

# 山徑里程 X 演算法— 以白姑大山及能高安東軍為實例

施冠州 \*

## 摘 要

山友們都在山上看過各試各樣的里程路標，但究竟里程是如何標定的？登山界有個江湖傳說是：拿著皮尺去一段一段量出來的。

過去使用的方法我們或許無從考證也無須考證，重要是展望未來。從規畫及執行中華民國山岳協會白姑大山的編碼路標架設，到接下來的能高安東軍編碼路標規畫，我們採用了 GPS 航跡技術、國家地形模型(DTM)開放資料、無數的三角函式及數學公式來進行里程標定。最終的目的就是要讓里程標定不再流於江湖傳說，而是可驗證、可重覆、可技術轉移的科學。

在本實務經驗分享，我們將說明：

- 如何由 GPS 航跡自動化產生每百公尺里程。
- 如何將里程賦予至重要地標。
- 如何將高程列入里程的考量，而非平面投影距離。

最後將會介紹因白姑大山編碼路標架設而誕生的航跡產生器，並在能高安東軍編碼路標的規劃過程中進一步改善並強化。台灣還有無數的山徑缺乏完善的路標，航跡產生器的存在即是將白姑大山及能高安東軍的實務經驗化為人人都能使用的網站服務，而非難以親近的工具。

## 關鍵字

GPS、軌跡探勘、自然步道、編碼路標、距離碼

---

\* 沛星互動科技研發團隊 主管

# 山徑里程 X 演算法— 以白姑大山及能高安東軍為實例

施冠州

## 登山里程的江湖傳說

登山界有很多江湖傳說常常是山友茶餘飯後的閒聊話題，其中之一就是：步道里程到底怎麼定的？為什麼有的走了好久都到不了，有的卻一下子就到了？

山友 A：「是用皮尺吧？」

山友 B：「是用 GPS 吧？」

我：「猜老半天不如上網寫信到營建署署長信箱<sup>1</sup>問！？」

山友 C：「噢？國家公園不是林務局管的嗎？」

其中，最令我自己印象深刻的莫過於前往南湖大山的路上，審馬陣草原的那幾支里程碑。當我從舊雲稜鞍部開始一路陡上審馬陣山，接下來就要爬上南湖北山了，但眼前的幾支里程碑卻讓人覺得是否是高山反應發作...

「怎麼里程碑之間會這麼靠近!？」



圖 1：審馬陣草原里程碑實景



圖 2：使用 OruxMaps 測量 17.4~17.5K 距離

1 “營建署署長信箱”，內政部營建署

[https://service.cpami.gov.tw/index.php?option=com\\_mailhead&view=mailhead&Itemid=2](https://service.cpami.gov.tw/index.php?option=com_mailhead&view=mailhead&Itemid=2)

拿起手機用 OruxMaps 在魯地圖上實測一下審馬陣草原上的 17.4K 到 17.5K，水平距離竟然只有 50 公尺。為免誤會我們也納入兩個里程柱的高程差異，使用直角三角形的斜邊長來計算，也就是所謂的**畢氏定理(Pythagorean Theorem)**<sup>2</sup>：

$$a^2 + b^2 = c^2$$

$$c = \sqrt{a^2 + b^2}$$

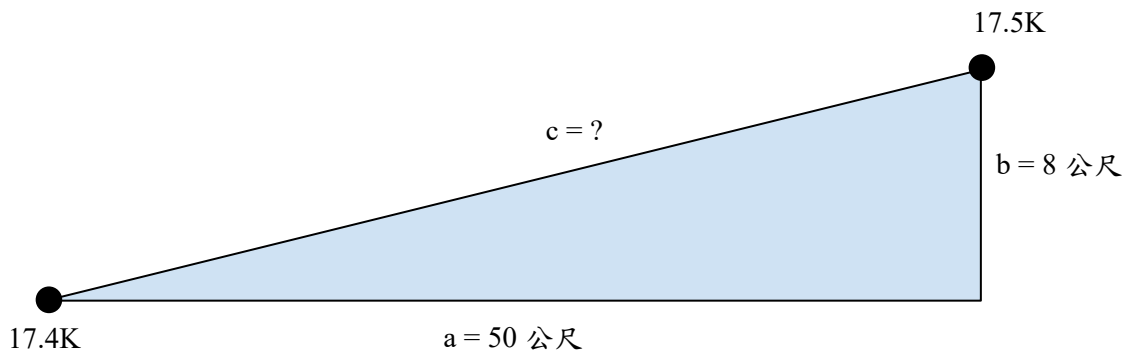


圖 3：使用畢氏定理計算地表(3D)距離(Terrain Distance)

結果是  $\sqrt{50^2 + 8^2} = 50.66$ 公尺，雖然有多了一點點，可是還是跟 100 公尺有很大的差距。

所以究竟這些里程柱為何這麼近？或許只能找當時的施作單位詢問，又或許永遠也不會有答案了。但這些原本都與我無關，因為太多太多的疑點讓我自己登山時總把這類的里程當作次要參考，直到有一天這些事都變得跟我有關。

## 科學化的登山里程計算

在因緣際會之下，有幸受中華民國山岳協會之邀參與了白姑大山反光路標的規劃及施作，並且也很榮幸能參加桃園市山岳協會的反光色差路標<sup>3</sup>訓練課程<sup>4</sup>，學習其科學化的登山里程計算方式。

桃園市山岳協會所推行的方式是以 GPS 航跡為基礎，並在施作前以適當的 GIS 軟體標示出每 100 公尺的編碼路標位置。而施作人員只要載入預先規劃好的 GPS 航跡，即可於現場按座標施作，確保山友對於每 100 公尺一個的編碼路標的感受相對穩定，不再是「江湖傳說」。

2 “畢氏定理”，Wikipedia.

<https://zh.wikipedia.org/zh-tw/%E5%8B%BE%E8%82%A1%E5%AE%9A%E7%90%86>

3 “由南一段及南三段編碼反光色差路標成效研提登山設施改進之建議”，張國雄，全國登山研討會 2020.

<http://taiwanmt.nchu.edu.tw/download/C2-2%E5%BC%B5%E5%9C%8B%E9%9B%84.pdf>

4 “2021 年體育志工特殊訓練—路標志工與種子教練設計與實作課程”，桃園市山岳協會粉絲團，Facebook.

