網格連接方法是每個網格點可往前與其他非障礙區之任意網格點相連接，因為在平面內兩點間之最短距離是該兩點之連接線段，換言之，將網格點往前與每一個可通行之網格點連接，其連接線段長度即是該兩點間之距離，因此採用線段連接方式將每個網格點往前連接其他非障礙區之網格點，並記錄其距離，假如有 n 列而每列有 m 個可通行之網格點，則總格點 $N=nm$ 其連接線段總數為等差級數 $(N-1)+(N-2)+\dots+(N-3)+3+2+1+0=(N)\times(N-1)/2$ ，如選用迪吉斯托法來計算則鄰接矩陣之計算次數為 $P=N(N-1)/2$ 而軌跡矩陣之計算次數為 $Q=(N)^2$ ，因此在此範圍中任意兩點之間之計算次數必小於等於 $O(P+Q)=O((N(N-1)/2)+(N)^2)$ 。第二種網格連接方法是每個格點可與其前、後、左、右等四個相鄰格點連接，此法適用於立體停車場及無人搬運車之路徑規劃，但在多面體之路徑規劃上利用此法所求得之路徑其誤差值較大，假如有 n 列而每列有 m 個可通行之網格點，則其連接線段總數為 $2\times 4+2\times 3\times(m-2)+2\times 3\times(n-2)+4(n-2)(m-2)=4N-2n-2m$ ，如選用迪吉斯托法來計算則鄰接矩陣之計算次數為 $P=4N-2n-2m$ 而軌跡矩陣之計算次數為 $Q=(N)^2$ ，因此在此範圍中任意兩點之間之計算次數必小於等於 $O(P+Q)=O(4N-2n-2m+(N)^2)$ 。第三種網格連接方法是將選定的網格點與其前、後、左、右、與其對角相臨非障礙區之網格點相連接，在凹面與輪廓較複雜之凸面多面體之路徑規劃時採用此法即可獲得最短

路徑軌跡，假如有 n 列而每列有 m 個可通行之網格點，則其連接線段總數為 $P=2 \times 3 \times n + 2 \times 5 \times (m-2) + 8(n-2)(m-2) = 8N - 10n - 6m + 12$ ，如選用迪吉斯托法來計算則鄰接矩陣之計算次數為 $P=8N - 10n - 6m + 12$ 而軌跡矩陣之計算次數為 $Q=(N)^2$ ，因此在此範圍中任意兩點之間之計算次數必小於等於 $O(P+Q) = O(8N - 10n - 6m + 12 + (N)^2)$ ，迪吉斯托法將於下節中敘述。

網格規劃法是將平面或曲面分割為若干等份，將每個方向之線結構分別置放在彼此正交之座標軸上，於二維平面之網格規劃即是在作業平面上，於水平 X 方向區分為 i 個網格點，在縱向 Y 方向區分為 j 個網格點，以 $P[i]$ 及 $P[j]$ 代表在平面上任意網格點的位置，平面上所有的點集合以 $P[i][j]$ 來表示如圖 2.1 所示。如在三維空間則增加 $P[k]$ 之每一個點集合視為一層，共有 k 層的 $P[i][j]$ 點集合，組成 $P[i][j][k]$ 之三維曲面，圖 2.4 即表示由多個不同高度平面所組成之多面體。

將障礙物以物體擴張法使每一障礙區之周邊擴大，成為多邊形的直線架構，捨棄與障礙區周邊相同之網格點以及障礙區內之網格點集合，條件符合者視為可通行點，由起點與目標點間之可通行點集合重新編排，並記錄其順序編號而得新的點集合。將可通行點互相連接並判斷該線段與障礙區周邊有無相交，捨棄與障礙區周邊相交之線段，並依序而不重覆記錄由任意點至其他點之編號與其間的距離。分別對平面、非共面平面多面體與自由輪廓多面體等為例做說明。

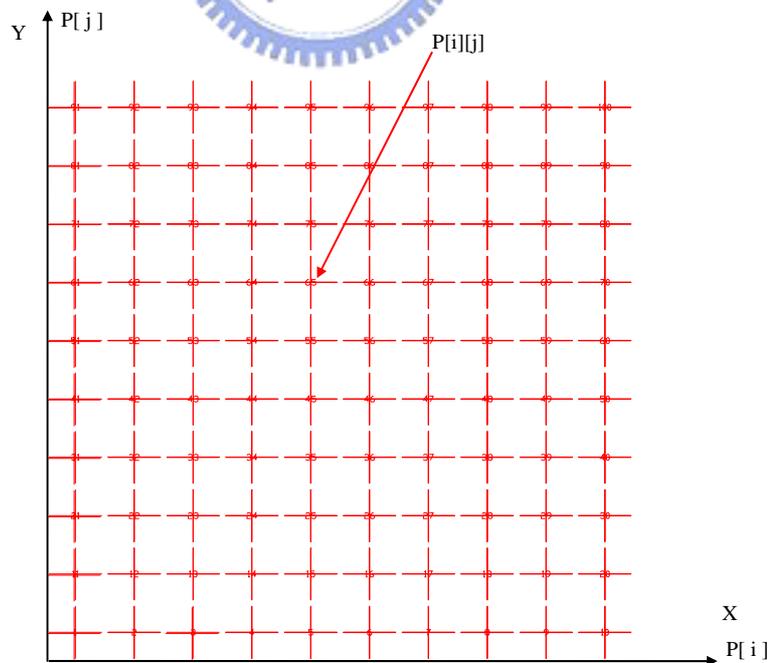


圖 2.1 由 $10 \times 10 = 100$ 個點集合所組成的網格平面

以二維平面結構在 X 及 Y 方向各分割為 10 等份為例，依網格規劃原理應有 $10 * 10$ 共 100 個點集合所組成的網格平面如圖 2.1 所示。假設任意給定的條件為：障礙區之周邊由 30 個網格點所組成，障礙區域內之網格內有 20 個點集合，因此由起點 1 至終點 100 之間可通行的點集合為 50 個網格點集合，將該 50 個網格點重新編排並付予名稱代號，如圖 2.2 所示。在圖 2.2 中粗線段所圍成的區域稱為障礙區，而障礙區總共由 30 個頂點分別組成七個障礙區，在障礙區範圍內不包括頂點的數目共有 20 個點，因此剩下 50 個網格點稱為可通行點，如圖示重新將所有可通行點依序編排並由 1 開始編號至 50 號為止，其中可以任取一點當作工作起點，同樣的也可以任意的取其中一點當成終點，假如取第 1 點為工作起點，取最後 50 點當成目標點，則路徑之向量即由第 1 點往第 50 點方向前進。

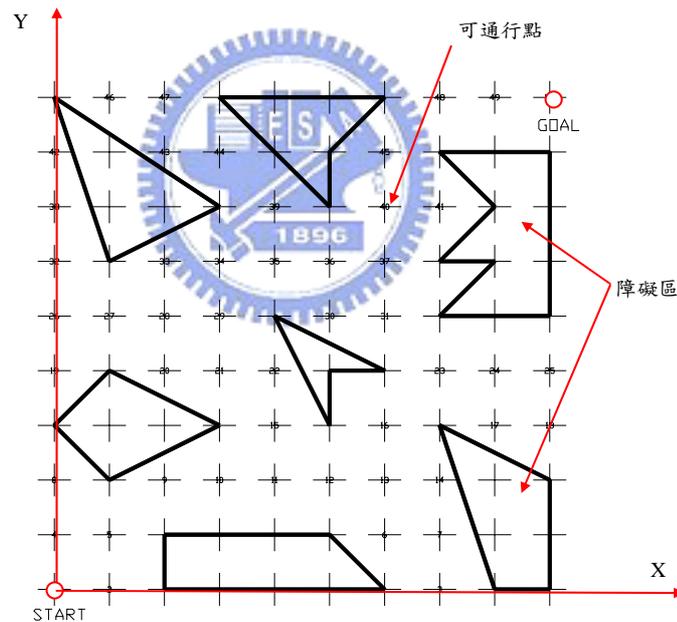


圖 2.2 依序將可通行點重新編排並付予名稱

當網格數目不同或障礙區範圍內之不可通行點不同時所產生之可通行數量會隨著網格數量而改變，在上例中經過重新編排之點集合依序排列而不重覆分別連接，理論上會產生 1225 個線段，但每個可通行點與其他可通行點集合之連線必須捨棄與障礙邊界相交之線段，因此可通行線段將減少甚多，如障礙愈多可通行線段就愈少，障礙擺放位置不同也會有不同的可通行線段，該線段的數目需視障礙邊

