

以 Google Map 為基礎的地理圖資數位化網頁程式

Web-based Geographical Image Data Digitizer on Google Map

張逸中

致遠管理學院電機工程學系

ycc@dwu.edu.tw

摘要

本文以 Google Map 開放之免費地圖與程式設計資源，搭配 ASP.NET 的伺服器端程式技術，製作出可以在網頁環境下運作的地理圖資數位化工具程式。此網頁系統可以指定網路上存在或使用者自行上傳的地理影像圖資，如衛星影像、航照圖或舊版地圖之掃描影像，並設定其邊界座標之後，將影像疊置於 Google Map 之上。接著使用以 JavaScript 開發的客戶端圖形化操作介面程式，在瀏覽器中進行圖資的數位化工作。

關鍵詞：Google, ASP, 地理, Web, 數位化, JavaScript

Abstract

A web-based Geographical Image Data digitizer program constructed on Google Map is introduced in this paper. Some ASP.NET program in server side is also implemented for the transition of user's data into internet available resources. The users can assign any geophysical related images existing on internet, like satellite images or old maps on internet, or upload their own images to the server, as an overlay on the Google Map. After that, a client side JavaScript program designed inside the web page takes over for map digitizing via standard GUI interfaces, just like many window-based software can do.

Keywords: Google, ASP, Map, Web, Digitizer, JavaScript

1. 前言

對於許多需要使用地理資訊的工作者而言，取得相關圖資，如衛星影像、航照圖或舊版地圖掃描影像，的位置座標是日常工作中極重要的項目之一。如標示衛星影像中颱風中心的位置、海岸線的變遷等等。此類工作通常需要搭配專業之地理資訊軟體，先匯入目標影像，設定邊界座標，再以圖形化操作界面，主要是滑鼠點選，以取得數位化資訊供後續處理使用。

這些工作通常也必須與現有的可靠圖資隨時進行比對，以避免明顯的謬誤，譬如衛星或航照中出現的已知地標必須與可靠的地圖所示相符。因此需要匯入的圖層將不只一層，還必須耗費額外的資源以取得正確的參考圖資。

自 Google Map 等網路地圖開放之後，相當精確的地圖與衛星影像已經可以輕易取得；不僅如此，Google 還提供免費程式設計組件(Application Programming Interface[3]，以下簡稱 API)供使用者自行設計其應用程式。目前多數應用集中於商業性的資訊展示，將產品或服務資訊與地理資訊連結。如房地產廣告、美食地圖或觀光導覽等等。但是對於地理圖資的專業使用者而言，如何建立與 Google Map 系統連結的應用程式，使其直接成為工作介面之一部分，應該是一個普遍的期待。

欲完成上述目標，在技術上首先面對的困難是 Google Map 以網頁架構(Web)為基礎，圖資資料庫與多數程式碼事實上都在遠端網站，運作過程中還牽涉到 HTML、JavaScript 與其他伺服器端程式語言等複雜的環境，這與一般地理資訊軟體使用本機資料，並以單一程式語言建構的方式差異頗大。程式設計人員除了須具備使用者介面的設計能力之外，對於資料庫、網路運作的細節以及多種網路相關的程式語言都必須有足夠的了解與素養。對於多數研究者而言，這是相當高的技術門檻。

其次是所謂 Rich Internet Application(以下簡稱 RIA)[4]的議題，希望讓網頁具備傳統桌面軟體的行為與功能。在網頁架構下，圖型化操作介面(Graphic User Interface, GUI)的運作效能往往遠遠遜於一般視窗軟體，如何使建構於網頁架構的軟體具有接近一般視窗軟體的效能，關鍵在於將使用者介面處理程序與暫存資料須盡量建構於客戶端的網頁之中，降低往返伺服端的網路通訊需求。

以目前的程式設計環境而言，如此施作的主要困難在於必須使用功能限制較多，且編輯除錯機制較不完整的草稿語言，通常是 JavaScript，進行使用者操作介面的程式設計。但功能非常豐富多樣的 Google Map 與其 API 函式庫就是以 JavaScript 為主要語言所建構，其可行性無庸置疑，只是必須忍受較傳統視窗語言簡陋許多的設計與除錯環境。

