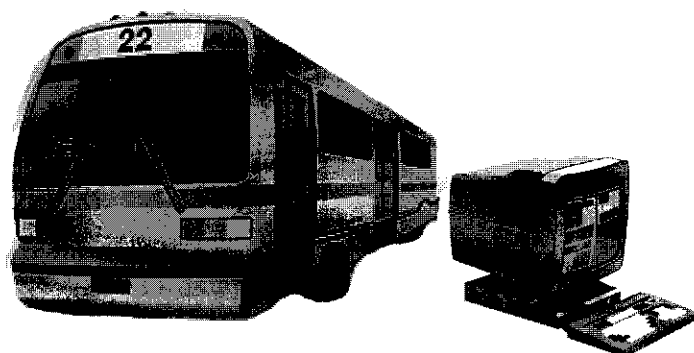


台北都會區公車運輸起迄調查

技術報告



委託單位： 台北市政府交通局
交通部運輸研究所
承辦單位： 台大慶齡工業研究中心
研究單位： 國立台灣大學土木工程學研究所

中華民國八十六年七月

台北都會區公車運輸起迄調查

技術報告

計畫主持人：張學孔 教授

協同主持人：龍天立 教授

王瑞民 博士

研究人員：朱家德

邱奕明

陳志勇

程玉萍

鍾秋弘

黃俐嘉

杜宗芳

林益民

許哲瑋

委託單位：台北市政府交通局
交通部運輸研究所

承辦單位：台大慶齡工業研究中心

研究單位：國立台灣大學土木工程學研究所

中華民國八十六年十月

摘 要

本技術報告為「台北都會區公車運輸起迄調查」研究計畫之部分成果，主要內容係配合研究計畫所構建之台北都會區公車地理資訊系統，針對系統初學者為設計對象所編纂之技術移轉與訓練手冊；使用者需具備基本 PC Windows95 操作觀念。技術報告共包括三部份，分別說明地理資訊系統之操作方式、公車運輸資料庫之建立程序、及公車旅客起迄資料表(O-D Table)之分析結果。

第一部份「地理資訊系統操作」介紹台北市公車地理資訊系統應用軟體 UfosNet 之功能特性與操作方式，包括檔案管理、圖層與資料庫編輯、相關分析功能、矩陣與資料庫之函數運算功能、及大眾運輸路線分析與展示功能等。第二部份「台北市公車地理資訊系統資料庫建立」係說明調查結果資料電腦化之程序，包括公車旅客起迄站資料庫、公車旅客特性資料庫、及家庭訪問活動日記資料庫，除 UfosNet 外，並應用 DELPHI、VISUAL BASIC、與 MS-ACCESS 等程式語言與套裝軟體進行介面整合，以增進調查結果彙整效率。第三部份「地理資訊系統應用範例說明」，為應用 UfosNet 進行數個大眾運輸系統分析範例之操作流程說明。第四部份「台北都會區公車旅次起迄表」，則為公車旅客起迄站資料庫應用 UfosNet 進行分析，所得台北都會區 388 個交通分區之公車旅次起迄矩陣。

本技術報告除可提供大眾運輸研發與實務單位進行調查工作之資料彙整工具，並提供規劃者應用地理資訊系統分析之操作指引。對於「台北都會區公車運輸起迄調查」專案研究之進行方式與旅次特性分析結果，請詳見另冊專案報告。

目 錄

摘要	I
目錄	II

第一部份 地理資訊系統操作

第一章 簡介	1
1-1 地理資訊系統作業概念	1
1-2 地理資訊系統分析環境	3
1-3 技術報告第一部份各章節說明	6
第二章 檔案管理	7
2-1 系統設定(Configure System)	9
2-2 檔案載入與儲存(Load and Save)	10
2-3 列印(Print)	11
第三章 圖層與資料庫編輯	12
3-1 路網編輯工具箱(Toolbox-Network)	12
3-2 分區編輯工具箱(Toolbox-Boundary)	19
3-3 註釋層編輯工具箱(Toolbox-Annotation)	25
3-4 點資料(Point Data)	29
3-5 點陣圖檔(Bitmap)	30
3-6 資料表格(Data Sheet)	30
3-7 定義資料庫(Database Definition)	30
第四章 圖層繪製(PLOT)	32

4-1 圖層結構(All Objects)	32
4-2 物件查詢(Query Objects)	33
4-3 預設圖層顯示格式(Define Default Style)	37
4-4 清除存區物件顯示(Clear Plot Buffer)	43
4-5 圖層控制(Layer Control)	43
4-6 視窗控制功能	44
第五章 相關分析功能	45
5-1 公路最短路徑查詢(Highway Path Query)	45
5-2 大眾運輸路徑查詢(Transit Path Query)	47
5-3 影響圈域分析(Buffer Region Analysis)	48
5-4 主題地圖顯示(Thematic Mapping)	49
5-4 巨集模組編輯器(Macro Editor)	49
第六章 資料庫計算功能	51
6-1 分區、節點、節線、點、F 係數及大眾運輸路網計算功能	51
6-2 計算函數(Calculation Functions)	51
6-3 計算語法	55
6-4 迴歸(Regression)	55
6-5 直方圖(Histogram)	56
6-6 節點進入量(Node Entering)	56
第七章 矩陣	57
7-1 矩陣形式(Matrix Types)	58
7-2 矩陣定義(Defining Matrix)	68
7-3 矩陣編輯(Editing Matrix)	72

7-4 矩陣運算(Calculating Matrix)	73
7-5 矩陣轉出與轉入(Exporting and Importing)	75
7-6 複製矩陣(Copying Matrix)	79
7-7 矩陣位置(Site Matrix)	81
7-8 矩陣統計(Matrix Statistics)	82
7-9 矩陣延伸(Expanding Matrix)	83
7-10 矩陣壓縮(Compressing Matrix)	84
7-11 期望線(Desire Lines)	85
7-12 轉向損失儲存(Save Turn Penalty Matrix)	89
第八章 大眾運輸路線	90
8-1 工具箱(Toolbox-Transit)	90
8-2 大眾運輸路徑(Transit Path)	99
8-3 大眾運輸指派(Transit Assignment)	99

第二部份 台北市公車運輸起迄資料庫建立

第九章 資料庫建立簡介	103
第十章 公車旅客起迄站資料庫	110
10-10-D 表建檔	110
10-2 建立編碼對照表	110
10-3 轉換矩陣資料	122
10-4 各時段旅次量加總	123
10-5 旅次量放大	131
10-6 校估旅次起迄表	132

第十一章 公車旅客特性資料庫	139
11-1 問卷分類	139
11-2 人工讀卡	140
11-3 賦予批號	140
11-4 問卷資料電腦化	142
第十二章 家庭訪問活動日記資料庫	150
12-1 聯繫問卷填寫不全之家戶	150
12-2 人工讀卡	150
12-3 問卷資料電腦化	150

第三部份 地理資訊系統應用範例說明

第十三章 地理資訊系統應用範例說明	165
13-1 道路路網應用範例	165
13-2 大眾運輸路網應用範例	176

第四部份 台北都會區公車旅次起迄表

第一部份

地理資訊系統操作

第一章 簡介

1-1 地理資訊系統作業概念

應用地理資訊系統進行路線規劃作業，分析程序如圖

1-1，其內容包括：

一、訂定作業計畫目標

應用地理資訊系統進行路線規劃設計，其目標在：

- 1.提昇溝通能力
- 2.空間資料處理與展示自動化
- 3.未來研究整合

二、資料庫設計

決定分析範圍、相關分析工具的應用與操作程序之整合、需應用哪些圖層，各圖層特徵(點、線、面)為何、所需屬性為何、如何編碼、如何組織與定義各種屬性。

三、圖層與地理資料庫建立

資料庫建立為地理資訊作業最耗時的步驟，資料庫的完整性與精確度攸關分析結果的品質，若可運用現有社經資料與土地使用之資料庫與道路、交通分區數位圖檔，將可大幅減少此步驟所需的時間；因而建立一都會區大眾運輸地理資料庫，以增加規劃資料之資源共享性，提昇分析效率，為運輸規劃資訊化的基礎。

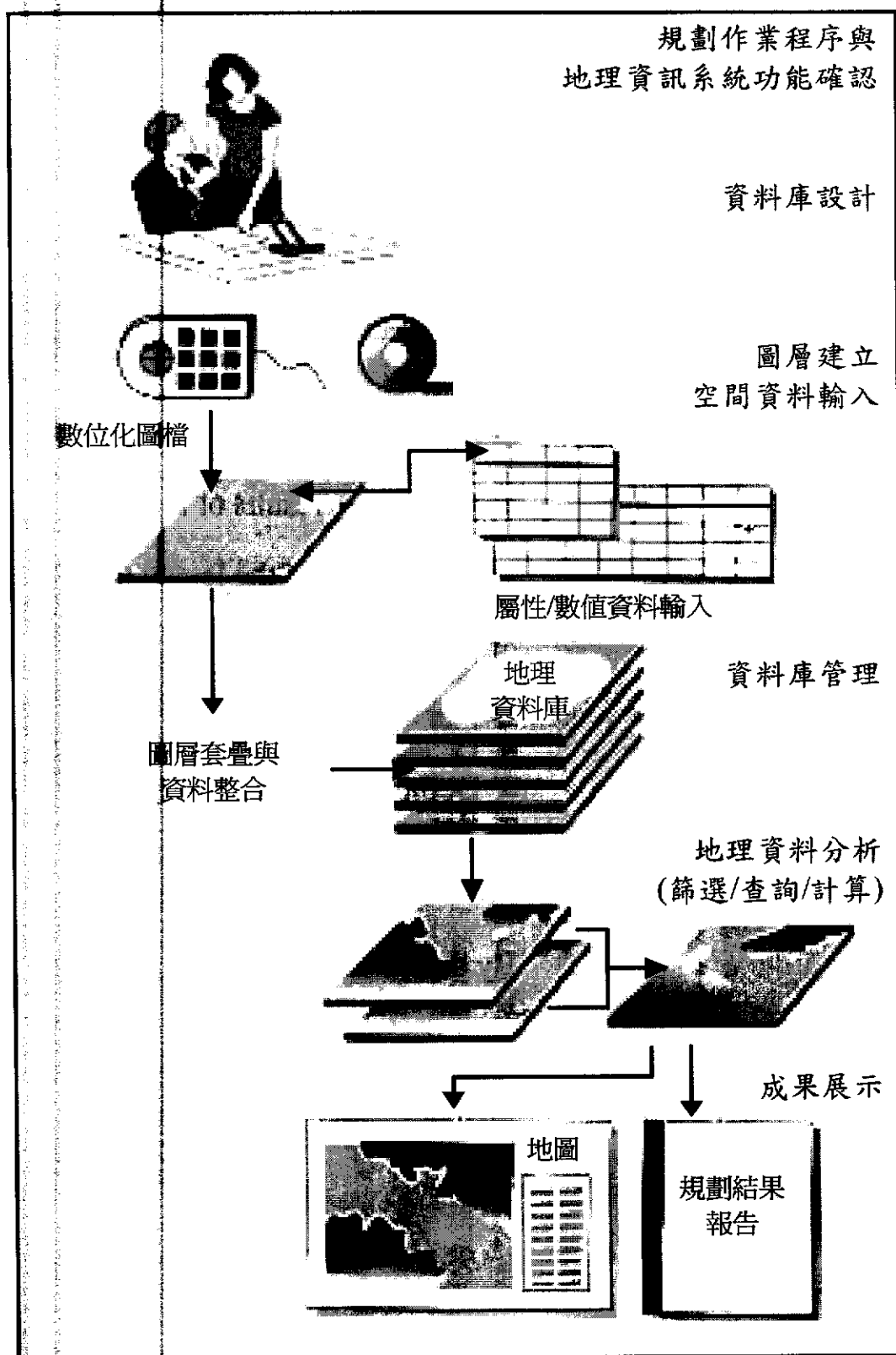


圖 1-1 地理資訊系統作業程序

[資料來源：ESRI, 1993]

四、資料分析

地理資訊系統之資料分析功能使原本十分費時費力的分析工作，得以較有效率地運作，藉由空間資料數位化以及屬性資料輸入等資料自動化方式，以利分析作業進行。使用者可藉由交談式的操作方式簡便地更改部份分析方法，以分析不同情境下之替選方案。分析步驟包括：

- 1.瞭解問題，建立分析目標與準則。
- 2.空間資料與分析功能確認。
- 3.執行空間資料分析。
- 4.輸出結果列表分析。
- 5.方案評估與成果展示。

本計畫依據公車旅客起迄調查結果，結合相關研究單位建立的地理資料圖層與資料庫，以軟體 UfosNet 構建台北市公車地理資訊系統，期能應用計畫研究成果，作為未來台北都會區相關大眾運輸研究課題分析之基礎，而在資料庫的設計上，力求與捷運局「台北都會區整體運輸規劃模式」之資料格式相同，其內容則依未來相關研究需求與本研究調查結果，補充並延伸若干新的屬性資料。

1-2 地理資訊系統分析環境

地理資訊系統分析環境主要包括圖層與資料庫，分別簡介如下，詳細操作程序請參閱後續章節：

1. 圖層

UfosNet 將空間資料以多個圖層予以區隔，以利使用者視分析過程與展示需要彈性運用，其圖層包括：結點(Node)、節線(Link)、分區(Boundary-1/2/3)、分區邊界點(Boundary Point)、附註標記(Annotation)、點陣圖(Bitmap)、公車路線(Transit Line)等，藉由視窗工具列的操作，使用者可以簡易的步驟，構建與修改分析環境，如圖 1-2 所示。就公車路線規劃而言，站位於平面環境中為點的形式，而可行路網與公車路線為點與線的組合，交通分區則屬於面的展示，善用各種資料的空間形式，將使地理資訊系統發揮極佳的輔助效果[朱家德，1997]。

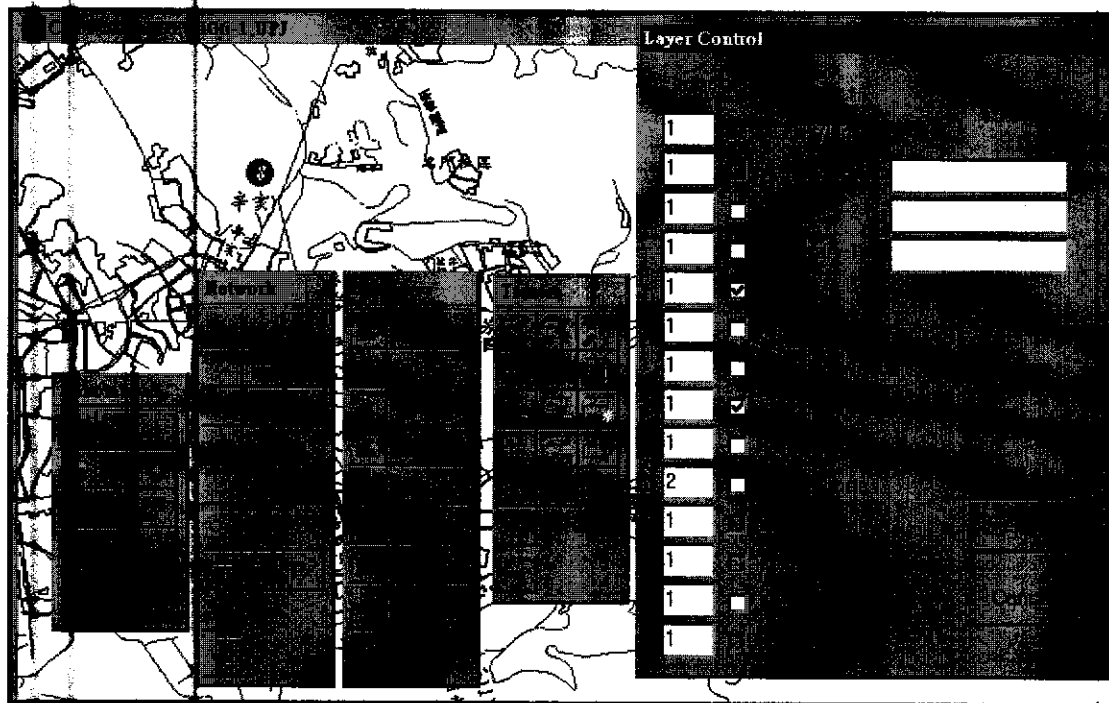


圖 1-2 地理資訊系統圖層建立與管理

2. 資料庫

UfosNet 對結點、節線、分區資料均提供使用者資料庫儲存與分析運算之功能，而公車路線目前僅提供資料儲存，包括路線與運具特性資料等，其他分析功能則尚在測試階段。

資料庫管理可以一般表格化方式構建與修改，亦可依據於地圖顯示狀態下依據其空間分佈特性，以線上作業方式給予適當屬性值，如圖 1-3 所示，各圖層與其資料庫可獨立轉出(Export)或轉入(Import)，圖層資料可與其他圖形處理軟體如 AutoCAD 相容，資料庫亦可與其他資料庫分析軟體如 Excel、dBase、Lotus1-2-3 等結合，配合目前電腦多工處理的運作，提昇分析效率，而其檔案架構亦可藉由轉檔方式應用於其他地理資訊系統軟體如 MapINFO、TransCAD 等。可提供相關研究資料間之共享與整合，以達到應用地理資訊系統提昇作業合理性、經濟性與效率性的目的。

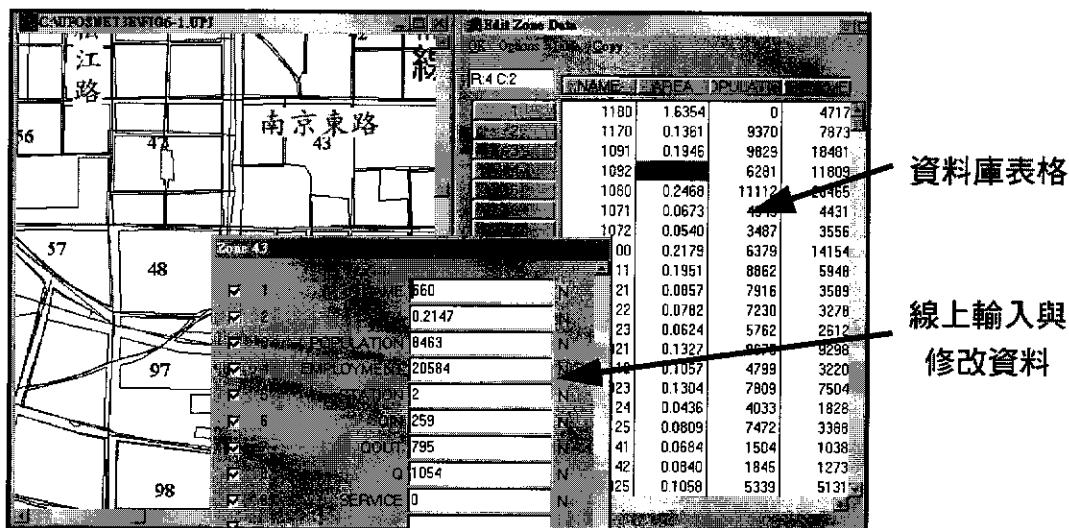


圖 1-3 地理資料庫建立與修改

1-3 技術報告第一部份各章節說明

第一部份共分七章，第一章簡介地理資訊系統之特性。第二章檔案管理則包括檔案儲存、載入形式與列印輸出等。第三章圖層與資料庫編輯介紹各式地理圖件製作與展示方式，以及各圖層對應資料庫之編輯與修改方法。第四章則介紹圖層與視窗控制技巧。第五章與第六章分別就各圖層與資料庫之分析運算功能進行介紹。大眾運輸系統分析與路線資訊功能則整合於第七章中介紹。

第二章 檔案管理

UFOSNET 將各種地理資料存於不同檔案中，各類型檔案包含資料庫、顯示格式緩衝記憶區與其他設定，而形成一圖層(Layer)，每一圖層與資料庫可獨立存檔，亦可合併存檔形成一工作計畫檔(Project File)。

各檔案類型簡述如下：

- **Network (.net)**-路網資料包括節點、節線、大眾運輸路線與路段資料、轉向損失及流量、及節點與節線的流量延滯函數(VDF)。此外，所有路網之顯示格式緩衝記憶區亦儲存於路網檔案。

注意：節點、節線及大眾運輸資料庫儲存於三個不同的圖層中。
- **Matrix (.tmx)**-矩陣檔案為一方陣型式，可用以儲存任何區對區之間的資料值，如旅次數、旅行時間、旅行距離、k 因子、運具承載率、停車轉乘之選擇及大眾運輸票價等。
- **Zone Data (.zon)**-用以設定各分區之屬性，如人口數、就業、社經資料、旅次產生/吸引、旅次起迄、區內旅行時間等。通常分區資料庫的分區數應與矩陣檔案(Matrix File)及第一層分區邊界檔案(Boundary-1 File)的個數一致。
- **Point Data (.upt)**-點資料為一簡單的資料庫，不像路網檔案(Network File)，點資料並無設定的地理上的限制，因此任何格式一致的 dBase-III 檔案皆可轉換成點資料檔案，如節點、節線、大眾運輸路段資

料庫等。點資料檔案可用以儲存家戶調查檔案、O/D調查檔案及儲存圖片檔案以備使用者點選。UFOSNET 藉由點資料檔案以進行一些特殊資料的操作，如計算任兩點間的最小成本路徑，點資料的排序、以文字格式輸出資料及顯示旅次鍊等。

注意：點資料庫儲存於一分隔獨立的圖層。

- **Boundary and Arclink (.zby)-分區檔案(Boundary File)**包含分區之空間座標與幾何特性，每一分區可以包含一或多個多角形。分區(Boundary)通常用以描述分區的形狀，如交通分區(TAZ)等。分區檔案的構成數目應與分區資料檔案(Zone Data File)一致。

記憶體可同時容納三個分區檔案，這些分區檔案將儲存於不同的圖層：第一層分區(Boundary-1)、第二層分區(Boundary-2)與第三層分區(Boundary-3)。然而，只有第一層分區之圖層可以編輯並連結至分區資料庫，其餘兩個分區檔案只用於顯示。

分區邊界(Arclink)為一特殊之分區檔案，用以儲存道路邊界形狀之點資料。如果分區檔案以節線圖層儲存，所有分區將以多邊形顯示。分區邊界檔案應只包含單一多角形分區。

注意：Boundary-1、Boundary-2 與 Boundary-3 及分區邊界檔案皆儲存於四個不同的圖層。

- **Annotation (.ann)-此檔案包含註釋物件如符號、線段、多邊形、文字與圖檔等。每一個註釋物件各包含其顯示格式。**

- **Bitmap (.bmp)**-點陣可以單色或多色之 BMP 格式儲存。
- **Project File (.us)**-除了點陣與矩陣外，巨集檔案可將數個檔案一起儲存成單一檔案。巨集檔案可以儲存所有的資料庫、顯示格式緩衝記憶區、檔案集合及圖層集合。

2-1 系統設定(Configure System)

UFOSNET 系統設定主要就資料庫大小適當限制，以兼顧系統運算效率與分析規模之需求，系統設定對話方塊如圖 2-1 所示。設定內容包括各種資料庫之大小、座標系統與距離量度單位等。各項資料庫大小上限值隨軟體版本略有不同。

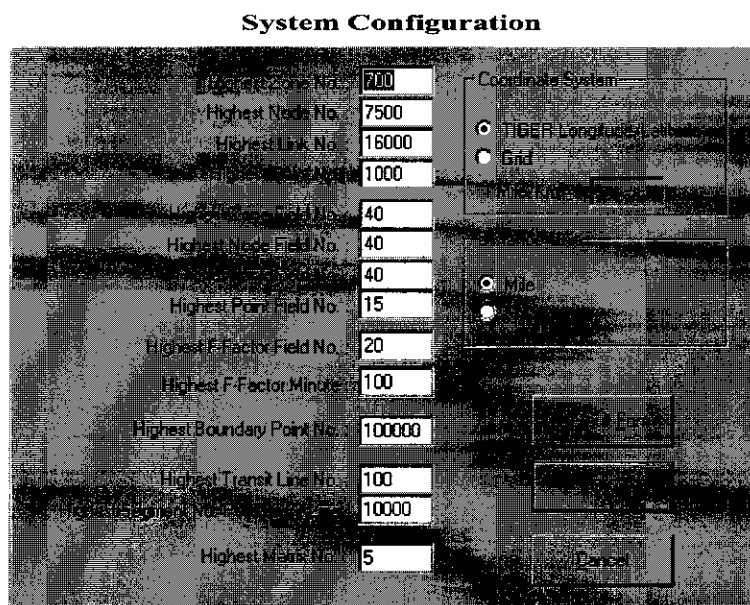


圖 2-1 系統設定對話方塊

2-2 檔案載入與儲存(Load and Save)

UFOSNET 各圖層檔案均可獨立儲存與載入，各圖層檔案型態如下：

- Network (.net)-路網檔案包括三圖層：節點(Node)、節線(Link)與大眾運輸路線(Transit)。
- Matrix (.tmx)-矩陣檔。
- Zone Data (.zon)-分區資料檔。
- Point Data (.upt)-點資料檔。
- Boundary-1 (.zby)-第一層分區並聯結分區資料(Zone Data)檔。
- Boundary-2 (.zby)-第二層分區不與分區資料(Zone Data)檔聯結。
- Boundary-3 (.zby)-第三層分區不與分區資料(Zone Data)檔聯結。
- Arlink (.zby)-分區邊界檔。
- Annotation-註釋層。
- Bitmap-點陣圖檔。

為使檔案管理更具彈性，UFOSNET 亦提供檔案整合管理之功能，使用者可將所有檔案儲存為工作計畫檔案(Project File; .upj)；另外亦提供密碼保護功能，以滿足使用者之需求。

2-3 列印(Print)

欲將螢幕呈現之畫面列印輸出時，選擇列印項目後，UFOSNET 將出現圖 2-2 之對話方塊。




圖 2-2 列印設定對話方塊

對話方塊之設定包括印表機設定、單色列印(無灰階、彩色功能)、外框框線設定、列印標題、標題字型與預覽列印等。

第三章 圖層與資料庫編輯

3-1 路網編輯工具箱(Toolbox-Network)

按快速鍵  即可開啟路網編輯工具箱，如圖 3-1 所示，此工具箱包含節點與節線之編輯功能。

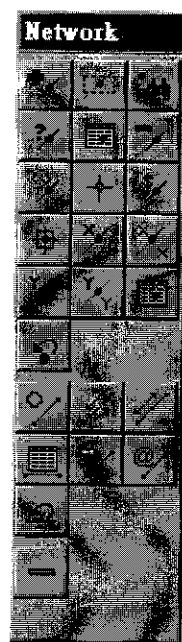


圖 3-1 路網編輯工具箱

3.1.1 加入節點(Node-Add)

按此鍵可於現有路網中加入新節點，節點編碼可由使用者在 Configuration 限制範圍內自行決定，或如圖 3-2 所示按下 Next#鍵，則可編入所有可用碼之最小值。

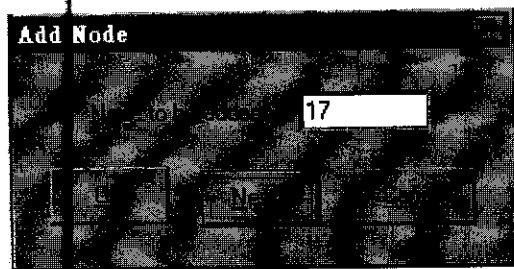





圖 3-2 加入節點對話方塊


3.1.2 移動節點(Node-Move)

按快速鍵  後，可由滑鼠點選節點並移動至新的位置，若需回復原狀，則按下復原鍵  即可。


3.1.3 搜尋節點(Node-Find)

按  鍵可尋找單一節點或編碼在某一範圍內之節點，搜尋工作完成後，螢幕游標將顯示於搜尋結果處，而搜尋所得節點亦將以高亮度及加註編碼方式顯示。

3.1.4 查詢節點(Node-Post)

按下  鍵後，再按下節點，則可查詢節點編碼，若需連續查詢，則選擇 Permanently 功能。

3.1.5 節點資料(Node-Data)

按  鍵可開啟節點資料對話框，點選節點即可針對特定節點進行線上(On-line)編輯，如圖 3-3 所示，對話框右側為各欄位資料型態(N:數值資料，C:字元資料)。

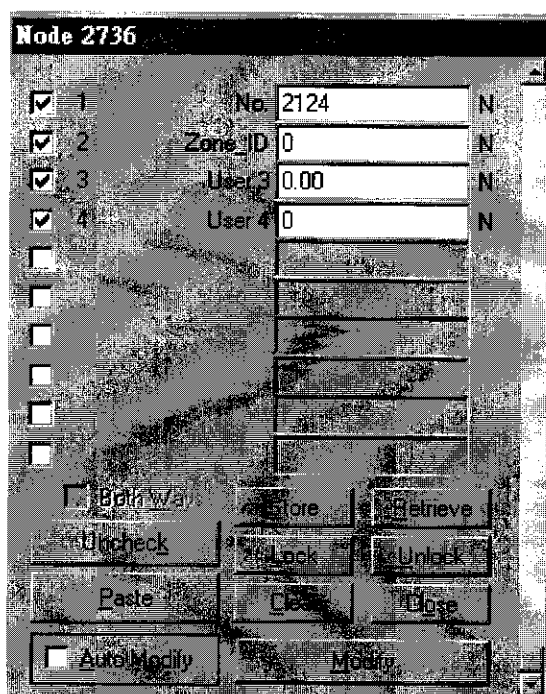



圖 3-3 節點資料庫線上修改

各項編輯功能顯示如下：


- **Store**-儲存現有該節點資料於暫存檔。
- **Retrieve**-以原存於暫存檔資料值取代現有資料。
- **Lock**-鎖定游標於特定欄位，以便使用者修改。
- **Unlock**-解鎖游標於特定欄位，以便使用者修改。
- **Modify**-永久儲存。

- **Paste**-複製緩衝記憶區(Buffer)資料到選定節點。
- **Auto Modify**-修改多點資料欄位，且資料值亦相同時，此鍵可取代 Store-Retrieve-Modify 程序。
- **Clear**-清除所有資料值。


3.1.6 移除節點(Node-Delete)

按  鍵移除節點前，需將其所連接之所有節線先行去除。

3.1.7 節線聯繫查詢(Node-Show Links)

按  鍵以高亮度顯示使用者點選之節點所連繫之節線。

3.1.8 轉向損失(Node-Turn Penalty)


 鍵提供路口轉向損失設定功能，藉由轉向損失之設定，可反映實際路網之轉向限制，如禁止左轉等。

注意：當一路口超過六條以上路段匯集時，則此功能將不提供。


3.1.9 復原(Node-Undo)

此鍵提供部份節點恢復修改前之狀態，包括刪除(Delete)、移動(Move)與比例調整(Rescale)。


3.1.10 加入節線(Link-Add)

按  鍵後，點選起始節點，拖曳至節線終點，將出現如下圖之對話框，確認節點編碼後，若此路段為雙向通行，則選取 Both Direction 功能。

3.1.11 合併節線(Link-Join)

按  鍵，將合併連續之節線為一條，刪除多餘節點後產生新節線，而其幾何特性(如：長度)資料將直接分割為二。

3.1.12 分割節線(Link-Split)

按  鍵後，首先拖曳點選欲分割之節線，而後點選欲分割之位置，編以有效編碼，即完成節線分割之工作，節線資料將同時存於兩分割段，而無資料分割情形。

注意：節線合併(Join)或分割(Split)均需先將其上之公車路線刪除，否則不支援此功能。

3.1.13 節線資料(Link-Data)

節線資料線上修改對話方塊如圖 3-4 所示。

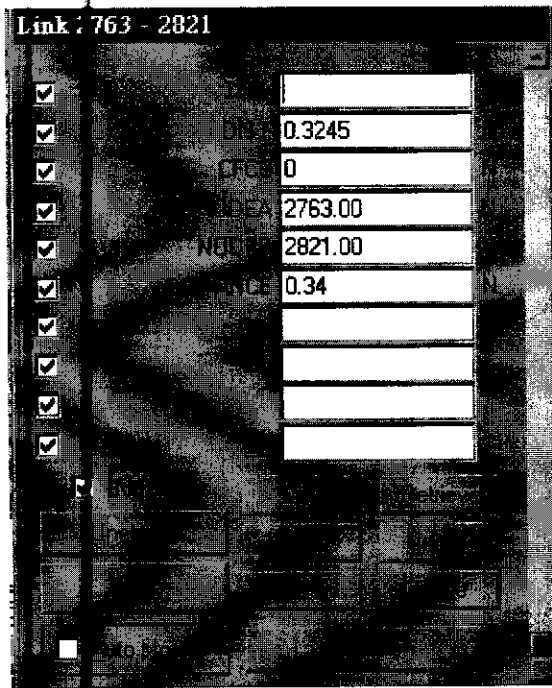



圖 3-4 節線資料線上修改對話方塊


- **Both Ways**-更改單向或雙向資料。
- **Store**-儲存現有該節線資料於暫存檔。
- **Retrieve**-以原存於暫存檔資料值取代現有資料。
- **Lock**-鎖定游標於特定欄位，以便使用者修改。
- **Unlock**-解鎖游標於特定欄位，以便使用者修改。
- **Modify**-永久儲存。
- **Paste**-複製所有緩衝記憶區(Buffer)資料到選定節線。
- **Auto Modify**-修改多線資料欄位，且資料值亦相同時，此鍵可取代 Store-Retrieve-Modify 程序。
- **Clear**-清除所有資料值。

3.1.14 刪除節線(Link-Delete)

按  鍵後，拖曳選取欲刪除節線，使用者可選擇刪除單向或雙向節線，如下圖所示。

注意：若節線上存有公車路線，則不支援此功能。

3.1.15 查詢線段距離(Link-Distance)

 鍵提供使用者查詢任兩節點間直線距離，如下圖所示。連續查詢時，提供累計距離功能，若需歸零時，再按一次該鍵即可。

3.1.16 節線編輯復原(Link-Undo)

此鍵提供和節點相同之復原功能，主要應用於節線合併(Join)、分割(Split)及刪除>Delete)等編輯功能。

3-2 分區編輯工具箱(Toolbox-Boundary)

台北市公車地理資訊系統提供三層分區顯示功能，但僅有第一層(Boundary-1)具有編輯功能。每個分區均為一或多個多邊形組合，在單一編碼下，聯結相關屬性資料，而多邊形乃由邊界點構成，在構建分區前，需定義邊界點位置。