界的位置與障礙區內之不可通行點數之多寡而定，則未與障礙區發生碰撞可通行之避碰軌跡即如圖 2.3 所示。

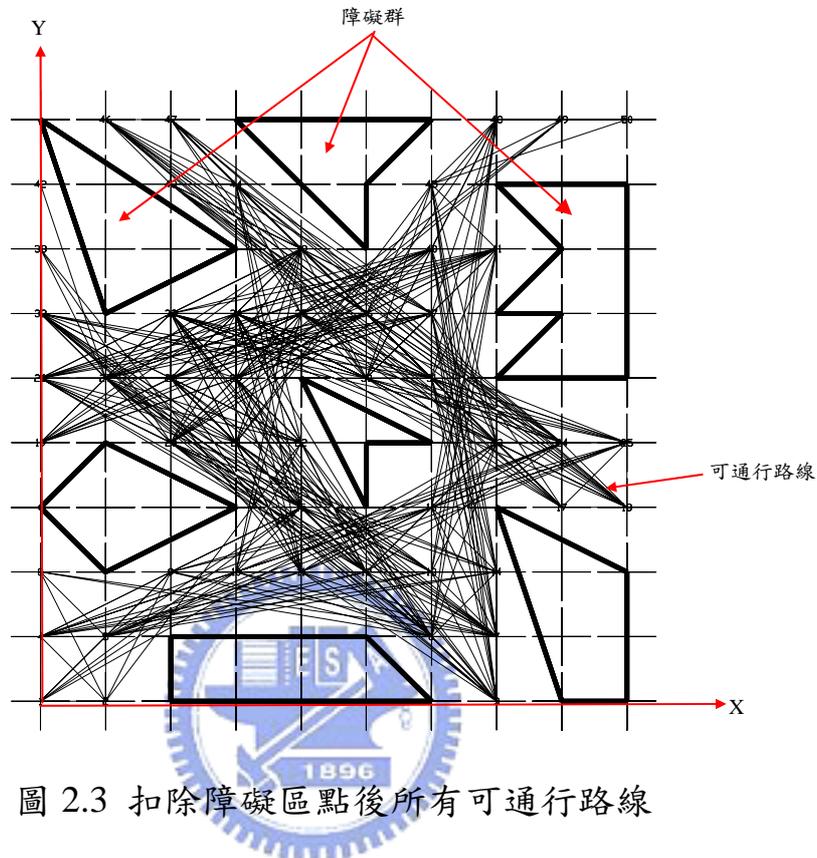


圖 2.3 扣除障礙區點後所有可通行路線

同理在兩個平面與一個斜面所建構之三維多面體如圖 2.4 所示，所有的通行路徑必須沿著多面體的表面是此路徑規劃之限制，因此可通行點之連線，除了前例所述之條件外，必須再刪除非貼於平面或斜面上之所有路線，限制增加符合條件之可通行路線則變少，將此例搭配與圖 2.2 完全相同的障礙邊界及座標位置，此例之限制為所有障礙位置應隨著平面變化，並沿著不同平面而附著於該平面上。在第一平面與斜面之間有 10 個網格點相重疊，因為必須去除不在斜面與平面上的通行路線，且第一平面上之可通行點不能直接與斜面或第二平面上之可通行點相連接，依表面規劃法第一平面之可通行點至斜面上之可通行點必須先經過該 10 個重疊點，相同的在斜面上之可通行點也不能直接與第二平面上可通行點相連接，欲連接第二平面上之可通行點時，亦須先經過斜面與第二平面之相重疊點，換句話說即是所有可通行點必須先經過面與面之轉折邊界重疊點才可與不同平面或斜面上之可通行點相連接，如前例所述，路徑規劃之向量仍為由起點至終點之線性方向。

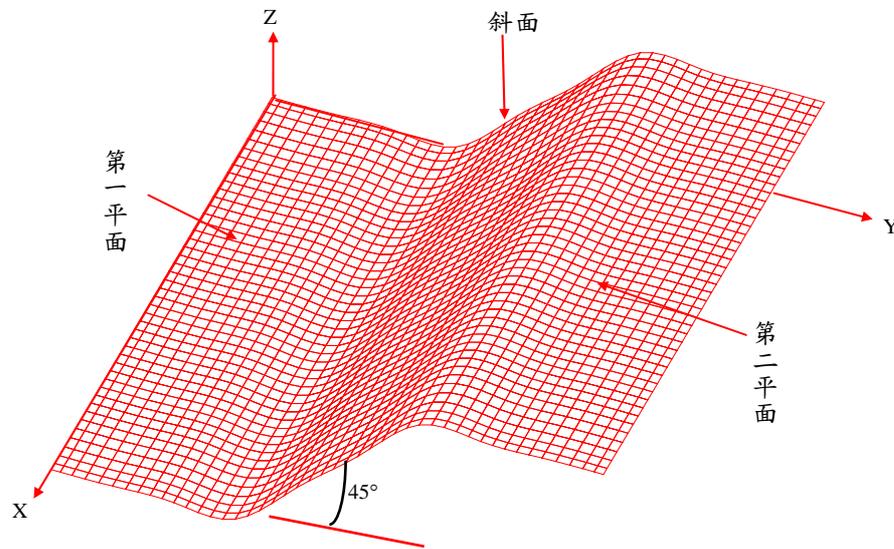


圖 2.4 由兩個平面與一個斜面所產生的非共面平面多面體

使用與第一個例子相同的 X 與 Y 障礙邊界條件，但在 Z 方向的點有所改變，第一平面全部由 $Z=0$ 所組成，第二平面則全部由 $Z=3$ 所組成，斜面則分別由 $Z=0,1,2,3$ 所組成如圖 2.5 所示。同理可得到 50 個可通行網格點，將該可通行點重新依序編號，然後去除與障礙區邊界相交及非附著於平面或斜面上的路線，記錄每條線的兩端點編號與距離，供求取最短路線時使用。

如圖 2.5 中在第一平面上有 18 個可通行點，表示 $N=1\text{---}18$ 並分別給予編號，為了避免重複計算，採用由小到大的計算方式，在編號 1 的可通行網格點需與其他 17 個網格點相連接，編號 2 需與其他的 16 個網格點相連接，依次類推，編號 16 需與 2 個網格點相連接，編號 17 需與 1 個網格點相連接，求得並記錄在第一平面上可通行點與點間之距離，結果在第一平面上共有 153 個可通行線段。在斜面上有 23 個可通行網格點，即表示 $N=15\text{---}37$ ，因為該可通行點經重新編號之後在斜面上的點是 15 到 37，利用前述方法求得並記錄斜面上可通行的點與點間之編號與距離，結果在斜面上共有 253 個可通行線段。在第二個平面中有 19 個可通行網格點，而經重新編號之後該點為 $N=32\text{---}50$ ，利用前述的方法求得並記錄斜面上可通行的點與點間之距離，在第二平面上共有 171 個可通行線段。因為所有可通行點必須先經過面與面之轉折邊界重疊點才可與不同平面或斜面上之可通行