<https://www.facebook.com/groups/1437202193160798/permalink/2925757570971912>

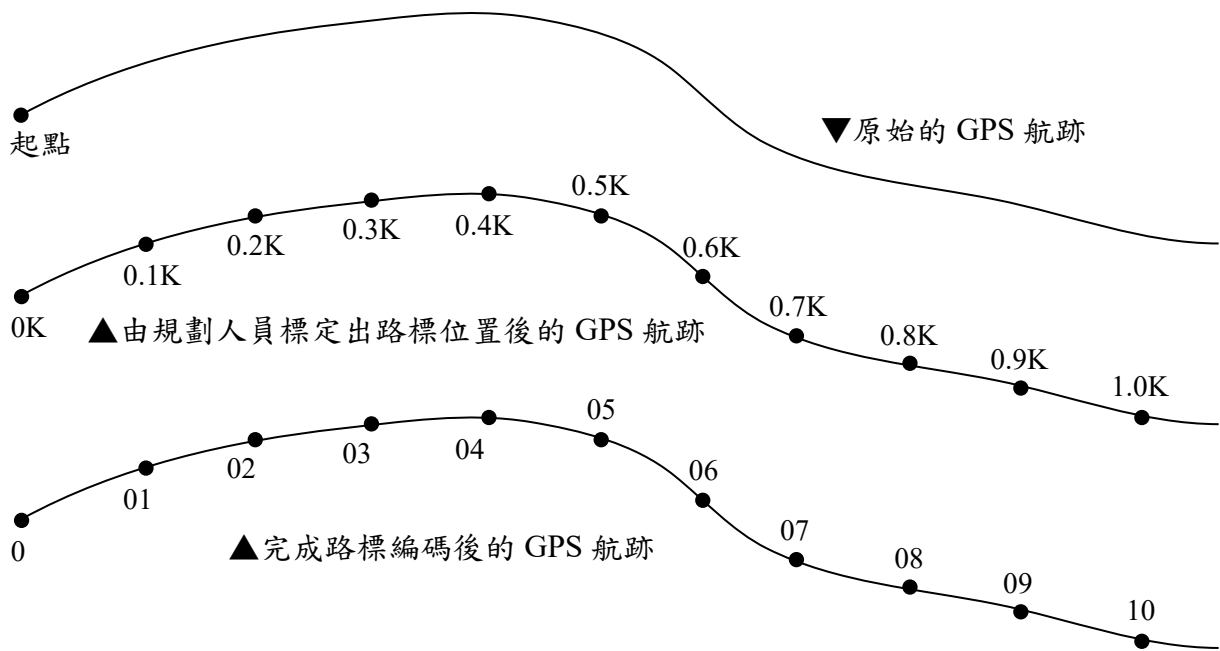


圖 4：使用 GPS 航跡於施作前進行路標位置規劃

這聽起來很不錯對吧？

唯一的問題是，如果每 100 公尺的編碼路標的定位是以人工進行，對於事前的規劃會是相當大的負擔，特別是長程縱走的路線。如果路線有所調整或改動的時候，整個過程或者需要重新來過。不論如何當然還是比拿著皮尺到現場重新量過還好，但如果是可以用軟體解決的為何要用人工來處理呢？而這時候就是演算法能派上用場的時候了。

## GPS 航跡的距離計算

相信很多人都有使用過 GPS 航跡的經驗，雖然它們看起來像是曲線，但實際上是由許多小小的直線線段組合而成。

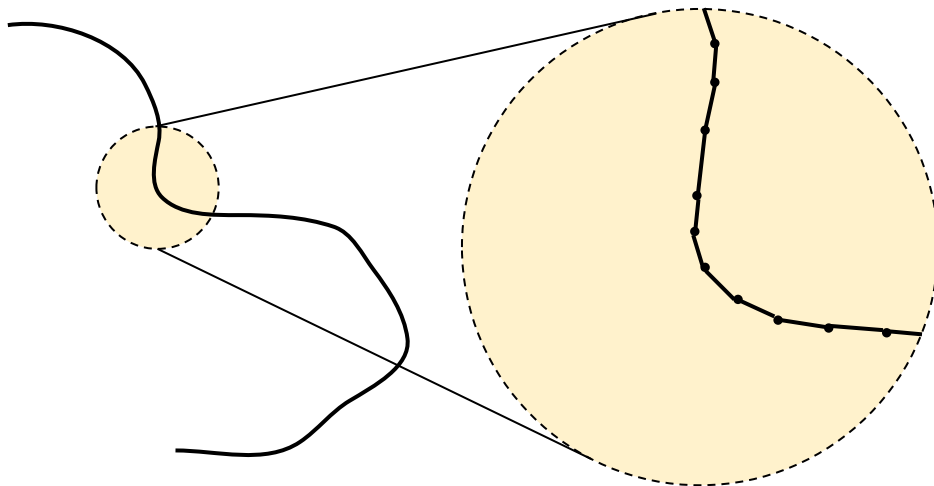


圖 5：放大後的 GPS 航跡

因此整條 GPS 航跡的長度即是由每個線段的長度累加而來，那麼每個線段的長度是怎麼計算的呢？

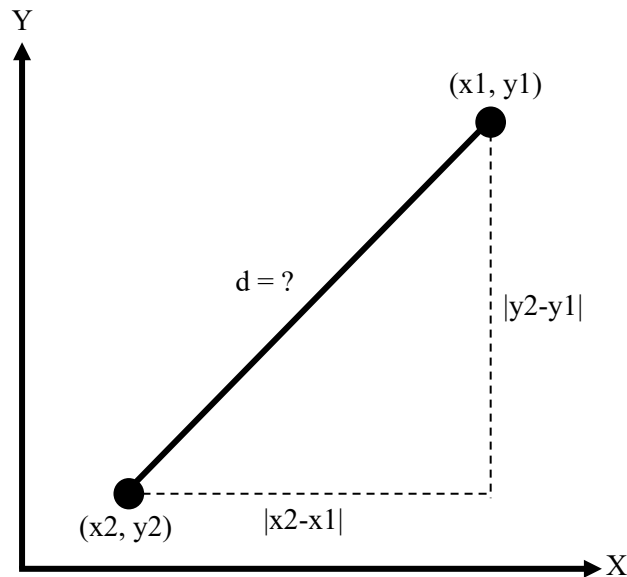


圖 6：使用畢氏定理計算線段長度

聰明的各位或許都想到前面的畢氏定理(Pythagorean Theorem)，因此可算出

$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ ，或稱**歐幾里得距離(Euclidean distance)**<sup>5</sup>。這個答案對也不對，

如果  $(x_1, y_1)$ ,  $(x_2, y_2)$  都是 TWD67/97 二度分帶(TM2)座標值，那我們就可以使用畢氏定理來計算線段長度。但畢氏定理是以平面為基礎的公式，實際上常見的 GPS 航跡都是以經緯度來記錄，而經緯度是以球面為基礎的。

所以在球面上要如何計算兩點間的距離？事實上在西元 1805 年為了大航海時代就已經有解法了，也就是**半正矢公式(Haversine formula)**<sup>6</sup>。

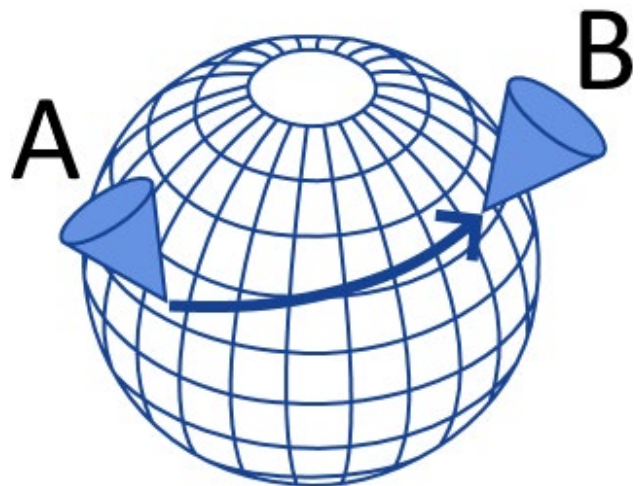


圖 7：球面上的距離（資料來源：Baeldung <https://www.baeldung.com/cs/haversine-formula>）

5 “歐幾里得距離”，Wikipedia.

<https://zh.wikipedia.org/wiki/%E6%AC%A7%E5%87%A0%E9%87%8C%E5%BE%97%E8%B7%9D%E7%A6%BB>

6 “半正矢公式”，Wikipedia.