按  鍵將出現分區編輯工具箱，如圖 3-5 所示，以下各小節將逐一介紹各按鈕對應之功能。



圖 3-5 分區編輯工具箱

3.2.1 構建新分區(Boundary-Build)

按  鍵後，螢幕將出現下圖之對話方塊，首先輸入分區之編碼(Boundary ID)與文字標記(Boundary Label)，關閉對話框後，點選預先建好的分區邊界點，構成一封閉多邊形後，按  鍵即告完成。

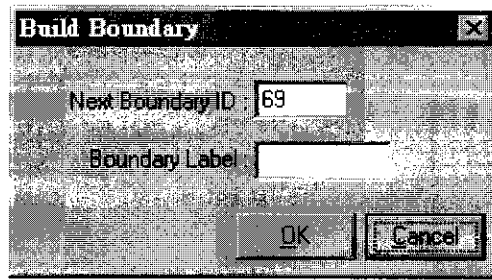







圖 3-6 構建新分區對話方塊


3.2.2 邊界選取與刪除(Boundary-Select)

1. 游標位於分區內之選取方法：按  鍵。
2. 依游標最鄰近之分區重心為選取標準：按  鍵。
3. 刪除分區：選取分區後，按  鍵將刪除該分區及相關資料。

3.2.3 分區重心查詢與重新編碼

1. 重心位置查詢：按  鍵
2. 重新編碼：按  鍵，若重新編碼為已使用之 ID，將會出現警告訊息，若強制編入，則原有該 ID 之分區將自動消失。

3.2.4 變更分區顯示型式(Boundary-Style)

按  鍵將出現圖 3-7 之對話方塊，以修改分區顯示型式，包括：編號、編界線條、填色之顯示方式。

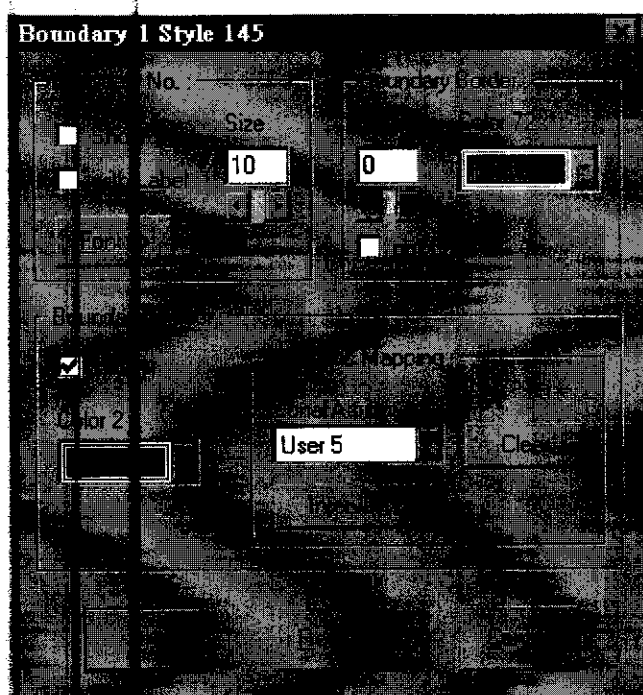







圖 3-7 變更分區顯示形式


3.2.5 分區面積與周長計算

- 1.分區面積計算：選取分區後，按  鍵，可計算面積。
- 2.周長計算：選取分區後，按  鍵，可計算周長。

3.2.6 分區邊界點之加入與刪除

- 1.加入新邊界點：選取分區後，按  鍵，可區入新分區邊界點，一旦加入點後即無法刪除。
- 2.移動分區邊界點：可運用以下方式進行。
 - (1)利用滑鼠直接移動：按  鍵。
 - (2)輸入確定之座標：按  鍵。

3.2.7 分區比例調整(Rescale)、組合 (Combine)與分割(Break)

- 1.分區比例調整：按  鍵，出現如下圖之對話方塊，以調整 X/Y 方向之比例。

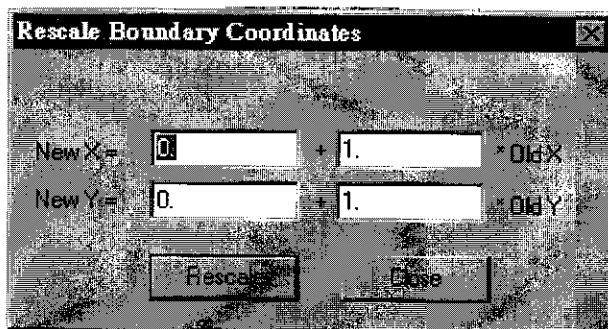



圖 3-8 分區比例調整對話方塊

- 2.分區組合：按  鍵可合併兩分區，如圖 3-9 所示。

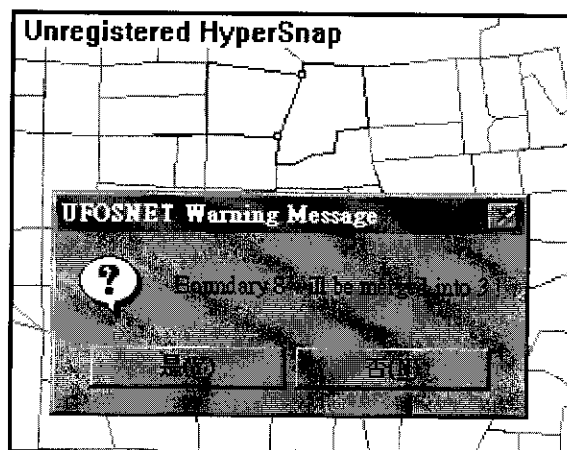



圖 3-9 分區組合

- 3.分區切割：按  鍵，將分區一分为二。

3.2.8 分區資料編輯(Boundary-Data)

按  鍵可開啟分區資料對話框，點選分區即可針對特定分區進行線上(On-line)編輯，如圖 3-10 所示，對話框右側為各欄位資料型態(N:數值資料，C:字元資料)。

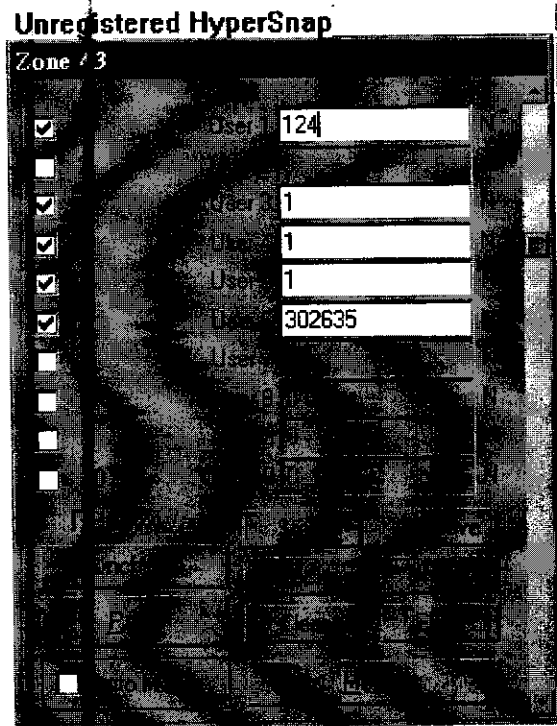


圖 3-10 分區資料線上修改對話方塊

各項編輯功能顯示如下：

- **Store**-儲存現有該分區資料於暫存檔。
- **Retrieve**-以原有儲存於暫存檔資料值取代現有資料。
- **Lock**-鎖定游標於特定欄位，以便使用者修改。
- **Unlock**-解鎖游標於特定欄位，以便使用者修改。
- **Modify**-永久儲存。
- **Paste**-複製所有緩衝記憶區(Buffer)資料到選定分

區。

- **Auto Modify**-修改多區資料欄位，且資料值亦相同時，此鍵可取代 Store-Retrieve-Modify 程序。
- **Clear**-清除所有資料值。

3.2.9 影響圈域(Boundary-Buffer Region)

選取分區後，按下此鍵，使該分區形成影響圈域，而後之分析與查詢將限制於此區域內，如圖 3-11 所示。


(3-21 頁圖)

圖 3-11 設定分區為影響圈域

3.2.10 復原(Boundary-Undo)

復原鍵提供部份分區修改前之狀態，適用於重心查詢、邊界點移動與比例重置、組合等。

3-3 註釋層編輯工具箱 (Toolbox-Annotation)

註釋層主要提供輔助說明功能，使地理資訊得以更明確地展示。按  鍵後螢幕出現其工具箱，如下圖所示。

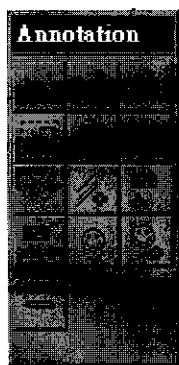




圖 3-12 註釋層編輯工具箱

3.3.1 設定註釋圖件群組 (Annotation-Select and setup Group)

欲編輯註釋圖件前，先按  鍵以確定編輯圖件之群組。

3.3.2 移動(Move)與選取(Select)

1. 選取物件：確定物件群組後按  鍵選取欲編輯之物件。

2. 移動單一物件：按  鍵點選物件。

3.移動物件群組：按  鍵後點選該群組任一物件。

3.3.3 加入註釋物件

(Annotation-Add Symbol Object)

註釋物件形式包括：符號(Symbol)、線段(Line)、文字(Text)、圖檔(Metafile)。加入物件之功能鍵包括：

1.符號：按  鍵。


2.線條：按  鍵。


3.文字：按  鍵。


4.圖檔：按  鍵。


3.3.4 定義註釋物件型式

(Annotation-Define Symbol Style)

1.符號：按  鍵，如圖 3-13 所示，可變更其大小、圖形、群組、線段與填色，內填數值與其大小等。

2.線段：按  鍵，如圖 3-14 所示，可變更其線型、顏色、寬度。

3.文字：按  鍵，如圖 3-15 所示，可變更其字型、大小、顏色、轉角、字體等。

4.圖檔：按  鍵，圖檔亦可由相關之圖形編輯器，如 Hijaak95(tm)設計使用者所需之 emf 圖檔放進

ufosnet\emf\子目錄下，再於 ufosnet 中載入即可。

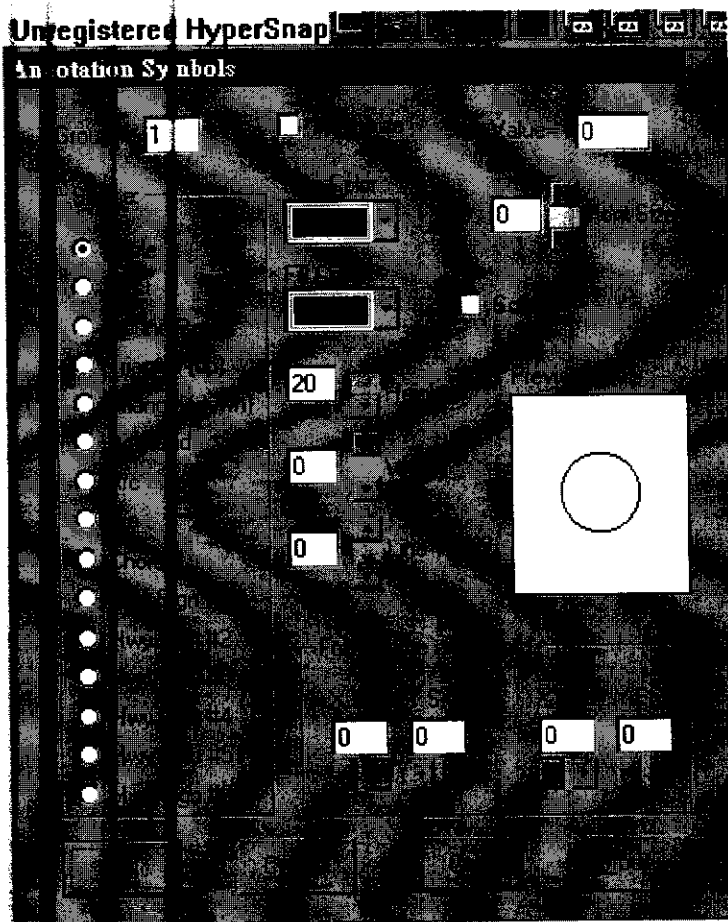


圖 3-13 定義符號形式對話方塊

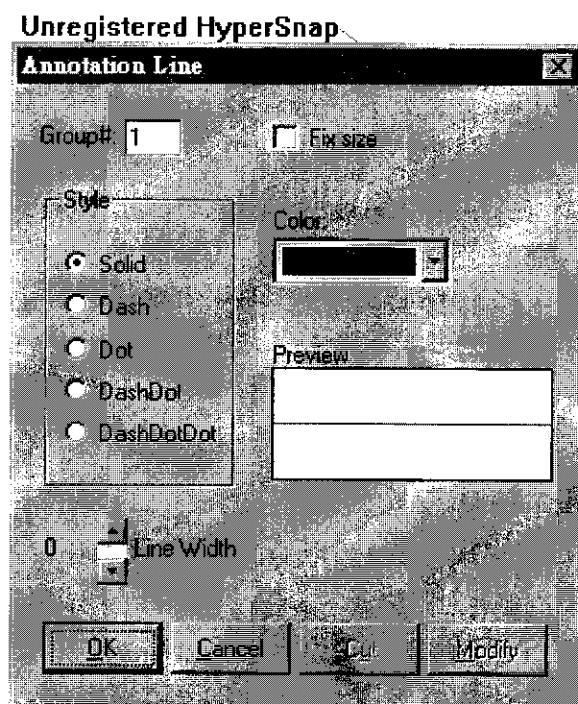


圖 3-14 定義線條形式對話方塊

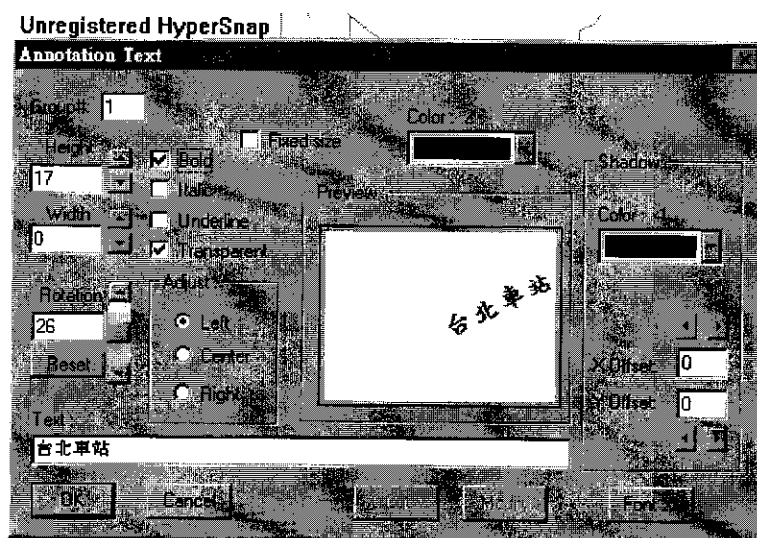


圖 3-15 定義文字形式對話方塊

3-4 點資料(Point Data)

編輯點資料僅需點選該點即可出現資料修改對話方塊，如圖 3-16 所示。對話方塊中各按鍵功能如下：

(3-29 頁圖)

圖 3-16 點資料編輯對話方塊

各按鍵功能說明如下：

- **Store**-儲存現有該點資料於暫存檔。
- **Retrieve**-以原存於暫存檔資料值取代現有資料。
- **Lock**-鎖定游標於特定欄位，以便使用者修改。
- **Unlock**-解鎖游標於特定欄位，以便使用者修改。
- **Modify**-永久儲存。
- **Paste**-複製所有緩衝記憶區(Buffer)資料到選定點。
- **Auto Modify**-修改多點資料欄位，且資料值亦相同時，此鍵可取代 Store-Retrieve-Modify 程序。
- **Clear**-清除所有資料值。

3-5 點陣圖檔(Bitmap)

載入點陣圖檔後，可以拖曳方式改變其點陣圖大小與位置，完成後再選一次 Bitmap 編輯功能鍵，即可離開點陣圖編輯模式。

3-6 資料表格(Data Sheet)

除前述之線上修改方式外，資料庫亦可以表格型式編輯，如圖 3-17 所示。畫面至多可出現五個資料欄位，完成後按 Modify/Close 鍵即可存檔離開。

Unregistered HyperSnap102.011

Data Table - Zone

Zone ID	NAME	POPULATION	EMPLOYMENT	STATION	QIN
11	1122	7230	3278	0	0
12	1123	5762	2612	0	0
13	1021	9675	9298	2	296
14	1112	4799	3220	0	0
15	1023	7809	7504	0	0
16	1124	4033	1828	0	0
17	1125	7472	3388	0	0
18	1141	1504	1038	0	0
19	1142	1845	1273	0	0
20	1025	5339	5131	2	163

Modify Close

圖 3-17 資料表格

3-7 定義資料庫(Database Definition)

在各種資料庫使用前，需定義各資料欄位之儲存型式，包括欄位各資料型式(字元、數值)長度、數值資料、小數位數，如圖 3-18 所示。

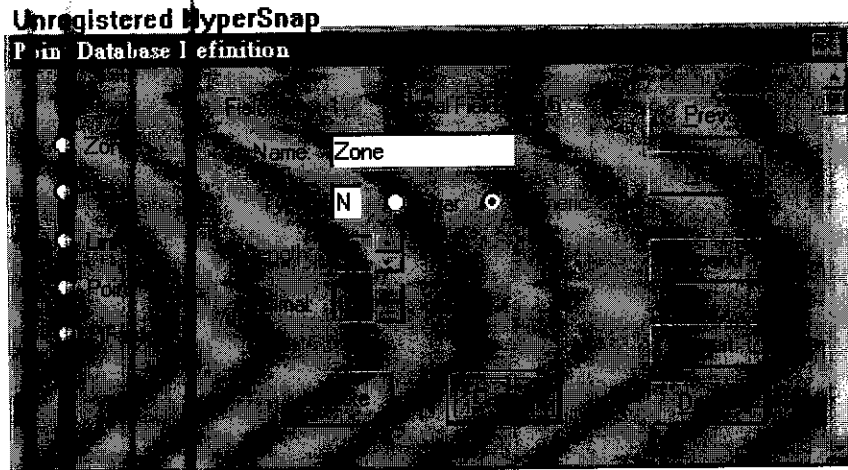


圖 3-18 定義資料欄位

台北市公車地理資訊系統資料庫結構，包括分區(Zone)、節點(Node)、節線(Link)、點(Point)、阻抗因子(F-Factor)，如下圖所示，各資料庫至多可開啟 65,000 欄，而資料筆數主要受限於電腦處理能力限制。

各按鍵功能簡介如下：

- Prev-到前一資料欄位。
- Next-到下一資料欄位。
- Append-加入新欄位。
- Insert-插入新欄位。
- Delete-刪除欄位。
- Modify-更新目前資料欄位型式。

第四章 圖層與視窗控制

4-1 圖層結構(All Objects)

UFOSNET 圖層結構如圖 4-1 所示：

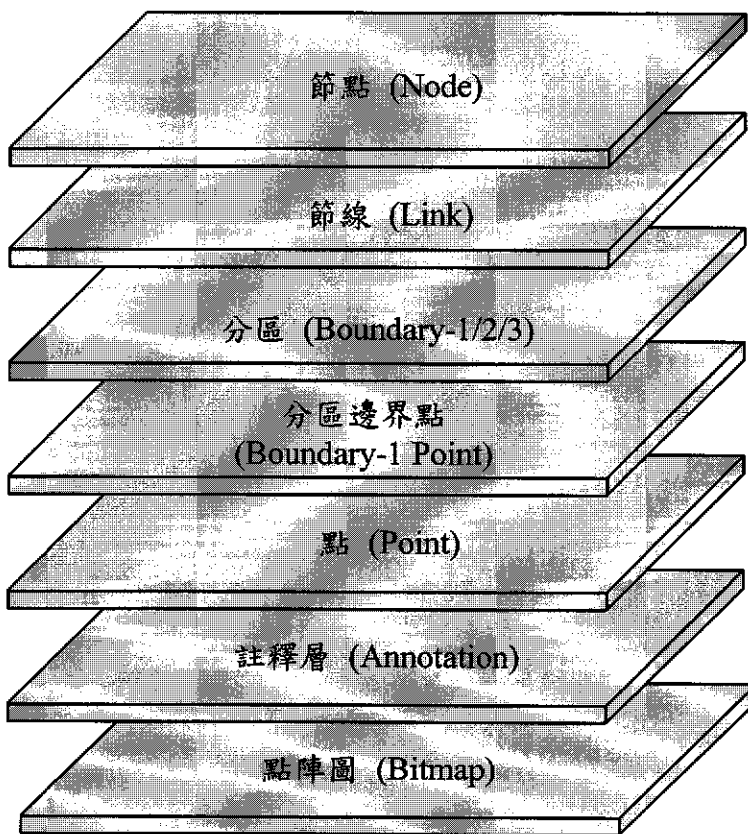


圖 4-1 UFOSNET 圖層結構

4-2 物件查詢(Query Objects)

物件查詢主要在提供使用者依據資料庫內容介定篩選準則，以不同於原有之資料顯示方式(如顏色)，展示篩選結果，本節將介紹分區、節點、節線之篩選。

4.2.1 物件查詢-分區

(Query Objects-Zones)

分區查詢對話方塊如圖 4-2 所示，主要針對 Boundary 而言。其操作功能簡介如下：

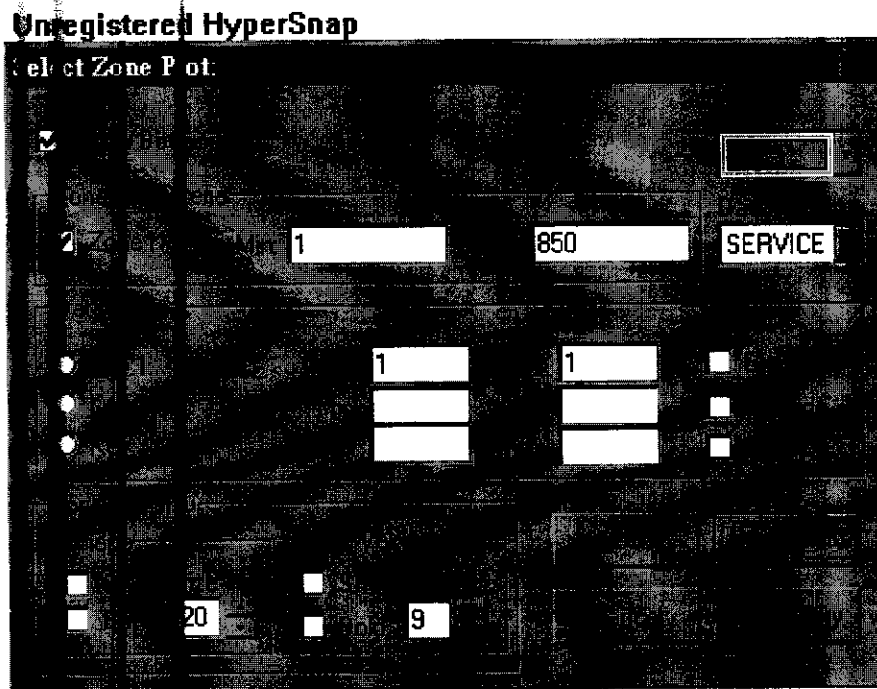


圖 4-2 分區查詢對話方塊

- **Show Boundary**-分區包括 Boundary-1 的多邊形與 Node 中的分區重心。選擇此項，分區將以設定之顏色顯示。
- **Color**-設定顯示顏色。
- **Zone No. Criteria**-以特定分區編碼範圍為篩選原則時使用。
- **Data Keys**-選特定資料欄位為篩選依據。
- **Data Criteria**-設定篩選準則，其程序如下：
 - 1.選擇資料欄位。
 - 2.選擇準則編碼(Key1/2/3)。
 - 3.設定範圍。
 - 4.若欲篩選範圍以外者，選取 exclude。
 - 5.若有二或三種準則，重複 1~4 步驟。
- **Shape**-可以餅圖或柱狀圖顯示各分區篩選標準欄位之資料值。
- **Data Value**-顯示資料值。
- **New Query**-查詢程序與前面操作無關時。
- **Sub Query**-延續上次篩選結果之查詢時。
- **Reset**-重設所有設定。

4.2.2 節點查詢(Query Objects)

節點查詢對話方塊如圖 4-3 所示。

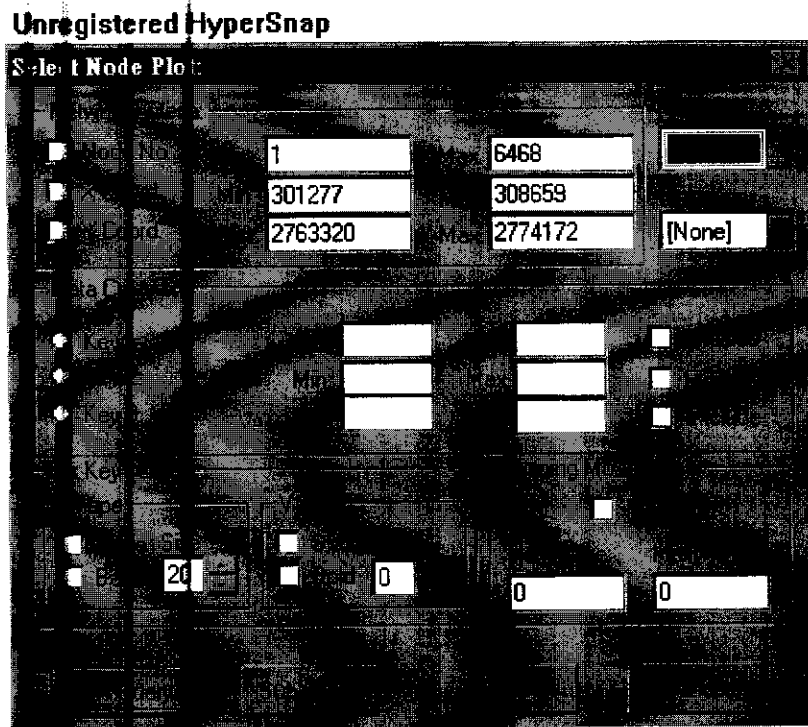


圖 4-3 節點查詢對話方塊

各項功能簡介如下：

- **Color**-設定顯示顏色。
- **Data Keys**-選特定資料欄位為篩選依據。
- **Network Criteria**-除以編碼外，亦可以 X,Y 座標作為篩選標準。
- **Data Criteria**-設定篩選準則，其程序如下：
 - 1.選擇資料欄位。
 - 2.選擇準則編碼(Key1/2/3)。
 - 3.設定範圍。

4.若欲篩選範圍以外者，選取 exclude。

5.若有二或三種準則，重複 1~4 步驟。

- **Shape**-可以餅圖或柱狀圖顯示各節點篩選標準欄位之資料值。
- **Data Value**-顯示資料值。
- **New Query**-查詢程序與前面操作無關時。
- **Sub Query**-延續上次篩選結果之查詢時。
- **Reset**-重設所有設定。

4.2.3 節線查詢(Query Objects-Links)

節線查詢對話方塊如圖 4-4 所示。

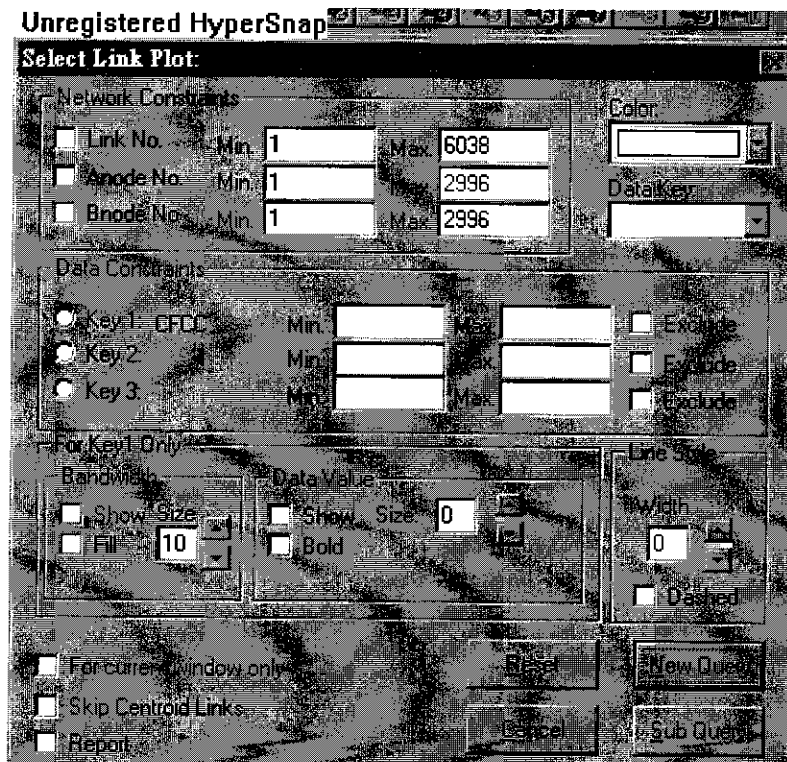


圖 4-4 節線查詢對話方塊

各項功能簡介如下：

- **Color**-設定顯示顏色。
- **Data Keys**-選特定資料欄位為篩選依據。
- **Network Criteria**-除以節線編碼外，亦可以節線起迄點之編碼作為篩選標準。
- **Data Criteria**-設定篩選準則，其程序如下：
 - 1.選擇資料欄位。
 - 2.選擇準則編碼(Key1/2/3)。
 - 3.設定範圍。
 - 4.若欲篩選範圍以外者，選取 exclude。
 - 5.若有二或三種準則，重複 1~4 步驟。
- **Shape**-可以餅圖或柱狀圖顯示各節線篩選標準欄位之資料值。
- **Data Value**-顯示資料值。
- **New Query**-查詢程序與前面操作無關時。
- **Sub Query**-延續上次篩選結果之查詢時。
- **Reset**-重設所有設定。

4-3 預設圖層顯示格式

(Define Default Style)

預設之圖層顯示格式包括各圖層元件之填色、線段顏色、編碼顯示與字型顏色、大小等。此格式會因其他查詢、分析操作而變更，以 Clear Plot Buffer 選項將可

恢復預設格式。

4.3.1 預設節點格式

(Define Default Style-Node)

節點格式設定對話方塊如圖 4-5 所示。

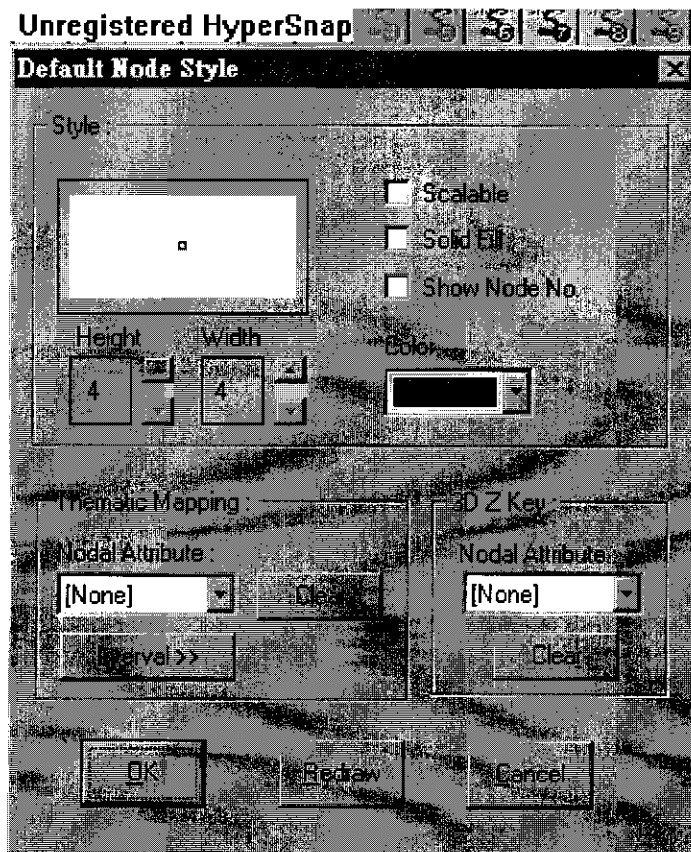


圖 4-5 節點格式設定對話方塊

相關功能簡介如下：

- **Scalable**-使節點大小可以調整，否則將以 4 點數顯示。
- **Solid Fill**-空心或實心節點。

- **Show Node No.**-是否顯示編碼。
- **Height**-若 Scalable 開啟，則可調整其高度。
- **Width**-若 Scalable 開啟，則可調整其寬度。
- **Nodal Attribute**-主題地圖顯示功能所應用之屬性值。
- **Interval**-各節線之選定屬性值依不同區間顯色。

4.3.2 節線預設格式

(Define Default Style-Link)

節線預設格式對話方塊如圖 4-6 所示。

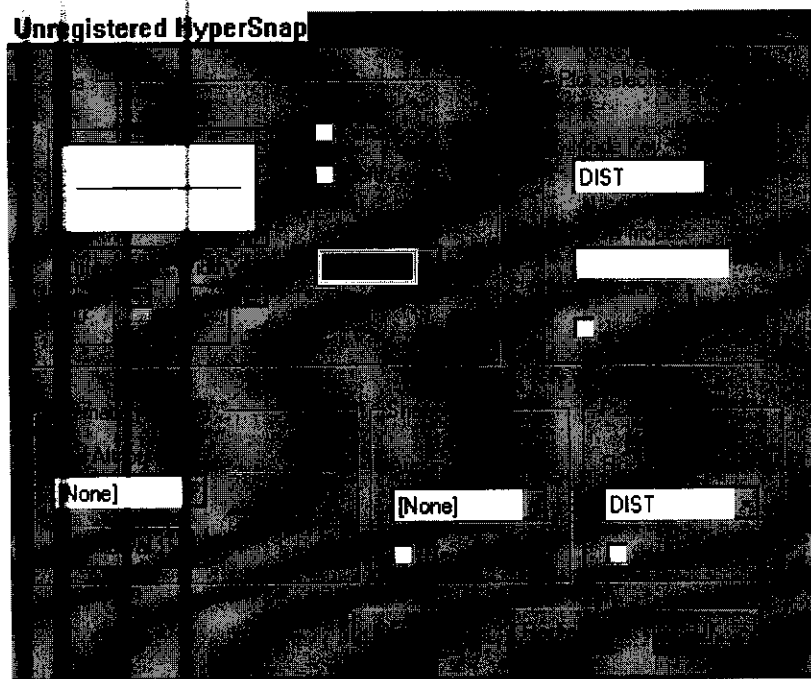


圖 4-6 節線預設格式對話方塊

簡介各項功能如下：

- **Arrow Head**-是否需顯示路段之方向性。

- **Left Handed**-檢核是否為靠左行駛路段。0
- **Color**-節線顏色。
- **Offset**-節線雙向分開顯示。
- **Width**-節線顯示寬度。
- **Link Attribute**-主題地圖顯示功能所需之節線屬性值。