本研究的主要目標就是開發一個具備 RIA 特性，以 Google Map 資源為基礎的圖資數位化軟體。目前已開發完成的包括進行圖資數位化的點圖網頁，以及可將文字座標資料檔案整批標示於 Google Map 的網頁。後續將持續依據使用者的可能需求進行研發，並預設為免費之網路資源[6]。

2. 系統架構與設計環境

本系統之主要架構包含：(1)以 ASP.NET2.0[1] 技術為主的伺服器端程式；(2)以 Google Map API[3] 為核心建構的客戶端網頁程式。

前者存在之主要目的為提供使用者自有的文字或影像檔上傳服務。因為以網頁為架構之程式無法直接存取客戶端電腦的一般資料，程式中使用的資料必須為具備網路位址(URL)之網路資源，但多數使用者並不具備讓自用資料取得網址的環境。因此本系統以資料上傳的方式，令使用者的資料成為本系統網站的暫時資料，具備網頁引用時必須的網址，隨後成為客戶端網頁顯示資料的一部分。

當然，在過程中也利用伺服器較為方便的資料處理程式設計環境，先行分析資料，如座標檔案的中心與邊界座標、資料量是否過大需要稀釋以免造成網路塞車等等，並將分析結果隱藏於網頁之中，可以減少客戶端處理程式的複雜度。如果使用者可將資料自行上網，也可用直接輸入網址的方式引用資料，無須上傳至本系統伺服器。

客戶端程式為本系統的核心，基本上操作功能與資料處理部份皆以 Google Map 提供之 HTML 搭配 JavaScript 語言的範例為藍本，參考 API 類別程式庫資訊改寫；並進一步將此程式片段插入 ASP 架構的網頁之中，如此即可很方便的使用到 ASP 的伺服器端程式功能與網頁頁面的設計環境。

3. 圖資上傳頁面



圖 1 圖資上傳頁面

本頁面可讓使用者指定已上網的圖資、自行上傳圖檔或完全不使用自定圖資，在 Google Map 預設的地圖或衛星影像上進行擷取座標資訊的工作。如不使用圖資可按畫面上方之『不使用自定影像直接進入』切換到以 Google Map 為基底的下一頁；若指定網路影像則需輸入網址以及影像檔左下

(S,W)及右上方(N,W)邊界點之經緯度座標；欲上傳自有圖片則由瀏覽鍵選檔上傳，一樣的，必須自行輸入邊界座標。完成資料選定後可按下方『使用上列指定影像進入』按鍵切換至工作網頁。

4. 圖資數位化頁面

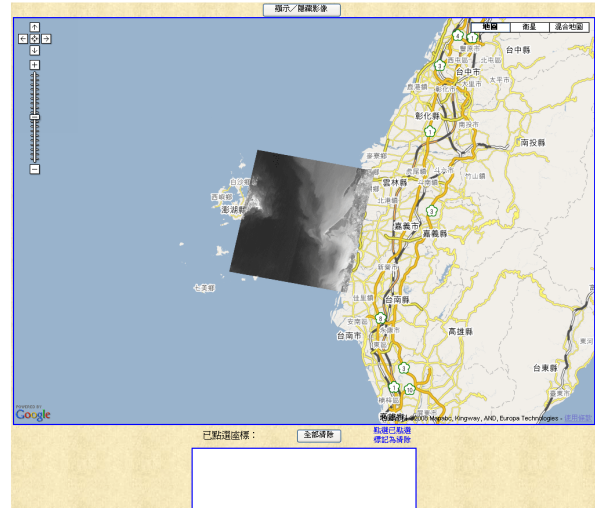


圖 2 數位化工作頁面全覽

如上圖 2 所示，選定之影像(此例為嘉義外海之歷史衛星影像)將依據指定座標浮貼於 Google Map 之預設畫面之上，上方並提供一『顯示/隱藏』切換鈕，可隨時比對此影像與背景地圖之差異。當然 Google 預設之衛星影像資料亦可切換比對，如下圖 3 所示；網路地圖原有之縮放與移動功能也都如常運作，自定貼圖會隨之縮放與移動。