<https://zh.wikipedia.org/zh-tw/%E5%8D%8A%E6%AD%A3%E7%9F%A2%E5%85%AC%E5%BC%8F>

$$h \sin(\theta) = \sin^2\left(\frac{\theta}{2}\right)$$

$$d = 2r \cdot \arcsin(\sqrt{h \sin(\text{lat}_1 - \text{lat}_2) + \cos(\text{lat}_1) \cdot \cos(\text{lat}_2) \cdot h \sin(\text{lng}_2 - \text{lng}_1)})$$

其中的 $r$ 為地球的平均半徑，我們使用的 WGS84 平均地球半徑 6,371,009 公尺<sup>7</sup>。

看到這麼多三角函數相信大家應該都不想思考了，大概也沒有人想要推導或證明？還好這些都交給電腦來計算就好。簡單來說，半正矢公式所計算出來距離的即為所謂的大圓距離 (Great-circle distance)<sup>8</sup>。以比較生活化的例子來說就像飛機的航線一樣，其航線不會是直線而會是貼著地表的弧線。

另外要特別說明的是半正矢公式是以完美球面為基礎，但眾所皆知地球稍微是一個橢球，只是地球的扁平程度並不影響半正矢公式在地球上的實用性。

## 地表(3D)距離 vs 平面(2D)距離

「可是我爬山時上上下下的，這些不算在距離裡嗎？」

的確，這也是長久以來山友或越野跑者心中過不去的坎。因此某些運動手錶或運動管理 App 裡會提供特別的功能稱作 3D 距離，主要用途是為了修補你的玻璃心讓你覺得自己跑蠻遠的符合我們在立體的山域中行進的感受。

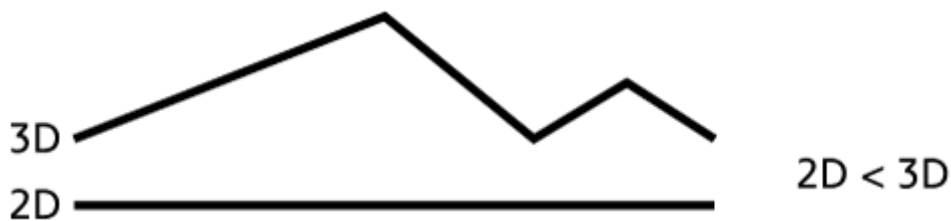


圖 8：3D vs 2D 距離<sup>9</sup>

這聽起來很不錯對吧<sup>10</sup>？我們只要使用畢氏定理算出直角三角型斜邊長度就好了啊！

7 “WGS84”, Wikipedia.

[https://en.wikipedia.org/wiki/World\\_Geodetic\\_System#WGS84](https://en.wikipedia.org/wiki/World_Geodetic_System#WGS84)

8 “大圓距離”, Wikipedia.

<https://zh.wikipedia.org/wiki/%E5%A4%A7%E5%9C%86%E8%B7%9D%E7%A6%BB>

9 “3D 距離”, Suunto.

[https://www.suunto.com/Support/Product-support/suunto\\_ambit3\\_peak/suunto\\_ambit3\\_peak\\_zh-hant/3/3d-/](https://www.suunto.com/Support/Product-support/suunto_ambit3_peak/suunto_ambit3_peak_zh-hant/3/3d-/)

10 “這聽起來很不錯對吧”, 網路迷因.

<https://www.dcard.tw/f/meme/p/239254543>

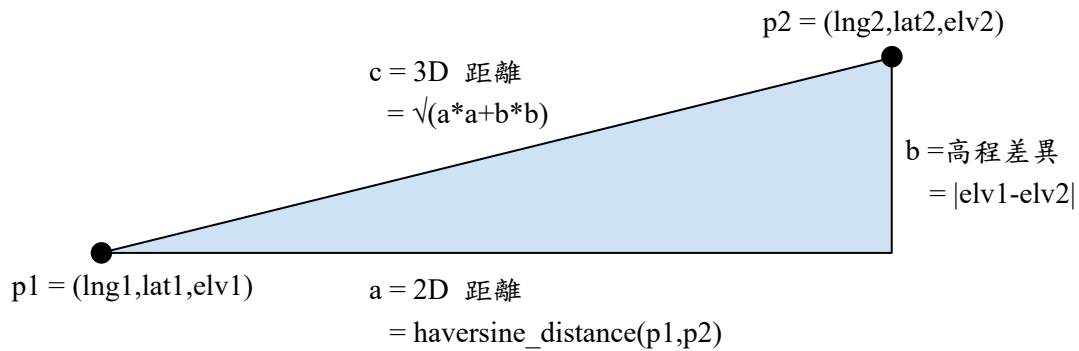


圖 9：使用 GPS 航跡中的經緯度及標高計算 3D 距離

然而 GPS 定位時的垂直誤差一般都大於水平誤差，特別是在山區，但如果要仔細解釋原因大概又要寫另一篇讓人看到睡著的文章。以結果來說，大多數的運動管理 App<sup>11 12</sup>都會使用 GPS 航跡裡的經緯度來查詢海拔(或稱標高)，而不會直接採用 GPS 航跡中測得的海拔。

那要怎麼查詢？我們使用的是所謂的數位地形模型 (Digital Terrain Model / DTM)<sup>13</sup>，另外也有所謂的數位地表模型 (Digital Surface Model / DSM) 但就不在本文的範圍內了，簡單的說法是 DSM 包含植被高度但 DTM 則不包含。目前可以無償取得的幾個 DTM 裡，其中有美國太空總署的太空梭雷達地形測量任務 (Shuttle Radar Topography Mission / SRTM)，另外也有內政部釋出的 OpenData 台澎金馬 20 公尺解析度 DTM。

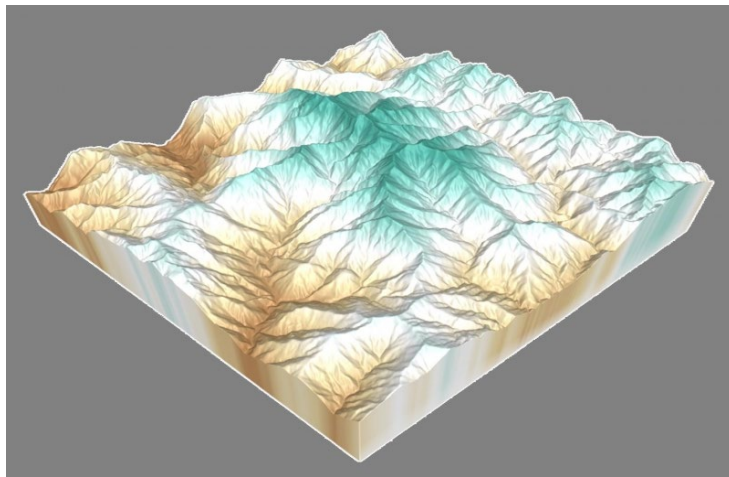


圖 10：全台 20 公尺網格地形數值檔

(來源：健行筆記/內政部 <https://hiking.biji.co/index.php?q=news&act=info&id=8028>)

11 “Elevation on Strava FAQs”, Strava.

<https://support.strava.com/hc/en-us/articles/115001294564-Elevation-on-Strava-FAQs>

12 “How does Runkeeper calculate elevation and climb?”, Runkeeper.

<https://support.runkeeper.com/hc/en-us/articles/201109736-How-does-Runkeeper-calculate-elevation-and-climb->

13 “Digital elevation model”, Wikipedia.

[https://en.wikipedia.org/wiki/Digital\\_elevation\\_model](https://en.wikipedia.org/wiki/Digital_elevation_model)

NASA 的 SRTM 涵蓋全球，但最佳解析度是經緯度裡度分秒裡的秒，在台灣地區來說解析度則是 30 公尺左右。而內政部的 DTM<sup>14</sup>雖然只有台灣，但據其說明原始資料解析度是 5 公尺，為了國安理由打馬賽克疏化成 20 公尺解析度<sup>15</sup>。20 公尺當然比 30 公尺好，因此我們也採用了內政部的 OpenData 來計算 3D 距離。然而還是有個小小的問題是，內政部的 DTM 採用的是 TWD97/TM2 的座標系統，因此在查詢前需要先將經緯度轉換為 TWD97/TM2<sup>16</sup>的平面座標，而這個轉換方式的細節也超出本文的範圍了。