4.3.3 分區重心預設格式(Define Default Style-Centroid)

分區重心為一群點集合，故其格式設定功能鍵與節點相同，其對話方塊如圖 4-7 所示。

Unregistered HyperSnap

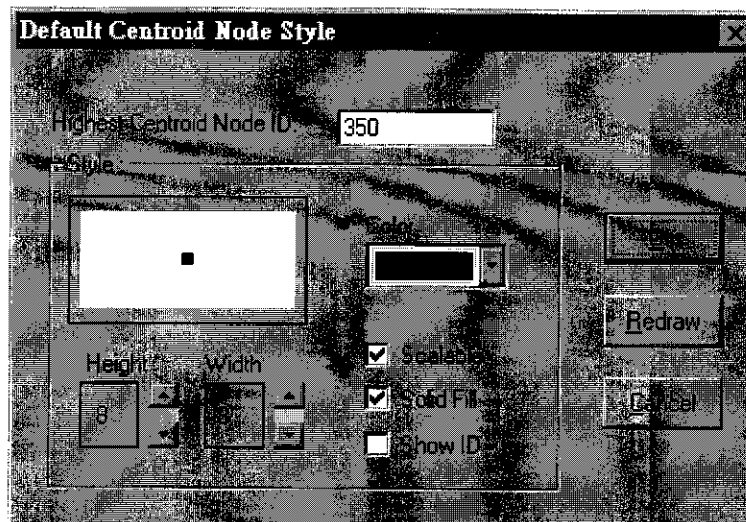


圖 4-7 分區重心預設格式對話方塊

4.3.4 第一層分區預設格式

(Define Default Style-Boundary-1)

分區為面之集合，其預設格式對話方塊如圖 4-8 所示。

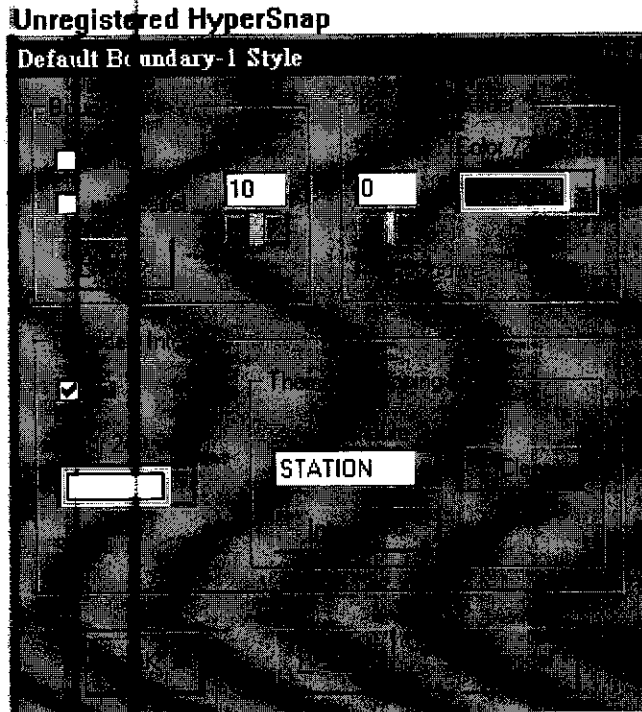


圖 4-8 第一層分區預設格式對話方塊

分區各項按鍵功能簡述如下：

- **Show Boundary No.**-檢核是否顯示分區編號。
- **Show Boundary No. with Label**-檢核是否顯示分區編號與標記。
- **Font**-選擇顯示字型。
- **Font Size**-選擇顯示字體大小。
- **Border Width**-決定分區邊界線條寬度。
- **Border Color**-決定分區邊界線條顏色。
- **Fill**-檢核是否需填色，亦為主題地圖功能提供之基

礎。

- **Fill Color**-選擇顯示顏色。
- **Zonal Attribute**-主題地圖顯示功能所需之分區屬性值。
- **Interval**-各屬性依不同區間顯色，如圖 4-9 所示。

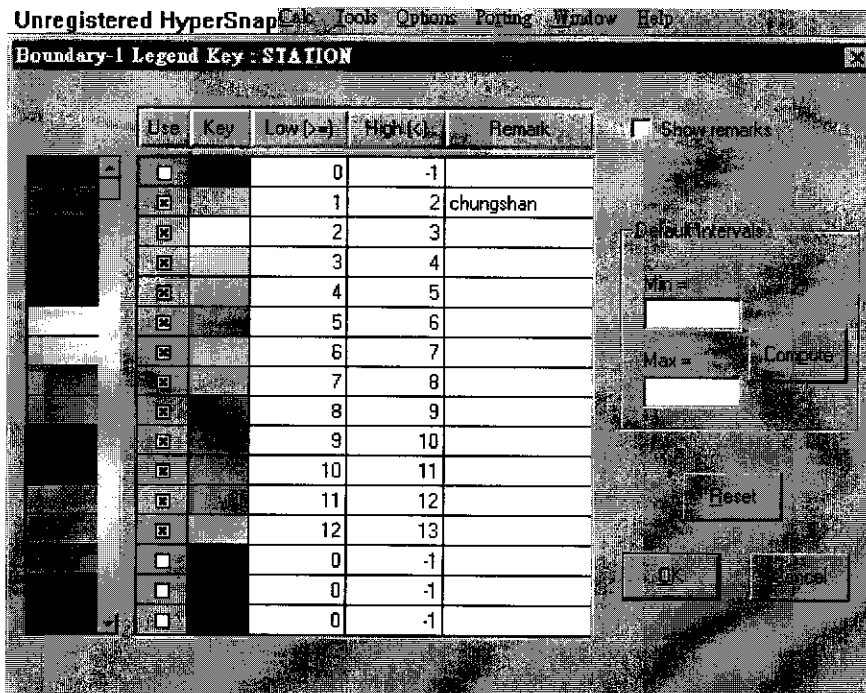


圖 4-9 各分區依屬性值不同顯色

4.3.5 第二、三層分區預設格式

(Define Default Style-Boundary-2/3)

對話方塊和第一層相同，但不提供主題地圖之功能。

4-4 清除存區物件顯示(Clear Plot Buffer)

因查詢、分析過程中留存於畫面中之圖件均儲存於暫存區中，需適時予以清除，清除方式包括：

- **Node**-清除暫存節點。
- **Link**-清除暫存節線。
- **Node/Link**-同時清除暫存之節點、節線。
- **Point**-清除點。

4-5 圖層控制(Layer Control)

圖層控制功能可依分析需要顯示所需圖層，並減少不必要的圖層顯示，減少電腦運算之負荷，按快速鍵

 將出現如圖 4-10 之圖層顯示控制對話方塊。

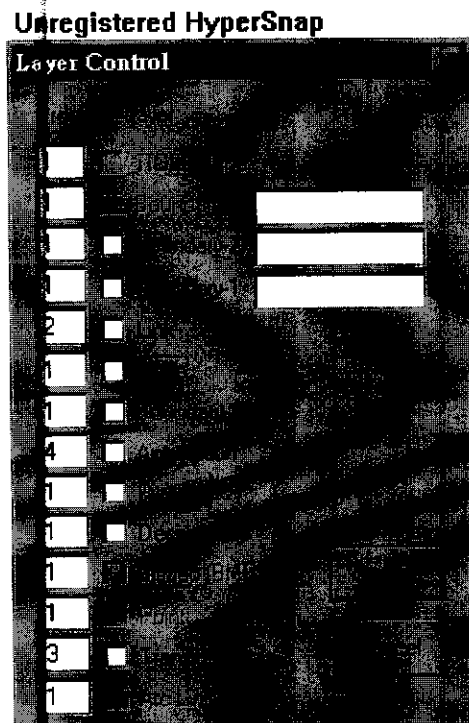






圖 4-10 圖層控制功能


4-6 視窗控制功能


視窗控制包括螢幕顯示比例調整與更新等操作，包括：


- 1.顯示選取範圍(Zoom In)：按  鍵。

2. 預設視窗縮放(Default Zoom In/Out)：按  與  鍵。

3. 設定顯示比例(Fix Map Scale)：按  鍵。

4. 視窗平移(Freehand Panning)：按  鍵。

5. 畫面重新顯示(Redraw Screen)：按  鍵。

6. 畫面更新(Refresh Screen)：按  鍵。

7. 畫面清除(Erase Screen)：按  鍵。

第五章 相關分析功能

5-1 公路最短路徑查詢

(Highway Path Query)

此功能提供使用者於螢幕上以交談方式查詢最短路徑，查詢方式包括一對一(One-to-One)最短路徑查詢、多對一(Many-to-One)路徑查詢、一對多(Many-to-One)路徑查詢等。路徑產生方法包括「樹狀演算法」(Tree Builder Algorithms)與「藤蔓狀演算法」(Vine Builder Algorithms)等，後者應用於有轉向限制的路網，計算較為耗時，且電腦處理能力要求亦高。

公路最短路徑查詢功能對話方塊如圖 5-1 所示：

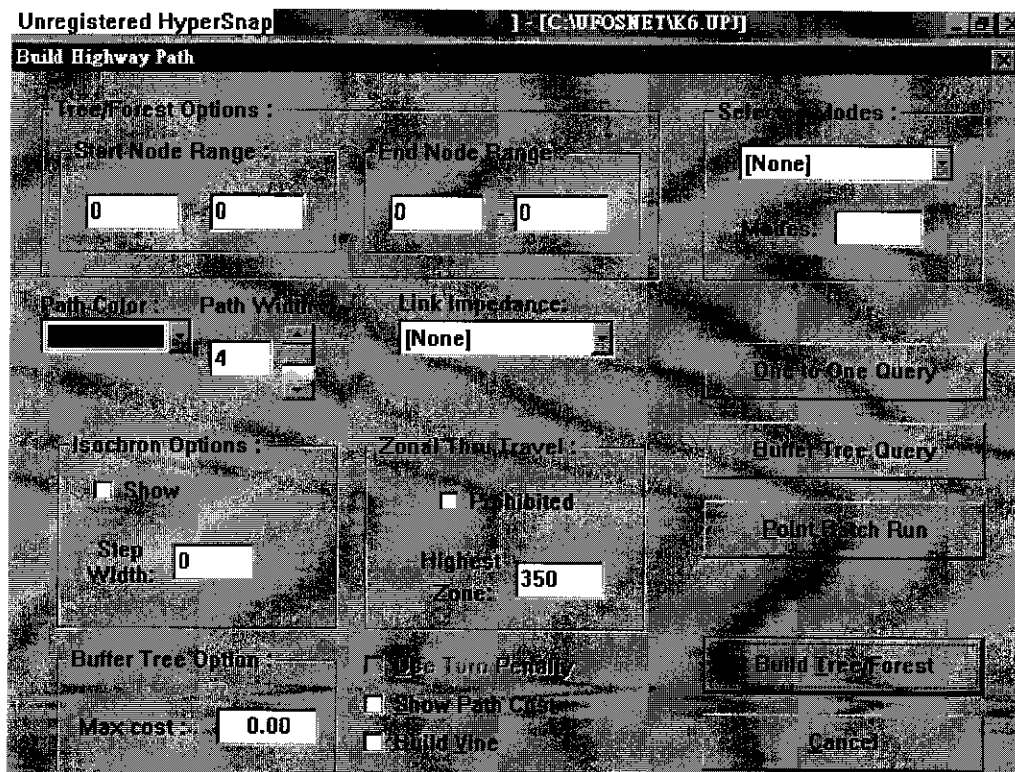


圖 5-1 公路最短路徑查詢對話方塊

各項功能鍵簡介如下：

- **Start Node Range-** 輸入路徑查詢之起點編碼範圍，若後者編碼小於前者，則僅前者有效，若為一對多路徑查詢，則僅需輸入前者。
- **End Node Range-** 輸入路徑查詢之迄點編碼範圍，若後者編碼小於前者，則僅前者有效，若為多對一路徑查詢，則僅需輸入前者。
- **Link Cost Field-** 選取適當的節線資料欄位作為最短路徑查詢之依據。
- **Select Modes-** 查詢公路依運具使用別作為分類，可縮小公路查詢範圍。
- **Path Color-** 選取路徑查詢結果展示顏色。
- **Path Width-** 選取路徑查詢結果展示寬度。
- **Isochron Options-** 路徑查詢結果可以不同顏色顯示其路徑成本大小之區間，最多 16 種顏色。
- **Use Turn Penalty-** 檢核是否需加入轉向損失作為最短路徑計算之考量。
- **Show Path Cost-** 檢核是否需顯示路徑成本值。
- **Build Vine-** 以藤蔓式演算法搜尋最短路徑，藤蔓式演算法在考慮轉向損失時始得應用。
- **One-to-One Query-** 一對一最短路徑查詢，不需輸入 Start Node Range 與 End Node Range，僅需輸入路徑成本屬性所屬欄位後，於螢幕上點選起點，拖曳至終點即可，如圖 5-2 所示。
- **Buffer Tree Query & Buffer Tree Option-** 主要應

用於多對一與一對多路徑查詢，使用者需輸入成本上限，並於螢幕點選一特定起(迄)點即可查詢。

- **Build Tree/Forest-** 主要應用於多對一、一對多與多對多路徑查詢，使用者需輸入起迄點範圍，可查詢各起迄點間之最短路徑，此時不提供 Isochron Options。

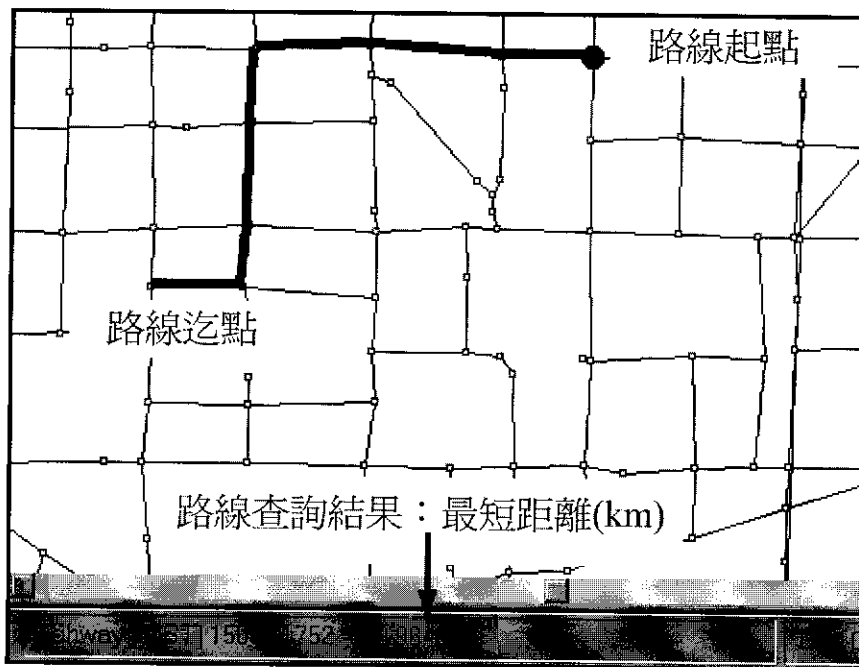


圖 5-2 一對一最短路徑查詢

5-2 大眾運輸路徑查詢

(Transit Path Query)

大眾路徑查詢功能提供使用者由大眾運輸營運觀點，查詢兩站間之最小成本路徑，做為路線規劃時參考，詳細功能說明請參見 7-2 節。

5-3 影響圈域分析(Buffer Region Analysis)

影響圈域區域分析可使規劃者由自訂基準點、線或地區一定範圍內進行進一步的資料分析工作，使距離或面積的量測自動化，對於路線規劃作業所需之車站位置、路線長度與影響範圍等資料，可藉由此功能輔助進行評估。同時，運用影響圈域可限制其他分析功能於此一範圍，除前面章節提及以分區為影響圈域外，使用者亦可於下拉視窗 Tool 中選取圓形(Circle Buffering)、沿線(Route Buffering)、多邊形(Polygon Buffering)等，其功能如圖 5-3 所示。

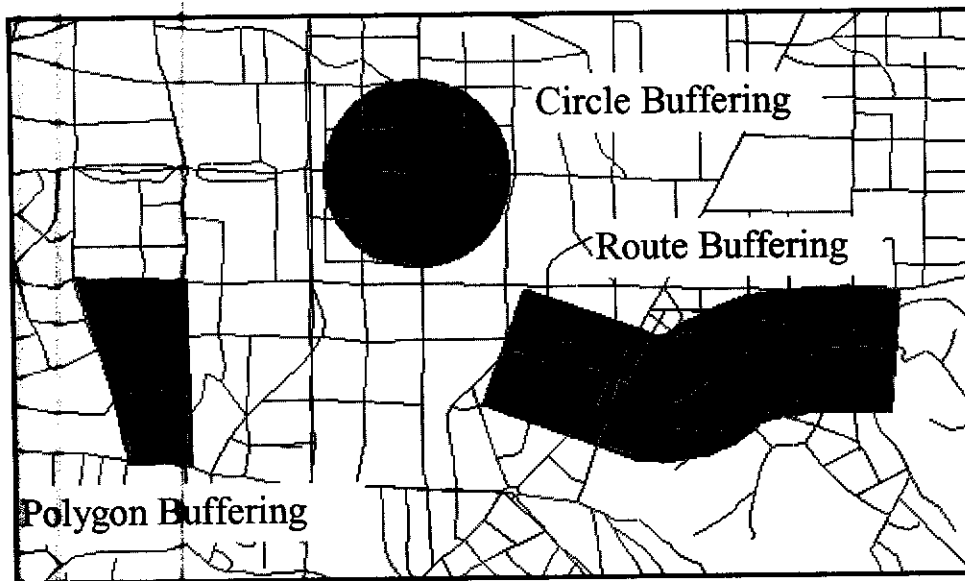


圖 5-3 影響圈域分析功能展示

5-4 主題地圖顯示(Thematic Mapping)

藉由各資料單元不同的屬性值，以不同顏色區別，使規劃者得以清楚辨識各其空間分佈之差異性，如圖 5-4 所示，以不同顏色代表不同的屬性值，以展示之捷運站的服務範圍。

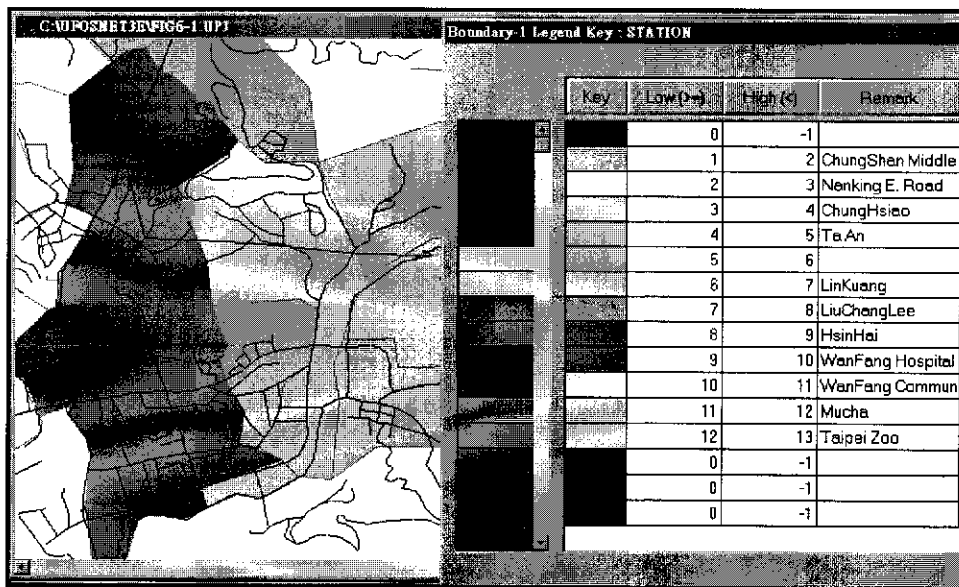
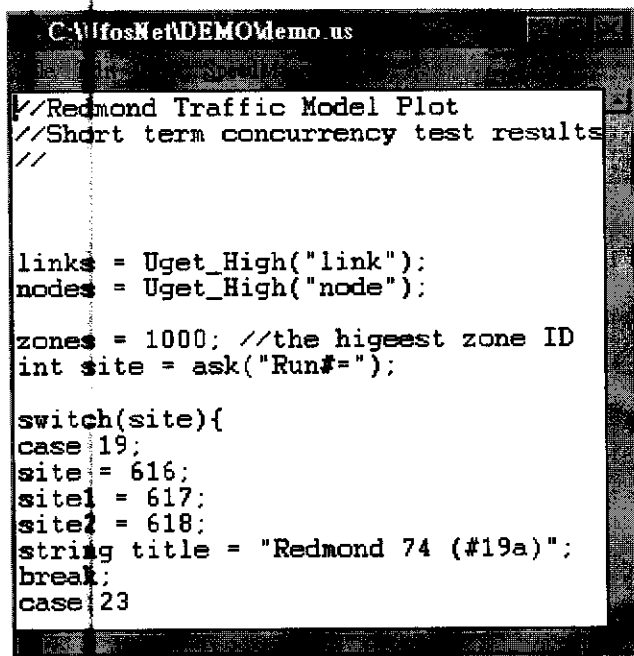


圖 5-4 主題地圖展示

5-4 巨集模組編輯器(Macro Editor)

UFOSNET 提供以 C/C++程式碼為基礎之巨集語言(Macro Scripting Language)，使用者可依分析需求建立巨集模組，使分析程序達到自動化的需求，增進資料庫運算之效率，並使複雜的運算形成模組化的程序。UFOSNET 並依系統特性設計 140 餘種運算函數(Function Calls)，加上各式資料庫之基本運算功能，簡

化程式撰寫之複雜度，另外並提供巨集快速鍵(Speed Macro)功能簡化巨集執行工作，其程式編輯器(Macro editor)如圖 5-5 所示。



```
C:\ufosnet\demo\demo.us
//Redmond Traffic Model Plot
//Short term concurrency test results
//

links = Uget_High("link");
nodes = Uget_High("node");

zones = 1000; //the higeest zone ID
int site = ask("Run#=");

switch(site){
case 19:
site = 616;
site1 = 617;
site2 = 618;
string title = "Redmond 74 (#19a)";
break;
case 23
```

圖 5-5 UFOSNET 巨集程式編輯器

第六章 資料庫計算功能

6-1 分區、節點、節線、點、F 係數及大眾運輸路網計算功能

此選項提供 UFOSNET 各資料表單之資料庫計算功能。每一資料庫計算器包含一組數學函數公式可以靈活的處理其中的每一資料值。計算功能由一組代數語法表示執行，其由主(main)及副(sub)函數組合而成，其中副語法為描述主語法的條件限制。兩者的關係可簡單表示成 if-then 的規則，例如：IF ”副語法”為真，THEN “儲存結果”等於”主語法”，若無副語法的顯示，則表單中所有記錄都將被計算。計算功能亦可跨越不同資料庫之間，如矩陣-分區、點-分區、節線-分區、節線-點、邊界-邊界或是節點-分區。需注意資料庫計算功能在於提供簡單及快速的計算，進階或複雜的計算建議使用 UScript macros 的功能。

注意：若點選了緩衝區域，則計算將侷限於此緩衝區域的資料。

6-2 計算函數(Calculation Functions)

約有 100 種函數可用於資料庫計算，其中一些適用於所有資料庫而某些只適用於特定之資料庫。一個函數可含有零到多個參數，參數可以為常數、字串或是語法，視各函數性質而定，但只有原本資料庫表單中存在的資料可以當作參數。點選函數名稱即會自動顯示函數

相關資訊。

一、一般函數(Common functions)：

- **ABS(a)** 將參數 a 以絕對值表示。
例： $\text{abs}(-1) = 1$
- **ACOS(a)** 將參數 a 以 arc cosine 值表示，a 需在 -1 至 1 之間。
例： $\text{acos}(-1) = 3.14$
- **AND(a,b,...)** 若資料(最多 10 筆)全為非 0，則為真(1)；反之則為假(0)。
例： $\text{and}(1,2,3) = 1$; $\text{and}(1,0,3,4) = 0$
- **ATAN(a)** 將參數 a 以 arc tangent 值表示。
例： $\text{atan}(-1) = 0.46$
- **INSTR(a,b)** 若字串 b 是字串 a 的一部份則為 1，否則為 0。
例： $\text{instr}(\text{"abc"}, \text{"a"}) = 1$
- **PUT1(a)** 將值 a 置入第一緩衝陣列(buffer array)內，且可利用函數 GET(1)再取出。
例： $\text{put1}(3) + \text{get}(1) = 6$
- **RAND()** 從 0 到 32767 中隨機抽取虛擬的亂數，此函數不需給予參數。

二、分區函數(Zone functions)：

- **AREA()** Boundary - 1 層的分區邊界之面積。

- CENTERX() Boundary - 1 層之分區中心的 X 軸座標。
- TMXROWSUM() 為矩陣-分區的計算，得矩陣中列的總和。
- TMXROW(a) 為矩陣-分區的計算，得到矩陣中給定列編碼為 a 的列總和。此結果可以儲存於分區資料庫中。

三、節點函數(Node functions)：

- PEEKX() 可獲得所有節點之邏輯上的 X 座標。
- TLINE() 可獲得使用某節點之大眾運輸路線數，只要節點編號和大眾運輸路段之 bnode 編號相符合即表示有路線使用了該節點。
- TUSER(a) 可獲得使用某節點之大眾運輸路段(segment)之使用者資料(user data)總和，其中 a 必須為 1 至 6 的數字以表示大眾運輸路段之使用者欄位。

四、節線函數(Link functions)：

- DIST() 可獲得節線兩端點的距離。
- TLINE() 可獲得使用某公路路網節線之大眾運輸路線數。

五、點函數(Point functions)：

- **FREQZBY(a)** 為點-分區之計算。此函數計算以分區編號為基礎的點記錄的頻率，此結果將記錄於分區資料表單中。
- **INZBYP(x,y,a)**可獲得某點所在位置之邊界編號(Boundary ID)，其中 x 與 y 為點座標值，而 a 為邊界編號，若 a 小於等於零，則所有的邊界編號都將被搜尋，反之則只有特定邊界之搜尋作用。這項功能可供進行將點資料加入分區資料前之或得分區編號使用。
- **NEARNODE(a,b)** 可獲得最接近給定座標位置之節點編號，a 及 b 即為點的 x 與 y 座標，此功能可作為路徑計算前需有起始、最終節點編號之預備工作，此外，使用此功能需先載入整個路網。
- **PATH(a,b)**可獲得由 a 節點起始、b 節點結束之最小路徑之成本值。

六、矩陣函數(Matrix functions)：

- **CFLAG()** 可得矩陣的欄位編號。此函數可用於多矩陣的計算。
- **RFLAG()** 矩陣-點的計算。此函數可用於多矩陣的計算。

計算後軟體會進行該記錄簡單的統計與加總工作，包含的項目有：記錄筆數(Objs)、所有記錄總值(Sum)、最小值(Min)、最大值(Max)、平均值(Mean)、

標準差(Std)，但若為字串之計算，則只有 Objs 欄會有有效值出現。

6-3 計算語法

- ‘0123456789’ 按數字鍵將使數字出現於游標所在之位置。
- ‘+*/()’ 按鍵將使數學符號出現於游標所在之位置。
- BACK 退位鍵。
- Sub Expression 為條件設定，只有符合設定條件之記錄才會進入 main expression 之處理。
- Main Expression 此處會產生計算結果並儲存。
- Save Result 選擇資料欄位以儲存 main expression 之計算結果，若此欄空白，則結果不會儲存。
- Clear Expression 清除游標所在欄位之內容。

6-4 迴歸(Regression)

此功能提供兩資料欄位之間簡單的統計工具，迴歸分析可用於分區、節點、節線的資料庫中。簡單的線性迴歸可以下式表現：

$$Y = a + bX + e$$

6-5 直方圖(Histogram)

此功能可繪出給定組間距之資料直方圖。

6-6 節點進入量(Node Entering)

給定節線資料欄位來源並據以計算某節點之進入
交通量。

第七章 矩陣

從 3.1 版開始，矩陣檔已擴大為包含任何維數 (Dimension) 的多重矩陣形式。矩陣的數目可於 System Configuration 對話方塊的矩陣檔中定義。每一矩陣檔可包含 20 種矩陣形式，並將於後討論。新的矩陣能提供更簡單及有力的功能來儲存及處理矩陣資料。

注意：所有矩陣值以 4 位元之浮點形式儲存。

7-1 Matrix Types

7-2 Defining Matrix

7-3 Editing Matrix

7-4 Calculating Matrix

7-5 Import/Export Matrix

7-6 Copy Matrix

7-7 Site Matrix

7-8 Matrix Statistics

7-9 Expanding Matrix

7-10 Compressing Matrix

7-11 Desire Lines

7-12 Save Turn Penalty Matrix

7-1 矩陣形式(Matrix Types)

軟體內建 20 種矩陣形式且每一形式被定義一種單獨的代號。當運用時，唯有適當的矩陣形式才能被引用。此外，每一種矩陣已預先被定義其維數限制。

■ 0 - MOTHR： 其他形式

此形式可提供任何行、列的組合，通常用來儲存尚未定義之應用型或臨時的使用。

■ 1 - MPATT： 旅次產生-吸引表

此形式矩陣用來儲存分區間旅次產生-吸引表，為行列數目為交通分區數的方陣形式。MPATT 形式矩陣主要於旅次分佈及運具分配過程使用。

■ 2 - MODTT： 旅次起-迄表

此形式矩陣用來儲存分區間旅次起-迄表，為行列數目為交通分區數的方陣形式。MODTT 形式矩陣主要於交通量指派過程使用。

■ 3 - MTIME： 旅次時間表

此形式矩陣用來儲存分區間旅次時間表，為行列數目為交通分區數的方陣形式。MTIME 形式矩陣主要於旅次分佈及交通量指派過程使用。

■ 4-MPROD：產生陣列

此形式矩陣用來儲存分區之總產生量，為MPATT矩陣之列總和為 $N \times 1$ 形式矩陣。MPROD形式矩陣主要於旅次分佈過程使用，如圖7-1所示。

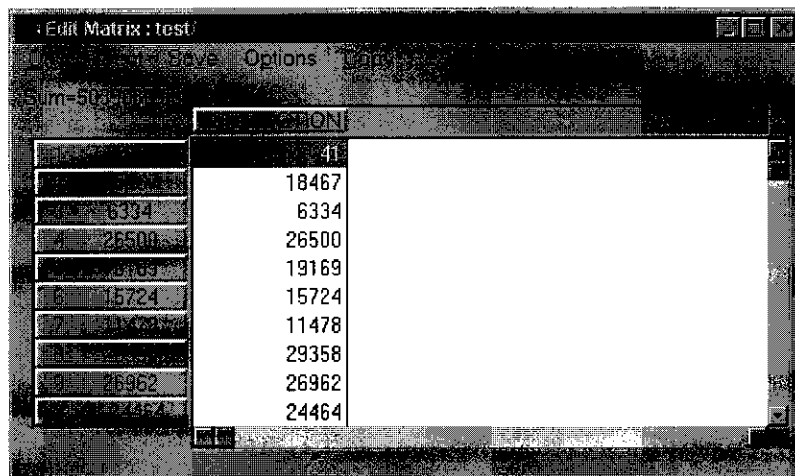
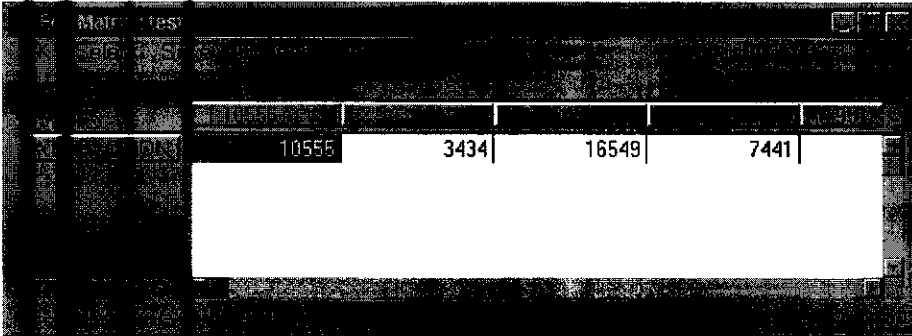