圖 3 自定影像與 Google 衛星影像

須進行擷取座標的點圖作業時僅需放大目標至所需程度，並以滑鼠點選目標即可。點選成功後會出現紅色十字標記，下方文字框並會依序出現該點的經緯度座標，如圖 4。為方便精細的點圖作業，在此將標記圖示由 Google 預設的較大氣球圖示更改成較小的紅十字。欲取消已標記資料時，僅需再按一次已選標記，它會自動消失，下方文字欄中相對應的該點座標資訊也會同步被刪除。



圖 4 目標點選與座標資訊處理

重要的是，這些操作過程中都不會有網頁常見的動作延遲現象，原因是它們完全由客戶端的 JavaScript 程式碼運作，不須與本系統伺服器互動，無關於網路效能。系統伺服器的工作在此只是產生一個將使用者定義的圖片與 Google Map 結合的網頁，點圖作業的程式碼已經附於原始網頁之內(可使用網頁原始碼功能檢視)。這是一個雙贏的結果：客戶端獲得與一般視窗程式類似的高效能圖形介面；而本服務網站並未增加任何網路流量或電腦使用時間，客戶上傳之暫存檔案也可以在工作階段結束後隨即刪除。相當符合 RIA 的設計理念。

當然，略有不便的是網頁程式無法提供直接存檔的功能，點選之座標文字資訊必須以複製方式轉貼於其他的文字編輯軟體再加以儲存。本網頁提供之文字欄在多數瀏覽器中皆可隨資料內容增加而自動放大，點選資料量不會受限於文字框的預設大小，並具備編輯、複製與刪除等功能，與記事本程式相似，相當方便。

5. 座標資料檔案檢視頁面

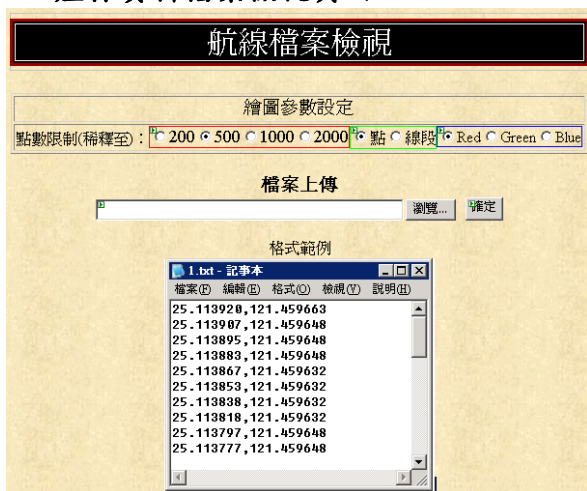


圖 5 座標文字資料上傳頁面

座標擷取的反向應用常常是需要將文字座標資料投影於正確的地圖上進行檢視。譬如野外施測

的測點位置分布是否與計畫相符；或取自歷史衛星影像的海岸線資訊想與目前的地圖進行比對等等。這種作業必須將相當大量的點位資訊以標記的方式顯示於網路地圖上。如前所述，網頁程式無法直接讀取使用者本身電腦的檔案，因此第一步仍是提供資料上傳服務。如圖 5 所示。

在此，除了多出點或連線的顯示方式與標記顏色的選項之外，主要的差別是要求選取最高顯示點數，這可以避免使用者資料太龐大時網頁處理過於耗時。本程式內定處理的方式是稀釋資料至要求的點數以下。譬如 5000 點資料要求顯示 500 點時，資料會每十點取一，大致上可以維持資料分布態勢，但點數較少。

與前述之影像點圖功能不同之處是伺服器程式在此並不將資料檔案上網，而是在伺服器自行開啟檔案進行資料稀釋等等處理，其中也包括尋找資料內的位置極值，最終將客戶端需要的經緯度座標資料寫入網頁之隱藏物件(如 ASP 之 HiddenField)中。客戶端網頁載入之後，網頁內的 JavaScript 程式會讀取這些頁內資料，決定 Google Map 應該顯示的中心點與縮放尺度，當然也包括各個必須顯示的點位會被製作為標記浮貼於地圖上。如圖 6。