點相連接，總共的可通行線段共有 577 條($153+253+171=577$)。

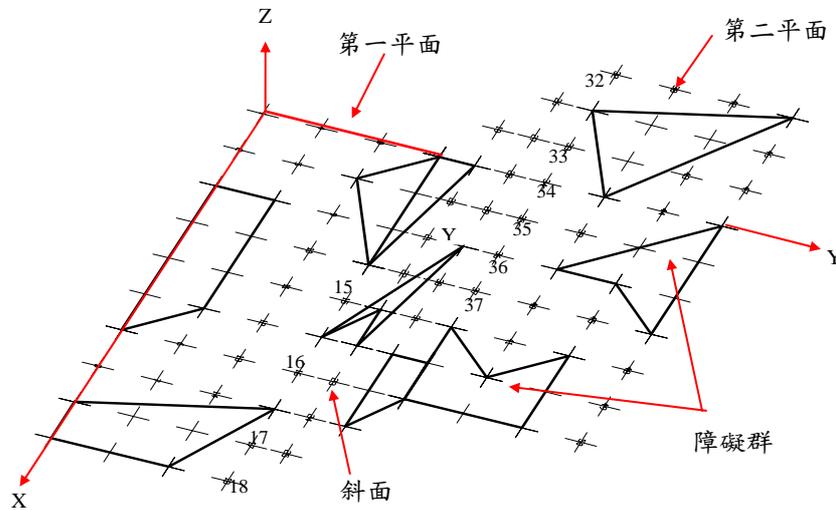


圖 2.5 在非共平面上所有可通行的點

因為路徑規劃的軌跡必須沿著平面與斜面上行進，因此在第一平面上的可通行點不能隨意的連接至斜面與第二平面上的任何網格點，因為有些連線並未貼附於多面體的表面上行走。所以在第一平面上的網格點要連接斜面的可通行點時必須經過編號 15、16、17、或 18 的網格點。同理在斜面上的點要連接到第二平面的可通行點時也必須要先經過編號 32、33、34、35、36、或 37 的網格點。因為第一平面上的網格點是不能直接與第二平面相連接，且限制連接的順序是由小到大，所以在斜面上的網格點與第二平面上的網格點是不能連接第一平面上的網格點如圖 2.5 所示。

將第一平面、斜面與第二平面所有符合條件的可通行路線相加如前述可得到 577 條可通行路線，而這些路線包括與障礙區相交的路線，再研判所有路線與障礙邊界有無相交，並捨棄有相交之線段，不過上述的 577 線段當中沒有互相重複的線段，例如由網格點編號 1 到網格點編號 2 的距離是 2.54 單位，當然由網格點編號 2 到網格點編號 1 的距離也是 2.54 單位，規劃由小到大的計算方式即可避免重複運算，而此限制在規劃時即需要做一個轉換計算，假如要計算由網格點 50 至網格點 1 的最短距離，就要轉換成由網格點 1 至網格點 50 的最短距離，因為這兩種方法所計算出來的結果應該是相同的，因此得到在非共面平面多面體上所有可通行的網格點編碼及可通行點至

其他可通行點的距離，最後取得距離最小值當成最短路徑軌跡，在非共面平面多面體中所有可通行的路線如圖 2.6 所示。以圖 2.6 及圖 2.3 之可通行路線做比較，因圖 2.6 的限制條件比圖 2.3 的限制多，雖然網格點的數量一樣，且障礙區也相同，但限制條件不同所產生的可通行路線也不同，圖 2.6 中所得到的可通行總路線較圖 2.3 少了許多。

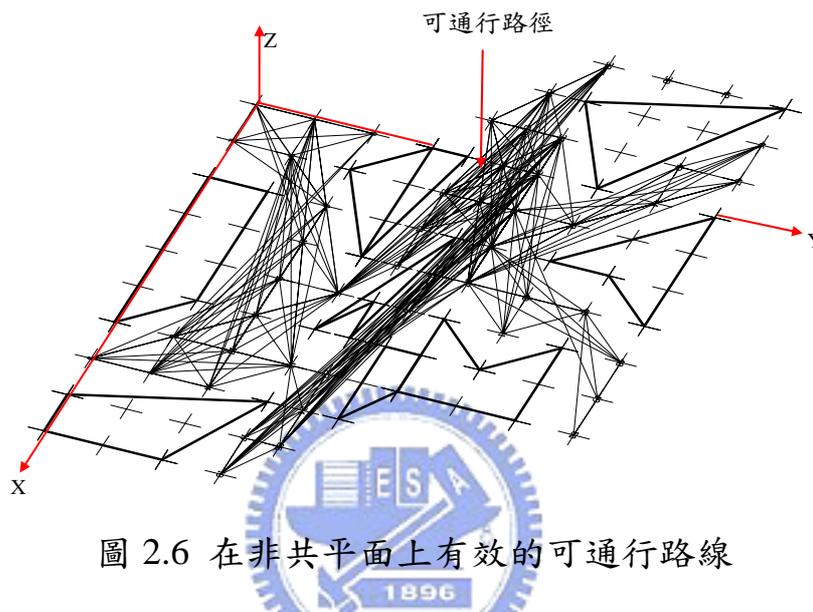


圖 2.6 在非共平面上有效的可通行路線

第二個例子以多個山形輪廓多面體做路線規劃說明如圖 2.7 所示，將該山形輪廓區分為山腳部份、山腰部份與山頂等部份來規劃可通行路線，依照表面搜尋法的限制，將山形輪廓部份當成是平面上的障礙區，並以山腳所座落的位置來尋找在平面上的障礙區，障礙區的頂點不能選在網格點上，因為這樣會使平面的網格點無法連接到山腳所座落的網格點，在平面上的網格點之間是可以相互連接，但必須捨棄會與山形輪廓障礙區相交的線段，而在平面上的任一網格點與山腰與山頂連接之前就必須先與山腳的網格點連接，相同的由山腳的網格點與山頂的網格點相連接時就必須先通過山腰上的網格點，為了使選擇的路線不脫離多面體的表面，在山形輪廓之底部區域以多邊形平面來建構障礙區(山腳、山腰、與山頂等部份)，連接並判斷網格點間的連線是否會與障礙區平面產生相交，捨棄所有與障礙區相交的線段即為可通行路線。

網格點的編號仍由起點至目標點依序增加，在規劃的初期連同平面與山丘一起編號，在計算平面之可通行路徑時需將山腰與山頂的網格點視為不可通行點，在計算由山腳至山腰與由山腰至山頂的路徑則

將平面的網格點視為不可通行點，最後將兩種的結果合併就可得到所有可通行之路徑。

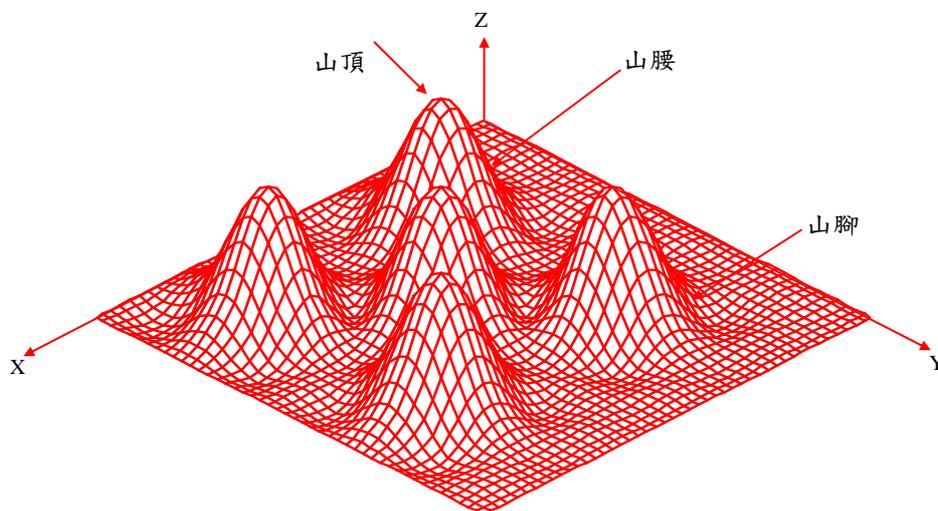


圖 2.7 自由輪廓多面體的座標空間示意圖

將圖 2.7 中自由輪廓多面體區分成兩個部份來處理，第一個部份是山丘以外之平面部份，其路徑規劃與上述平面避障規劃相同，必須先找出山腳座落在平面上之網格點，將山丘當成是網格平面之障礙區，並刪除障礙區內的所有網格點，除了山腳座落於平面上的網格點外，其餘山丘上的網格點皆當成不可見點。為了讓平面上之點群能夠連接山腳之點集合，在規劃障礙區時，選擇現有山腳頂點間之中點，再連接各中點成封閉之多邊形如圖 2.8 所示。