圖 7-1 MPROD 矩陣

■ 5-MATTR：吸引陣列

此形式矩陣用來儲存分區之總吸引量，為MPATT矩陣之行總和為 $1 \times N$ 形式矩陣。MATTR形式矩陣主要於旅次分佈過程使用，如圖7-2所示。

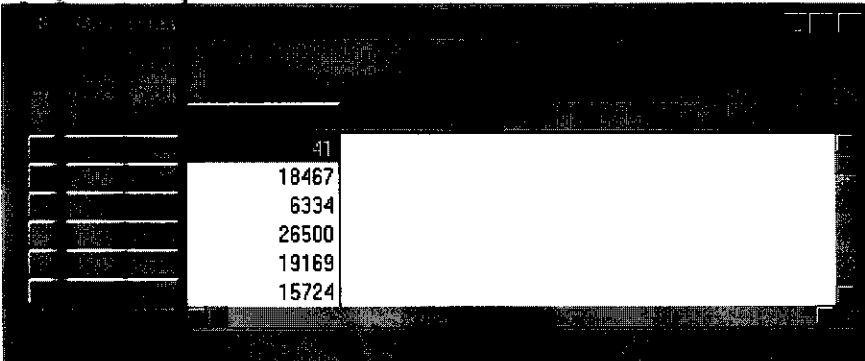


10555	3434	16549	7441
-------	------	-------	------

圖 7-2 MATTR 矩陣

■ 6-MORIG：起點陣列

此形式矩陣用來儲存分區之總起點量，為MODTT矩陣之列總和為 $N \times 1$ 形式矩陣。MORIG形式矩陣主要於旅次分佈過程使用，如圖7-3所示。



41
18467
6334
26500
19169
15724

圖 7-3 MORIG 矩陣

■ 7-MDEST：迄點陣列

此形式矩陣用來儲存分區之總迄點量，為MODTT矩陣之列總和為 $1 \times N$ 形式矩陣。MDEST形式矩陣主要於旅次分佈過程使用，如圖7-4所示。

FROM	TO	MDEST	SUM
10555	3434	16549	7441

圖 7-4 MDEST 矩陣

■ 8-MSPAS：散佈矩陣

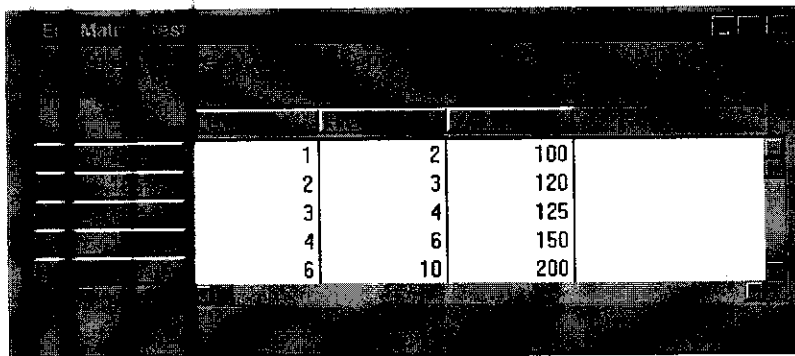
此形式矩陣用來儲存各分區之間的起迄組合量，為 $N \times 3$ 形式矩陣，如圖7-5所示。MSPAS矩陣主要於期望線繪圖及點對點旅次指派（未來補充功能）時使用。

FROM	TO	MSPAS	SUM
1	3	10	20
20	1	30	1
1	7	14	2
2	8	13	3
3	8	11	3

圖 7-5 MSPAS 矩陣

■ 9-MLOOK：條列表

此形式矩陣儲存某區間範圍的數值，為 $N \times 3$ 形式矩陣，如圖7-6所示。



1	2	100
2	3	120
3	4	125
4	6	150
6	10	200

圖 7-6 MLOOK 矩陣

- **LOW** - 區間下限。
- **HIGH** - 區間上限。
- **VALUE** - 若數值大於等於LOW值，而又小於HIGH值，則該數值將被存於該欄位。

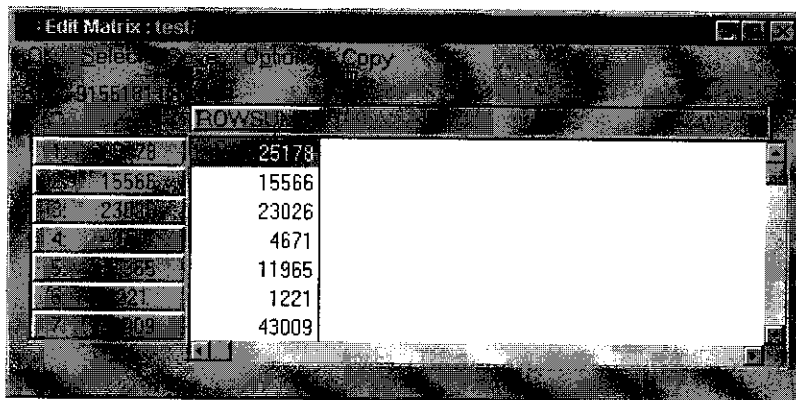
MLOOK矩陣形式主要於計算LOOKUP函數時使用。

■ 10 - MKFAC: K-Factor 表

此形式矩陣用來儲存K-Factor值，為行列數目為交通分區數的方陣形式。此形式矩陣通常於旅次分佈過程中使用。

■ 11 - MROWS: 列總和陣列

此形式矩陣用來儲存任何矩陣的列總和，為Nx1的矩陣形式，如圖7-7所示。MROWS矩陣主要於計算ROWSUM函數時使用。

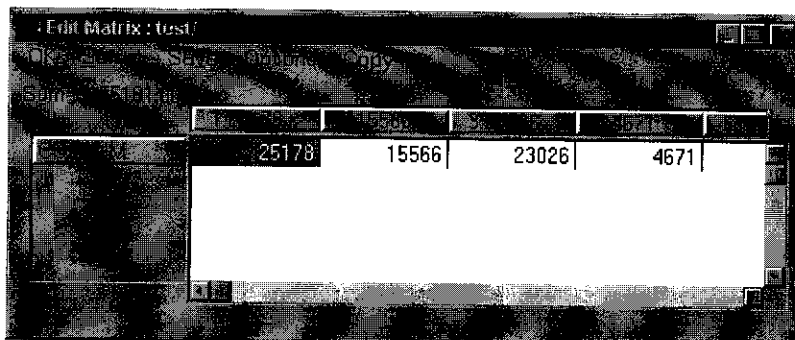


Row	Value
1	25178
2	15566
3	23026
4	4671
5	11965
6	1221
7	43009

圖 7-7 MROWS 矩陣

■ 12 – MCOLS：行總和陣列

此形式矩陣用來儲存任何矩陣的行總和，為 $1 \times N$ 的矩陣形式，如圖 7-8 所示。MCOLS 矩陣主要於計算 COLSUM 函數時使用。

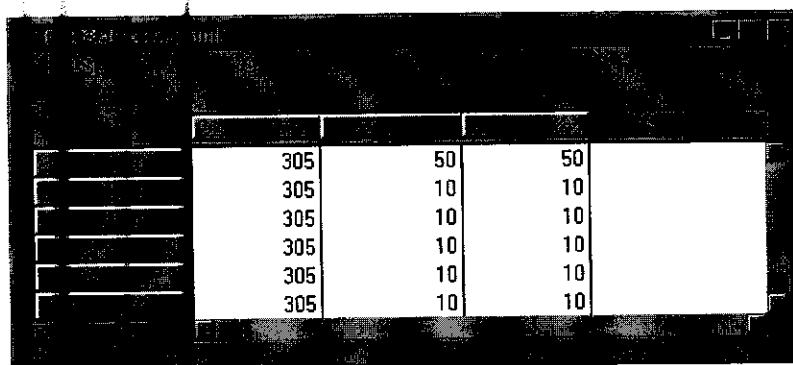


25178	15566	23026	4671
-------	-------	-------	------

圖 7-8 MCOLS 矩陣

13-MEQEX: 擴大等值因子

此矩陣形式用來儲存將一維數較小矩陣擴張為較大矩陣之等值放大因子值，由於此過程為將一分區切割為許多小分區，故此因子也被稱為分割因子。分割因子於行或列皆可使用，MEQEX為 $N \times 3$ 矩陣，如圖7-9所示。



305	50	50
305	10	10
305	10	10
305	10	10
305	10	10
305	10	10

圖 7-9 MEQEX 矩陣

- **OLD ID**: 已存在之分區代碼。
- **ROWFAC%**: 對已存在分區之列的分割因子（百分比）。
- **COLFAC%**: 對已存在分區之行的分割因子（百分比）。

MEQEX 矩陣在矩陣擴大模式時使用。

■ 14 – MEQSQ： 縮小等值因子

此矩陣形式用來儲存將一維數較大矩陣縮小為較小矩陣之等值縮小因子值，為 $N \times 1$ 矩陣，如圖7-10所示。

MEQSQ矩陣在矩陣縮小模式時使用。

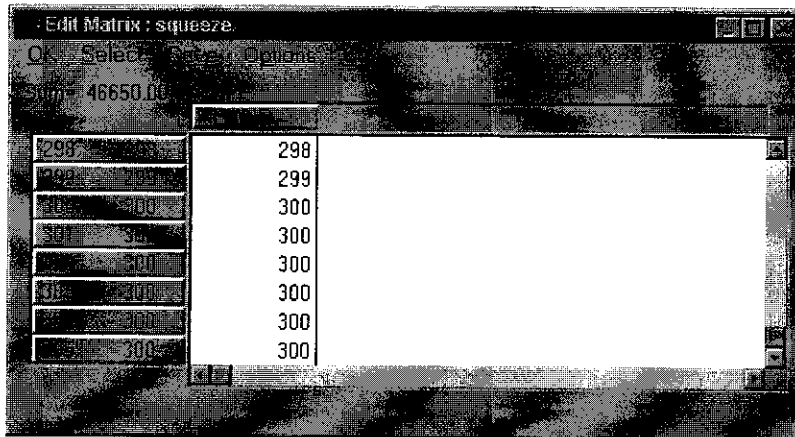


圖 7-10 MEQSQ 矩陣

■ 15 – MMODE： 大眾運輸模式表

此形式矩陣用來儲存大眾運輸模式間的penalty或是轉移資料，為 10×10 矩陣。

■ 16 – MTURN： 轉向移動表

此形式矩陣儲存交叉口之轉向資料為 $N \times 5$ 矩陣，如圖7-11所示。

	7554	7641	7633	233	1
	7554	7641	7640	520	2
	7554	7640	7641	280	0
	7554	7640	7633	35	0
	7554	7559	7641	69	0

圖7-11 MTURN矩陣

- **PIVOT NODE**：交叉口代碼。
- **FROM NODE**：上游路口代碼。
- **TO NODE**：下游路口代碼。
- **TOTVOL**：此轉向之總量。
- **SZ VOL**：選擇分區於此轉向之量。

MTURN矩陣用於公路指派以及交叉口容量分析。

■ 17 - MTPEN：轉向損失表

此形式矩陣儲存節點之轉向損失，為 $N \times 4$ 矩陣，通常用於公路指派，如圖7-12所示。

PIVOT NODE	FROM NODE	TO NODE	PENALTY
1523	8000	8001	99
1523	8001	8000	99
1824	8007	8031	99
1824	8008	8007	99
1824	8008	8021	99

圖 7-12 MTPEN 矩陣

- **PIVOT NODE**：交叉口代碼。
- **FROM NODE**：上游路口代碼。
- **TO NODE**：下游路口代碼。
- **PENALTY**：此流向會增加的秒數或是轉向妨礙。

■ 18 – MCOST：分區間旅行成本表

此形式矩陣儲存分區間旅行成本資料，如SOV成本、HOV成本或大眾運輸成本等，為 $N \times N$ 矩陣，通常用於運具選擇模式中。

■ 19 – MSHAR：分區間運具分配表

此形式矩陣儲存來自運具選擇模式之分區間運具分配資料，為 $N \times N$ 矩陣，通常用於運具選擇模式中。

■ 20-MPARK: 分區間轉乘選擇表

此形式矩陣儲存來自轉乘選擇模式之分區間轉乘數資料，為 $N \times N$ 矩陣，通常用於轉乘選擇模式中。

7-2 矩陣定義(Defining Matrix)

當接受使用某矩陣之前，必須開啟一給定某值的新矩陣，如此才能在硬體中預留處理空間。定義及開啟一個矩陣必須使用如圖 7-13 的對話方塊。

Matrix Name	Value
M1: Tpen: Tern Penalties	
M2: ALL 94 T J: 1994 D	
M3: AON Time: Inter zone	
M4: ADJ Time: Inter zone	
M5: NF 94 KJS: =m108 to	305
M6: WF 94 KJS: =m109 to	305
M7: DF 94 KJS: =m110 to	
M8: KFAC KJS: KJS K F	
M9: NF 12 KJS: =m148 to	
M10: WF 12 KJS: =m149 to	
M11: DF 12 KJS: =m150 to	
M12: expand:	0.00
M13: test:	
M14: squeeze:	617485.88
M15: test1:	9155189.87

圖 7-13 矩陣定義對話方塊

每一矩陣有一個標題來描述矩陣的細節。該敘述

包含下列資訊：

- **Name**：關於該矩陣的簡短敘述（最多 10 字元）。
- **Remark**：關於該矩陣的詳細敘述（最多 100 字元）。
- **Type**：矩陣型式代碼（0 到 20）。
- **Default**：矩陣之初始值。
- **Read Only**：若矩陣為唯讀，當資料被覆蓋時將會發出警告訊息。
- **Rows**：矩陣的列數。
- **Cols**：矩陣的行數。
- **Min**：矩陣最小值。
- **Max**：矩陣最大值。
- **Sum**：矩陣總值。

當矩陣定義之對話方塊開啟，所有可以取得的矩陣都會列示於方塊中，點選表列的矩陣將會立即展示其標題資訊，而之後的動作可以進行選擇、增加、刪除、更改維數或刪除矩陣。

7.2.1 選擇(SELECT)

點此按鍵可複製反白的矩陣至暫存緩衝記憶體中以進行其他動作，如：矩陣編輯、矩陣轉入/轉出、初設轉向表、初設轉向損失表等。

7.2.2 建立新矩陣(NEW)

按下此鍵可開啟一新的矩陣，建議步驟如下：

- 選擇矩陣形式
- 調整矩陣之行、列數，某些特定矩陣其行列數為固定值，無法變更
- 為矩陣命名
- 加註說明(非必要)
- 設定預設值(非必要)

7.2.3 更新矩陣(UPDATE)

按下此鍵可更新矩陣資訊，包括：

- 矩陣形式 (僅適用於維度相同之矩陣)
- 矩陣名稱

- 矩陣註解
- 預設值
- 唯讀標記

7.2.4 調整矩陣維度(REDIM)

按下此鍵可調整矩陣維度，放大矩陣時，原有各矩陣元素值將被保留，而新增元素位置將以預設值填入。若縮小矩陣，則部分矩陣元素值將隨矩陣元素一起清除。使用此功能改變矩陣形式時，所有原素質將還原成預設值。

7.2.5 清除矩陣(DELETE)

按下此鍵將使矩陣自資料庫中清除，所有資料值將一併去除。

7.2.6 還原初值(INITIALIZE)

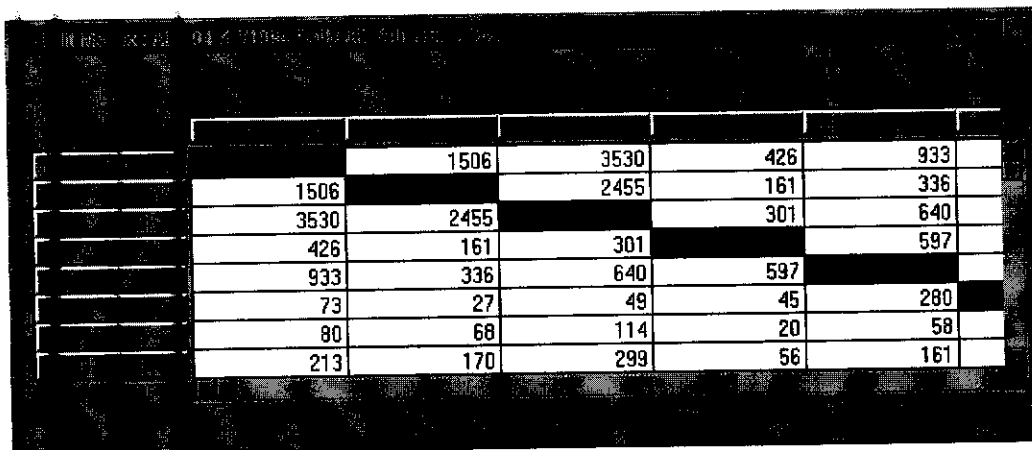
按下此鍵將使矩陣各元素還原成預設值。

7.2.7 關閉矩陣設定功能(CLOSE)

按下此鍵將關閉矩陣及設定功能對話方塊。

7-3 矩陣編輯

矩陣編輯器如圖 7-14 所示，其工作環境如同一般試算表。



The screenshot shows a window titled '矩陣編輯器' (Matrix Editor) with a grid of data. The grid has 10 rows and 6 columns. The data is as follows:

		1506	3530	426	933
	1506		2455	161	336
	3530	2455		301	640
	426	161	301		597
	933	336	640	597	
	73	27	49	45	280
	80	68	114	20	58
	213	170	299	56	161

圖 7-14 矩陣編輯器

上圖之矩陣編輯器包含下列各項編輯功能：

- OK：編輯完成後離開。
- SELECT：選擇特定矩陣檢視、或開啟編輯對話視窗。
- SAVE：將所有改變存檔。

■ **OPTIONS**：附屬編輯功能，包括：

- 到最上列(GoTo Top)
- 到最下列(GoTo Bottom)
- 分離各行/列以便檢視(Separation Lines On/Off)
- 每 5 行以不同顏色標記以便檢視(Color Every 5th Line On/Off)
- 列游標開啟關閉(Row Cursor On/Off)
- 設定欄寬(Set Width)
- 設定小數位數(Set Decimal)
- 對角元素以顏色標記(Color Diagonal On/Off)

- **COPY**：將目前矩陣畫面複製至剪貼簿，以便其他文書軟體使用。

7-4 矩陣運算

矩陣運算功能不僅應用於單一矩陣，亦可應用於多矩陣之運算。

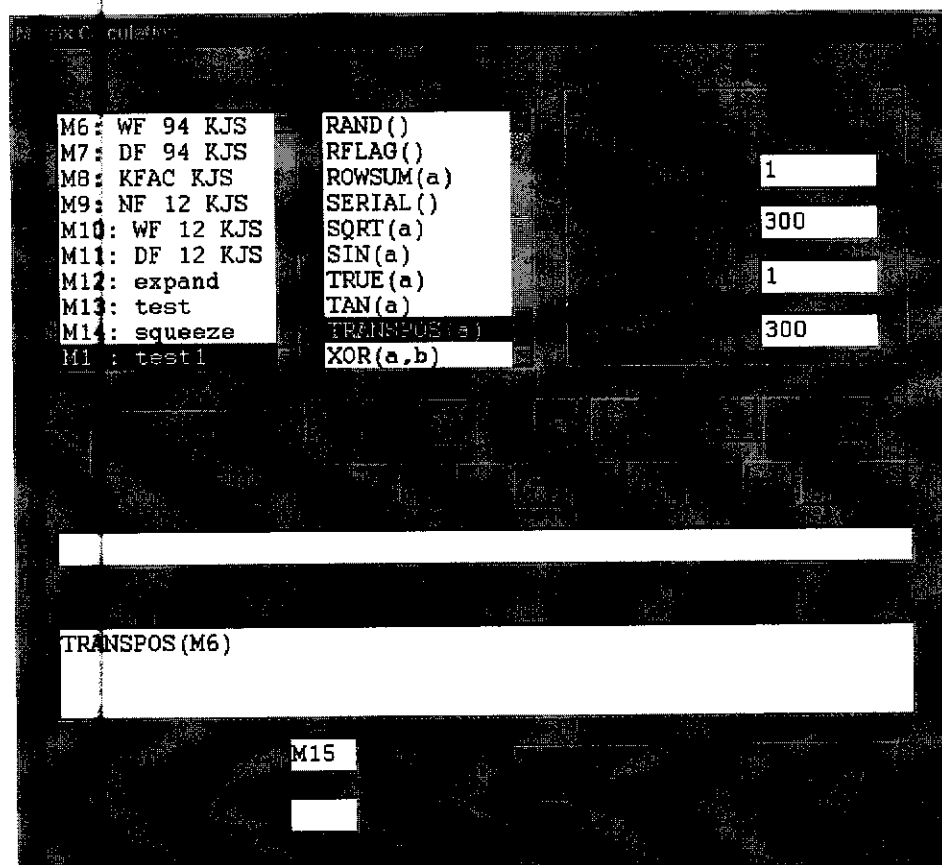


圖 7-15 矩陣運算對話視窗

矩陣運算對話視窗功能包括：

- **Matrices**：列出目前所有可應用矩陣與其編碼。
- **Functions**：內建函數運算功能，括號內為其所需之引數。
- **Ranges**：設定矩陣運算範圍。
- **Sub Expression**：設定運算條件，若符合該條件則提供予以計算處理。
- **Main Expression**：設定運算方式，可結合內建函數與相關運算元進行運算。

- **Clear**：清除運算設定，包括 Sub Expression 及 Main Expression。
- **Output Matrix**：選定矩陣儲存輸出結果。

建議運算步驟設定程序如下：

- 選定運算條件(Sub Expression)或運算方式(Main Expression)。
- 選擇所需之內建函數或運算元。
- 設定所需運算矩陣位置。
- 選定結果輸出之矩陣，並設定輸出範圍。
- 按下計算鍵開始運算

7-5 矩陣轉出(Export)與轉入(Import)

這項功能主要文字檔(*.txt)提供矩陣轉出與轉入，以增加使用彈性。其文字檔格式包括：

- 1.矩陣形式為：8(MSPAS), 9(MLOOK), 13(MEQEX), 16(MTURN), 17(MTPEN)，第一行寫入格式、第二行註解，第三行起條列各元素預設值。

Tpen, 200, 4, 200, 17, 0.000000 (Header 1: 名稱,
列數, 欄數, 列數, 形式名, 預設值)

Turn Penalties (Header 2: 註解)

1523, 8000, 8001, 99.000000
1523, 8001, 8000, 99.000000
1824, 8007, 8031, 99.000000
1824, 8008, 8007, 99.000000
1824, 8008, 8021, 99.000000
1824, 8008, 8031, 99.000000
1824, 8021, 8007, 99.000000
1824, 8021, 8008, 99.000000
1824, 8021, 8031, 99.000000
1824, 8029, 8030, 99.000000
1824, 8030, 8029, 99.000000
1824, 8031, 8007, 99.000000
1824, 8031, 8008, 99.000000
1824, 8031, 8021, 99.000000
1824, 8007, 8008, 99.000000
1824, 8007, 8021, 99.000000
2575, 8024, 8025, 99.000000
2575, 8025, 8024, 99.000000
2890, 8027, 8004, 99.000000

.....

2.其他矩陣形式：第一行寫入格式、第二行註解，第三行起條列各元素預設值。

ALL 94 KJ, 10, 10, 99, 2, 0.0000000 (Header 1:名稱,列數,欄數,資料筆數,形式名,預設值)

1994 Daily All Veh Trips, O&D(Header 2: 註解)

1, 1, 8380.479492

1, 2, 1505.514648

1, 3, 3530.317139

1, 4, 426.318787

1, 5, 933.471924

1, 6, 72.633904

1, 7, 80.183197

1, 8, 213.276993

1, 9, 694.099121

1, 10, 228.580597

2, 1, 1505.514648

2, 2, 3919.717773

2, 3, 2455.020508

2, 4, 161.073807

2, 5, 336.138611

2, 6, 27.113001

2, 7, 68.311798

2, 8, 170.027496

2, 9, 606.361023

2, 10, 167.314194

.....

7.5.1 矩陣轉入

其步驟包括：

- 選定一已定義之矩陣，以載入轉入矩陣，其大小需與來源矩陣相同。
- 選定矩陣檔案(.txt)轉入。
- 視需要更新矩陣名稱及註解。

7.5.2 矩陣轉出

其步驟包括：

- 選擇欲轉出矩陣。
- 選擇一目標文字檔(.txt) 以載入轉出矩陣。

7-6 複製矩陣

此功能提供使用者複製矩陣元素群至同一矩陣不同位置或另一矩陣。

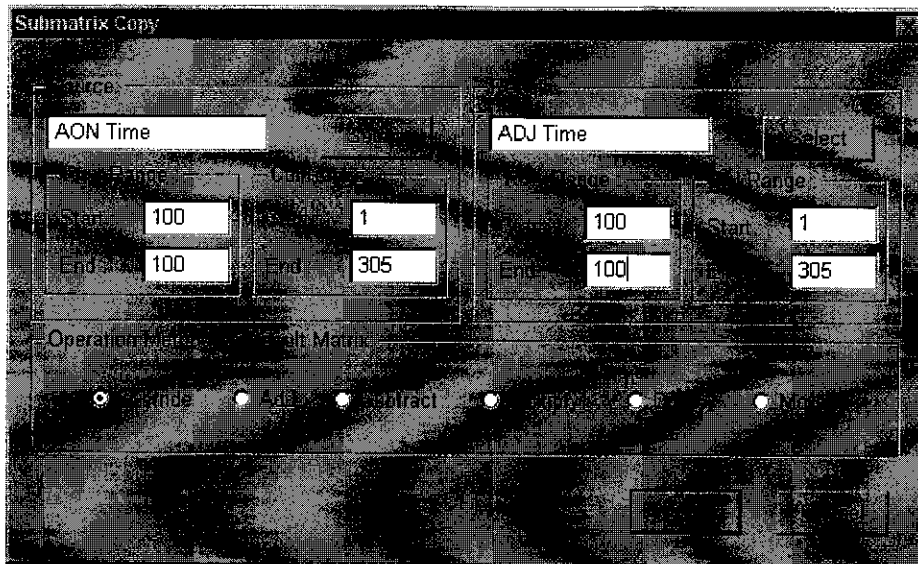


圖 7-16 矩陣對話方塊功能說明

7.6.1 來源矩陣(SOURCE)

- 選定矩陣來源。
- 選定欲複製之行列範圍。

7.6.2 複製結果儲存(RESULT)

- 選擇結果儲存矩陣名稱。
- 選定欲儲存之行列範圍。

7.6.3 運算種類(OPERATION)

- **Override**：覆寫功能，來源矩陣元素值將覆寫至目標矩陣。
- **Add**：加入，來源矩陣元素值將加至目標矩陣對應元素，行程新值。
- **Subtract**：減掉，目標矩陣元素將減去來源矩陣元素值。
- **Multiply**：乘積，來源矩陣與目標矩陣元素相乘。
- **Divide**：相除，目標矩陣元素將除以來源矩陣元素值。
- **Move**：移動，來源矩陣元素值將移至覆寫至目標矩陣，且原矩陣元素重設為 0。

7-7 矩陣位置(Site Matrix)

給一個行列總和的配對，此工具可被使用來計算介於在來源分區上的每一分區配對值的流量交互作用(flow interchanges)。使用步驟如下：

- 選擇要被計算的矩陣，必須是 MPATT 或 MODTT 型式。
- 指定來源分區號碼(Parent Zone ID)。
- 指定要建立的分區號碼，通常是新的分區號碼。
- 指定行的總和為內值域的旅次(Inbound Trips)。
- 指定行的總和為外值域的旅次(Outbound Trips)。

而且，你也可以選擇建立一個減去來源分區的新旅次，不過其值不可為負值，若為負值，則來源分區及新旅次的值將被設定為零。

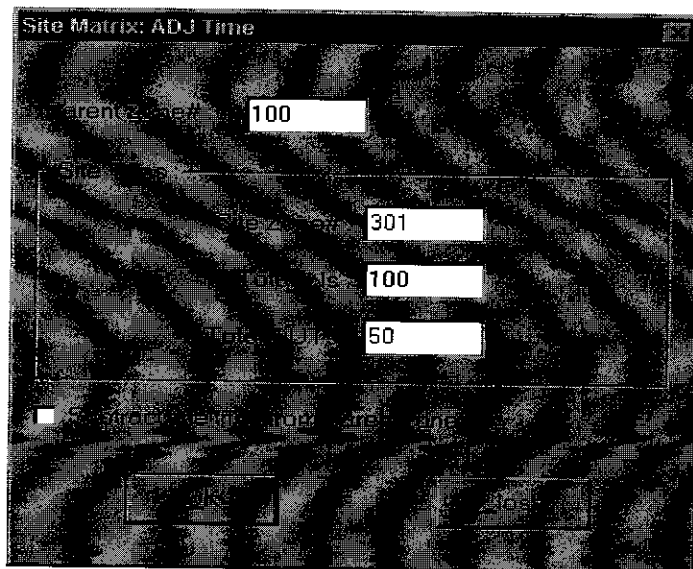


圖 7-17 矩陣位置對話方塊

7-8 矩陣統計(Matrix Statistics)

此工具則可用來計算統計矩陣值，同時可以用來繪圖出矩陣內各元素的長條圖，可以自己定義各值的區間，其內定值為各值的平均值，你也可以調整某些矩陣值的區間，如圖 7-18 所示。

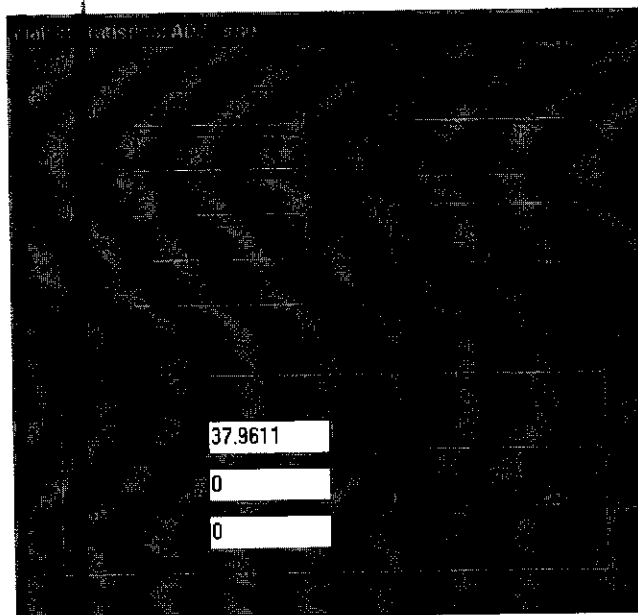


圖 7-18 矩陣統計對話方塊

- Sum - 總計矩陣的所有元素。
- Min - 所有元素中的最小值。
- Max - 所有元素中的最大值。
- Mean - 所有元素中的平均值。
- Std - 所有元素中的標準差。

■ Interval - 用以建立矩陣值的長條圖區間。

■ **Histogram**：用特別的區間值去畫長條圖。

注意長條圖是一種最新的的畫表工具，所以有

Histogram 工具列以補助之。

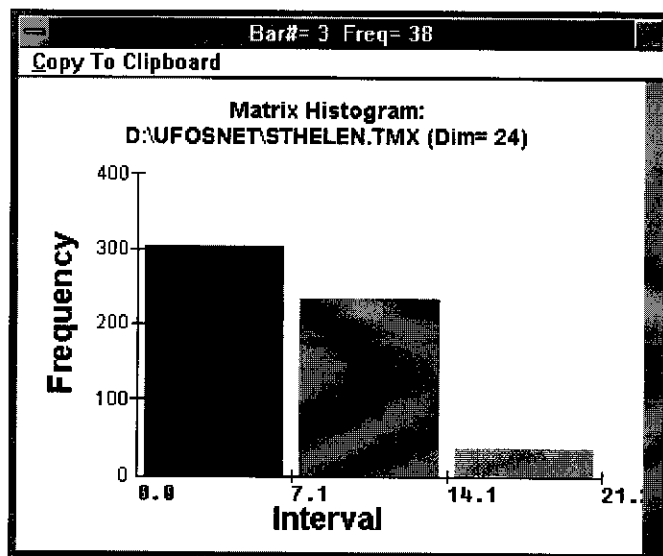


圖 7-19 長條圖功能

7-9 矩陣延伸(Expanding Matrix)

此工具列可用以分割或延伸多維矩陣。矩陣的延伸是要先提供矩陣中各行各列的分割因子(Splitting Factors)。通常，由土地的使用狀況及預定的旅次目的來決定分割因子。

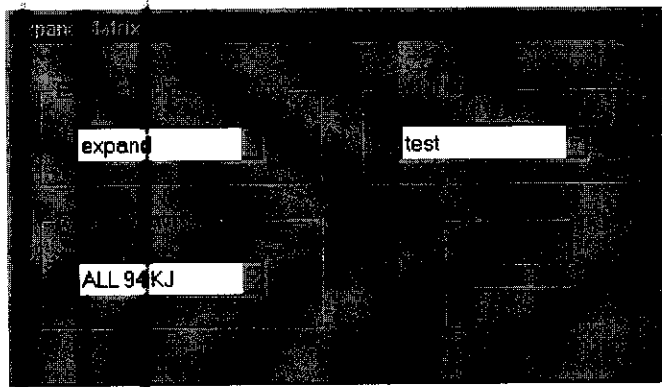


圖 7-20 矩陣延伸對話方塊

- **Expansion Factor Matrix**：選擇包含分割因子的 MEQEX 矩陣，延伸因子矩陣 (Expansion Factor Matrix) 的矩陣數必須和結果矩陣 (Result Matrix) 的矩陣數一致。
- **Source Matrix**：選擇一個要被延伸的矩陣，必須是 MODTT 或 MPATT 型式。
- **Result Matrix**：選擇一個要被輸出的矩陣，必須是 MODTT 或 MPATT 型式。所以，結果矩陣 (Result Matrix) 的列數必須和延伸因子矩陣 (Expansion Factor Matrix) 的列數一致。

7-10 矩陣壓縮(Compressing Matrix)

此工具列可用以壓縮矩陣至少維的矩陣。矩陣的壓縮是將分區號碼 (zone ID) 一致的各矩陣元素給予合併。

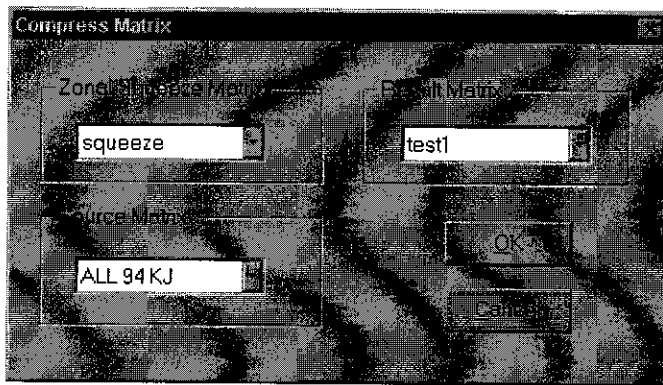


圖 7-21 矩陣壓縮對話方塊

- **Zonal Squeeze Matrix**：選擇包含區內等值 (zonal equivalency) 的 MEQSQ 矩陣，壓縮矩陣 (Squeeze Matrix) 的列數必須和來源矩陣 (Source Matrix) 的列數一致。
- **Source Matrix**：選擇一個要被壓縮的矩陣，必須是 MPATT 或 MODTT 型式。
- **Result Matrix**：選擇一個要被輸出的矩陣，必須是 MPATT 或 MODTT 型式。所以，結果矩陣 (Result Matrix) 的行列數必須配合高分區號碼的壓縮因子矩陣 (Squeeze Matrix) 的行列數。

7-11 期望線 (Desire Lines)

利用此項工具以展示分區間矩陣之流量，這些流量將以直線連接兩個分區組合，而非顯示真實路網的路徑。期望線可以多種方式呈現，如一對一、一對多、多對一及多對多。要使用期望線的功能，必須已載入矩陣、路網及邊界圖層。