如果要將高程修正的過程簡單表示的話，首先是將經緯度轉換為平面座標，接下來再使用平面座標來查詢該座標位置的高程(elevation)：

$$(x, y) = TWD97(lng, lat)$$

$$elv = MOIDTM(x, y)$$

有了修正後的高程，我們就可以算出誤差更低的地表(3D)距離。

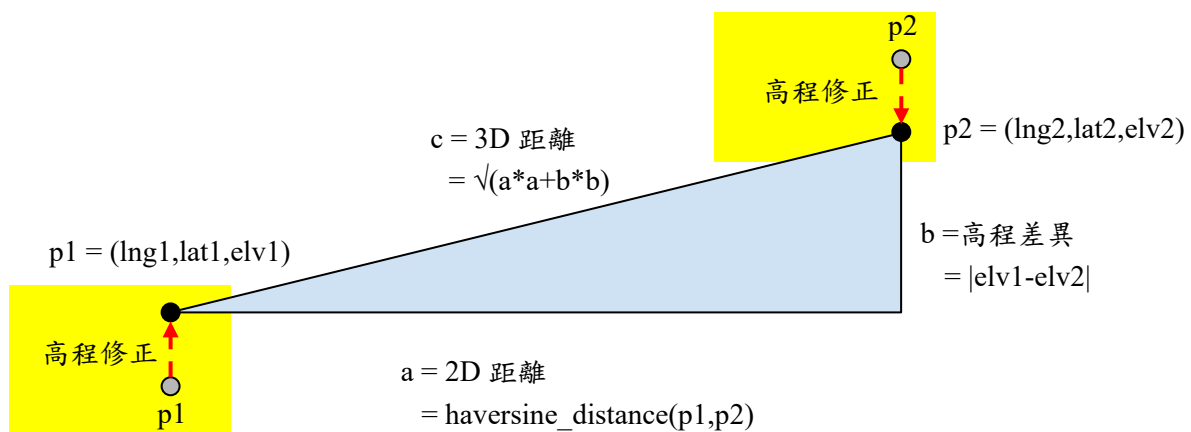


圖 11：使用 GPS 航跡中的經緯度及修正後的標高計算 3D 距離

再次特別說明使用基於球面的大圓距離作為三角形底邊來計算地表(3D)距離是有數學缺陷的。然而在 GPS 航跡線段一般都只有幾公尺或幾十公尺的前提下，此誤差是可被忽略的。

## 里程碑的訂定

至此，已知道如何計算每線段的距離，甚至還考量高程的落差。但里程碑的間距都是固定，可能是每 100 公尺一個或每 500 公尺一個。那 GPS 航跡的線段長度累加起來都會是 100 公尺或 500 公尺的倍數嗎？當然不會是，或者說在使用浮點數的二進位世界裡幾乎不可能。

14 “2020 年版全臺灣及部分離島 20 公尺網格數值地形模型 DTM 資料”，政府資料開放平台。

<https://data.gov.tw/dataset/35430>

15 “網格重製 20 米 28 行詮釋資料格式說明表”，內政部。 [https://data.moi.gov.tw/MoiOD/default/20m\\_readme.doc](https://data.moi.gov.tw/MoiOD/default/20m_readme.doc)

16 “EPSG:3826”，epso.io。 <https://epsg.io/3826>

實際上，GPS 航跡的線段長度累加後都會是像這樣：

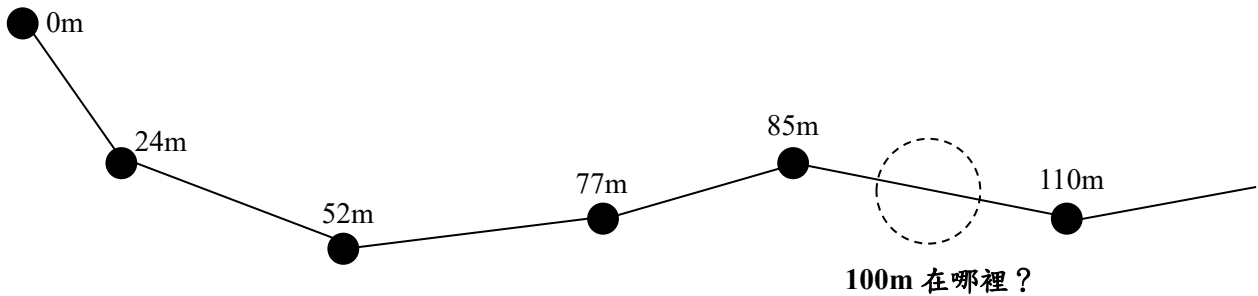


圖 12：典型的 GPS 航跡累加距離不會剛好是里程的倍數

只有 85 公尺航跡點跟 110 公尺航跡點，那 100 公尺的里程碑要設置在哪裡？在沒有演算法的幫助之下，只能依靠人眼的視覺來標定。那要如何使用演算法來處理這個問題？此時我們採用簡單的**線性內插(Linear interpolation)**<sup>17</sup>。雖說簡單，但稍有不慎還是很可能運算錯誤。

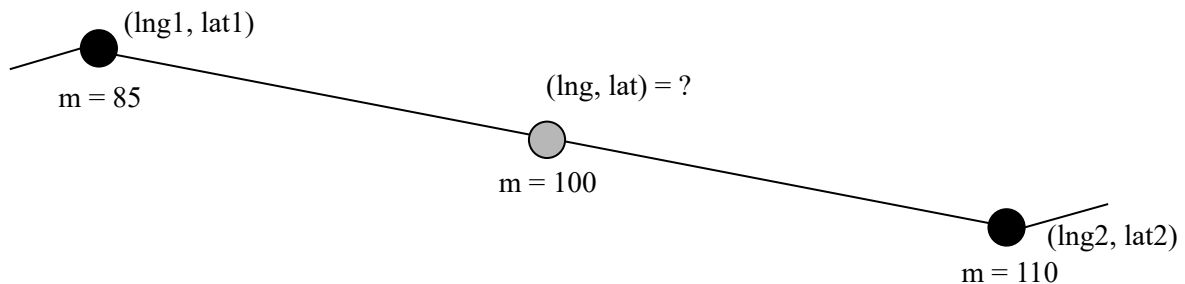


圖 13：使用線性內插取得里程碑座標

以上圖為例，我們可以使用線性內插求得第 100 公尺的里程碑座標：

$$lng = \frac{100 - 85}{110 - 85} \cdot (lng_2 - lng_1)$$

$$lat = \frac{100 - 85}{110 - 85} \cdot (lat_2 - lat_1)$$

同樣需要特別說明的是，使用以平面計算為基礎的內插法來計算以球面為基礎的經緯度座標是有數學缺陷的。然而跟前述章節中同樣的理由，此誤差在我們的應用情境下是可忽略的。

## 地標的里程

終於在運用了各種不同的演算法之後，我們可以透過 GPS 航跡來計算出各里程碑的位置了。但實際上這樣就可以了嗎？事情沒這麼簡單。在實際架設路標時，我們也會在重要的地標上標示出里程。以下圖為例，在 0.6K 跟 0.7K 之間有一個地標。

17 “線性插值”, Wikipedia.

<https://zh.wikipedia.org/zh-tw/%E6%8F%92%E5%80%BC%E7%BA%BF%E6%80%A7%E6%8F%92%E5%80%BC>

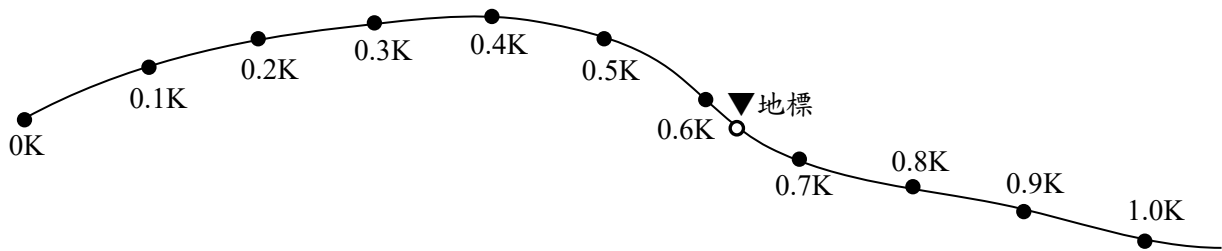


圖 14：GPS 航跡上的地標

要怎麼知道地標的里程呢？聰明的你或許也想到了前面的內插，只要前後內插就可以了對嗎？基本上這樣的想法是正確的，但實際上在 GPS 航跡中地標通常不會剛剛好在路徑上，特別像是營地或水源這樣的地標，常常距離路徑還有一點距離。就算是看起來好像在路徑上的地標，放大再放大之後就可以看出事實上並非如此。