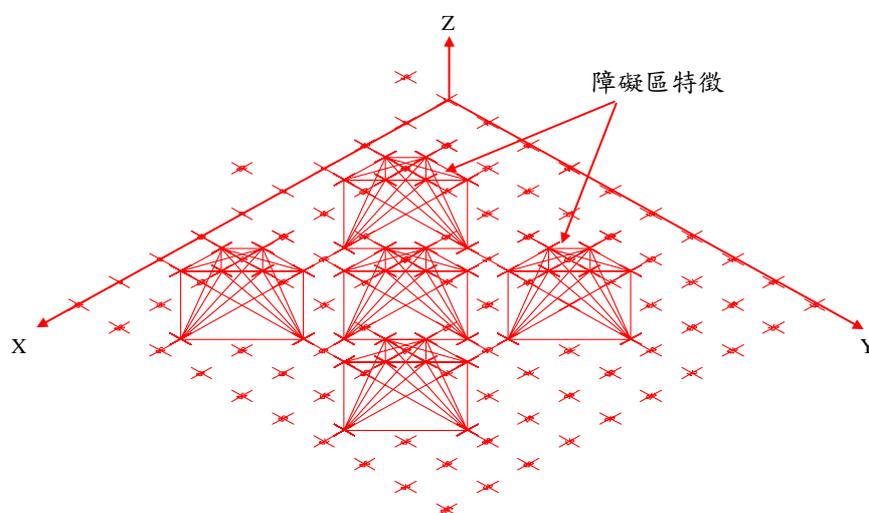


圖 2.8 自由輪廓多面體的障礙特徵

圖 2.8 顯示已經編號的網格點，其編號的順序是由左至右由下至上，而障礙區的限制條件是讓平面上之任意點在到達山腰及山頂之前先經過山腳部位，由網格點間之連線來判斷該線段有無與障礙邊界相交，並捨棄與障礙邊界相交之線段。設定障礙區之後，刪除障礙區內之網格點再重新編號，依據網格規劃法來求出所有可通行的網格點，避開與障礙邊界相交的路線，找出所有可通行的路線如圖 2.9 所示，並記錄各可通行點至其他可通行點之編號與距離。

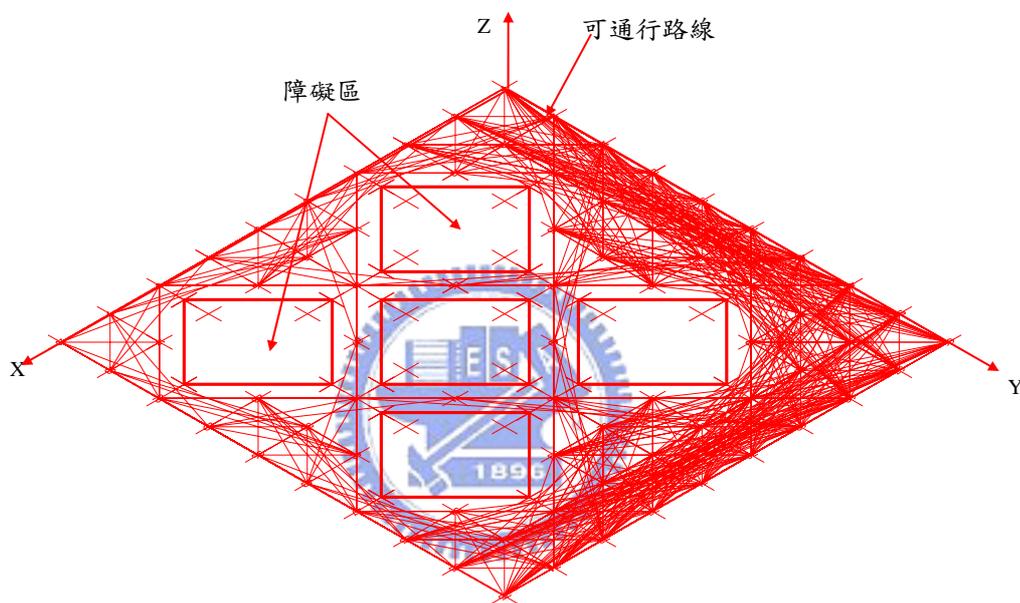


圖 2.9 自由輪廓多面體中同平面部份的可通行路線

第二個部份就是山丘多面體部份，其規劃方法是由線段與線段求交點轉換為線段與平面求交點。下列將針對山丘的路線規劃作探討，如圖 2.7 所示山丘分成三個階段來討論：第一階段為山腳部份由平面之網格點所組成，該網點與平面部份的可通行點相重疊，平面上的點必須經過該網點才能與山腰與山頂部份之可通行點連接。第二階段山腰部份是由高度為一個單位的網點所構成，第三個階段為山頂部份由高度為三個單位的網格點所組成。在此例中有五個山丘，因此有五個第二部份。在山丘路線規劃的限制即是可通行點不能任意的連接，因為在山腳或山腰的部份不能直接穿透連接，因此山腳或山腰的網點要先經過山腰或山頂的可見點，由山頂與山腰的網格點分別建構多個平面、再由山腰與山腳的網格點分別建構多個平面，以做為障礙邊界之

特徵，隨後判斷在山丘上網格點間之連線與障礙區有無相交，並捨棄相交路線，即可得到所有在山丘上的可通行路線，自由輪廓多面體山丘部份所有的可通行路線如圖 2.10 所示。

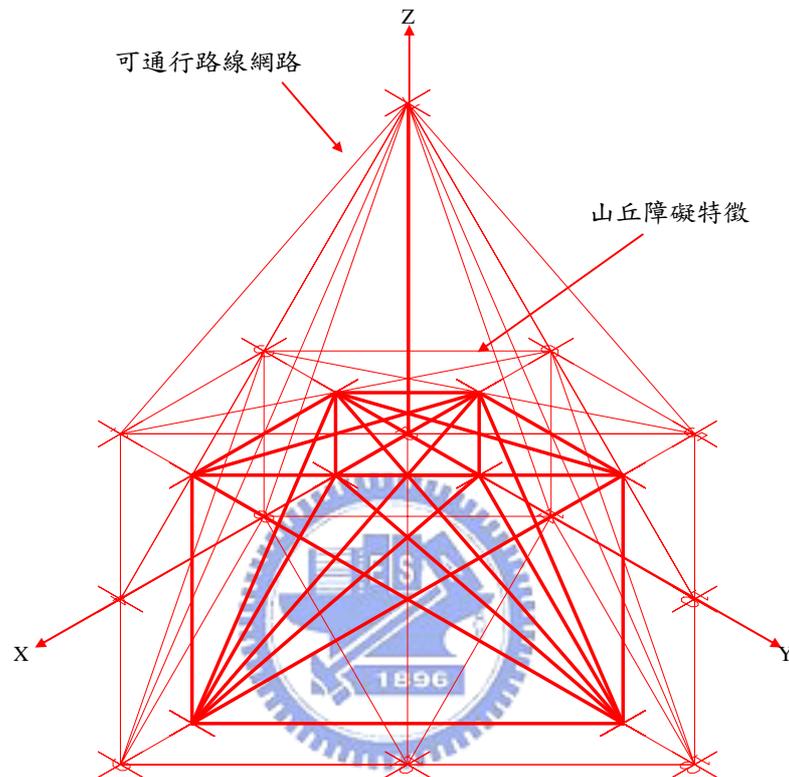


圖 2.10 自由輪廓多面體山丘部份的可通行路線

總而言之，山丘路線規劃必須根據下列的三個法則：第一個法則，山腳與山腰及山頂的可通行網格點必須是可見點才能連接，此條件在障礙區的設計時即需加以考慮，意即在判斷與障礙邊界是否相交時就能滿足該條件。第二個法則，平面與山丘之問題必須分開處理，但最後仍需結合整體計算。第三個法則，因為可通行點是先經過排列編號，因此平面及山丘相重疊的網格點編號必須相同，且點集合也必須依大小順序來編排，包括平面與山丘之點集合。至於可通行路線是否與障礙相交之問題與前述方法一樣，結合平面與山丘之可通行點與路線，在結合過程必須要確認網點編號是否依序編排，如果順序錯誤則所求得之最短路徑即不正確，結合平面與山丘之可通行路徑，即成為整個自由輪廓多面體之可通行路徑如圖 2.11 所示，以檔案儲存方