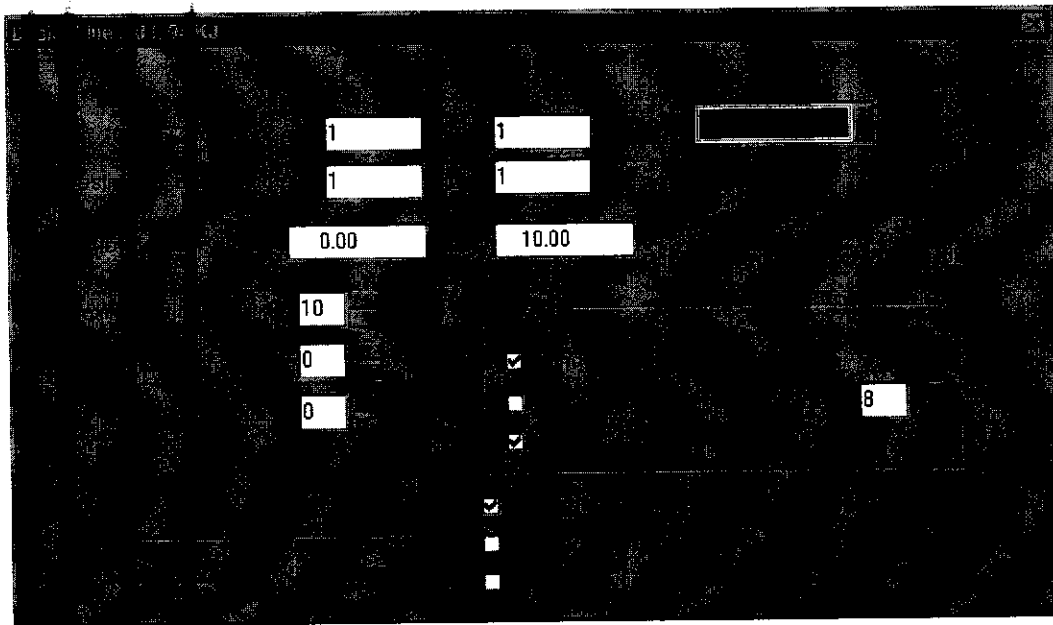


圖 7-22 期望線對話方塊

- **Origin Zone Range:** 指定特定範圍的分區編號作為起點分區。
- **Destination Zone Range:** 指定特定範圍的分區編號作為迄點分區。
- **Flow Value Range:** 指定特定的流量值使得只有該範圍內的流量才被執行。較高的數值將被用以當作展示時流量寬度的分母尺度。
- **Min flow value:** 矩陣中最小的流量值。
- **Max flow value:** 矩陣中最大的流量值。
- **Color:** 選擇流量的展示顏色。
- **Maximum Flow Width:** 選擇螢幕上繪製流量的最大帶寬。此寬度和前面定義的最高值相關。

- **Inbound Radius**：於迄點處選擇緩衝圓的範圍，以避免線與迄點重疊。
- **Outbound Radius**：於起點處選擇緩衝圓的範圍，以避免線與起點重疊。
- **Font Size**：選擇流量值展示的文字尺寸。
- **Bandwidth Style**：流量可以箭頭及帶寬兩種方式展示，其中箭頭方式適合表示流量的方向，而帶寬方式則同時以並排的兩個矩形表示兩個方向流量。若此按鍵被選擇，則表示使用帶寬方式。
- **Show Flow Value**：選擇此按鈕以展示沿此期望線的流量值。
- **Fill Flow**：選擇此按鈕以使流量帶寬或箭頭填滿。
- **Use Boundary Center**：若路網、邊界檔案同時使用中，使用者可以選擇使用邊界中心(Boundary Centers)或分區重心(Centroid)當作起迄點來展示流量值。
- **Boundary Clipping at Origin**：分區邊界可以當作屏柵線使用，故向外的期望線將由邊界開始。此選項只有在選取邊界中心功能時使用。

- **Boundary Clipping at Destination**：分區邊界可以當作屏柵線使用，故向外的期望線將至邊界即停止。此選項只有在選取邊界中心功能時使用。
- **OK**：選此按鈕以繪製期望線，且將開啟一個暫存檔儲存該繪圖之資訊，若原先已存在檔案，使用者可覆蓋舊檔或增加資訊於舊檔中，且所有繪圖將會一直累積。

注意：期望線將會置於「期望線繪製」圖層，使用者可以運用圖層控制功能視需要開啟/關閉圖層。

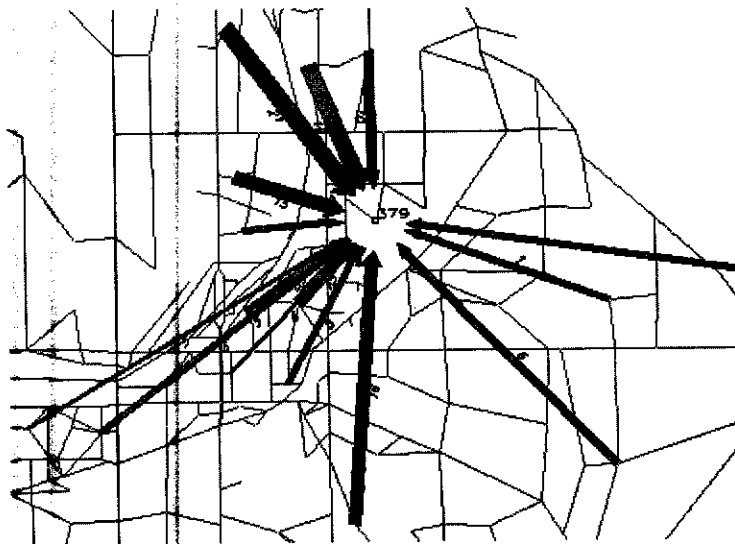


圖 7-23 多對一方式期望線範例

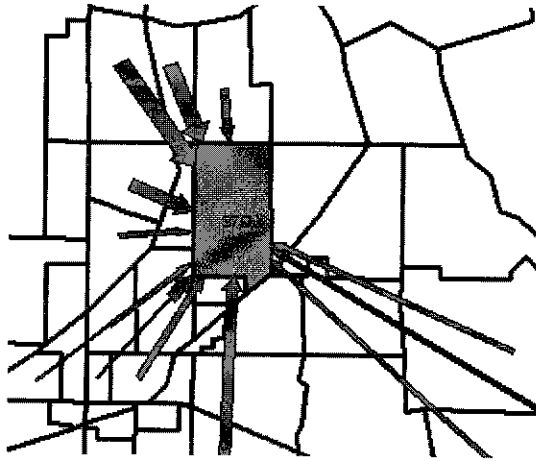


圖 7-24 使用邊界屏柵功能之多對一期望線範例

7-12 轉向損失矩陣儲存(Save Turn Penalty Matrix)


在轉向損失編輯過程中，所有的改變將儲存於暫存區，為了使變更長久保持，必須使用此功能將變更儲存至矩陣資料庫。

儲存轉向矩陣可由下述步驟進行：

- 定義及選擇一個 MTPEN 矩陣以儲存轉向損失。
- 選擇此選單開始進行此步驟。

第八章 大眾運輸路線

8-1 工具箱(Toolbox-Transit)

按快速鍵  即可開啟此工具箱，如圖 8-1 所示。此工具箱包含建立大眾運輸路線(Transit Lines)、編輯大眾運輸路段(Transit Segment)、編輯查詢大眾運輸資料等功能。但此工具箱必須在已建立正確之公路路網的條件下才能正確運作。

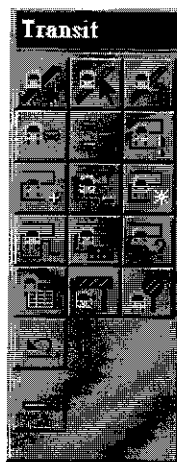



圖 8-1 大眾運輸路線編輯工具箱

8.1.1 運具型式選擇(Mode)

按快速鍵  選取，出現如圖 8-2 的對話方塊。提供大眾運輸路網 10 種的運具選擇，如公車、鐵路、渡輪等，每一大眾運輸路線必須給予其一種運具模式。每種運具必須給予其一個字母長度的代號，同時亦可於其後進行說明，此外，每種運具可給定不同的顏色代表。

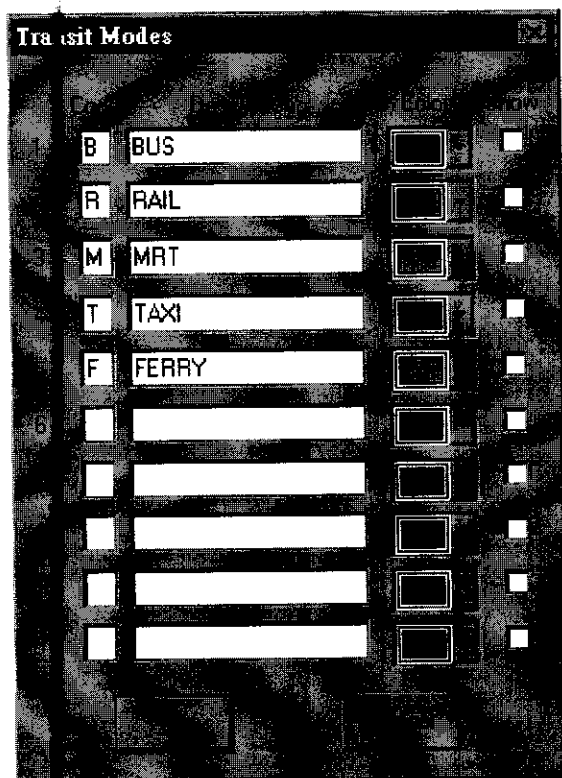



圖 8-2 運具型式選擇對話方塊

8.1.2 車輛型式選擇(Vehicle)

按快速鍵  點選，出現如圖 8-3 之畫面。此功能提供 99 種車輛型式，每一型式包含數種描述營運之屬性可用來計算績效指標。屬性包括：

- **No**：車輛型式（1 至 99）。
- **Name**：最多 20 字元的車輛名稱說明。
- **Mode**：形式代號。
- **Fleet Size**：車隊大小。
- **Total Capacity**：每車之容量（含座位及立位）。
- **Seating Capacity**：每車座位容量。

- **Default Speed**：該型車輛之平均速度。
- **Cost/Length**：單位里程之營運成本。
- **Cost/Hour**：單位小時之營運成本。
- **Energy/Length**：單位里程之能源消耗。
- **Energy/Hour**：單位小時之能源消耗。

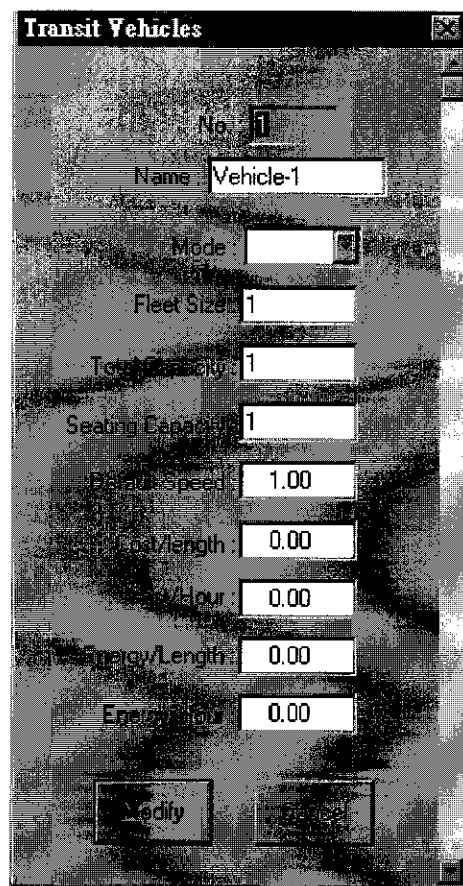




圖 8-3 車輛型式選擇對話方塊

8.1.3 建立新路線(New Line)

按快速鍵  點選後，出現如圖 8-4 之對話方塊。軟體可在已存在的大眾運輸路網上增加新的大眾運輸路線，建立新路線時，必須先進行一些資訊的設定，

之後才可細部修改路線行經路段之屬性資料。需先設定之屬性包括：路線編碼(Line No)、最多 8 個字元的路線標號(Line Label)、運具種類(Line Mode)、車輛型式(Vehicle Type)、班距(Default Headway)、平均營運速度(Default Speed)、單位為分鐘的停站時間(Default Dwell Time)、大眾運輸時間函數指標(Default TTF)、不超過 60 個字元之路線名稱(Line Name)詳細說明。設定完成，選擇  鍵以完成一條路線之編輯，每一路線將以一系列的節點、節線表示，而每一節線將會變成一大眾運輸路段，最後，需選擇 End - Of - Line 鍵以完成大眾運輸路線之編輯。

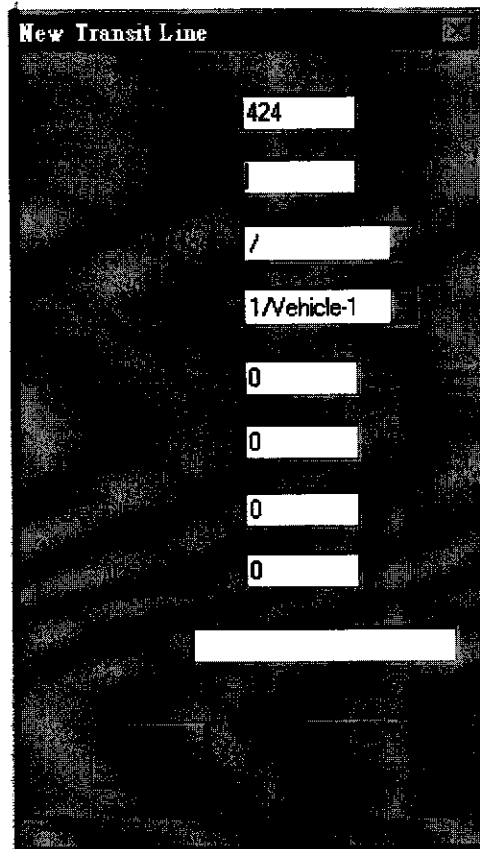




圖 8-4 建立新大眾運輸路線對話方塊

8.1.4 選擇/刪除路線(Select / Delete Line)

按  快速鍵，出現如圖 8-5 之對話方塊。可選擇及顯示一條大眾運輸路線，此功能可以定義顯示的形式，所有的路線都將顯示於對話箱中，包括路線編碼(ID)、標號(Label)、及名稱(Name)，此外，單程(One-way)之路線以"*"號標示於前。此外，本選項可刪除已存在之大眾運輸路線，最後一次刪除之路線可用選擇  鍵之功能來回復。

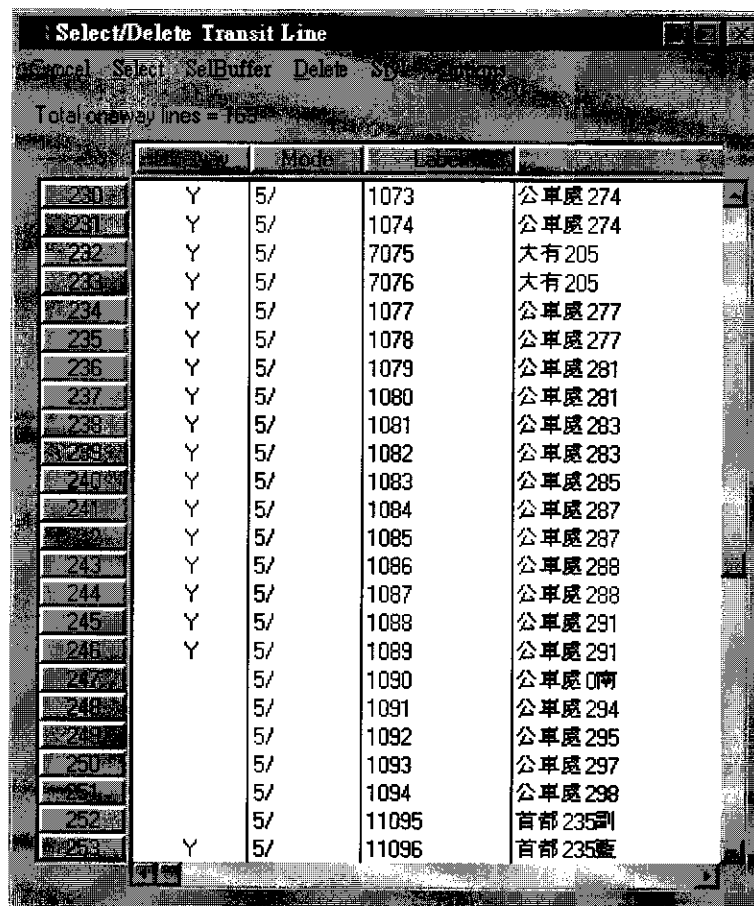


圖 8-5 選擇/刪除路線功能之對話方塊

8.1.5 路線資訊(Line Info)

按  快速鍵，顯示及變更大眾運輸路線屬性資料，如圖 8-6 所示。

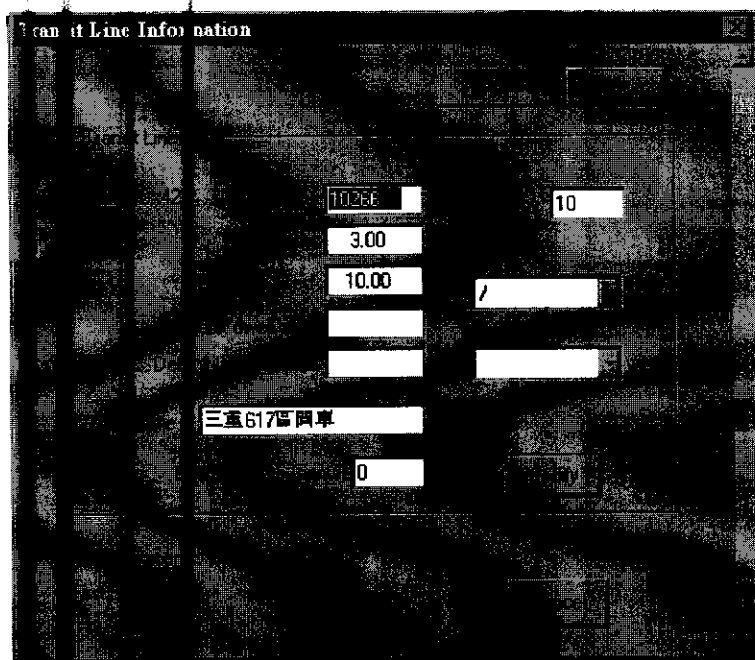




圖 8-6 大眾運輸路線資料

8.1.6 完成編輯(End of Line)


需點選  快速鍵，以完成一條新路線的編輯，否則將無法使該路線編入大眾運輸資料庫中。若該路線為單程(One-way)，則可直接編入資料庫，若為來回(Two-way)，則最終之節點應與起始節點相同。

8.1.7 增加路段(Add Segment)


可點選  快速鍵執行，為更正已存在路線之第

一步驟，需由已存在之節點開始增加新的路段。


8.1.8 移除路段(Remove Segment)

可點選  快速鍵執行，為更正已存在路線之第二步驟，刪除路段為針對現存路段去除連結或區隔成新的節線的方式之一。

8.1.9 重建路線(Rebuild Line)

可點選  快速鍵執行，為更正已存在路線之第三步驟，此功能可重建與檢查新舊路段之間連結的正確性，若無法成功的重建路線，則會出現錯誤訊息告知。

8.1.10 路段資料(Segment Data)

點選  快速鍵，出現如圖 8-7 之對話方塊，可編輯或更正每一路段之屬性資料，藉由點選路段之起始節點並將游標拖曳至終點之節點可以完成選擇路段之動作，若該路段之停靠時間(Dwell Time)為零，即表示該路線於終點節點處未設有停靠站；中途停留時間(Layover Time)通常指該路線第一個起始節點或路段中間的費時；Boarding 與 Exiting 指終點節點之停靠站上下車時間；Volume 欄位存放該路段之承載人數，此外，每一路段尚留 6 個欄位可定義使用。

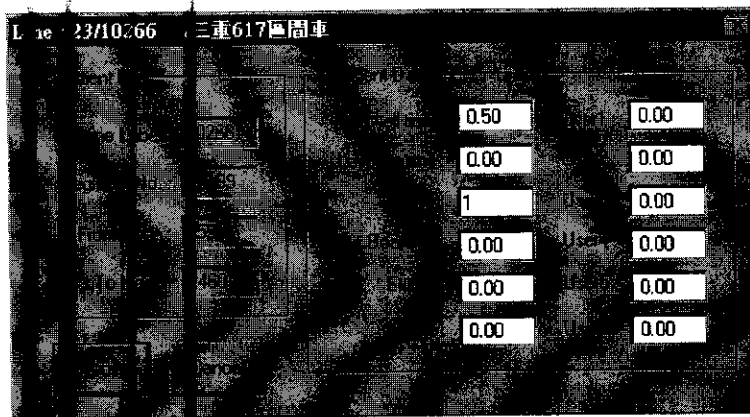



圖 8-7 路段資料編輯對話方塊

8.1.11 顯示形式(Default Style)

可點選  快速鍵執行，大眾運輸路線可以不同的形式展示，包含定義 Color、Offset、Width 及 Font Size，其對話方塊如圖 8-8 所示。

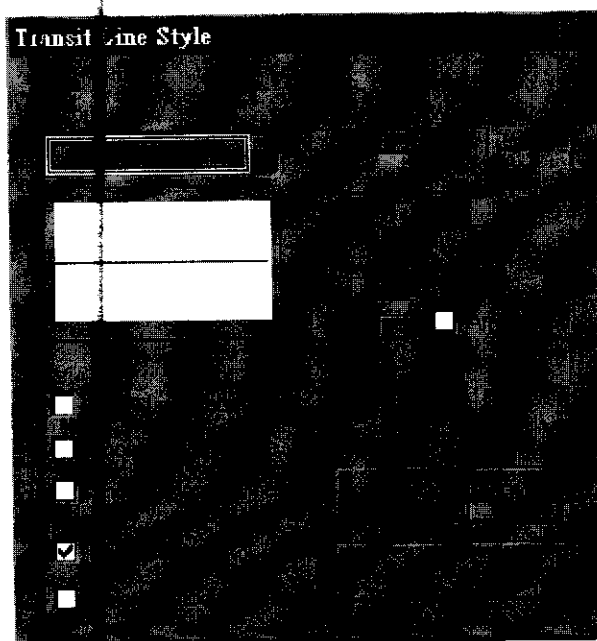



圖 8-8 路線顯示型式對話方塊

8.1.12 大眾運輸時間函數(TTF)


在都市地區，路段之大眾運輸時間會受小汽車交通條件的影響，通常，可以線性因子來表示大眾運輸時間與小汽車時間的關係，TTF 即表示大眾運輸時間與小汽車時間的比值，而大眾運輸時間可用設定之大眾運輸速度來計算。

8.1.13 路段表單(Segment Table)

點選  快速鍵執行，此功能提供表單方式顯示及編輯路段資料。


8.1.14 以節線顯示路線

(Show Lines by Link)


點選  快速鍵執行，可顯示共同使用某節線之所有路線，可由此點選展示其中之大眾運輸路線，該路線將保留於螢幕直到選擇 Clear Screen 鍵為止。

8.1.15 以站位顯示路線

(Show Lines by Stop)

點選  快速鍵執行，可顯示共同使用某站位之所有路線，唯有 Dwelling Time 不為零之路線才會顯示，可由此點選展示其中之大眾運輸路線，該路線將保留於螢幕直到選擇 Clear Screen 鍵為止。

8.1.16 復原(Undo)

點選  鍵可回復上一動作。

8-2 大眾運輸路徑(Transit Path)

由下拉選單之 Tools 部份選擇 Transit Path 即可進入如圖 8-9 之畫面。可使用此功能找尋兩站位之間的最短路徑，首先必須選擇表示主大眾運輸阻抗(Main Transit Impedance)的項目，如距離、時間等，路徑之找尋乃基於每一路段之阻抗及其他停靠時間、停留時間、轉車等候等總成本大小，其中，轉車等候時間大小為將搭乘路線班距之一半，當最短路徑找出後，總成本、轉車次數將會顯示於螢幕，而路徑的計畫會填入設定之檔案中，同時，亦可用此路徑建立給定寬度之路程緩衝區域。

8-3 大眾運輸指派(Transit Assignment)

如同公路指派，大眾運輸指派將大眾運輸旅次指派至大眾運輸路網之上，可由下拉選單之 Models 之 Trip Assignment 選項中選擇 Transit Assignment 進入如圖 8-10 之對話方塊。

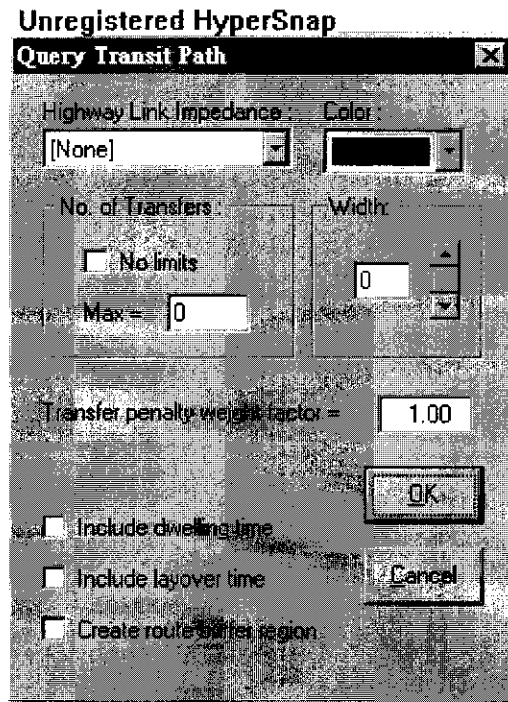


圖 8-9 大眾運輸路徑之對話方塊

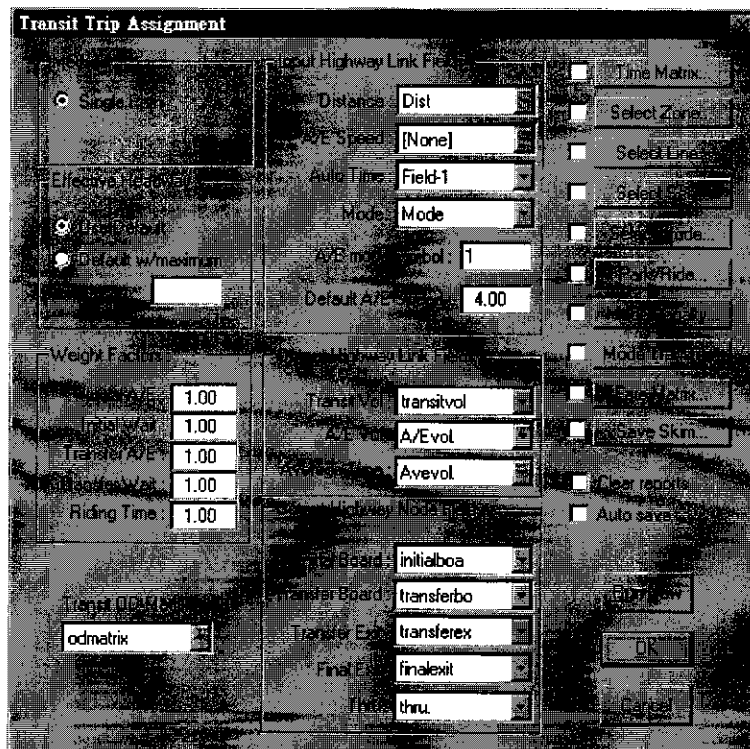


圖 8-10 大眾運輸指派功能對話方塊

大眾運輸旅次以兩分區之最小路徑加以分派，而最小路徑建立於總成本的計算，該成本包括步行時間、乘車時間及等候時間。指派結果將存放於大眾運輸路段資料屬性及總計後置於公路節線表單中。使用者可以運用展示公路路網之工具展示大眾運輸之結果。

- 方法：提供單一路徑(Single Path)指派法。
- 輸入公路節線資料：共同屬於公路路網與大眾運輸路網的節線屬性需輸入，其中包含距離(Distance)、步行速度(Walk Speed)、小汽車時間(Auto Time)、運具形式(Mode)、步行方式(Walk Mode Symbol)、步行速度(Default Walk Speed)。
- 有效班距(Effective Headway)：通常為最大班距之一半時間，亦可引入最大時間之限制來反應大眾運輸使用者平均等候型態之經驗。
- 權數：決定最小路徑之總成本乃由數種時間決定，權重在於反應典型大眾運輸使用者的阻抗函數。這些權重包括：
 1. **Initial walk**：起點至第一大眾運輸停靠站。
 2. **Initial wait**：於第一停靠站之等候時間，定義為有效班距值。
 3. **Transfer walk**：若無法於同一站位直接轉乘所需之步行時間，通常給予較大的權重。
 4. **Transfer wait**：轉乘候車時間，定義為將搭乘路線之有效班距值，具有較大之權重。
 5. **Riding Time**：車內時間。
- 公路節線資料輸出：

1. **Tot. Volume**：儲存使用該節線之所有大眾運輸路線之總流量。
 2. **Ave. Time**：儲存使用該節線之所有大眾運輸路線之平均旅行時間。
- 公路節點資料輸出：
 1. **Boarding**：儲存利用該節點之所有大眾運輸路線之總上車流量。
 2. **Exiting**：儲存利用該節點之所有大眾運輸路線之總下車流量。
 3. **Transfer**：儲存利用該節點之所有大眾運輸路線之總轉乘流量。
 - 時間矩陣：提供儲存不同時間組成之成本函數，包括總步行時間、初始等候時間、總轉車等候時間、總轉車步行時間、總車內時間及總旅行時間。這些時間矩陣可引入運具選擇模式使用。
 - 選擇分區：此功能可獨立儲存各分區產生或吸引之大眾運輸旅次，最多提供 100 個分區選擇，且該分區之旅次起迄將被存入公路節線資料中。
 - 選擇路線：此功能可獨立儲存各路線之大眾運輸旅次，使用該路線之大眾運輸旅次將被儲存至公路節線資料中，此外，該路線各成對分區之旅次數將以旅次表單的形式產生與顯示。
 - 轉運：此功能提供於正規大眾運輸旅次表增加轉運之旅次表，大眾運輸轉運旅次將在上午尖峰於旅次產生分區被指派，下午尖峰於旅次吸引分區指派。所需輸入資料包括：

1. **P/R Zone Range**：詳載分區編號範圍以表示轉運之分區。此編號必須建立於正規之分區編號之上。
2. **Park/Ride Transit Trip Table**：詳載大眾運輸旅次轉運表以顯示成對分區間之轉運需求。
3. **Park/Ride Lot Choice Matrix**：詳載最佳轉運選擇矩陣之檔案，通常為轉運選擇模式之輸出。

第二部份

台北市公車運輸資料庫建立

第九章 資料庫建立簡介

台北都會區公車運輸起迄調查資料庫建立部份可分為下列三項工作：

1.公車旅客起迄站資料庫(車上調查問卷上聯)：

為建立公車使用者之起迄位置及其他相關資訊資料庫，採用車上調查的方式直接獲得，並將調查員在車上調查所回收的問卷上聯進行資料電腦化的處理。本計畫針對台北市聯營公車業者共計 10 家（含公車處及 9 家民營業者，共經營 220 條一般路線）採用全面普查的原則，並將調查工作分成兩階段進行，部分特殊路線則予以篩選剔除。此外，非聯營公車部份亦採抽樣調查，以蒐集其乘客起迄資料，所得資料共計 831 個班次。

2.公車旅客特性資料庫(車上調查問卷下聯與車站問卷)。

為建立公車旅客特性資料庫，除了蒐集由民眾寄回的車上調查問卷下聯之外，本調查亦選擇台北都會區主要的大眾運輸系統車站作為車站問卷的發放地點，以了解台北都會區大眾運輸搭乘者的旅次特性，其內容包含個人基本資料、旅次特性、運輸系統狀況。其中，由乘客對一天中的所有活動填寫，可以知道各運輸系統所轉乘的乘客數，以及各運輸系統間的轉乘情形，以作為未來各運輸系統轉乘路線設計的參考。

3. 家庭訪問活動日記資料庫。

除了旅次起迄與乘客特性資料庫外，本研究計畫亦嘗試以活動理論為基礎，選取適當樣本進行家戶訪問，並將所得之資料電腦化。

本計畫為使調查所得之問卷資料得以效率化、系統化的方式整理，並便於台北市公車地理資訊系統資料庫之整合，結合 Visual Basic、Delphi 程式語言與 Microsoft ACCESS 資料整理系統，構建台北市公車旅次起迄調查資料整理程式 OD1、OD2、ODOB、ODHOME 與 ODTRIP。OD1、OD2 為公車旅客起迄調查資料輸入整理系統，ODOB 適用於處理旅客特性資料之登入工作；ODHOME 及 ODTRIP 則用於處理家庭訪問調查問卷資料。圖 9-1 為 OD1 資料輸入畫面；圖 9-2 為 OD2 資料轉接畫面；圖 9-3 與圖 9-4 為 ODOB 旅客特性資料輸入畫面；圖 9-5 及圖 9-6 為 ODHOME 及 ODTRIP 家庭訪問問卷資料輸入畫面；圖 9-7 則為系統架構與台北市公車地理資訊系統。