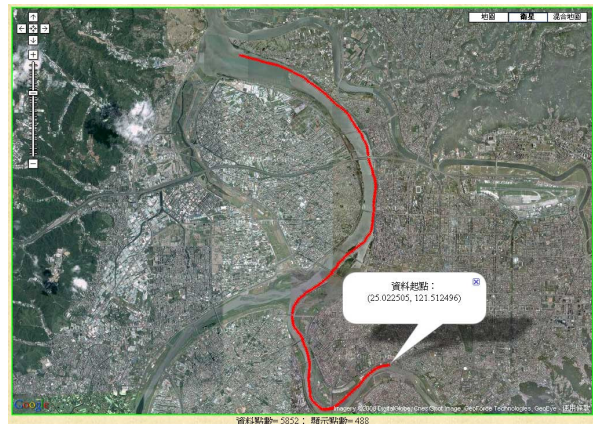


圖 6 淡水河測量工作之測點分布圖

令 Google Map 依據上傳資料自動決定顯示的位置與縮放尺度是一個必要的動作，否則使用者必須自行移動地圖尋找『自己』的資料相當不合理。但在此製作這個功能具有相當的難度，主因是伺服端的 ASP 程式雖可取得相關資料的座標資訊，卻無法與網頁內的 JavaScript 程式取得直接的聯繫，譬如使用共同的變數或陣列等等。因此必須以網頁上的伺服器端控制項為介面，令 ASP 將資料寫入控制項，再經過 JavaScript 可以藉由 DOM 架構讀取網頁控制項內容的功能，間接將伺服端的資訊傳遞到控制 Google Map 的 JavaScript 程式中。本系統內目前的兩個功能網頁都具備此自動調整的功能，如圖 7 所示，即使範圍很小的資料也可以自動以最佳化方式迅速顯示。



圖 7 自動完成之小區域資料最佳化顯示畫面

6. 地理圖資加值展示

本研究使用的技術亦可提供許多地理資訊相關的研究計畫或機構作為網路展示成果的一種方式，較諸傳統的靜態地圖影像將有更高的附加價值。譬如網路上展示的地圖在選擇以『Google Map 檢視』(如圖 8)的選項後即可將該圖疊置於 Google Map 系統中讓使用者自由檢視與比對資料。事實上就是切換到如圖 2 的網頁，如此使用者便可以自由縮放移動影像、與 Google 的背景地圖或衛星影像比對，甚至自行擷取需要的特徵點座標。



圖 8 選擇以原尺寸圖或置於 Google Map 檢視

在此種應用中，圖的邊界座標資訊必須由展示網站預先設定，因為使用者必然的不會知道。這可以使用圖形之外的文字檔或資料庫建立，也可以寫入影像檔案內部適當的欄位，譬如目前數位影像中的 Exif (Exchangeable Image File Format)[2] 資訊欄。本系統目前則直接將邊界座標附加於影像檔案尾端，不會傷及影像結構，也不必考慮影像檔案的多種可能格式，亦是一種可行的方式。

7. 討論與結論

Google Map 的出現是網路資訊發展史上的一大突破，其免費開放的 API 資源更促進了近年如雨後春筍般的各式應用程式開發與應用。另一方面，RIA 的網路程式設計理念亦在近幾年大為風行，包

括 AJAX[5]在內的網頁程式技術使得網頁程式效能逐步擺脫網路效能與 HTML 格式的限制，與傳統的視窗程式並駕齊驅。

在目前已出現的 Google Map 應用程式中，多數為資訊展示之用，直接作為資料處理工具之研發較少。本研究以地理資訊工作者的需求為起點，研究使用 Google Map 的免費圖資與程式工具發展類似地理資訊資料處理軟體之功能，目前已獲致初步的成果[6]，預計將繼續使程式更精緻與多元化，續往 RIA 的設計理念邁進，成為地理資訊專業人員樂於使用之免費網路資源。

參考文獻

- [1] ASP.NET Web Site.
<http://www.microsoft.com/taiwan/msdn/asp.net/default.aspx>
- [2] Exchangeable image file format, Wikipedia.
<http://en.wikipedia.org/wiki/Exif>
- [3] Google Maps API Web Site
<http://code.google.com/apis/maps/documentation/index.html>
- [4] J. Allaire, Macromedia White Paper, 2002.
<http://www.adobe.com/devnet/flash/whitepapers/richclient.pdf>
- [5] J. J. Garrett, Ajax: A New Approach to Web Applications, 2005.
<http://ycc.dwu.edu.tw/S1/AJAX.mht>
- [6] YCC Google 應用程式, 2008.
<http://ycc.dwu.edu.tw/MapApp/>