式記錄每個可通行網格點至與障礙區無相交的其他可通行網格點的距離，方便爾後計算最短路徑之用。

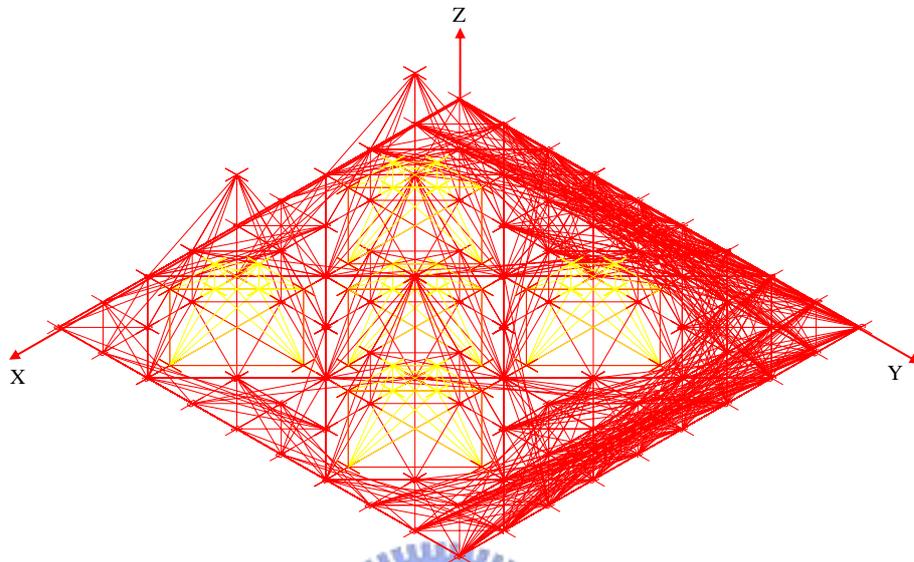


圖 2.11 自由輪廓多面體可通行路線的結合圖形

2.2 迪吉斯托演算法

在可通行之軌跡中尋找最短路線的求法，即由起點至目標點間之可通行連線，以其距離之累加來求出最短距離之路線，即是所謂迪吉斯托演算法。該演算法之基本觀念是設定並逐步闡充集合 S ，存放已求出最短路線之頂點，而尚未確定最短路線之頂點集合 V ，演算法初始化時 S 只有一個源點就是預先設定的起點，以後每一步都是按最短路線遞增的順序，逐個的把 V 中的下一個最近距離之頂點加入到 S 中，若從起點到另一點的路線不存在，則可假設由起點至該點的路線是一條長度為無窮大的虛擬路線，在演算法中必定是最後才會放入 S 中，而尋找到目標點時終止。亦即在目標點被選取之前，無窮大的虛擬路線是不被選到，而沒有被選取之點即視為不可通行點。首先將所有的矩陣元素設定為一個很大值，再以網格規劃法所求得並記錄之每個連線起點與其他點之連線距離分別加入矩陣元素內，而新得到的矩陣結果就是所謂的鄰接矩陣。迪吉斯托演算法的運算原理就是以鄰接矩陣之第一行元素為基準，選擇該行最小值，與其下一行之各元素相加運算後並與該行矩陣相對位置之元素相比較，以較小的值來寫入下一行位置而得軌跡矩陣，最後從已完成軌跡矩陣之內容即可求得最短的通行路線。其詳細的計算過程如下說明：

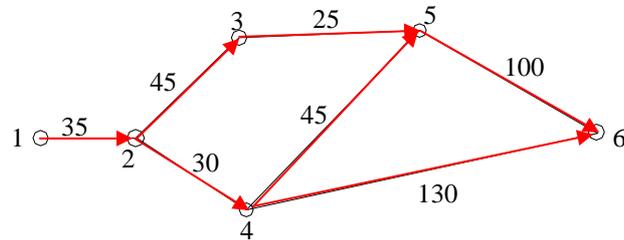


圖 2.12 可通行路線與節點間之距離

(1)、將鄰接矩陣之第一個元素更改為零，因為起點至起點的距離為零，然後將第一列內容拷貝至軌跡矩陣之第一列，記錄第一個起點編號，並將該起點編號編入 S 中。

(2)、接著執行頂點總數減一次迴路：

a. 從起點至同列之各元素中選擇最短距離值之頂點稱為點 w，且此頂點是在 V 內。然後將該點編入 S 中，在比較之過程中是所有元素均要反覆比較。

b. 將軌跡陣列中 w 點之值加上鄰接矩陣中 w 點該列之所有元素值再與軌跡矩陣中之相對元素比較大小，並將距離較小者存入軌跡矩陣下一列的每一個位置。

以六個頂點為例如圖 2.12 所示，各頂點間均有其方向性，頂點 1 只有至頂點 2 一個路線其距離為 35，頂點 2 則可至頂點 3 其距離為 45 與頂點 4 其距離為 30，頂點 3 只可至頂點 5 其距離為 25，頂點 4 可至頂點 5 其距離為 45 與頂點 6 其距離為 130，頂點 5 也只可至頂點 6 其距離為 100，如圖 2.12 所示。由 1 到 3、4、5、6 均要先通過 2，亦即由 1 到 3、4、5、6 之距離為無窮大 ∞ ，相同的由 2 到 5、6 之距離亦為無窮大 ∞ ，依此類推可通行路線可分別列出如下：

$$\begin{aligned}
 1 \rightarrow 2 &= 35, & 2 \rightarrow 3 &= 45, \\
 2 \rightarrow 4 &= 30, & 3 \rightarrow 5 &= 25, \\
 4 \rightarrow 5 &= 45, & 4 \rightarrow 6 &= 130, \\
 5 \rightarrow 6 &= 100,
 \end{aligned}$$

設定一個矩陣其初始元素皆為 ∞ ($p[i][j] = \infty ; i = j = 6$)，將上述

之可通行路線之第一列編號代入 i ，第二列編號代入 j ，第三列編號之距離值替代 ∞ ，則可得鄰接矩陣之內容如下表所示：

表 2.1 從起點 1 至終點 6 的鄰接矩陣

Points	1	2	3	4	5	6
1	∞	35	∞	∞	∞	∞
2	∞	∞	45	30	∞	∞
3	∞	∞	∞	∞	25	∞
4	∞	∞	∞	∞	45	130
5	∞	∞	∞	∞	∞	100
6	∞	∞	∞	∞	∞	∞

資料來源 Reference [17]

由表 2.1 鄰接矩陣之內容中，先將第一個元素更改為 0，再將第一列的值全部拷貝至軌跡矩陣中，依上述計算過程之操作步驟即可求得軌跡矩陣中每一個元素，其結果如表二所示。