起迄點輸入

工讀生編號: 999 問卷批號: 788

路線: 222 時間: 5 月: 29 日: 1645 時

發出問卷份數: 121 回收問卷份數: 118

拒填人數: 07

開始輸入 離開視窗

起點 迄點 次數 資料

02 06 01

已輸入 6 份問卷

圖 9-1 OD1 資料輸入畫面

轉換資料

輸入文字檔名: 036226.TXT

開始轉換 離開

選擇檔案

圖 9-2 OD2 資料轉換畫面

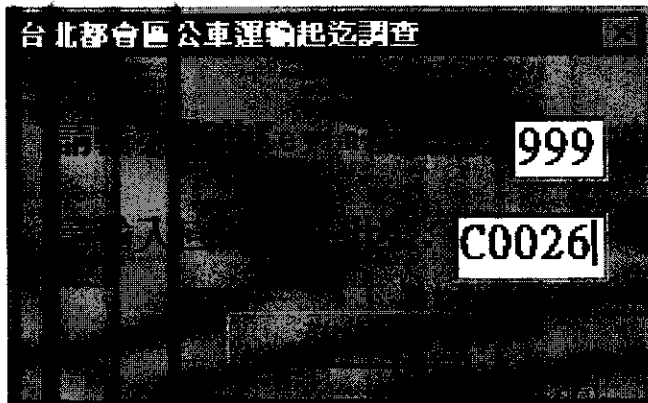


圖 9-3 ODOB 資料輸入畫面之一

台北都會區公車運輸起迄調查問卷輸入

119297

1 1 0

1 010000

0101000

2 1

300 1000

1

1300 2

3 5

0 0 0

3 10

100 10

0 10

100 10 20

XXX XX XX

XX XX XX

0529

1

73	604	176	XXX	XXX	XXX	117	109	XXX	XXX	5	24
90	604	117	109	XXX	XXX	176	XXX	XXX	XXX	3	24
XXX	XXXXXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	X	XXX
XXX	XXXXXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	X	XXX
XXX	XXXXXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	X	XXX
XXX	XXXXXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX	X	XXX

圖 9-4 ODOB 資料輸入畫面之二

圖 9-6 ODTRIP 資料輸入畫面

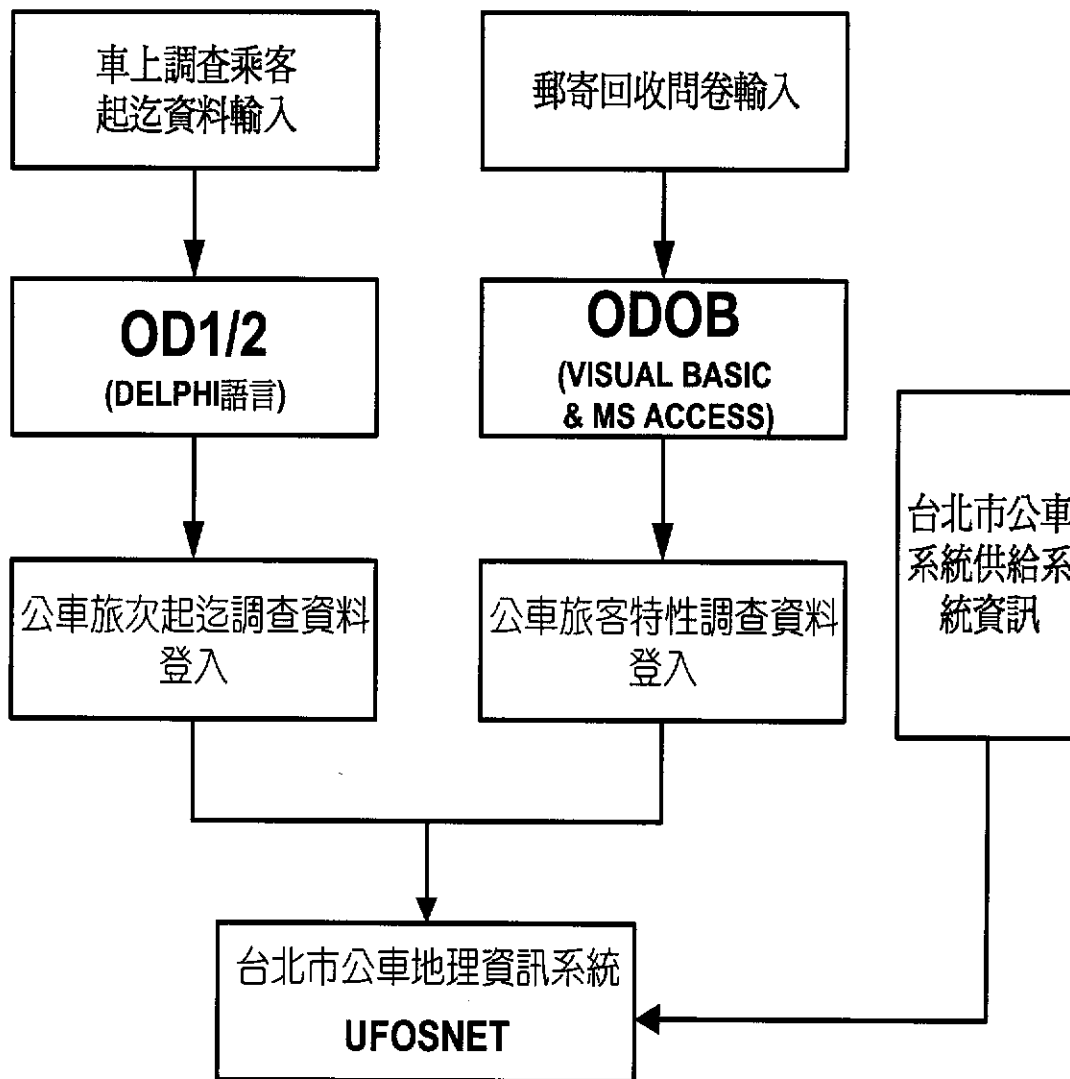


圖 9-7 資料整理系統架構與地理資訊系統結合

第十章 公車旅客起迄站資料庫

車上問卷上聯之上下車站起迄卡回收之後，公車旅客起迄資料庫的建立步驟如圖 9-1 所示。

10-1 O-D 表建檔

1. 各班次旅次起迄矩陣(O-D 表)整理：除了記錄該班次的 OD 矩陣外，亦記錄了路線名、調查日期、時間、調查員、問卷發出張數、實際回收張數以及搭乘人數等基本資料(參見表 10-1)。

台北市都會區公車運輸起迄調查技術報告
表 10-1 公車旅次起迄矩陣整理表

調查路線：										日期：		出發時間：										起站站名：																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													
										發出張數：		實際回收張數：										拒拿人數：																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																																													

2.將各班次調查結果賦予批號(ID Code)：例如五月二十九日下午四點四十五分發車的 222 路線，依問卷回收的順序賦予第 788 號

3.資料輸入：利用程式 OD1 進行資料建檔工作(程式 OD1 請參見表 10-2)，建檔完成後以批號為檔名、文字檔(*.txt)格式儲存。例如：由編號 229 之工讀生負責輸入，批號 788 的資料存檔為 788229.TXT，檔案格式說明如下：

以檔案 788229.TXT 為例，檔案內容為：

222529164512111807
010101
010301
010501
010701
011101
020401
020601
021301
030102
030601
031101
031501
040101
041101

.....

(以下省略)

檔案內容說明如下：

1.檔案第一行內容為：

222 5 29 1645 121 118 07

(路線名 月 日 時間 發出張數 收回張數 拒拿人數)

表示此檔案為路線 222 於 5 月 29 日 16 時 45 分發車，調查員共發出 121 張問卷，實際回收 118 張，有 7 位乘客拒拿問卷。

2.檔案第二行至檔尾內容為:

01 01 01

(起點編碼 迄點編碼 人次)

表示乘坐此班車的乘客從分區 01 上車並在分區 01 下車的計 01 人次。

表 10-2 OD1 程式列表及程式說明

unit Od11;	—
interface	
uses	
SysUtils, WinTypes, WinProcs, Messages,	
Classes, Graphics, Controls,	
Forms, Dialogs, StdCtrls, Buttons;	
type	變
TODInput = class(TForm)	數
Label1: TLabel;	宣
Edit1: TEdit;	告
Label2: TLabel;	
Edit2: TEdit;	
Label3: TLabel;	
Edit3: TEdit;	
Label4: TLabel;	
Edit4: TEdit;	
Label5: TLabel;	

```

Label6: TLabel;
Edit5: TEdit;
Label7: TLabel;
Edit6: TEdit;
Label8: TLabel;
Edit7: TEdit;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Edit8: TEdit;
Edit9: TEdit;
Edit10: TEdit;
BitBtn3: TBitBtn;
BitBtn4: TBitBtn;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Edit11: TEdit;
Label16: TLabel;
Edit12: TEdit;
procedure FormCreate(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn4Click(Sender: TObject);
procedure BitBtn3Click(Sender: TObject);
procedure Edit1Exit(Sender: TObject);
procedure Edit2Exit(Sender: TObject);
procedure Edit3Exit(Sender: TObject);
function IsNumber(InputStr: String): Integer;
procedure Edit4Exit(Sender: TObject);
procedure Edit5Exit(Sender: TObject);
procedure Edit6Exit(Sender: TObject);
procedure Edit7Exit(Sender: TObject);
procedure Edit8Exit(Sender: TObject);
procedure Edit9Exit(Sender: TObject);
procedure Edit10Exit(Sender: TObject);
procedure Edit11Exit(Sender: TObject);

```

變
數
宣
告

```

    procedure Edit12Exit(Sender: TObject);
    procedure Edit8KeyPress(Sender: TObject; var
Key: Char);
    procedure Edit9KeyPress(Sender: TObject; var
Key: Char);
    procedure Edit10KeyPress(Sender: TObject;
var Key: Char);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    ODInput: TODInput;
    data: TextFile;
    Count, Check: Integer;
    file_name: String;

implementation

{$R *.DFM}

procedure TODInput.FormCreate(Sender: TObject);
begin
    ODInput.Height := 250;
    Count := 1;
end;

procedure TODInput.BitBtn2Click(Sender:
TObject);
begin
    Close;
end;

procedure TODInput.BitBtn1Click(Sender:
TObject);
var temp1: String;
begin
    file_name := Edit12.Text + Edit11.Text + '.txt';
    If FileSearch(file_name, '.') = '' then
        begin

```

變
數
宣
告

開
啟
待
輸
入
檔
案

```

ODInput.Height := 400;
BitBtn1.Enabled := false;
BitBtn2.Enabled := false;
AssignFile(data,file_name);
ReWrite(data);
temp1 := Edit1.Text + Edit2.Text + Edit3.Text
+ Edit4.Text + Edit5.Text +
      Edit6.Text + Edit7.Text;
WriteLn(data,temp1);
BitBtn3.Enabled := true;
Check := 0;
CloseFile(data);
Edit8.SetFocus;
end
else
begin
  ShowMessage('檔案已存在');
  ODInput.Height := 400;
  BitBtn1.Enabled := false;
  BitBtn2.Enabled := false;
  Edit8.SetFocus;
end;
end;

procedure TODInput.BitBtn4Click(Sender:
TObject);
begin
  ODInput.Height := 250;
  Count := Count + 1;
  BitBtn1.Enabled := true;
  BitBtn2.Enabled := true;
  Edit1.Text := '000';
  Edit2.Text := '5';
  Edit3.Text := '01';
  Edit4.Text := '0630';
  Edit5.Text := '000';
  Edit6.Text := '000';
  Edit7.Text := '00';
  Edit8.Text := '01';
  Edit9.Text := '01';
  Edit10.Text := '01';

```

開啟待輸入檔案

設定變數初始值

Edit12.Text := '000';	
Edit12.SetFocus;	
end;	—
 procedure TODInput.BitBtn3Click(Sender: TObject);	—
var temp1: String;	
begin	
AssignFile(data,file_name);	
Append(data);	將
Check := Check + StrToInt(Edit10.Text);	資
Label13.Caption := IntToStr(Check);	料
temp1 := Edit8.Text + Edit9.Text + Edit10.Text;	寫
WriteLn(data,temp1);	入
Edit8.Text := '01';	檔
Edit9.Text := '01';	案
Edit10.Text := '01';	
CloseFile(data);	
Edit8.SetFocus;	—
end;	
 procedure TODInput.Edit1Exit(Sender: TObject);	—
begin	
If Length(Edit1.Text) > 3 then	
begin	
ShowMessage('路線最多只有三碼');	
Edit1.SetFocus;	
end	
else if Length(Edit1.Text) = 0 then	錯
begin	誤
ShowMessage('請輸入路線');	檢
Edit1.SetFocus;	查
end	程
else if Length(Edit1.Text) = 1 then Edit1.Text := '00' + Edit1.Text	序
else if Length(Edit1.Text) = 2 then Edit1.Text := '0' + Edit1.Text;	
end;	
procedure TODInput.Edit2Exit(Sender: TObject);	
begin	
If (Length(Edit2.Text) < 1) or (Edit2.Text > '9')	
or (Edit2.Text < '2') then	

```

begin
  ShowMessage('月份輸入有誤');
  Edit2.SetFocus;
end;
end;

procedure TODInput.Edit3Exit(Sender: TObject);
begin
  If (Length(Edit3.Text) > 2) or (Length(Edit3.Text)
= 0) or (IsNumber(Edit3.Text) = 0) then
    begin
      ShowMessage('日期輸入有誤');
      Edit3.SetFocus;
    end
  else if Length(Edit3.Text) = 1 then Edit3.Text :=
'0' + Edit3.Text;
end;

function TODInput.IsNumber(InputStr: string):
Integer;
var i, j, temp1: Integer;
begin
  i := Length(InputStr);
  temp1 := 0;
  for j := 1 to i do
    begin
      if (InputStr[j] < '0') or (InputStr[j] > '9')
      then
        begin
          result := 0;
          temp1 := 1;
        end;
      end;
  if temp1 = 0 then result := 1;
end;

procedure TODInput.Edit4Exit(Sender: TObject);
begin
  If (Length(Edit4.Text) < 4) or
(IsNumber(Edit4.Text) = 0) then
    begin

```

錯
誤
檢
查
程
序

```

    ShowMessage('時間輸入有誤');
    Edit4.SetFocus;
end;
end;

procedure TODInput.Edit5Exit(Sender: TObject);
begin
    If (Length(Edit5.Text) > 3) or (Length(Edit5.Text)
= 0) or (IsNumber(Edit5.Text) = 0) then
        begin
            ShowMessage('發出份數輸入有誤');
            Edit5.SetFocus;
        end
    else if Length(Edit5.Text) = 1 then Edit5.Text :=
'00' + Edit5.Text
    else if Length(Edit5.Text) = 2 then Edit5.Text :=
'0' + Edit5.Text;
end;

procedure TODInput.Edit6Exit(Sender: TObject);
begin
    If (Length(Edit6.Text) > 3) or (Length(Edit6.Text)
= 0) or (IsNumber(Edit6.Text) = 0) then
        begin
            ShowMessage('回收份數輸入有誤');
            Edit6.SetFocus;
        end
    else if Length(Edit6.Text) = 1 then Edit6.Text :=
'00' + Edit6.Text
    else if Length(Edit6.Text) = 2 then Edit6.Text :=
'0' + Edit6.Text;
end;

procedure TODInput.Edit7Exit(Sender: TObject);
begin
    If (Length(Edit7.Text) > 2) or (Length(Edit7.Text)
= 0) or (IsNumber(Edit7.Text) = 0) then
        begin
            ShowMessage('拒拿份數輸入有誤');
            Edit7.SetFocus;
        end
    else if Length(Edit7.Text) = 1 then Edit7.Text :=
'0' + Edit7.Text;

```

錯
誤
檢
查
程
序

end;

procedure TODInput.Edit8Exit(Sender: TObject);

begin

If (Length(Edit8.Text) > 2) or (Length(Edit8.Text) = 0) or (IsNumber(Edit8.Text) = 0) then

begin

ShowMessage('起點輸入有誤');

Edit8.SetFocus;

end

else if Length(Edit8.Text) = 1 then Edit8.Text := '0' + Edit8.Text;

end;

procedure TODInput.Edit9Exit(Sender: TObject);

begin

If (Length(Edit9.Text) > 2) or (Length(Edit9.Text) = 0) or (IsNumber(Edit9.Text) = 0) then

begin

ShowMessage('迄點輸入有誤');

Edit9.SetFocus;

end

else if Length(Edit9.Text) = 1 then Edit9.Text := '0' + Edit9.Text;

end;

procedure TODInput.Edit10Exit(Sender: TObject);

begin

If (Length(Edit10.Text) > 2) or (Length(Edit10.Text) = 0) or (IsNumber(Edit10.Text) = 0) then

begin

ShowMessage('次數輸入有誤');

Edit10.SetFocus;

end

else if Length(Edit10.Text) = 1 then Edit10.Text := '0' + Edit10.Text;

end;

procedure TODInput.Edit11Exit(Sender: TObject);

begin

If (Length(Edit11.Text) > 3) or (Length(Edit11.Text) = 0) then

錯
誤
檢
查
程
序

```

begin
  ShowMessage('工讀生編號輸入有誤');
  Edit11.SetFocus;
end
else if Length(Edit11.Text) = 1 then
  Edit11.Text := '00' + Edit11.Text
else if Length(Edit11.Text) = 2 then
  Edit11.Text := '0' + Edit11.Text;
end;
procedure TODInput.Edit12Exit(Sender: TObject);
begin
  If (Length(Edit12.Text) > 3) or
  (Length(Edit12.Text) = 0) then
  begin
    ShowMessage('問卷批號輸入有誤');
    Edit12.SetFocus;
  end
  else if Length(Edit12.Text) = 1 then
    Edit12.Text := '00' + Edit12.Text
  else if Length(Edit12.Text) = 2 then
    Edit12.Text := '0' + Edit12.Text;
  end;
procedure TODInput.Edit8KeyPress(Sender:
TObject; var Key: Char);
begin
  If Key <> #13 then exit
  else Edit9.SetFocus;
end;

procedure TODInput.Edit9KeyPress(Sender:
TObject; var Key: Char);
begin
  If Key <> #13 then exit
  else Edit10.SetFocus;
end;
procedure TODInput.Edit10KeyPress(Sender:
TObject; var Key: Char);
begin
  If Key <> #13 then exit
  else BitBtn3.SetFocus;
end;

end.

```

錯
誤
檢
查
程
序

資
料
輸
入
程
序
控
制

10-2 建立編碼對照表

- 1.運用 Excel 建立各路線之 O-D 編碼與 388 個交通分區編碼對照表。
- 2.對照表建立完成後，以 dBASE 檔案格式(*.dbf) 儲存。例如：將 222 路之路線編碼對照表存檔為 222.dbf，檔案格式如表 10-3 所示：

表 10-3 222 路線編碼對照表

1	1	172		
2	2	171	170	
3	2	180	171	
4	2	180	184	
5	2	181	184	
6	2	181	183	
7	2	183	74	
8	2	182	74	
9	1	74		
10	2	73	74	
11	1	73		
12	1	69		
13	1	56		
14	1	53		
15	1	61		
16	1	65		
17	1	68		
18	2	11	88	
19	2	11	12	
20	2	2	10	
21	2	6	10	
22	1	6		
23	2	5	7	
24	1	7		

說明:N1 為 222 之路線編碼

N2 為 222 路線編碼對照交通分區區碼數

N3~N6 為所對照之交通分區碼

10-3 轉換矩陣資料

運用程式 OD2，根據分區編碼對照表(*.dbf)
將所有班次路線之 O-D 資料全部轉換為 388 交通
分區之 O-D 矩陣資料。例如：原始資料
788229.TXT，經 OD2 轉換後存檔為 788229.db，
檔案格式如表 10-4 所示，程式 OD2 請參見表 10-5：

表 10-4 788229.db 內容說明

788	172	172	
788	172	180	
788	172	171	
788	172	181	
788	172	184	
788	172	183	
788	172	74	
788	172	73	
788	171	180	
788	171	184	
788	171	181	
788	171	183	
788	171	56	
788	180	172	
788	180	181	
788	180	183	
788	180	73	
788	180	61	
788	180	56	
788	180	3	
788	180	6	
788	181	65	
788	181	171	
788	181	170	

(以下省略)

說明:QNo 為批號

O,D 分別為起點與迄點之 388

交通分區區碼

Times 為在該起迄點上下車的

乘客人次*

*註: 由於許多公車站牌位置均在分區界線中間,因此路線編碼在對照交通分區區碼時,至少一組,最多則有四組,遇到大於一組的情形時轉換程式便自動將乘客人次平均分

表 10-5 OD2 程式列表及程式說明

unit Od21;

interface

uses

SysUtils, WinTypes, WinProcs, Messages,
Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, Buttons, DB, DBTables;

type

TForm1 = class(TForm)
Label1: TLabel;
Edit1: TEdit;
BitBtn1: TBitBtn;
BitBtn2: TBitBtn;
DataSource1: TDataSource;
Query1: TQuery;
BitBtn3: TBitBtn;
OpenDialog1: TOpenDialog;
procedure BitBtn2Click(Sender: TObject);
procedure CopyFile(Dest1, Source1: String);
procedure BitBtn1Click(Sender: TObject);
procedure WriteMatrix(Line, O, D, Tim, QNo:
String);
procedure Edit1Exit(Sender: TObject);
procedure Edit1KeyPress(Sender: TObject; var
Key: Char);
procedure BitBtn3Click(Sender: TObject);
private
{ Private declarations }
public

變
數
宣
告

```

    { Public declarations }
end;

var
    Form1: TForm1;

implementation

{$R *.DFM}

procedure TForm1.BitBtn2Click(Sender: TObject);
begin
    Close;
end;

procedure TForm1.CopyFile(Dest1, Source1:
String);
var
    Dest, Source: File;
    NumRead, NumWritten: Integer;
    Buf: array[1..2048] of char;
begin
    AssignFile(Source,Source1);
    ReSet(Source,1);
    AssignFile(Dest,Dest1);
    ReWrite(Dest,1);
    repeat
        BlockRead(Source, Buf, SizeOf(Buf),
NumRead);
        BlockWrite(Dest, Buf, NumRead,
NumWritten);
        until (NumRead = 0) or (NumWritten <
NumRead);
        CloseFile(Source);
        CloseFile(Dest);
    end;

procedure TForm1.BitBtn1Click(Sender: TObject);
var
    temp1, sql: String;
    Reject, O, D, Tim: String[2];
    Line, Day, Out, Inp: String[3];
    Ti: String[4];

```

變
數
宣
告

開
啟
原
始
資
料
、
區
碼
對
照
表
及
待
輸
入
檔
案

```

filename: String[6];
i: Integer;
data: TextFile;
begin
  If FileSearch(Edit1.Text, '.') = '' then
    begin
      ShowMessage('檔案不存在');
      Edit1.SetFocus;
    end
  else
    begin
      i := 1;
      filename := '000000';
      repeat
        filename[i] := Edit1.Text[i];
        i := i + 1;
      until i = 7;
      temp1 := 'temp.db';
      CopyFile(temp1, 'oda.db');
      CopyFile('temp1.db', 'odb.db');
      AssignFile(data, Edit1.Text);
      Reset(data);
      ReadLn(data, Line, Day, Ti, Out, Inp, Reject);
      Query1.Close;
      Query1.sql.Clear;
      sql := 'INSERT INTO
      Ques(QNo, Line, Days, Times, Out, Inp, Reject)
      VALUES (:1, :2, :3, :4, :5, :6, :7)';
      Query1.sql.Add(sql);
      temp1 := filename[1] + filename[2] +
      filename[3];
      Query1.ParamByName('1').AsString := temp1;
      Query1.ParamByName('2').AsString := Line;
      temp1 := '1996/' + Day[1] + '/' + Day[2] +
      Day[3];
      Query1.ParamByName('3').AsDateTime :=
      StrToDate(temp1);
      Query1.ParamByName('4').AsString := Ti;
      Query1.ParamByName('5').AsInteger :=
      StrToInt(Out);
      Query1.ParamByName('6').AsInteger :=
      StrToInt(Inp);
      Query1.ParamByName('7').AsInteger :=
      StrToInt(Reject);
    end
  end
end

```

讀取原始資料並根據區碼對照表進行資料轉換

```

Query1.ExecSQL;
while not EOF(data) do
begin
    ReadLn(data,O,D,Tim);
    Query1.Close;
    Query1.sql.Clear;
    temp1 := 'temp';
    sql := 'INSERT INTO ' + temp1 + '
(QNo,O,D,Times) VALUES (:1,:2,:3,:4)';
    Query1.sql.Add(sql);
    temp1 := filename[1] + filename[2] +
filename[3];
    Query1.ParamByName('1').AsString :=
temp1;
    Query1.ParamByName('2').AsInteger :=
StrToInt(O);
    Query1.ParamByName('3').AsInteger :=
StrToInt(D);
    Query1.ParamByName('4').AsInteger :=
StrToInt(Tim);
    Query1.ExecSQL;
    WriteMatrix(Line,O,D,Tim,temp1);
end;
CloseFile(data);
temp1 := filename + 'a.db';
CopyFile(temp1,'temp.db');
temp1 := filename + 'b.db';
CopyFile(temp1,'temp1.db');
DeleteFile('temp.db');
DeleteFile('temp1.db');
end;
end;

procedure TForm1.WriteMatrix(Line, O, D, Tim,
QNo: String);
var
    sql, temp1: String;
    O2, D2, temp2, i, j: Integer;
    Origin, Desti: array[1..4] of Integer;
    temp3: single;
begin
    i := 1;
    j := 1;
    temp2 := StrToInt(Line);
    temp1 := IntToStr(temp2);

```

讀取原始資料並根據區碼對照表進行資料轉換

資料輸出程序

```

CopyFile('line.dbf',temp1+'.dbf');
Query1.Close;
Query1.sql.Clear;
sql := 'SELECT N2,N3,N4,N5,N6 FROM line
WHERE N1 = :1';
Query1.sql.Add(sql);
Query1.ParamByName('1').AsInteger      :=
StrToInt(O);
Query1.Open;
O2 := Query1.Fields[0].AsInteger;
Origin[1] := Query1.Fields[1].AsInteger;
If O2 > 1 then Origin[2] :=
Query1.Fields[2].AsInteger;
If O2 > 2 then Origin[3] :=
Query1.Fields[3].AsInteger;
If O2 > 3 then Origin[4] :=
Query1.Fields[4].AsInteger;
Query1.Close;
Query1.sql.Clear;
sql := 'SELECT N2,N3,N4,N5,N6 FROM line
WHERE N1 = :1';
Query1.sql.Add(sql);
Query1.ParamByName('1').AsInteger      :=
StrToInt(D);
Query1.Open;
D2 := Query1.Fields[0].AsInteger;
Desti[1] := Query1.Fields[1].AsInteger;
If D2 > 1 then Desti[2] :=
Query1.Fields[2].AsInteger;
If D2 > 2 then Desti[3] :=
Query1.Fields[3].AsInteger;
If D2 > 3 then Desti[4] :=
Query1.Fields[4].AsInteger;
DeleteFile('line.dbf');
while ( i <= O2 ) do
begin
  while ( j <= D2 ) do
  begin
    Query1.Close;
    Query1.sql.Clear;
    sql := 'SELECT Times FROM temp1
WHERE O = :1 AND D = :2';
    Query1.sql.Add(sql);
    Query1.ParamByName('1').AsInteger :=
Origin[i];
    Query1.ParamByName('2').AsInteger :=
Desti[j];

```

資料輸出程序

```

Query1.Open;
temp3 := Query1.Fields[0].AsFloat;
Query1.Close;
Query1.sql.Clear;
If temp3 = 0 then
begin
    sql := 'INSERT INTO temp1
(QNo,O,D,Times) VALUES (:1,:2,:3,:4)';
    Query1.sql.Add(sql);
    Query1.ParamByName('1').AsString :=
QNo;
    Query1.ParamByName('2').AsInteger :=
Origin[i];
    Query1.ParamByName('3').AsInteger :=
Desti[j];
    If Tim < '0' then
        Query1.ParamByName('4').AsFloat :=
StrToInt(Tim) / (i*j)
    else
        Query1.ParamByName('4').AsInteger := 0;
    Query1.ExecSQL;
end
else
begin
    sql := 'UPDATE temp1 SET Times =
Times + :4 WHERE O = :2 AND D = :3';
    Query1.sql.Add(sql);
    Query1.ParamByName('2').AsInteger :=
Origin[i];
    Query1.ParamByName('3').AsInteger :=
Desti[j];
    If Tim < '0' then
        Query1.ParamByName('4').AsFloat :=
StrToInt(Tim) / (i*j)
    else
        Query1.ParamByName('4').AsInteger := 0;
    Query1.ExecSQL;
end;
j := j + 1;
end;
i := i + 1;
end;
end;

```

資
料
輸
出
程
序

```

procedure TForm1.Edit1Exit(Sender: TObject);
begin
    If Length(Edit1.TExt) = 6 then
        Edit1.TExt := Edit1.Text + '.txt';
end;

procedure TForm1.Edit1KeyPress(Sender: TObject;
var Key: Char);
begin
    If Key = #13 then BitBtn1.SetFocus;
end;

procedure TForm1.BitBtn3Click(Sender: TObject);
begin
    If OpenFileDialog1.Execute then
        Edit1.Text := OpenFileDialog1.filename;
end;

end.

```

輸入
程序
控制

10-4 各時段旅次量加總

第一階段調查資料分為四個時段，分別為上午尖峰 AM6:30~AM9:30、白天離峰 AM9:30~PM4:30、下午尖峰 PM4:30~PM8:30 以及夜間離峰 PM8:30~PM11:00，第二階段調查資料則均為下午尖峰 PM4:30~PM8:30，各時段分別以 DELPHI 語言所寫的程式 ODSum 執行加總工作，程式 ODSum 請參見表 10-6。

10-5 旅次量放大

旅次量放大須依照下列步驟進行：

10-5-1 計算第一階段各路線下午尖峰調查值比例：

根據本計劃的調查時段，上午尖峰 AM6:30~AM9:30 計 3 小時、白天離峰 AM9:30~PM4:30 計 7 小時、下午尖峰 PM4:30~PM8:30 計 4 小時以及夜間離峰 PM8:30~PM11:00 計 2.5 小時，與 10-4 節所得之各路線各調查時段時段加總旅次量相乘，再將下午尖峰之乘積除以各時段乘積總合，即分別得到第一階段十一條路線的下午尖峰載客比例，取其平均值 28.18% 即為下午尖峰調查值佔總調查值比例。

10-5-2 計算各路線調查值佔全日總運量比例：

將各路線第一階段之調查值與各時段之時間相乘，再除以各路線之平均每日運量（由聯營公車營運資料而得），即可得到各路線第一階段調查值佔總運量之比例，所得之十一條路線平均值為 6.81%。

10-5-3 計算全日放大係數：

將 10-5-2 與 10-5-3 兩數字相乘可得各路線平均下午尖峰時間調查值佔全日運量之比例：1.3053%，此數字的倒數 76.6134 即為第二階段調查值之放大係數，將第二階段調查原始資料放大為代表全日之旅次起迄資料。

10-6 校估旅次起迄表

本計劃因限於人力、人力與經費的限制,無法將台北都會區各公車路線進行全面性的調查,故經 10-5 節所得之全日旅次起迄矩陣仍有部分無調查值之空格有待填補,為使整個台北都會區公車運輸起迄表能具有代表性,本計劃採用重力模式,根據亞聯工程顧問股份有限公司八十四年度台北都會區社經屬性資料,本計劃以分區人口數為旅次產生指標(P_{ij});分區就業人數為旅次吸引指標(A_{ij}),分區間之旅行長度則採用 Ufosnet 對大台北都會區之各交通分區間之最小旅行距離(Skim Tree)的運算結果(R_{ij}^n)(*註 1),求出台北都會區之重力矩陣,以重力矩陣中指數的最大值(起點分區 208;迄點分區 8)之全日旅次數為基準,按比例填滿無調查值之空格,最後得到依調查結果推估之台北市公車旅次起迄矩陣,完整之旅次起迄矩陣資料請參見本技術報告附錄(*註 2)。

*註:

- 1.經測試 2, 1.5, 1 三種 n 值後,本計劃認為以 $n=1$ 所得之矩陣最為合理。
- 2.分區 113 在基隆河截彎取直工程進行至今住戶極少,84 年之社經指標均為 0,故無法進行校估;分區 383~388 為台北都會區界外分區,故亦無社經指標。

表 10-6 程式 ODSum 列表

```

unit Odsum1;
interface
uses
  SysUtils, WinTypes, WinProcs, Messages,
  Classes, Graphics, Controls,
  Forms, Dialogs, DB, DBTables, StdCtrls, Buttons;
type
  TSum = class(TForm)
    OpenDialog1: TOpenDialog;
    DataSource1: TDataSource;
    Query1: TQuery;
    Label1: TLabel;
    Edit1: TEdit;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    ListBox1: TListBox;
    Label2: TLabel;
    BitBtn3: TBitBtn;
    DataSource2: TDataSource;
    Query2: TQuery;
    Label3: TLabel;
    procedure BitBtn1Click(Sender: TObject);
    procedure CopyFile(Dest1, Source1: string);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure WriteTable(rtemp, ctemp: Integer);
    function Fill0(InNum, DNum: Integer): String;
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Sum: TSum;
  file_name: String;
implementation
  {$R *.DFM}
  procedure TSum.CopyFile(Dest1, Source1: String);
  var
    Dest, Source: File;
    NumRead, NumWritten: Integer;
    Buf: array[1..2048] of char;
  begin
    AssignFile(Source, Source1);
    ReSet(Source, 1);
    AssignFile(Dest, Dest1);
  
```

變
數
宣
告

設
定

```

ReWrite(Dest,1);
repeat
    BlockRead(Source, Buf, SizeOf(Buf),
NumRead);
    BlockWrite(Dest, Buf, NumRead,
NumWritten);
until (NumRead = 0) or (NumWritten <
NumRead);
    CloseFile(Source);
    CloseFile(Dest);
end;

procedure TSum.BitBtn1Click(Sender: TObject);
var
    stemp1: String;
    itemp1, i: Integer;
begin
    If OpenFileDialog1.Execute then
    begin
        file_name := OpenFileDialog1.FileName;
        stemp1 := '';
        itemp1 := Length(file_name);
        stemp1 := file_name[itemp1-9] +
file_name[itemp1-8] + file_name[itemp1-7];
        {itemp1 := ListBox1.Columns;
        i := 1;
        while ( i <= itemp1 ) do
        begin
            ListBox1.ItemIndex := i;
            If ListBox1.Items = stemp1 then
            begin
                ShowMessage('這份問卷已經在這一次加
總過了!');
                exit;
            end;
        end;
        Edit1.Text := stemp1;
    end;
end;

procedure TSum.BitBtn2Click(Sender: TObject);
var
    sql: String;
    O1, D1: Integer;
    Times1, Times2: Real;
begin
    CopyFile('temp.db',file_name);
    Query1.Close;
    Query1.sql.Clear;
    sql := 'SELECT O,D,Times FROM temp';
    Query1.sql.Add(sql);
    Query1.Open;
    while not Query1.EOF do
    begin
        O1 := Query1.Fields[0].AsInteger;
        D1 := Query1.Fields[1].AsInteger;
        Times1 := Query1.Fields[2].AsFloat;
        Query2.Close;
        Query2.sql.Clear;
        sql := 'SELECT Times FROM SUMS WHERE
O = :1 AND D = :2';
        Query2.sql.Add(sql);
        Query2.ParamByName('1').AsInteger := O1;
        Query2.ParamByName('2').AsInteger := D1;
        Query2.Open;
    end;
end;