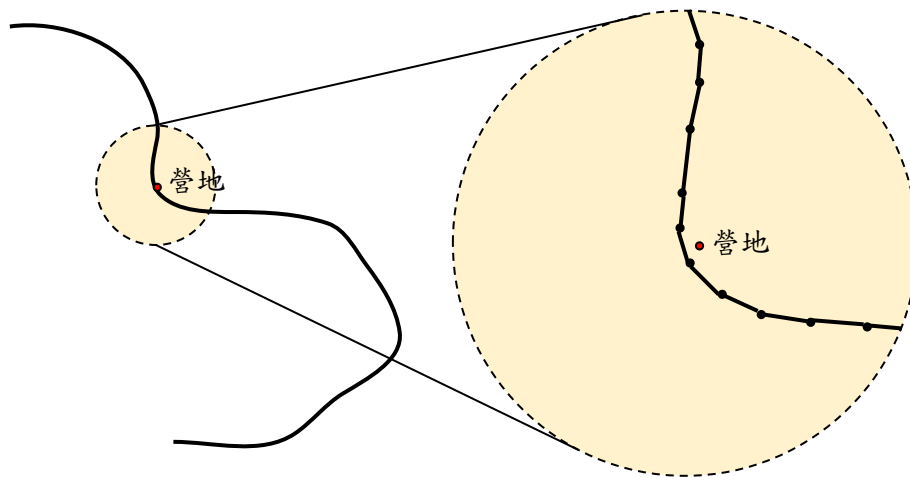


圖 15：GPS 航跡裡的地標並不會剛好在路徑上

為了克服這個問題，我們需要找出路徑上離地標最近的點，再以這個點來計算里程。

對人腦來說很直覺的一件事，用電腦來運算其實有點複雜。要怎麼找出路徑上離地標最近的點？如前面所說，GPS 航跡是由非常多的線段組成，因此我們需要把地標拿來對所有線段計算出最近距離，再取其中距離最小的線段，即可找到地標最接近航跡的對應點；然後每個地標都要重覆對所有的線段進行類似的運算。

我們先討論如何推導地標至線段的最短距離。實際上地標與線段的關係有以下三種情況：

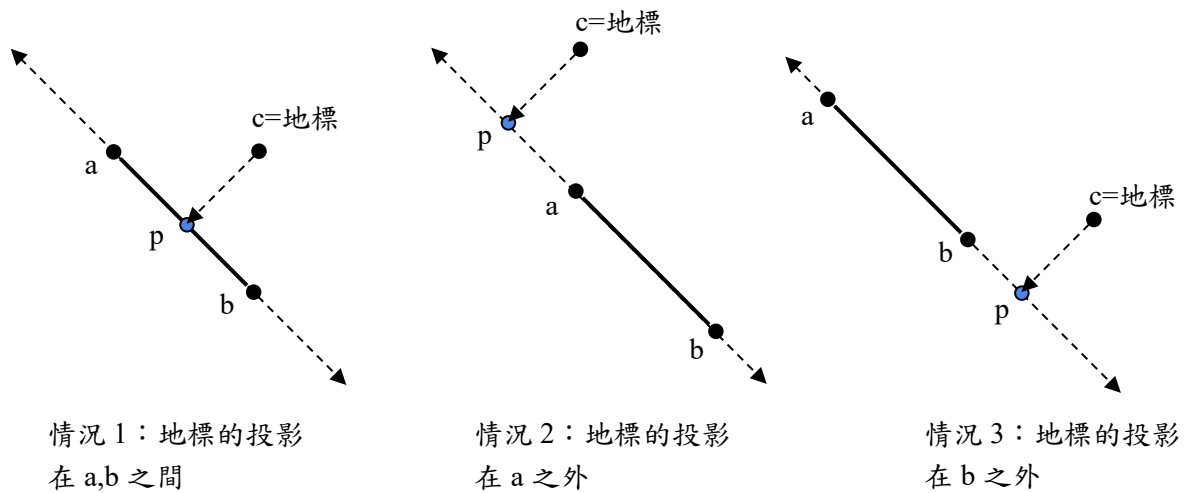


圖 16：地標與線段的三種關係

在這三種情況中，地標與線段的最短距離為：

- 情況 1：地標 c 及投影 p 的距離
- 情況 2：地標 c 及起點 a 的距離
- 情況 3：地標 c 及終點 b 的距離

但要如何判斷這三種情況呢？這時需要使用**向量內積(Dot product)**<sup>18</sup>，當向量內積為負值時，向量夾角大於 90 度。

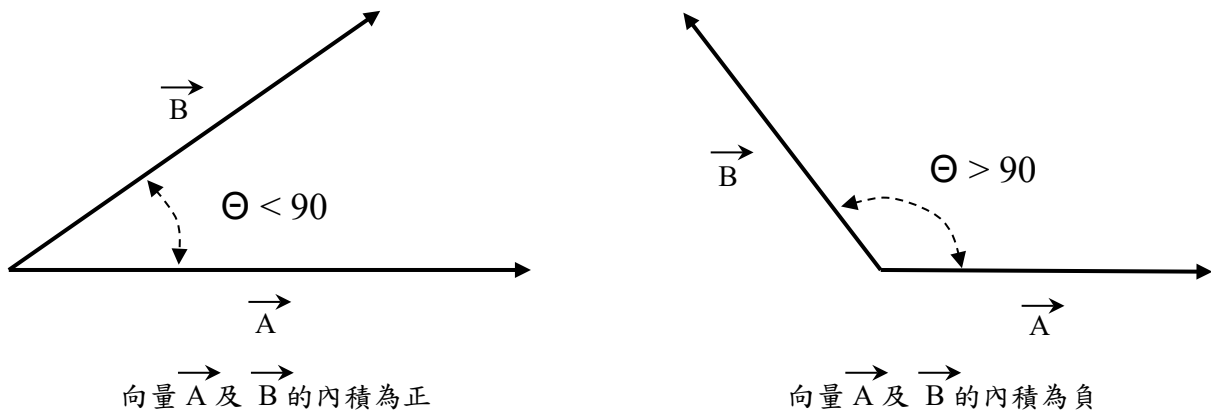


圖 17：使用向量內積判斷夾角範圍

呃？可是我們在高中數學裡學到的內積不是  $|\vec{A}||\vec{B}|\cos(\theta)$  嗎？還不知道  $\theta$  的話要怎麼算內積？實際上以  $\cos(\theta)$  來運算的是內積的**幾何定義 (Geometric definition)**<sup>19</sup>，但此時我們要使

18 “內積”, Wikipedia. <https://zh.wikipedia.org/zh-tw/%E7%82%B9%E7%A7%AF>

19 “內積#幾何定義”, Wikipedia.

<https://zh.wikipedia.org/zh-tw/%E7%82%B9%E7%A7%AF#%E5%87%A0%E4%BD%95%E5%AE%9A%E4%B9%89>

用的是內積的代數定義 (Algebraic definition)<sup>20</sup>。

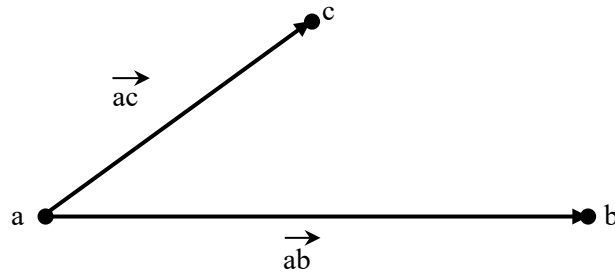


圖 18：使用代數定義計算內積

$$\vec{ab} \cdot \vec{ac} = (x_b - x_a)(x_c - x_a) + (y_b - y_a)(y_c - y_a)$$

竟然只要把每個點的 $(x, y)$ 座標帶入即可？這個代數公式是怎麼來的？很可惜再討論下去大家就要睡成一片就超出本文的範圍了，我們直接進結論：

- 若  $\vec{ab} \cdot \vec{ac} \geq 0$  為情況 2：地標 c 的投影在 a 點之外，最短距離為地標 c 及起點 a 的距離。
- 若  $\vec{ba} \cdot \vec{bc} \leq 0$  為情況 3：地標 c 的投影在 b 點之外，最短距離為地標 c 及終點 b 的距離。
- 否則為情況 1：地標 c 的投影在 a, b 點之間，最短距離為地標 c 及投影 p 的距離。