表 2.2 從任意起點至任意終點的軌跡矩陣

迴路	陣列 dist						最短距離 頂點 w
	1	2	3	4	5	6	
1	0	35	∞	∞	∞	∞	---
2	0	35	80	65	∞	∞	2
3	0	35	80	65	110	195	4
4	0	35	80	65	105	195	3
5	0	35	80	65	105	195	5

資料來源 Reference [17]

從軌跡矩陣中可得到由起點至終點距離最短之路線，如表 2.2 中由點 1 至點 6 其最短距離為 195，而其所經過的軌跡為 1→2→4→6，軌跡的選取步驟為在軌跡陣列中編號 6 列中選取第一個最小值為其最小距離值 195，然後往左選取與前點最短距離的點編號為 4，其次以軌跡陣列中編號 4 列中選取第一個最小值為其最小距離值 65，而往左選取與前點最短距離的點編號為 2，相同的在軌跡陣列中編號 2 列中選取第一個最小值為其最小距離值 35，再往左選取與前點最短距離之點編號為 1，因此得到最佳路線軌跡為 1→2→4→6。同理，如果將終點改成編號 5，則最短距離值為 105，然後依上述的方法可以

找出最短路徑的軌跡為 1→2→3→5。依次類推，則可得到由任一點至其他點的最短距離及路線。

2.3 範例說明

有幾個不同外形之障礙區域座落在網格平面區域內，為了計算方便以多個網格點來界定障礙區之外緣，將障礙區以多邊形來表示，在作業平面內貼附網格，而避碰區域以網格搜尋技巧來尋得，依序標示網格點之編號，扣除在障礙邊緣及障礙區內部之網格點，再重新標示依序標示可通行網格之編號，以迪吉斯托理論從可通行路段中來搜尋最短路徑如圖 2.13 表示，此例子表示所提出的方法是一個在複雜與龐大的領域內有系統搜尋最短路徑之方法。而迪吉斯托理論在所有可通行路徑中統計其路徑距離，計算由起點 1 至目標點 50 之最短路徑，其最短路徑軌跡為 1→10→35→49→50，如果以每一間隔為一個單位，則在 100 個網點的網格平面上所計算的最短路徑，其距離總合為 13.7287 單位。

範例一、在平面上有一些障礙區域中找出最佳的路徑規劃：

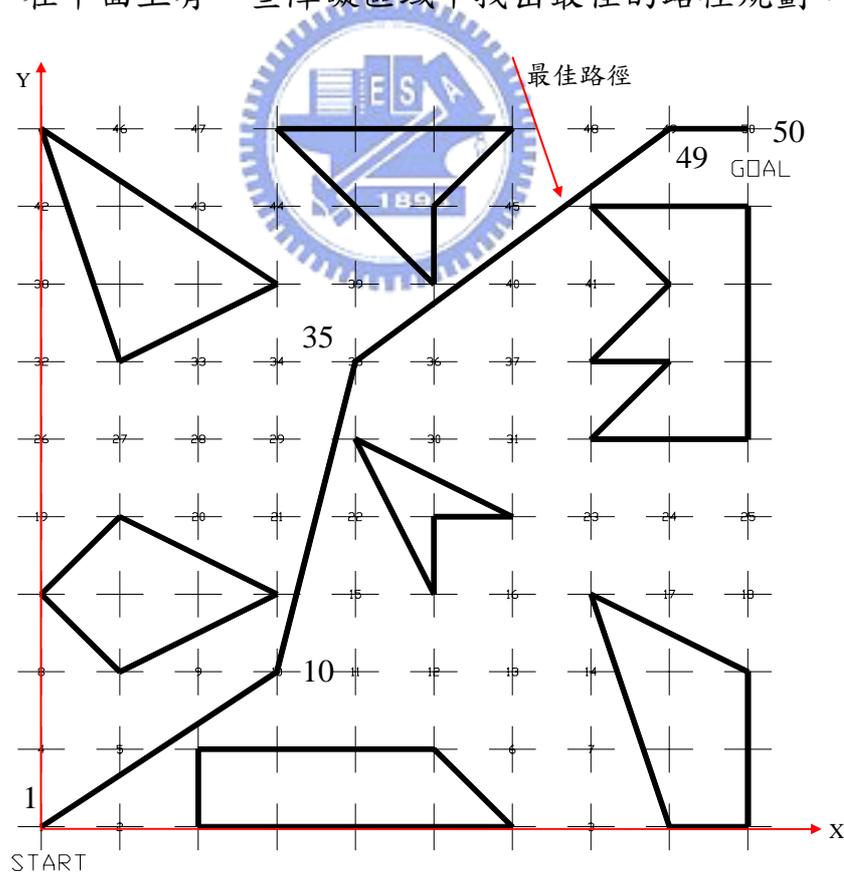


圖 2.13 在平面上從 1 到 50 的最短避碰軌跡

如果將圖 2.13 的網格間距縮小，由原先的 100 個網格點擴充至 361 個網格點，相同的利用網格規劃法在 361 個網格點之中計算所有可通行路徑，扣除障礙邊界頂點以及障礙區內的網格點，共有 225 個可通行點，並使用迪吉斯托理論在所有可通行路徑中計算其加總距離，計算由起點 1 至目標點 225 的最短路徑，並且求得其最短軌跡是 $1 \rightarrow 56 \rightarrow 112 \rightarrow 180 \rightarrow 211 \rightarrow 225$ 如圖 2.14 所示，如果以每一間隔為一個單位，則在 361 個網點的網格平面上所計算的最短路徑，其距離總合為 12.8707 單位，很顯然的其最短路徑的距離總合較 100 個網格點所求得的最短距離小 0.858 個單位。

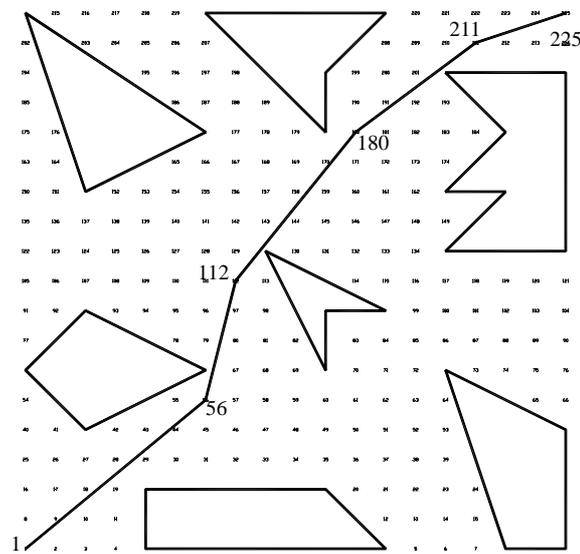


圖 2.14 從起點 1 到目標點 225 的最短避碰軌跡

如果將網格的間距再縮小，由原先 361 個網格點擴充至 784 個網格點，利用網格規劃法在 784 個網格點之中計算所有可通行路徑，扣除障礙邊界頂點以及障礙區內的網格點，共有 519 個可通行點，判斷所有的網格間之連線是否與障礙邊界相交，並使用迪吉斯托理論在所有可通行路徑中計算其加總距離，計算由起點 1 至目標點 519 的最短路徑，並且求得其最短軌跡是 $1 \rightarrow 55 \rightarrow 175 \rightarrow 271 \rightarrow 383 \rightarrow 478 \rightarrow 519$ 如圖 2.15 所示，如果以每一間隔為一個單位，則在 519 個可通行網點所計算的最短路徑，其距離總合為 12.1773 單位，其距離總合比 361 個網格點所求得的最短路徑距離總合小 0.6934 個單位，又比 100 個網格點所求得的最短路徑距離總合小 1.5514 個單位。當網格數量增加時其最短路徑就更精確，但是當網格大小增加到一定數量時其誤差會愈來愈小，而產生收斂效果，網格繼續增加但其結果卻只有些微的