```

暫
存
檔

選
擇
加
總
檔
案

資
料
加
總

```

        Times2 := Query2.Fields[0].AsFloat;
        If Times2 = 0 then sql := 'INSERT INTO
SUMS(O,D,Times) VALUES(:1,:2,:3)'
        else sql := 'UPDATE SUMS SET Times =
Times + :3 WHERE O = :1 AND D = :2';
        Query2.Close;
        Query2.sql.Clear;
        Query2.sql.Add(sql);
        Query2.ParamByName('1').AsInteger := O1;
        Query2.ParamByName('2').AsInteger := D1;
        Query2.ParamByName('3').AsFloat := Times1;
        Query2.ExecSQL;
        Query1.Next;
    end;
    DeleteFile('temp.db');
    ListBox1.Items.Add(Edit1.Text);
    Edit1.Text := "";
end;

procedure TSum.BitBtn3Click(Sender: TObject);
var
    rtemp, ctemp, texttemp: Integer;
    sql: String;
begin
    rtemp := 1;
    ctemp := 1;
    texttemp := 1;
    while rtemp < 400 do
        begin
            while ctemp < 400 do
                begin
                    WriteTable(rtemp,ctemp);
                    ctemp := ctemp + 50;
                    texttemp := texttemp + 1;
                    Label3.Caption := IntToStr(texttemp);
                end;
                rtemp := rtemp + 50;
                ctemp := 1;
            end;
            CopyFile('SUM1.db','SUMS.db');
            Query1.Close;
            Query1.sql.Clear;
            sql := 'DELETE * FROM SUMS';
            Query1.sql.Add(sql);
            Query1.ExecSQL;
            Close;
        end;

    procedure TSum.WriteTable(rtemp, ctemp: Integer);
    var
        ODTable: TextFile;
        i, j, t4: Integer;
        t1, t2, t3: String;
    begin
        i := 0;
        j := 0;
        t1 := Fill0(rtemp,3);
        t2 := Fill0(ctemp,3);
        t3 := t1 + t2 + '.txt';
        AssignFile(ODTable,t3);
        ReWrite(ODTable);
        Write(ODTable,' ');
        while i < 50 do
            begin
                t3 := '' + Fill0(ctemp+i,3) + ' ';
                Write(ODTable,t3);
            end;
        end;
    end;
end;

```

資
料
加
總

建
立
資
料
加
總
後
暫
存
檔

產
生
旅
次
起
迄
表

```

i := i + 1;
end;
WriteLn(ODTable,"");
i := 0;
While j < 50 do
begin
t3 := Fill0(rtemp+j,3) + '';
Write(ODTable,t3);
while i < 50 do
begin
Query1.Close;
Query1.sql.Clear;
t3 := 'SELECT Times FROM SUMS
WHERE O = :1 AND D = :2';
Query1.sql.Add(t3);
Query1.ParamByName('1').AsInteger :=
rtemp+j;
Query1.ParamByName('2').AsInteger :=
ctemp+i;
Query1.Open;
t4 := Query1.Fields[0].AsInteger;
t3 := Fill0(t4,4) + '';
Write(ODTable,t3);
i := i + 1;
end;
WriteLn(ODTable,"");
i := i + 1;
i := 1;
end;
CloseFile(ODTable);
end;

function TSum.Fill0(InNum, DNum: Integer):
String;
var
temp: String;
templen: Integer;
begin
temp := IntToStr(InNum);
templen := Length(temp);
while templen < DNum do
begin
temp := '0' + temp;
templen := templen + 1;
end;
result := temp;
end;
end

```

產生
旅次
起迄
表

整
數
處
理
副
程
式

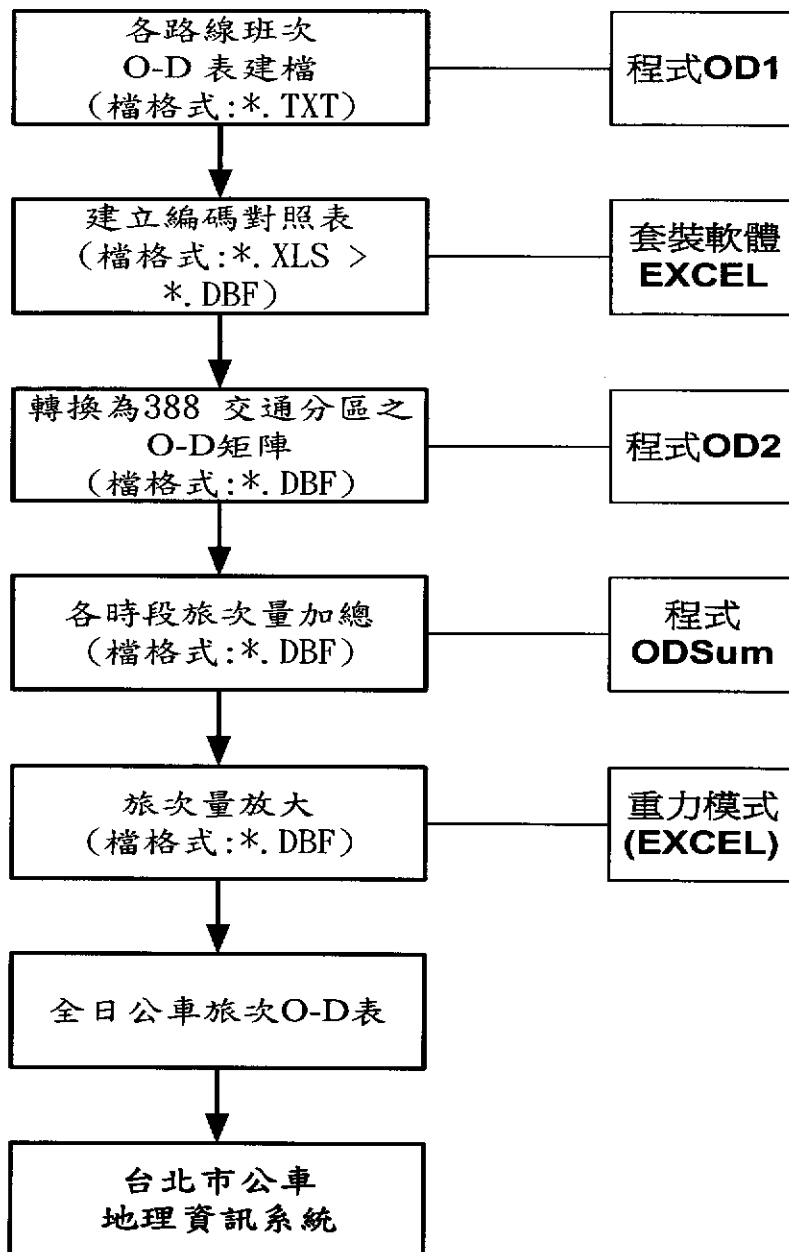


圖 10-1 公車旅客起迄資料分析步驟

第十一章 公車旅客特性資料庫

公車旅客特性資料為車上調查問卷下聯與車站發放問卷調查所得，經由郵寄問卷回收後，其資料庫建立步驟如圖 11-1 所示。

11-1 問卷分類

收到民眾寄回或傳真之問卷後，先粗略分為有效問卷、資料不全可聯絡、以及無效問卷三類，其中資料不全可聯絡的回收問卷均經過研究人員以電話與受調者聯繫後，視資料補充之內容再分為有效與無效兩類，而問卷有效性之判別則以下列三點為原則：

- 1.問卷正面第 10 題未答者暫定為無效問卷。
- 2.問卷背面活動路程均未答者暫定為無效問卷。
- 3.以上兩點經電話聯絡兩次失敗或遭拒一次者，則歸類為無效問卷。

接著再依問卷發出方式、有效性與受訪民眾接受進一步家庭訪問意願進行問卷分類，共分為 A、B、C、D，如表 11-1 所列，其中 A、C 兩類在經資料電腦化後即進行家庭訪問調查。

表 11-1 問卷分類方法

問卷發放點與有效性		接受家訪意願	願意接受家訪	不願意接受家訪
車上問卷	有效問卷		A	B
	無效問卷		-	
車站問卷	有效問卷		C	D
	無效問卷		-	

11-2 人工讀卡

以人工方式將問卷內容中答案為文字格式之地址或地點轉化為交通分區代碼後，交予資料輸入人員鍵入。

11-3 賦予批號

為便於進行資料輸入作業與日後偵錯查詢之用，在資料電腦化前先以回函問卷整理表(參見表 11-2)將問卷分批。

表 11-2 回函問卷整理表

台北都會區公車運輸起迄調查
回函問卷整理表

以下由讀卡員填寫				以下由 Key in 員填寫					
批 號	讀卡日期	讀卡員 簽名	備註	Key in 日期	Key in員簽 名	本批 數量	Key in 完 成否?		備註
							是	否	
					Key in員編 號				

批號編寫方式: a: 車上問卷願意接受家訪: A0001 開始
 b: 車上問卷不願意接受家訪: B0001 開始
 c: 車站問卷願意接受家訪: C0001 開始
 d: 車站問卷不願意接受家訪: D0001 開始

11-4 問卷資料電腦化

1. 應用 ODOB 資料整理程式，將數化後之問卷資料鍵入電腦，並以 Microsoft Access 資料庫的儲存格式儲存為*.mdb 檔，檔格式參見表 10-3，程式 ODOB 請參見表 11-4。
2. 透過地理分區規劃處理，此資料檔不僅可依調查樣本所處位置做 Geo Coding，更可做不同程度的整合處理(Aggregation)，其社經屬性與旅次特性得以與環境相結合，作為更精確交通分析之用。

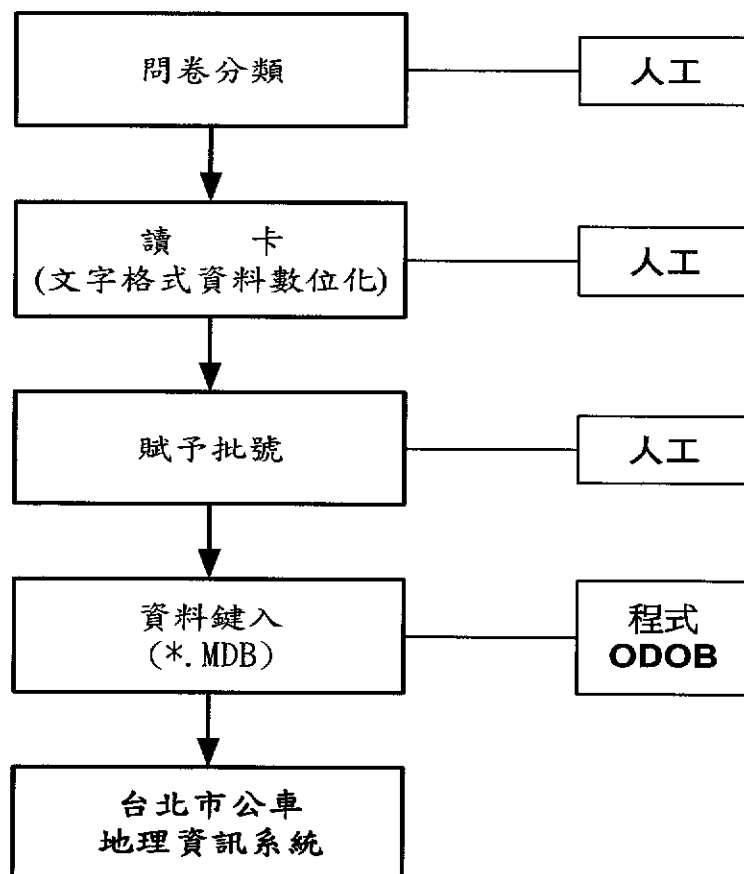


圖 11-1 公車旅客特性資料庫建立步驟

表 11-3 ODOB 資料輸入檔案格式說明表

資料欄位編號	問卷欄位名稱	MDB 檔欄位名稱	欄位大小
1	問卷編號	問卷編號	6
2	年齡	年齡	2
3	性別	性別	1
4	職業	職業	2
5	每月薪資	每月薪資	6
6	全家人數	全家人數	2
7	居住地址	居住區碼 1	3
8		居住區碼 2	3
9		居住區碼 3	3
10		居住區碼 4	3
11	工作地址	工作區碼 1	3
12		工作區碼 2	3
13		工作區碼 3	3
14		工作區碼 4	3
15	第一題	t-1	1
16		t-1-1	1
17		t-1-2	1
18	第二題	t-2	1
19	第三題	t-3	6
20	第四題	t-4	7
21	第五題	t-5	1
22	第六題	t-6	1
23	第七題	t-7-1	5
24		t-7-2	5
25	第八題	t-8	1
26	第九題	t-9-1	5
27		t-9-2	2
28	第十題	t-10-1	3
29		t-10-2	2
30	第十一題	t-11-1	4
31		t-11-2	1
32		t-11-3	2
33		t-11-4	2
34		t-11-5	3
35	第十二題	t-12-1	4
36		t-12-2	3
37	第十三題	t-13-1	3
38		t-13-2	3

39	第十四題	t-14-1	3
40	第十四題	t-14-2	3
41		t-14-3	2
42		t-15-1	3
43	第十五題	t-15-1-2	2
44		t-15-2	2
45		t-15-3	2
46	第十六題	t-16	2
47	受訪日	受訪日	4
48	活動行程 1 搭乘時間	P1time	4
49	活動行程 1 搭乘路線	P1 路線	6
50	活動行程 1 上車地點區碼 1	P1-O-1	3
51	活動行程 1 上車地點區碼 2	P1-O-2	3
52	活動行程 1 上車地點區碼 3	P1-O-3	3
53	活動行程 1 上車地點區碼 4	P1-O-4	3
54	活動行程 1 下車地點區碼 1	P1-D-1	3
55	活動行程 1 下車地點區碼 2	P1-D-2	3
56	活動行程 1 下車地點區碼 3	P1-D-3	3
57	活動行程 1 下車地點區碼 4	P1-D-4	3
58	活動行程 1 旅次目的	P1 旅次目的	1
59	活動行程 1 票價	P1 票價	3
...

*註:活動路徑 2~6 資料格式同活動路徑

1, 在此省略。

表 11-4 程式 ODOB 列表及程式說明

Private Sub Command1_Click()

```

Dim fn As String, FL As Long
fn = Text1.Text + Text2.Text
sourcefile = "c:\odob\odob.mdb"
targetfile = "c:\odob\" + fn + ".mdb"
On Error GoTo FileNotExist
FL = FileLen(targetfile)
GoTo DoRest

```

開啟待輸入資料檔

FileNotExist:

```

Resume CopyFile

CopyFile:
    FileCopy sourcefile, targetfile

DoRest:
    On Error GoTo 0
    Load Form1
    Form1.data1.DatabaseName = targetfile
    Form1.data1.Refresh
    Form1.Label44.Caption = fn
    Form1.Show
    Unload Me
End Sub

Private Sub Text1_GotFocus()
    Text1.SelStart = 0
    Text1.SelLength = Len(Text1.Text)
End Sub

Private Sub Text2_GotFocus()
    Text2.SelStart = 0
    Text2.SelLength = Len(Text2.Text)
End Sub

Option Explicit
Private UpdateCancelled As Boolean
Private Sub add_Click()
    Dim i As Integer
    data1.Recordset.AddNew
    Text2 = "xxxxxx"
    Text3 = "x"
    Text4 = "xx"
    Text5 = "xxxxxx"
    For i = 0 To 3
        Text6(i) = "xxx"
        Text7(i) = "xxx"
    Next
    Text8 = "x"
    Text9 = "x"
    Text10 = "x"
    Text1 = "x"

```

|
 —
 —
 |
 指定並重整資料庫
 |
 —
 —
 |
 防錯程序
 |
 —
 —
 |
 |
 設定變數初始值
 |
 |

```

Text11 = "xxxxxx"
Text12 = "xxxxxxx"
Text13 = "x"
Text14 = "x"
Text15 = "xxxxx"
Text16 = "xxxxx"
Text17 = "x"
Text18 = "xxxxx"
Text19 = "xx"
Text20 = "xx"
Text21 = "xx"
Text22 = "xxx"
Text23 = "x"
Text24 = "x"
Text25 = "xx"
Text26 = "xxx"
Text27 = "xxx"
Text28 = "xxx"
Text29 = "xxx"
Text30 = "xxx"
Text31 = "xxx"
Text32 = "xxx"
Text33 = "xx"
Text34 = "xxx"
Text35 = "xx"
Text36 = "xx"
Text37 = "xx"
Text38 = "xxx"
For i = 0 To 5
  Text39(i) = "xxx"
  Text40(i) = "xxxxxxx"
  Text41(i) = "xxx"
  Text42(i) = "xxx"
  Text43(i) = "xxx"
  Text44(i) = "xxx"
  Text45(i) = "xxx"
  Text46(i) = "xxx"
  Text47(i) = "xxx"
  Text48(i) = "xxx"
  Text49(i) = "x"
  Text50(i) = "xxx"
Next i

```

設
定
變
數
初
始
值

Text51 = "xx"	
Text52 = "xx"	
Text53 = "xx"	
Text54 = "x"	
Text2.SetFocus	
End Sub	—
Private Sub Command3_Click()	—
Unload Me	
Load OFile	
OFile.Show	
End Sub	
	主程式
	—
Private Sub data1_Reposition()	—
If data1.Recordset.EditMode = 0 Then	
Label42.Caption	
data1.Recordset.AbsolutePosition + 1	
End If	
End Sub	
	料筆數
	—
	顯示資
	—
Private Sub Data1_Validate(Action As Integer, save As Integer)	—
Dim msg As String	
If save = True Or Action = vbDataActionUpdate _	
Or data1.Recordset.EditMode =	
dbEditAdd _	
Then	
If Text2 = "" Then	
msg = "必須輸入問卷編號"	
Text2.SetFocus	
End If	
End If	
If msg <> "" Then	
MsgBox msg, vbExclamation	
Action = vbDataActionCancel	
UpdateCancelled = True	
Else	
UpdateCancelled = False	
End If	
End Sub	—
	錯誤檢查
	—
Private Sub del_Click()	


```

Dim msg As String
msg = "確定要刪除這筆資料?"
msg = msg & If(Text2 <> "", Text2, "這筆資料") &
""
If MsgBox(msg, vbQuestion + vbYesNo +
vbDefaultButton2) = vbYes Then
    data1.Recordset.Delete
    data1.Recordset.MoveNext
    If Data1.Recordset.EOF Then
        Data1.Recordset.MoveLast
    End If
End Sub

```

—
|
資料刪除程序
|
—

```

Private Sub exit_Click()
    Unload Form1
End Sub

```

```

Private Sub Form_Activate()
    add_Click
End Sub

```

```

Private Sub Form_Load()
    data1.Refresh
    Me.Show
End Sub

```

```

Private Sub Form_Unload(Cancel As Integer)
    If UpdateCancelled Then
        Cancel = True
    ElseIf data1.Recordset.EditMode = dbEditAdd Then
        data1.Recordset.Update
    End If
End Sub

```

```

Private Sub Save_Click()
    data1.Recordset.Update
    add_Click
End Sub

```

—
|
存程序
|
— 資料儲

```

Private Sub Text1~Text54_GotFocus()
    MarkAll Text1
End Sub

```

```
Public Sub MarkAll(Txt As TextBox)
    Txt.SelStart = 0
    Txt.SelLength = Len(Txt.Text)
End Sub
```

```
Private Sub undo_Click()
    If data1.EditMode = dbEditAdd Then
        data1.Recordset.CancelUpdate
        data1.Recordset.MoveLast
    Else
        data1.UpdateControls
    End If
End Sub
```

—
|
資料
復原
程序
|
—

第十二章 家庭訪問活動日記資料庫

整個家庭訪問調查大致可分為資料發放與資料回收，而資料回收後之資料庫建立步驟如圖 12-1 所示。

12-1 聯繫問卷填寫不全之家戶

問卷回收後先檢查該家戶是否填寫完全，若資料不全，則由研究人員以電話或親自前往該家戶進行補調工作，期使得來不易的家庭訪問調查資料能具有完整性。

12-2 人工讀卡

確定問卷資料完整後，以人工方式將家庭訪問問卷內容中答案為文字格式之地址或地點轉化為查通分區代碼。

12-3 問卷資料電腦化

1. 為便於資料的歸類，家庭訪問資料的輸入工作根據問卷設計分為表一與表二，表一住戶基本資料部分由程式 ODHOME 輸入，表二個人旅次資料則透過程式 ODTRIP 進行輸入工作，表一及表二數化後之問卷資料均以 Microsoft Access 資料庫的儲存格式儲存為*.mdb 檔，

ODHOME 檔格式請參見表 12-1，程式請參見表 12-2，ODTRIP 檔格式請參見表 12-3，程式則請參見表 12-4。

2. 將*.mdb 檔經由轉檔程序轉為 UfosNet 可讀取的*.dbf 檔格式，完成問卷資料電腦化。

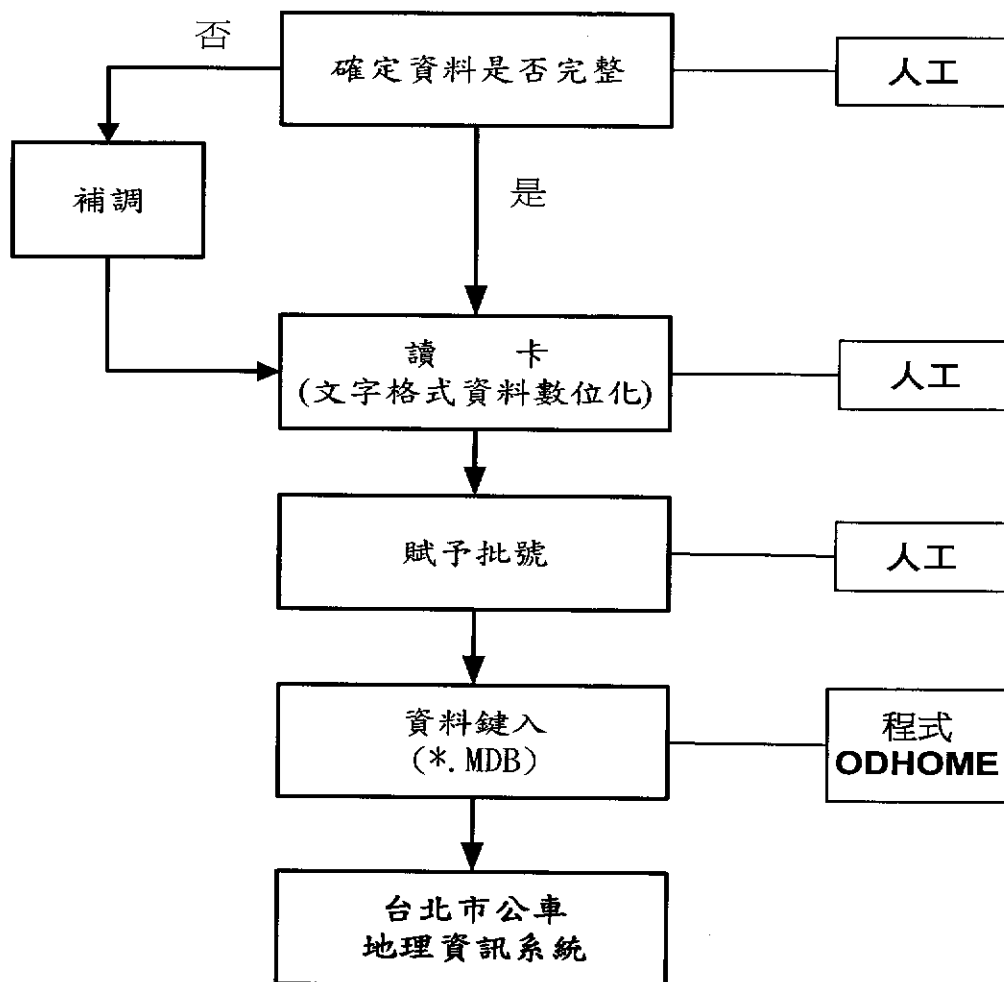


圖 12-1 家庭訪問資料庫建立步驟

表 12-1 ODHOME 資料輸入檔案格式說明表

資料欄位編號	問卷欄位名稱	MDB 檔欄位名稱	欄位大小
1	抽樣住戶編號	抽樣住戶編號	4
2	填表人	填表人	8
3	填表日期(月)	填表月	1
4	填表日期(日)	填表日	2
5	住址	住址	40
6		住址區碼1	3
7		住址區碼2	3
8		住址區碼3	3
9		住址區碼4	3
10	1-與戶長關係	1-與戶長關係	
11	1-年齡	1-年齡	
12	1-性別	1-性別	
13	1-教育程度	1-教育程度	
14	1-職業	1-職業	
15	1-每週使用公車次數	1-每週使用公車次數	
16	1-殘障	1-殘障	
17	1-月薪	1-月薪	
18	1-地點名稱	1-地點名稱	
19	1-地址	1-地址	
20	1-工作就學區碼1	1-工作就學區碼1	
21	1-工作就學區碼2	1-工作就學區碼2	
22	1-工作就學區碼3	1-工作就學區碼3	
23	1-工作就學區碼4	1-工作就學區碼4	
24	1-是否在家工作	1-是否在家工作	
25~135		
136	電話號碼	電話號碼	12
137	小客車_輛	小客車數	1
138	機車_輛	機車數	1
139	腳踏車_輛	腳踏車數	1
140	小貨車_輛	小貨車數	1

141	是否有自用停車位選項	是否有自用停車位	1
142	自用停車位_車位	自用停車位數	1
143	是否租用停車位選項	是否租用停車位	1
144	租用停車位_車位	租用停車位數	1
145	必需在路邊尋找停車位選項	需在路邊尋找停車位	1
146	小巷中可找停車之空間選項	需在巷子裡尋找停車位	1
147	其他選項	使用其他停車位	1
148	其他__(請說明)	其他停車位說明	20
149	每週七天需並排停車之天數, _天	每週需並排停車天數	1
150	自有選項	住所是否自有	1
151	租用選項	住所是否租用	1
152	親戚擁有選項	住所是否親戚所有	1
153	其他選項	住所其他	1
154	住所其他說明__	住所說明	20
155	已在現地居住日期	已居住年	2
156		已居住月	2
157	全家無共乘選項	全家無共乘	1
158	上班共乘選項	上班共乘	1
159	上班共乘人員_	上班共乘人員	1
160	上班偶爾共乘選項	上班偶爾共乘	1
161	上班偶爾共乘人員_	上班偶爾共乘人員	1
162	下班共乘選項	下班共乘	1
163	下班共乘人員_	下班共乘人員	1
164	下班偶爾共乘選項	下班偶爾共乘	1
165	下班偶爾共乘人員_	下班偶爾共乘人員	1
166	是居住在捷運線附近選項	是居住在捷運線附近	1
167	否居住在捷運線附近選項	否居住在捷運線附近	1
168	為公寓選項	是否為公寓	1
169	為大廈選項	是否為大廈	1
170	為透天厝選項	是否為透天厝	1
171	其他選項	是否其他	1
172	其他, 請說明__	住所其他說明	14
173	33萬元以下選項	33以下	1

174	33萬元-59萬元選項	33-59	1
175	60萬元-89萬元選項	60-89	1
176	90萬元-119萬元選項	90-119	1
177	120萬元-149萬元選項	120-149	1
178	150萬元-178萬元選項	150-178	1
179	179萬元-334萬元選項	179-334	1
180	334萬元以上選項	334以上	1

*註:家庭成員結構個人編號 2~9, 資料欄位編號 25~135,

因格式與個人編號 1 相同, 在此省略。

表 12-2 程式 ODHOME 列表及程式說明

Option Explicit

Private UpdateCancelled As Boolean

Private Sub add_Click()

Dim i As Integer

Data1.Recordset.AddNew

Data2.Recordset.AddNew

Text1 = "xxxx"

Text2 = "xxxxxxxxx"

Text3 = "x"

Text4

"xx"

Text5 = "xxxxxxxxxxxxx"

Text6 = "xxx"

Text7 = "xxx"

Text8 = "xxx"

Text9 = "xxx"

For i = 0 To 8

Text10(i) = "xxx"

Text11(i) = "xx"

Text12(i) = "x"

Text13(i) = "x"

Text14(i) = "xx"

Text15(i) = "xx"

Text16(i) = "x"

Text17(i) = "xxxxx"

```

Text18(i) = "xxxxxx"
Text19(i) =
"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
Text20(i) = "xxx"
Text21(i) = "xxx"
Text22(i) = "xxx"
Text23(i) = "xxx"
Text24(i) = "x"
Next i
Text25 = "x"
Text26 = "x"
Text27 = "x"
Text28 = "x"
Text29 = "x"
Text30 = "xxxxxxxxxxx"
Text31 = "x"
Text32 = "xxxxxxxxxxxxxxxx"
Text33 = "xx"
Text34 = "xx"
Text35 = "xx"
Text36 = "xx"
Text37 = "xx"
Text38 = "xx"
Text39 = "xxxxxxxxxxx"
Text40 = "xx"

Text1.SetFocus

End Sub

```

設
定
變
數
初
始
值


```
Private Sub Data1_Validate(Action As Integer, save
As Integer)
```

```
Dim msg As String
```

```
If save = True Or Action = vbDataActionUpdate _
Or Data1.Recordset.EditMode =
```

```
dbEditAdd _
```

```
Then
```

```
If Text1 = "" Then
```

```
msg = "必須輸入抽樣住戶編號"
```

```
Text1.SetFocus
```

```
End If
```

```
End If
```

```
If msg <> "" Then
```

```
MsgBox msg, vbExclamation
```

```
Action = vbDataActionCancel
```

```
UpdateCancelled = True
```

```
Else
```

```
UpdateCancelled = False
```

```
End If
```

```
End Sub
```

```
Private Sub del_Click()
```

```
Dim msg As String
```

```
msg = "確定要刪除這筆資料?"
```

```
msg = msg & If(Text2 <> "", Text2, "這筆資料") &
""
```

```
If MsgBox(msg, vbQuestion + vbYesNo +
vbDefaultButton2) = vbYes Then
```

```
Data1.Recordset.Delete
```

```
Data1.Recordset.MoveNext
```

```
If Data1.Recordset.EOF Then
Data1.Recordset.MoveLast
```

```
End If
```

```
End Sub
```

```
Private Sub exit_Click()
```

```
Unload Form1
```

```
End Sub
```

```
Private Sub Form_Load()
```

```
Load Form1
```

```
Form1.Data1.Refresh
```

```
Form1.Data2.Refresh
```

```
Form1.Show
```

錯
誤
檢
查

資
料
刪
除
程
序

```
add_Click
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
If UpdateCancelled Then
    Cancel = True
ElseIf Data1.Recordset.EditMode = dbEditAdd Then
    Data1.Recordset.Update
ElseIf Data2.Recordset.EditMode = dbEditAdd Then
    Data2.Recordset.Update
End If
End Sub
```

```
Private Sub Save_Click()
```

```
Rem data1.Recordset.Update
    Data2.Recordset.Update
    add_Click
End Sub
```

—
|
存 資
程 料
序 儲
|
—

```
Private Sub Text1~Text40_GotFocus()
MarkAll Text1
End Sub
```

```
Public Sub MarkAll(Txt As TextBox)
```

```
    Txt.SelStart = 0
    Txt.SelLength = Len(Txt.Text)
```

```
End Sub
```

```
Private Sub undo_Click()
If Data1.EditMode = dbEditAdd Then
    Data1.Recordset.CancelUpdate
    Data1.Recordset.MoveLast
Else
    Data1.UpdateControls
End If
End Sub
```

—
|
資 料
料 復
復 原
原 程
程 序
|
—

表 12-3 ODTRIP 資料輸入檔案格式說明表

資料欄位編號	問卷欄位名稱	MDB 檔欄位名稱	欄位大小
1	抽樣住戶編號	抽樣住戶編號	4
2	個人編號	個人編號	1
3	日期(月)	日期(月)	1
4	日期(日)	日期(日)	2
5	星期	星期	2
6	出發起點為家選項	出發起點為家	1
7	出發起點為其他選項	上一起點為其他	1
8	出發起點其他地址	出發起點其他地址	40
9		其他區碼1	3
10		其他區碼2	3
11		其他區碼3	3
12		其他區碼4	3
13	活動編號	活動編號	1
14	前往地點名稱	前往地點名稱	20
15	地址	地址	40
16		地址區碼1	3
17		地址區碼2	3
18		地址區碼3	3
19		地址區碼4	3
20	1轉車選項	1轉車	1
21	2回家選項	2回家	1
22	3上學選項	3上學	1
23	4工作選項	4工作	1
24	5例行性購物選項	5例行性購物	1
25	6非例行性購物選項	6非例行性購物	1
26	7專程接送親友選項	7專程接送親友	1
27	8順道接送親友選項	8順道接送親友	1
28	9商務, 洽公選項	9商務, 洽公	1
29	10社交, 休閒娛樂選項	10社交, 休閒娛樂	1
30	11私人事務選項	11私人事務	1
31	12用餐選項	12用餐	1

32	13就醫選項	13就醫	1
33	14其他選項	14其他	1
34	活動出發時	出發時	2
35	活動出發分	出發分	2
36	活動到達時	到達時	2
37	活動到達分	到達分	2
38	活動停留時	停留時	2
39	活動停留分	停留分	2
40	活動旅行時	旅行時	2
41	活動旅行分	旅行分	2
42	公車可否到達選項	公車可否到達	1
43	1-步行選項	1-步行	1
44	1-步行時間_	1-步行時間	2
45	1-等候選項	1-等候	1
46	1-等候時間_	1-等候時間	2
47	1-搭乘車種_	1-搭乘車種	1
48	1-公車路線_	1-公車路線	12
49	1-再轉乘其他交通工具選項	1-再轉乘其他交通工具	1
50	2-步行選項	2-步行	1
51	2-步行時間_	2-步行時間	2
52	2-等候選項	2-等候	1
53	2-等候時間_	2-等候時間	2
54	2-搭乘車種_	2-搭乘車種	1
55	2-公車路線_	2-公車路線	12
56	2-再轉乘其他交通工具選項	2-再轉乘其他交通工具	1
57	3-步行選項	3-步行	1
58	3-步行時間_	3-步行時間	2
59	3-等候選項	3-等候	1
60	3-等候時間_	3-等候時間	2
61	3-搭乘車種_	3-搭乘車種	1
62	3-公車路線_	3-公車路線	12
63	3-再轉乘其他交通工具選項	3-再轉乘其他交通工具	1
64	4-私人運具停車費_	4-私人運具停車費	3
65	4-乘員共_人	4-乘員	2