下一個問題來了，情況 1 時地標 c 跟線段的最近距離是地標 c 與其投影點 p 之間的距離。那要如何計算投影點 p 的位置呢？這時候要使用的是**正射影公式**<sup>21</sup>。

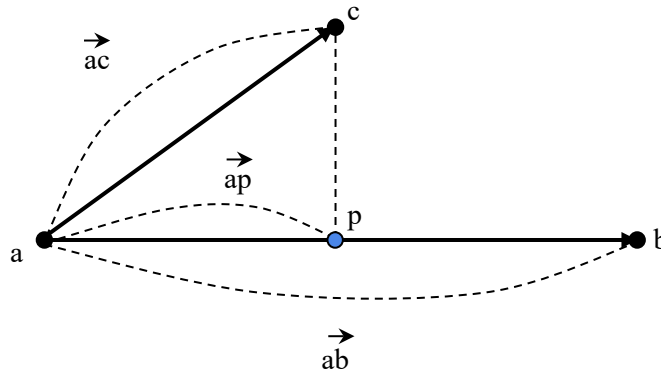


圖 19：計算投影點座標

$$p = a + \vec{ap}$$

$$\vec{ap} = \frac{\vec{ab} \cdot \vec{ac}}{|\vec{ab}|^2} \cdot \vec{ab} = \frac{(x_b - x_a)(x_c - x_a) + (y_b - y_a)(y_c - y_a)}{(x_b - x_a)^2 + (y_b - y_a)^2} \cdot (x_b - x_a, y_b - y_a)$$

20 “內積#代數定義”, Wikipedia.

<https://zh.wikipedia.org/zh-tw/%E7%82%B9%E7%A7%AF%E4%BB%A3%E6%95%B0%E5%AE%9A%E4%B9%89>

21 “Vector projection”, Wikipedia. [https://en.wikipedia.org/wiki/Vector\\_projection](https://en.wikipedia.org/wiki/Vector_projection)

竟然又只要把每個點的 $(x, y)$ 座標帶入即可？這個正射影公式又是怎麼來的！？其實高中數學都教過，但老師也不想面對就快速帶過，大家應該跟我一樣都還給老師了。沒關係，我們終於可以標定出地標的里程了。透過前面的公式，我們可以計算出以下幾種情況的地標里程：

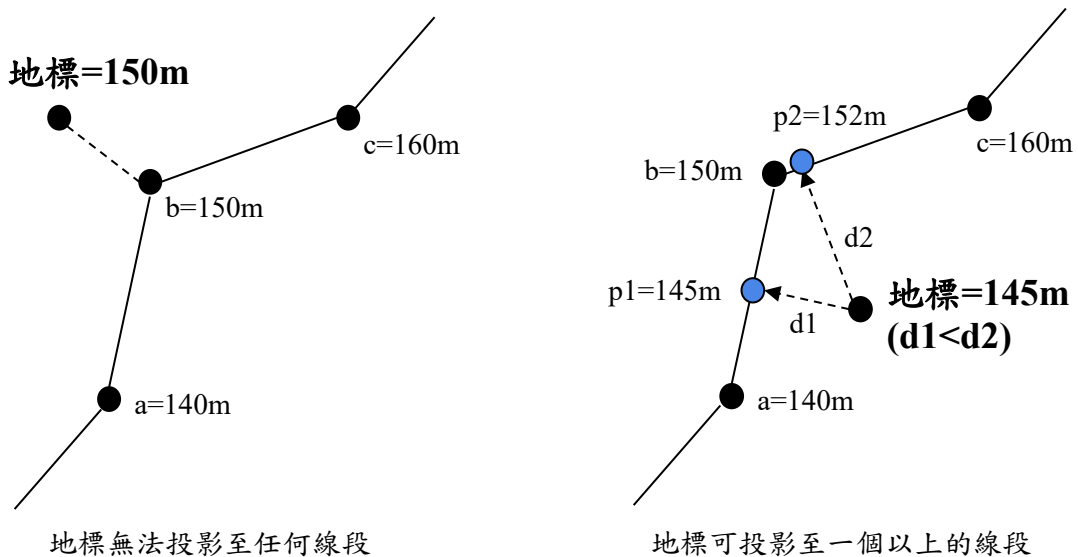


圖 20：正常的地標里程推算

然而現實世界裡總是沒有這麼簡單，實際上在 GPS 航跡中我們還需要對幾種地標進行特別的處理。

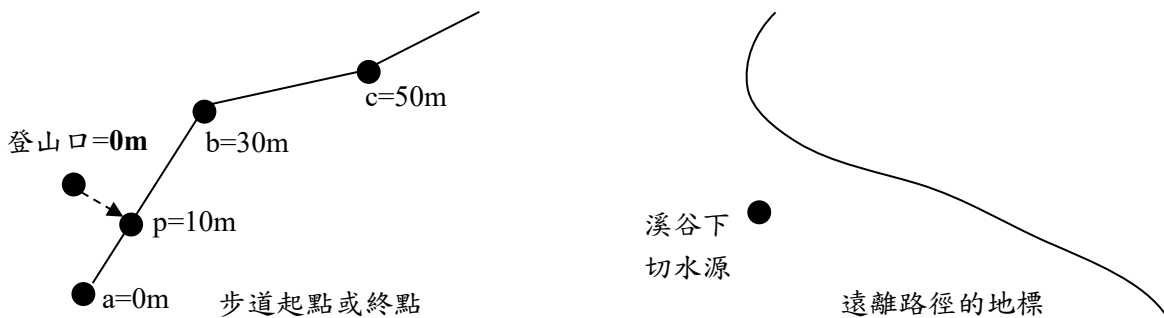


圖 21：需要例外處理的地標

皇天不負苦心人，我們把早就還給老師的高中數學跟大學線性代數硬是從潛意識裡挖出來之後，終於可以標定出每 100 公尺的里程碑，同時也可以計算出地標的里程。但實際上路標的精神是以人為本，一切不是數學說了算。

## 里程遷就地標

施作路標時有一個「里程遷就地標」的原則，這是什麼意思呢？例如「地標 1」如果在 0.56K 的位置，那麼我們會把 0.6K 的路標移到「地標 1」的位置來。

為何要這麼做？

實際上我們並不使用 0.6K 作為路標編號，而是每 100 設置一個並遞增編號，因此 0.6K 的編號會是 6 號。但「地標 1」的 0.56K 要怎麼處理編號？方法一是不處理，給它沒編號的路牌。但這樣的話不好溝通，例如在溝通「地標 1」時只能說「6 號跟 7 號之間的地標 1」。方法二就是採用「里程遷就地標」的原則，因此在溝通「營地 A」時就可以只說「6 號的地標 1」。

問題來了，實際上的「地標 1」跟真正的「6 號」路標差了 40 公尺，因此若只是讓**里程遷就地標**的話，登山者會有一種「怎麼這麼快(或久)就到下個路標？」的感覺。因此比較理想的方式是在「里程遷就地標」後，前前後後的所有路標都跟著微調，也就是**均攤前後里程以遷就地標**。這麼一來登山者的感受差異就不致於太大，而里程路標還算是平均地分散設置的。