改變。如何選擇適當網格點數來規劃最短路徑？則以所需要精度要求來決定，如大賣場之逃生路線、船隻的航行路線、電子地圖、汽車駕駛塞車時之衛星導行系統等場合則可選擇較少之網格點數量，但用於機器人之運動軌跡設計等精度較高之路線規劃時則應選擇較多之網格點。

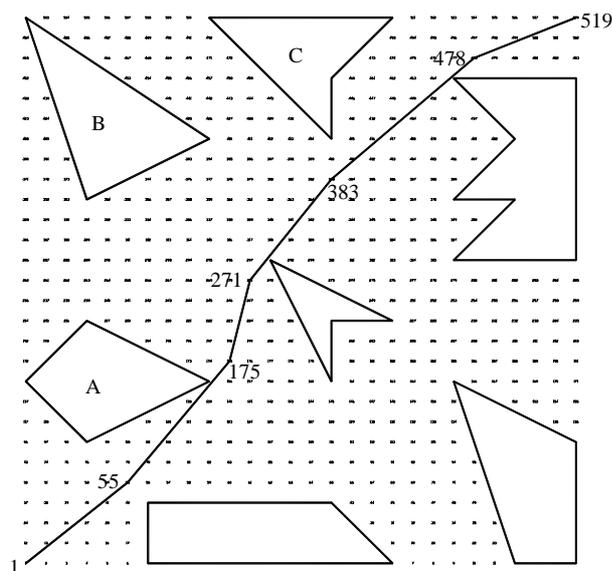


圖 2.15 從起點 1 到目標點 519 之最短避碰軌跡

如果網格點之間距大小不變，依起點與終點之向量而言，在圖 2.15 中左上方將障礙區 A、礙區 B、礙區 C 等加以合併，不影響最短路徑之搜尋，如圖 2.16 所示改變左上方的障礙區大小與數量，即可以獲得共有 374 個可通行網格點，判斷網格間之連線是否與障礙邊界相交，並使用迪吉斯托理論在可通行路徑中計算其距離，由起點 1 至目標點 374 的最短路徑，並且求得其最短軌跡是 1→55→172→233→299→350→374 如圖 2.16 所示，如果以每一間隔為一個單位，則在 374 個可通行網點所計算的最短路徑，其距離總合為 12.1773 單位，其距離總合與相同於在 519 個可通行網格點之最短路徑距離。

如在圖 2.16 中將障礙區 A、礙區 B、礙區 C 等加以合併，如圖 2.17 所示改變障礙區大小與數量，即可以獲得共有 280 個可通行網格點，判斷所有的網格間之連線是否與障礙邊界相交如圖 2.18 所示，並使用迪吉斯托理論在所有可通行路徑中計算其加總距離，計算由起點 1 至目標點 280 的最短路徑，並且求得其最短軌跡是 1→30→113→146→218→256→280 如圖 2.19 所示，如果以每一間隔為一個單位，則在 280 個可通行網點所計算的最短路徑，其距離總合為 12.1773

單位，其距離總合與相同於前兩例所求得的最短路徑距離總合。

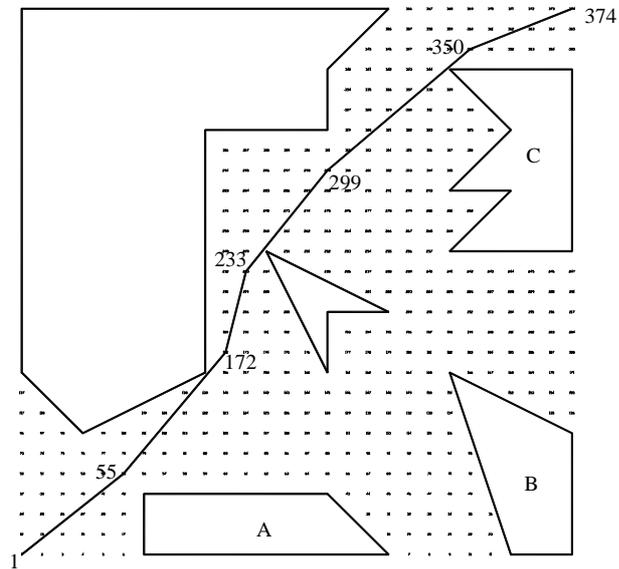


圖 2.16 從起點 1 到目標點 374 的最短避碰軌跡

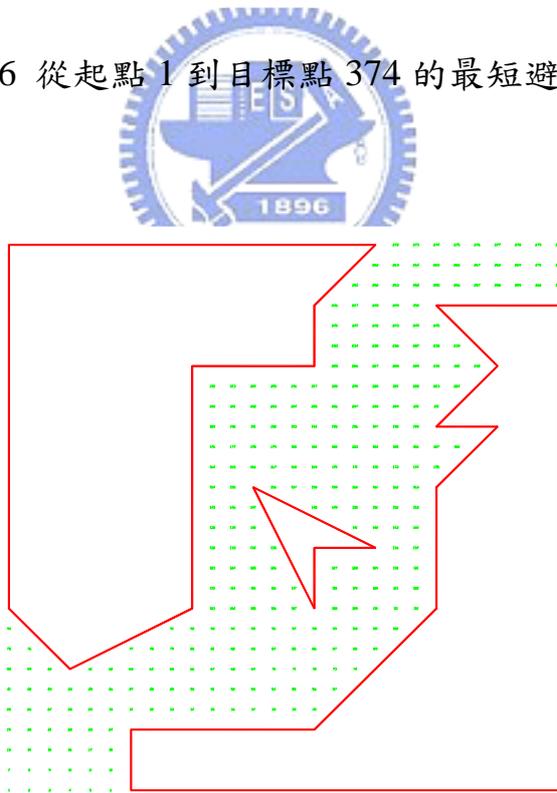


圖 2.17 合併右側障礙區後共有 280 個通行點

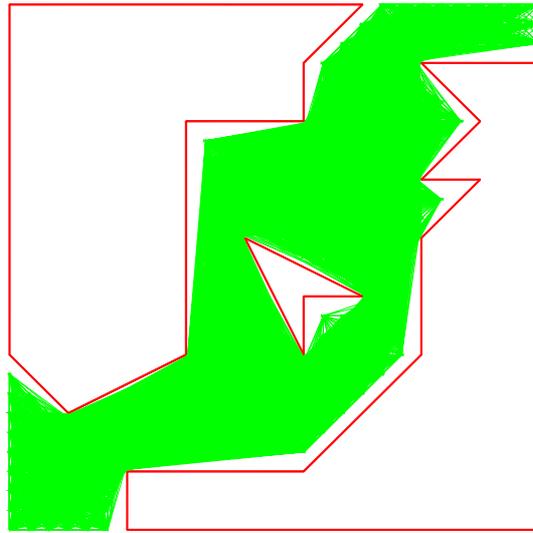


圖 2.18 從起點 1 到目標點 280 的所有避碰軌跡

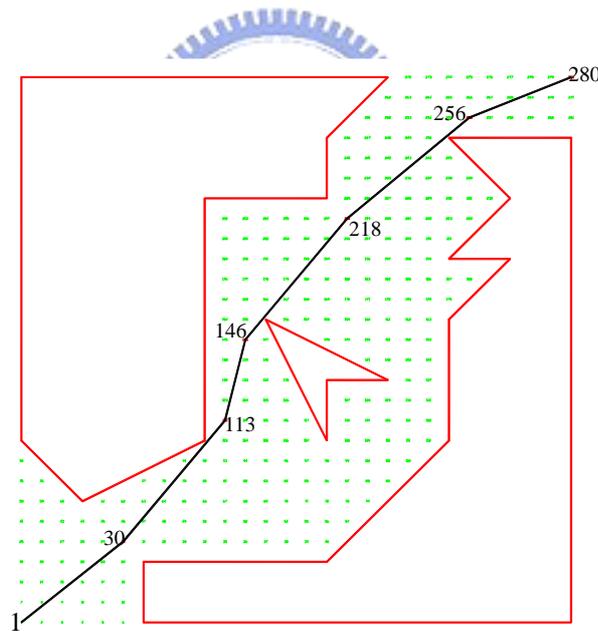


圖 2.19 從起點 1 到目標點 280 的最短避碰軌跡

由上述可得網格分割的數量愈細，所得到的距離總合愈小之結論，而且障礙區的改善亦可節省大量之計算次數，如圖 2.15 之中障礙區與圖 2.14 及圖 2.13 的障礙區相同，在 784 網格點中有 519 個可通行點，當合併若干障礙區後將減少其可通行點之數量，但所得到的路徑與距離總值不改變如表 2.3 所示。上述三種路徑之比較如圖