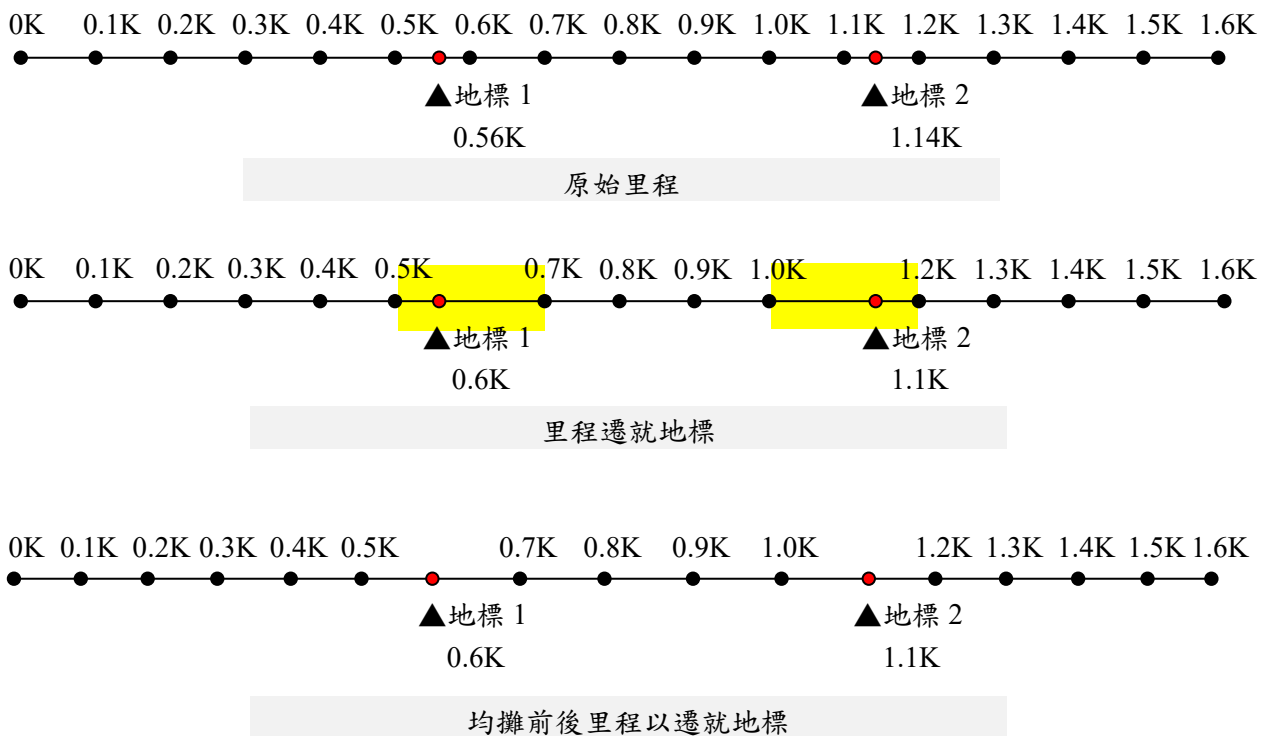


圖 22：原始里程及遷就地標的作法

以白姑大山的實例來說，司晏池營地實際上的里程是在 4.76K 左右，但為了「里程遷就地標」最後印製的里程路標為 48 號。

圖 23：司晏池營地路標<sup>22</sup>打樣

實際上白姑大山的路標是用人工的方式現場處理，也就是施作現場會有熟悉 GPS 及路標規劃的人員進行微調。主因是白姑大山主線只有 8.6 公里，因此基本上不會有什麼大問題。但接下來的能高安東軍路線就不一樣了，從光被八表後的能高越嶺岔路開始，到奧萬大登山口結束約莫 34 公里，還要額外加上支線安東軍山的 1.4 公里。最保險的作法還是事先就規劃好，因為現場可能因為沒有適合固定路標的條件還會臨時變更位置。

但要手動去微調 340 個路標的座標也太耗時，特別是如果原本決定好的路線臨時調整了該怎麼辦呢？這時候，如果可以用演算法來處理就再好不過了。而事實上，能高安東軍的路標施作路線還真的調整了。原先會經過溪谷的路線全數都改成高繞路線，如果原先是以手動方式來處理的話，規劃人員恐怕是要翻桌了？

以演算法來說要怎麼做呢？大致上可分為幾個步驟：

1. 計算出地標里程，除以地標間距後再四捨五入可以得到一個整數。例如「地標 1」的  $0.56K \div 0.1K = 5.6$ ，四捨五入後則為 6。
2. 用地標來將航跡「切割」成許多分段，例如前面的「地標 1」及「地標 2」可將航跡切割成 3 段：

22 “路標產生器”, GitHub.

<https://github.com/outdoorsafetylab/trail-sign-generator>

- a. 起點-地標 1: 長度  $0.56K \div 6$  個里程碑 = 每 93.3m 一個里程碑
  - b. 地標 1-地標 2: 長度  $0.58K \div 5$  個里程碑 = 每 116m 一個里程碑
  - c. 地標 2-終點: 長度  $0.46K \div 5$  個里程碑 = 每 92m 一個里程碑
3. 為每個分段依上述參數產生出里程碑後，最後再重新結合為一個 GPS 航跡。

然而上述演算法看似合理，但實際上會在地標太接近時發生邊緣案例(Edge Case)，造成里程重覆的問題。

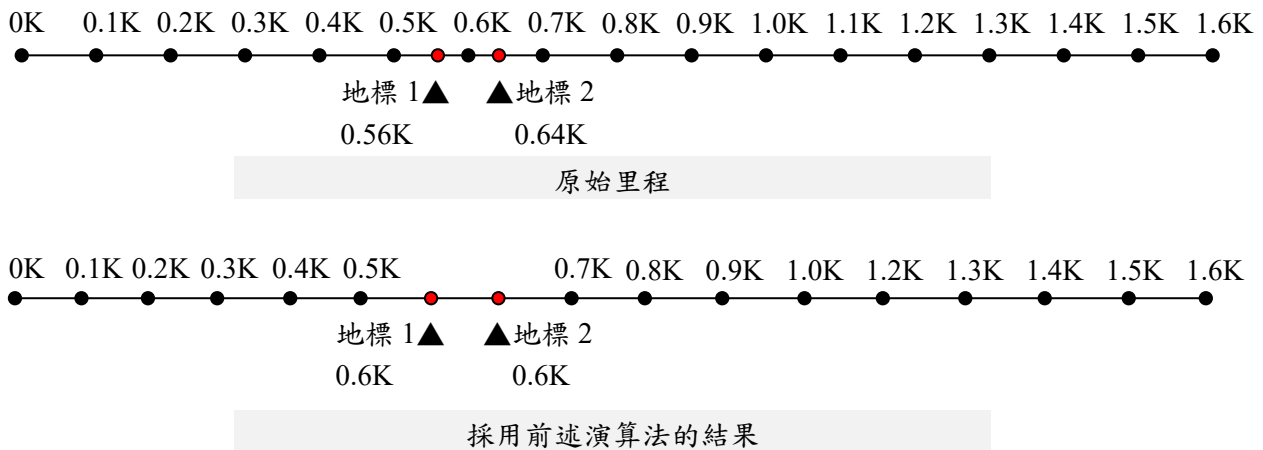


圖 22：前述演算法的邊緣案例(Edge Case) - 里程重覆

因此我們需要調整「里程遷就地標」的演算法來處理所有情境，首先即用地標來將航跡「切割」成許多分段。例如邊緣案例的「地標 1」及「地標 2」同樣可將航跡切割成 3 段：

1. 起點-地標 1: 長度 0.56K
  - a. 里程碑個數:  $0.56K \div 0.1K = 5.6$ , 四捨五入後為 6 個里程碑
  - b. 里程碑間距:  $0.56K \div 6 = 93.3$  公尺
2. 地標 1-地標 2: 長度 0.08K
  - a. 里程碑個數:  $0.08K \div 0.1K = 0.8$ , 四捨五入後為 1 個里程碑
  - b. 里程碑間距:  $0.08K \div 1 = 80$  公尺
3. 地標 2-終點: 長度 0.96K
  - a. 里程碑個數:  $0.96K \div 0.1K = 9.6$ , 四捨五入後為 10 個里程碑
  - b. 里程碑間距:  $0.96K \div 10 = 96$  公尺

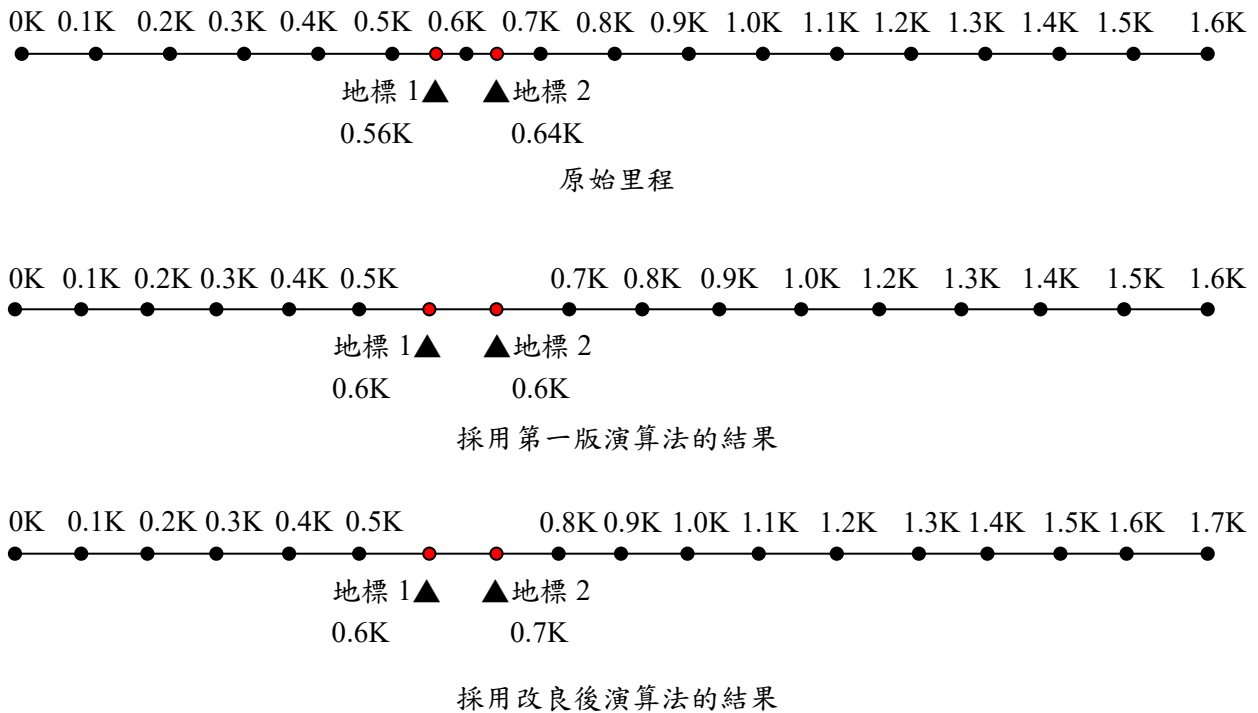


圖 22 邊緣案例(Edge Case)及不同「里程遷就地標」演算法的結果

從上述範例我們可以發現，原本距離 1.6K 的路線原設置的里程碑為 16 個，但在套用「遷就地標」的演算法後會變成 17 個。雖然會有這樣的副作用(Side Effect)，但並沒有第一版演算法的邊緣案例嚴重，也不一定會出現在所有的航跡中。

另外「里程遷就地標」的出發點原已是犧牲準確度來妥協實際山友的需求，因此即使有不完美的部份，仍然是利大於弊的演算法。

## 里程產生器

「難道每條路線的路標都要叫大家自己算!？」

好消息是，以上的所有演算法都已經實作為網頁服務甚至還公開原始碼<sup>23</sup>了。現在只要上里程產生器網站<sup>24</sup>把航跡拖拉到地圖上，點一點選項就可以產生你想要的里程。

舉例來說，能高安東軍的路標為了要延續能高越嶺道的里程要從 157 號(15.7K)開始，又為了要讓山友容易記憶總里程因此要以 500 號結束。其中還使用了 3D 距離納入高程差異，也運用了里程遷就地標的演算法。以手動方式來處理的話，規劃人員恐怕是要一怒之下退出團隊了。

<sup>23</sup> “里程產生器原始碼”，GitHub. <https://github.com/outdoorsafetylab/gpxtoolkit>

<sup>24</sup> “里程產生器網站”，Outdoor Safety Lab. <https://gpxtoolkit.outdoorsafetylab.org/>

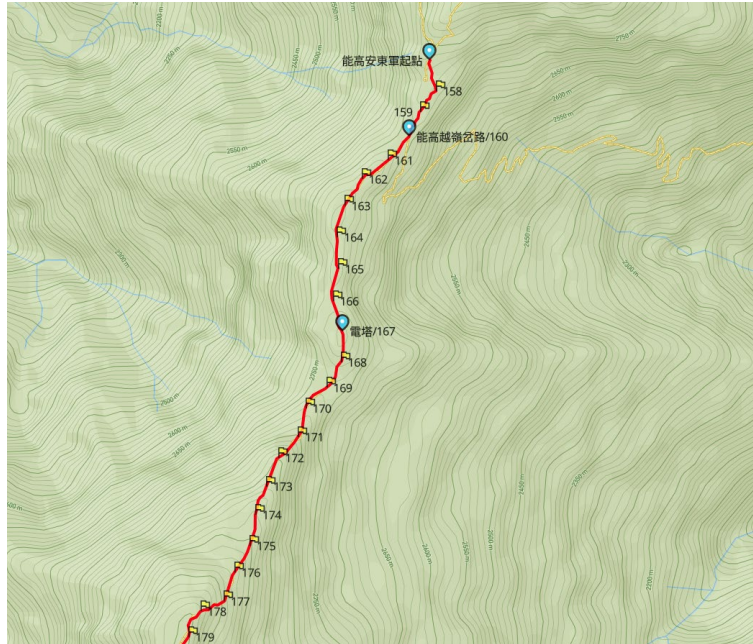


圖 24：能高安東軍的路標延續能高越嶺道將以 157 號開始

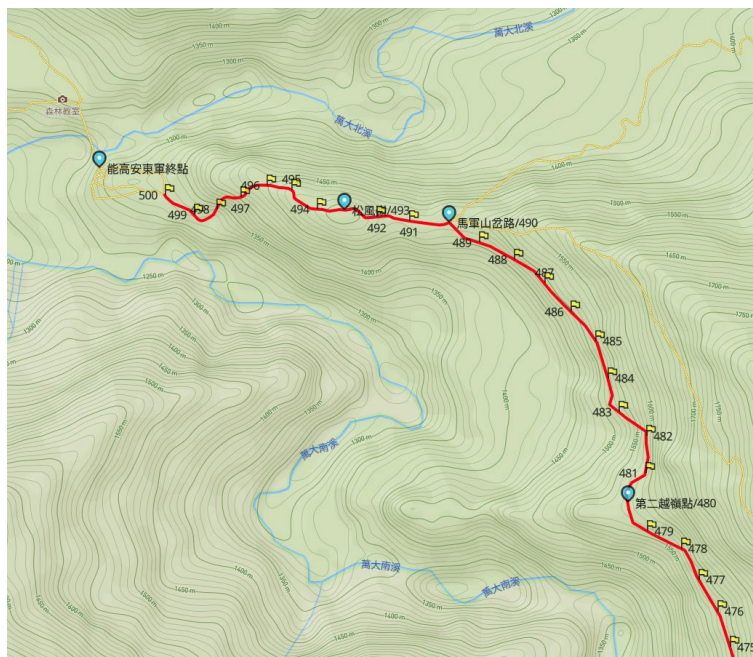


圖 25：能高安東軍的路標為方便山友記憶將以 500 號結束

既然這些莫名其妙的演算法都已經有現成的解決方案了，充滿熱忱的你還在等什麼？快點加入台灣山徑編碼路標的志工行列吧！（有軟體開發背景的山友更歡迎一同加入改善里程產生器！）