2.20，綠色部分為 100 網格點所建構之路徑，黑色部分為 361 網格點所建構之路徑，而紅色部分為 784 網格點所建構之路徑，如格點再細分則將收斂於紅色之路徑上。

表 2.3 從起點至終點的路徑軌跡

圖示	網格點	可通行點	路徑軌跡	總距離
圖 2.13	100	50	1→10→35→49→50	13.7287
圖 2.14	361	225	1→56→112→180→211→225	12.8707
圖 2.15	784	519	1→55→175→271→383→478→519	12.1773
圖 2.16	784	374	1→55→172→233→299→350→374	12.1773
圖 2.19	784	280	1→30→113→146→218→256→280	12.1773

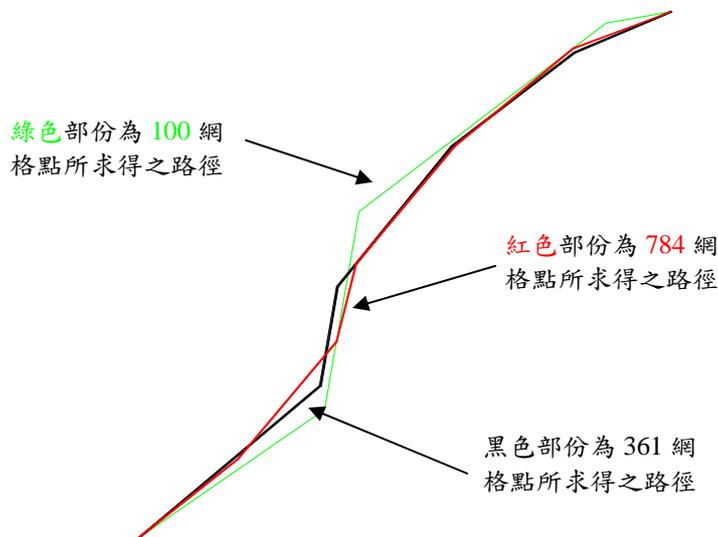


圖 2.20 三種(100、361、784)網格點由起點至終點之路徑比較

範例二、在 3D 自由輪廓多面體上參酌能量損耗的最短路徑規劃：

如圖 2.11 可通行軌跡，利用迪吉斯托理論來計算最短路徑，在圖 2.11 中有五個山頂網格點、20 個山腰網格點，以及 75 個山腳網格點來搜尋其最短路徑軌跡，且其起始點與終點是可以任意指定，假如任意選擇一個起始點之編號為 15，其位置在圖上之山腳平面上，任意選擇一個終止點其編號為 64，其位置在圖上之山頂點處，以圖 2.11 之可通行路徑，再透過迪吉斯托理論可求得該兩點間之最短路徑，用等高線來表示該最短路徑即如圖 2.21 所示，又以幾何模型之自由曲面來表示該最短路徑即如圖 2.22 所示。

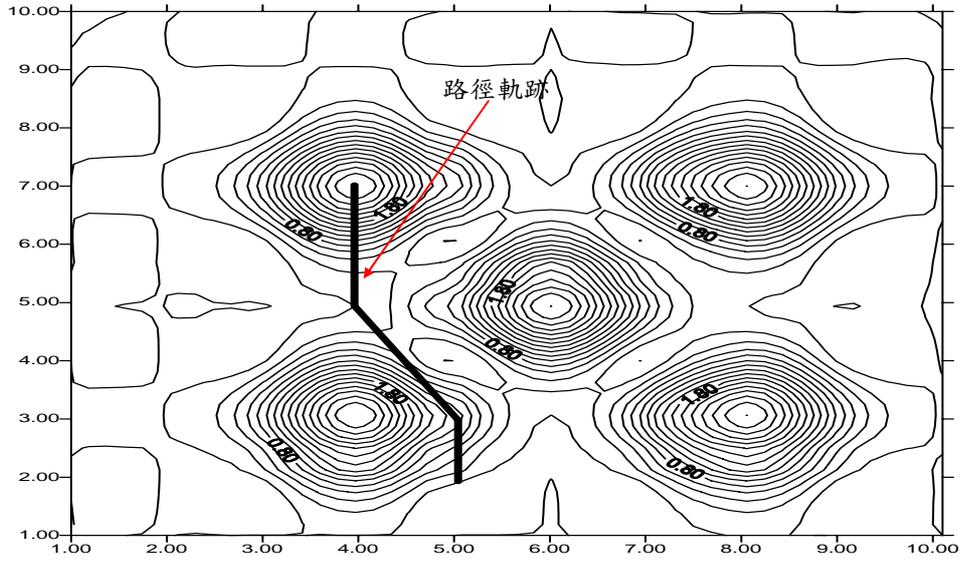


圖 2.21 在等高線上之最短路徑軌跡

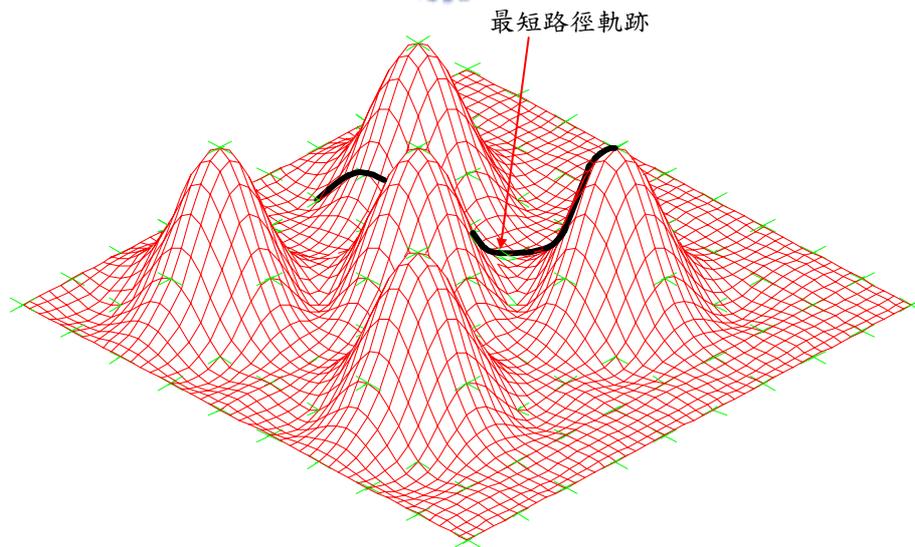


圖 2.22 在自由輪廓多面體上之最短路徑軌跡

2.4 討論

此結果運用在無人搬運車之路徑規劃、機械手臂之抓取與放置及山洞隧道之開挖問題等非常有參考價值，先前的學者在路徑規劃方面大多數的研究成果是在二維平面模式，我們將其運用擴展至三維自由輪廓多面體，將網格規劃法應用於避碰軌跡之路徑規劃，並以迪吉斯托法計算並獲得被最短路徑通過之所有邊界轉折點。

網格設置大小直接影響路徑之精確度，寬廣之網格設置所得到的路徑遠較細密網格設置之路徑大，不過當網格之密度達一定程度後所得到的結果就會產生收斂的作用，但網格設置越密則其計算量就越大，透過起點與目標點之向量估算，距離該兩點間之連線，方向較遠或經由簡單判斷不可能被通過之網格點可先刪除，即可節省部份之計算次數。