66	4-計程車資__元	4-計程車資	3
67	4-公車車資__元	4-公車車資	3
68	4-台汽車資__元	4-台汽車資	3
69	4-捷運車資__元	4-捷運車資	3
70	4-停車難易度選項	4-停車難易度	1
71	中山北路選項	中山北路	1
72	中山南路選項	中山南路	1
73	松江路選項	松江路	1
74	新生南路選項	新生南路	1
75	敦化北路選項	敦化北路	1
76	敦化南路選項	敦化南路	1
77	民權東路選項	民權東路	1
78	南京東路選項	南京東路	1
79	仁愛路選項	仁愛路	1
80	信義路選項	信義路	1
81	重陽橋選項	重陽橋	1
82	台北橋選項	台北橋	1
83	中興橋選項	中興橋	1
84	華江橋選項	華江橋	1
85	福和橋選項	福和橋	1
86	中正橋選項	中正橋	1
87	永福橋選項	永福橋	1
88	忠孝橋選項	忠孝橋	1
89	秀朗橋選項	秀朗橋	1

表 12-4 程式 ODTRIP 列表及程式說明

Private Sub Command1_Click()	
Dim fn As String, FL As Long	
fn = Text2.Text	
sourcefile = "c:\odtrip\odtrip.mdb"	
targetfile = "c:\odtrip\trip" + fn + ".mdb"	
On Error GoTo FileNotExist	
FL = FileLen(targetfile)	
GoTo DoRest	
FileNotExist:	
Resume CopyFile	開啟待輸入資料檔
CopyFile:	
FileCopy sourcefile, targetfile	
DoRest:	
On Error GoTo 0	
Load Form1	
Form1.data1.DatabaseName = targetfile	
Form1.data1.Refresh	
Form1.Show	
Unload Me	
End Sub	
Private Sub Text1_GotFocus()	
Text1.SelStart = 0	
Text1.SelLength = Len(Text1.Text)	
End Sub	
Private Sub Text2_GotFocus()	
Text2.SelStart = 0	
Text2.SelLength = Len(Text2.Text)	
End Sub	
Option Explicit	
Private UpdateCancelled As Boolean	
Private Sub add_Click()	
data1.Recordset.AddNew	
Text1 = "xxxx"	

```
Text2 = "x"  
Text3 = "x"  
Text4 = "xx"  
Text5 = "xx"  
Text6  
"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"  
Text7 = "xxx"  
Text8 = "xxx"  
Text9 = "xxx"  
Text10 = "xxx"  
Text11 = "x"  
Text12 = "xxxxxxxxxxxxx"  
Text13  
"xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"  
Text14 = "xxx"  
Text15 = "xxx"  
Text16 = "xxx"  
Text17 = "xxx"  
Text18 = "xx"  
Text19 = "xx"  
Text20 = "xx"  
Text21 = "xx"  
Text22 = "xx"  
Text23 = "xx"  
Text24 = "xx"  
Text25 = "xx"  
Text26 = "xx"  
Text27 = "xx"  
Text28 = "k"  
Text29 = "xxxxxxxxxxxxx"  
Text30 = "xx"  
Text31 = "xx"  
Text32 = "k"  
Text33 = "xxxxxxxxxxxxx"  
Text34 = "xx"  
Text35 = "xx"  
Text36 = "x"  
Text37 = "xxxxxxxxxxxxx"  
Text38 = "xxx"  
Text39 = "xx"  
Text40 = "xxx"  
Text41 = "xxx"  
Text42 = "xxx"
```

設定變數初始值

```

Text43 = "xxx"
Text44 = "x"
Text1.SetFocus

End Sub

Private Sub Data1_Validate(Action As Integer, save
As Integer)
Dim msg As String
If save = True Or Action = vbDataActionUpdate _
Or data1.Recordset.EditMode =
dbEditAdd _
Then
If Text1 = "" Then
msg = "必須輸入抽樣住戶編號"
Text1.SetFocus
End If
End If
If msg <> "" Then
MsgBox msg, vbExclamation
Action = vbDataActionCancel
UpdateCancelled = True
Else
UpdateCancelled = False
End If
End Sub

Private Sub del_Click()
Dim msg As String
msg = "確定要刪除這筆資料?"
msg = msg & IIf(Text1 <> "", Text1, "這筆資料") &
"?"
If MsgBox(msg, vbQuestion + vbYesNo +
vbDefaultButton2) = vbYes Then
data1.Recordset.Delete
data1.Recordset.MoveNext
If Data1.Recordset.EOF Then
Data1.Recordset.MoveLast
End If
End Sub

Private Sub exit_Click()
Unload Form1
End Sub

```

錯
誤
檢
查

資
料
刪
除
程
序


```
Private Sub Form_Activate()  
    add_Click  
End Sub
```

```
Private Sub Form_Load()  
    data1.Refresh  
    Me.Show  
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)  
If UpdateCancelled Then  
    Cancel = True  
ElseIf data1.Recordset.EditMode = dbEditAdd Then  
    data1.Recordset.Update  
End If  
End Sub
```

```
Private Sub Save_Click()  
    data1.Recordset.Update  
    add_Click  
End Sub
```

```
Private Sub Text1~Text44_GotFocus()  
MarkAll Text1  
End Sub
```

```
Public Sub MarkAll(Txt As TextBox)  
    Txt.SelStart = 0  
    Txt.SelLength = Len(Txt.Text)  
End Sub
```

```
Private Sub undo_Click()  
If data1.EditMode = dbEditAdd Then  
    data1.Recordset.CancelUpdate  
    data1.Recordset.MoveLast  
Else  
    data1.UpdateControls  
End If  
End Sub
```

—
|
載入
—
離開
輸入主程式
|
—

—
存
程
序
|
—
資料儲

—
|
資料
復
原
程
序
|
—

第三部份

地理資訊系統應用範例說明

第十三章 地理資訊系統

應用範例說明

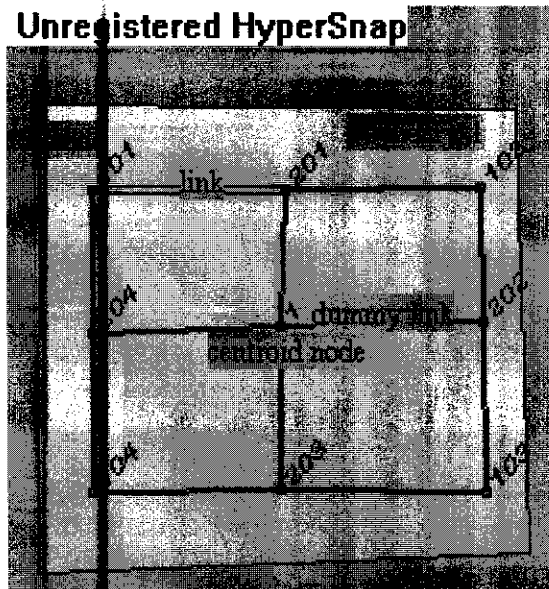
本章節將利用地理資訊系統 UFOSNET 應用程式為操作介面，分別針對道路路網及大眾運輸路網的構建方式，列舉幾個簡單的範例說明。

13-1 道路路網應用範例

將應用前面章節所敘述 UFOSNET 之基本功能，由節點(Node)、節線(Link)、分區(Zone)、註釋(Annotation)及邊界(Boundary)之建立至構成一簡單路網，而各屬性資料的輸入及查詢也將做一說明。範例 13.1 為基本路網的建立、範例 13.2 為各資料的輸入方式、範例 13.3 為介紹各資料如何查詢，以下分別敘述之。

範例 13.1：

構建一「口」字型路網，應包含分區(Zone)、節點(Node)、節線(Link)、邊界(Boundary-1)、註釋(Annotation)、中心點(Centroid Node)、虛擬節線(Dummy Link)基本構成要素，其中各節線皆為雙向。(路網上 Node、Link、Boundary-1、Centroid Node、Dummy Link 為註釋層所繪出)，結果如下圖：



步驟：

1. 進入 UFOSNET 應用程式開啟一新視窗，必須由構建節點開始。

而在構建網絡節點之前，必須預設節點多少以下為中心區節點編號，如圖 13-1 對話視窗，同時可設定分區節點的顏色。待後面節點編號輸入後系統自動會辨識此節點為分區節點或路網節點。

(本範例即可假設 100 以下為中心區節點號碼，100 以上為道路節點編號)

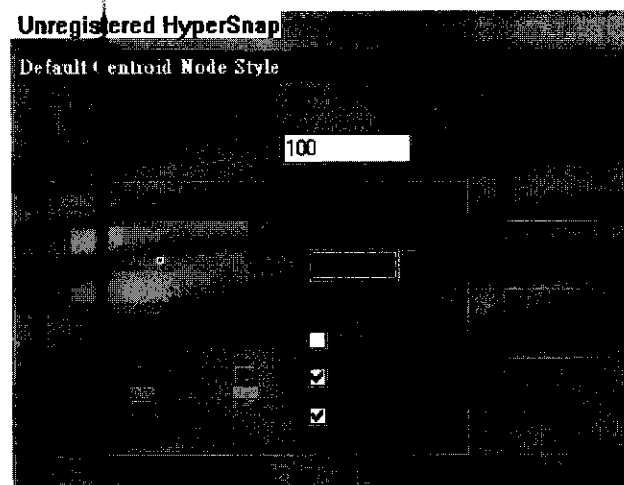




圖 13-1 預設中心區節點視窗

2. 由 UFOSNET 視窗上方工具列按快速鍵  即可開啟路網編輯

工具箱，此工具箱包含節點、節線及分區之編輯功能。

- 3.由路網編輯工具箱中按鍵即  可增加所需的節點數，同時會出現加入節點的對話方塊，如圖 13-2，分別輸入節點編號 1、101、102、103、104、201、202、203、204，並將各節定位於路網空間上相對應位置。(系統自動認定 1 為中心區節點)。

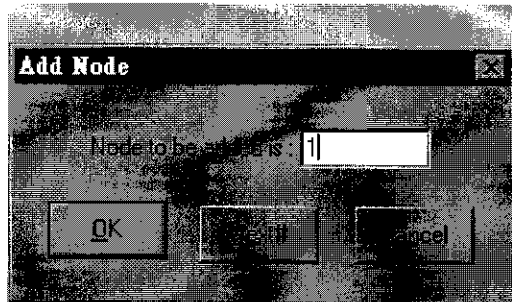



圖 13-2 加入節點視窗

- 4.同樣由路網編輯工具箱中按  鍵後，再選擇起始節點，拖曳至節線終點，將出現如圖 13-3 對話視窗，確認節點編碼後，若全節線接設為雙向通行，則選取 2-Way 功能。而重複此工作，分別把各節點所對應的節線連結起來，如「田」字。(本範例共計路網節線 16 條、虛擬節線 8 條)

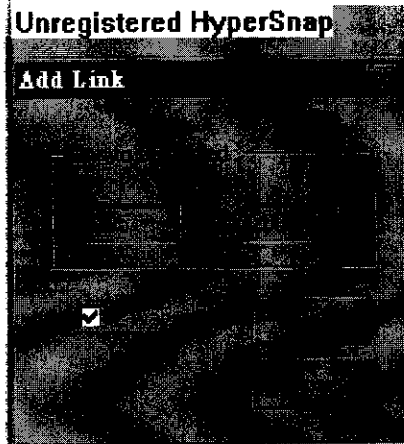


圖 13-3 增加節線視窗




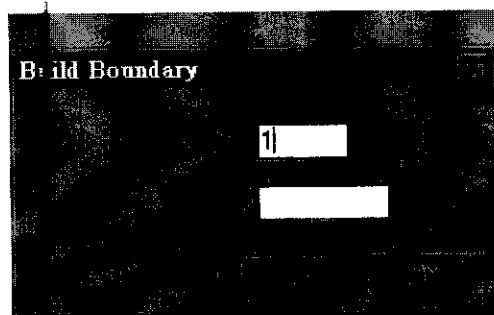

5. 接著為構建分區(Boundary)，於工具列上按  鍵將出現分區編輯工具箱。接著在此工具箱中按鍵加入分區邊界  所需的點數(本範例為4點)，再按  鍵後，螢幕將出現如圖 13-4 對話視窗方塊，輸入分區之編碼為1與文字標記(可不輸入)，選擇 OK 鍵後，再一次點選預先建好的分區邊界點，即可構成一個閉多邊形後，最後按鍵即告完成。

圖 13-4 構建新分

區對話方塊



6. 接下來即為各節點、節線、中心點、分區邊界構建註釋層。

按 UFOSNET 視窗上方工具列  鍵後螢幕出現其工具箱，再


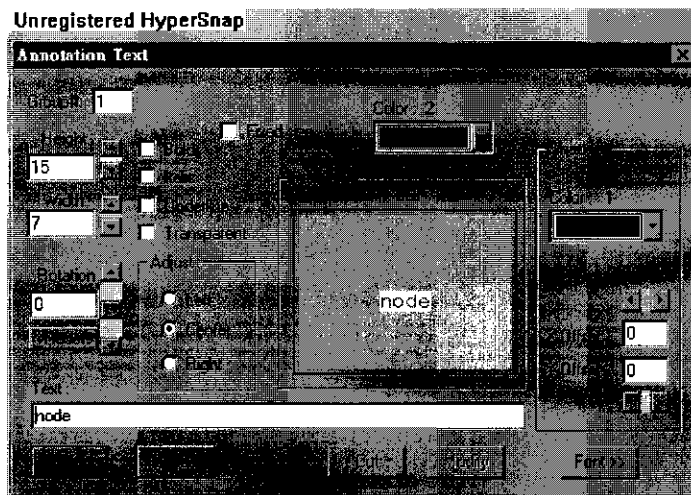

按註釋層工具箱中的按  鍵，將出現編輯文字對話視窗，如

圖 3-15 所示，可選擇所需輸入文字、字型、顏色及大小。

圖 3-5 定義文字型式對話視窗



7. 接者於註釋工具箱內按  鍵後，移動游標於適當位置按一下，


即可加入該註釋。重複步驟 6、7 分別加入 link、centroid node、

dummy link、boundary -1 註釋於適當位置，即告完成。

範例 13.2

以範例 13.1 所構建路網，分別定義各節點(Node)、節線(Link)、分區(Zone)之欄位名稱及資料的輸入。

步驟：

1. 按工具列中鍵， 此鍵為資料定義庫(Database Definition)，會

出現如圖 13-6 對話視窗，可針對節點(Node)、節線(Link)、分區(Zone)、點(Point)、F 因子分別定義各資料所需欄位、欄位名、儲存型式，每欄位定義完後要按 UPDATE 才算完成。

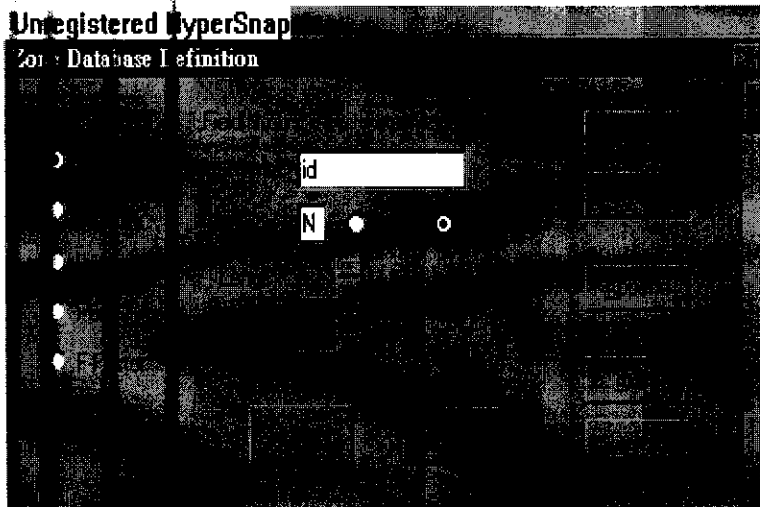



圖 13-6 定義各資料欄位

2. 節點(Node)、節線(Link)、分區(Zone)欄位定義完成後，即可分別進行各欄位的輸入，其輸入的方法有兩種，以下分別敘述之。

(一) 利用 UFOSNETg 視窗工具列上(Open  Datable Editor)快速鍵，即可開啟各資料欄位的輸入視窗，如圖 13-7 所示。同時可利用該視窗 Table 選項選擇所要輸入節點(Node)、節線(Link)、分區(Zone)等各欄位資料表，此方法可同時輸入多筆資料而不用更換該視窗。

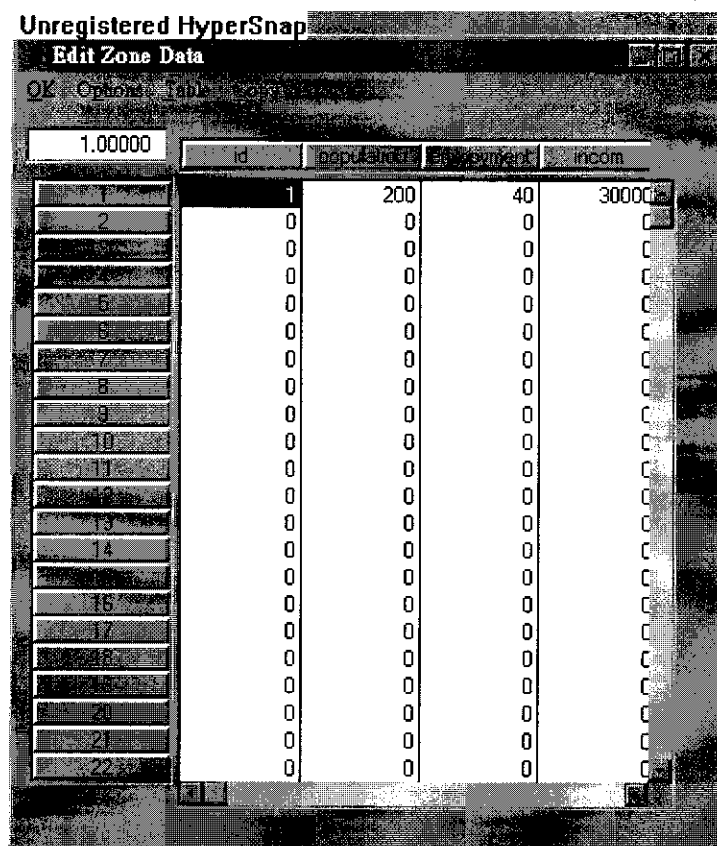



圖 13-7 資料編輯視窗

(二) 利用 UFOSNET 視窗工具列，按快速鍵  即可開啟路網

編輯工具箱，如圖 13-8 所示。此工具箱包含各節點、節線

及分區之編輯功能。

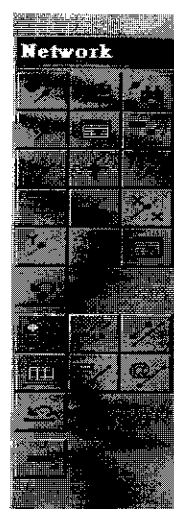
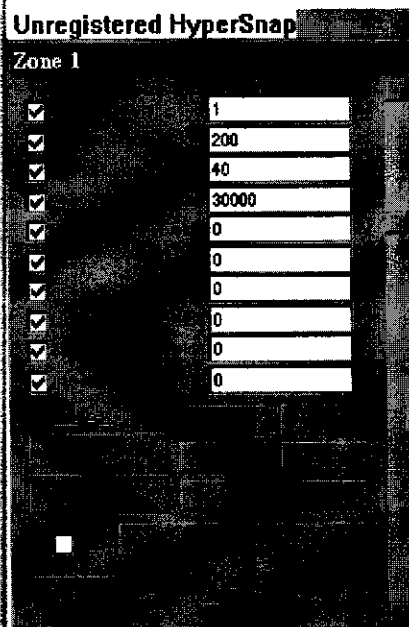


圖 13-8 路網編輯工具箱


(1) 按  鍵，即可進行分區(Zone)資料編輯視窗，如圖 13-9

所示。即可按照各分區輸入各欄位資料值，等視窗資料全部輸入完畢後，按 Modify 已完成。

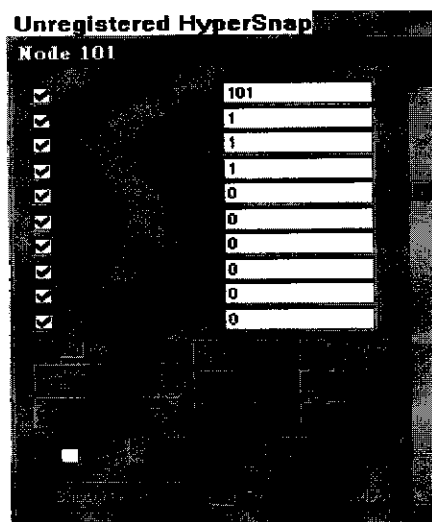


Zone 1	Value
<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	200
<input checked="" type="checkbox"/>	40
<input checked="" type="checkbox"/>	30000
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0

圖 13-9 分區(Zone)資料編輯視窗


(2) 按  鍵，即可進行節點(Node)資料編輯視窗，如圖

13-10 所示。即可按照各節點輸入各欄位資料值，等視窗資料全部輸入完畢後，按 Modify 以登入。



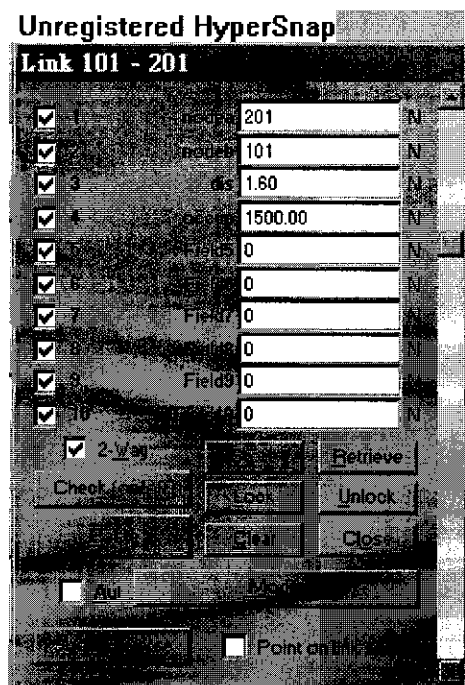
Node 101	Value
<input checked="" type="checkbox"/>	101
<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	1
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0
<input checked="" type="checkbox"/>	0

圖 13-10 節點(Node)資料編輯視窗

(3) 按  鍵，即可進行節線(Link)資料編輯視窗如圖 13-11 所

示。即可按照各節線輸入各欄位資料值，等視窗資料全部輸入

完畢後，按 Modify 以登入。



Field	Value	Unit
nodea	201	N
nodeb	101	N
dis	1.60	N
occup	1500.00	N
Field5	0	N
Field6	0	N
Field7	0	N
Field8	0	N
Field9	0	N
Field10	0	N

圖 13-11 節線(Link)資料編輯視窗

3.待節點(Node)、節線(Link)、分區(Zone)等全部欄位資料輸入完成。

範例 13.3

以範例 13.1 所構建路網中，查詢道路路網上節線(Link)上，
容量(occupancy)大於 2000 者，並於實際道路路網中展示出來。

步驟：

1.由 UFOSNET 視窗上方 Plot 選項下的 QueryObjects/Links，出現

節線選擇展示(Select Link Plot)視窗，如圖 13-12 所示。

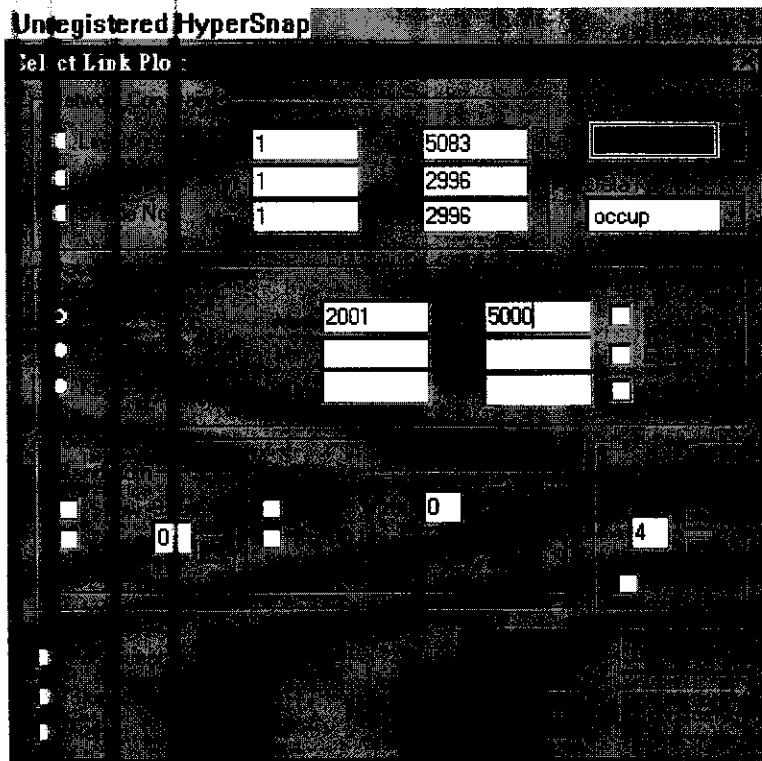


圖 13-12 節線選擇展示對話視窗

2. 由主要資料(Data Key)中選擇節線篩選欄位為容量(Occupancy)及選擇資料限制為 Key1 中 Min 值為 2001、Max 值為 5000 (最大值可使用者自訂)，同時可選則該節點之顏色及線條寬度。

3. 按 New Query 鍵，即可展示出道路容量在 2000 以上的路網，如

■ 13-13 所示。

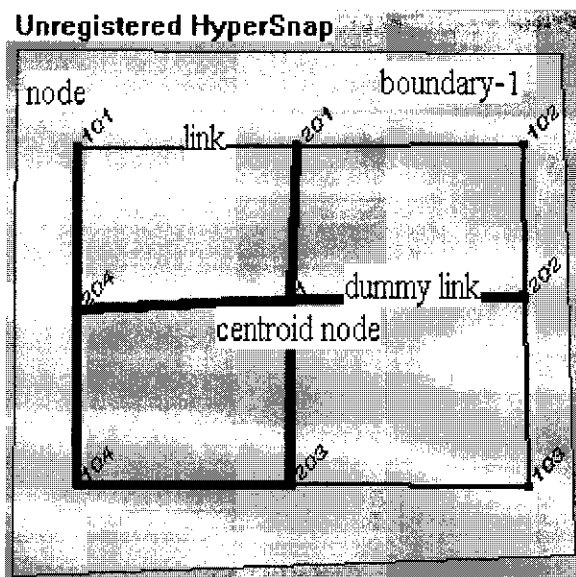


圖 13-13 節線容量大於 2000 路網

4.同時可由編輯節線資料視窗中，得知其篩選結果。如圖 13-14 所示。

Unregistered HyperSnap

Edit Link Data

Copy

1.00000

nodes	nodes	nodes	nodes
101	1	0.10	3000.00
1	203	0.10	3000.00
202	1	0.10	3000.00
201	1	0.10	3000.00
101	204	1.60	2300.00
201	101	1.60	1500.00
202	102	1.90	1800.00
102	201	1.10	1800.00
203	103	1.80	1800.00
103	202	1.60	1900.00
204	104	1.80	2200.00
104	203	1.50	2300.00
201	1	0.10	3000.00
102	201	1.10	1800.00
201	101	1.60	1500.00
202	1	0.10	3000.00
103	202	1.60	1900.00
202	102	1.90	1800.00
1	203	0.10	3000.00
104	203	1.50	2300.00
203	103	1.80	1800.00
204	1	0.10	3000.00

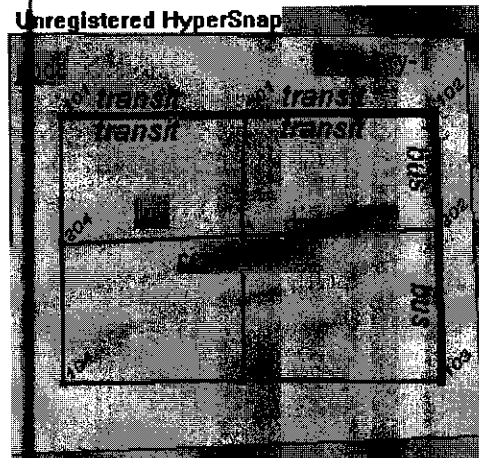
圖 13-14 節線容量大於 2000 詳細展示視窗

13-2 大眾運輸路網應用範例


大眾運輸路網基本上為架構於道路路網上，所以其路線的建立較為簡單，本節將選擇捷運及公車路線，說明其如何構建於道路路網上及如何查詢、展示該系統所經過路線。範例 13.4 為介紹大眾路網的建立、範例 13.5 為介紹如何查詢及展示大眾路網，以下分別敘述之。

範例 13.4

於範例 13.1 所繪出路網中，上方路網中加入「雙向捷運路線(transit)」及右邊路網中加入「單向公車路線(bus)」，如下圖所示。



步驟：

1. 於 UFOSNET 視窗上按快速鍵  即可開啟此工具箱，此工具箱包含建立大眾運輸路線(Transit Lines)、編輯大眾運輸路段(Transit Segment)、編輯查詢大眾運輸資料等功能。

2.在建立新大眾運輸路網之前有一點必須注意的，就是必須

先定義大眾運輸系統的運具(Mode)型式及車輛(Vehicle)型式，才能進行大眾運輸路網的構建。關於運具型式定義方面按快速鍵



選取，出現如圖 13-15 的對話視窗。輸入如公車、捷運、計程車等，每一大眾運輸路線必須給予其一種運具模式，每種運具必須給予其一個字母長度的代號，而每種運具可給定不同的顏色

代表。車輛型式方面按快速鍵



，出現如圖 13-16 對話視窗。

此功能提供 99 種車輛型式，車輛型式 1(Vehicle-1)即為公車型式之一種，同時可輸入每一型式包含數種描述營運之屬性可用來計算績效指標。

圖 13-15 運具型式選擇對話視窗

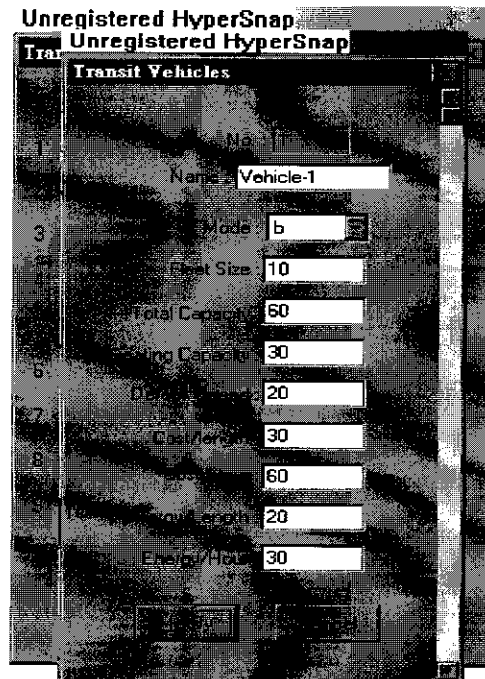



圖 13-16 車輛型式選擇對話視窗

3.接下來，進行捷運路線的建立，按快速鍵  點選後，

出現如圖 13-17 對話視窗，於屬性中 line label 值輸入「transit」

和 line mode 中選擇「m/mrt」運具模式，而其他屬性資料使用

者可自行定義輸入值。

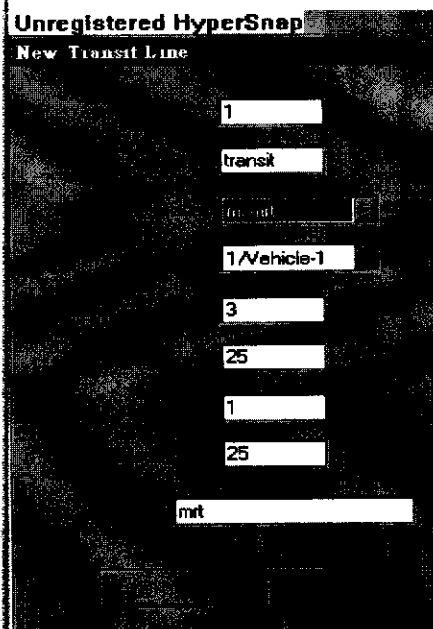


圖 13-17 建立捷運路線視窗

4.於步驟 3 視窗按 OK 之後，移動游標點於節點 101、201、102、

201、101 五點即可構成一雙向捷運路網，最後需按鍵，才算完成，

結果如圖  13-18。

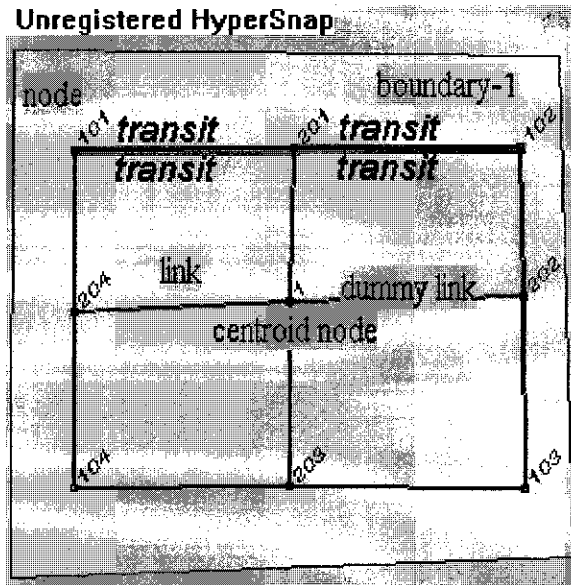



圖 13-18 捷運路網的構建

5.重複步驟 3，將圖 13-17 對話視窗中的 line no 設為「2」，於屬性中 line label 值輸入「bus」及 line mode 中選擇「b/bus」運具模式，而其他屬性資料使用者依然可自行選擇輸入值。

6.於步驟 5 視窗中按 OK 之後，移動游標點於節點 102、202、103 三點即可構成一南北單向公車路線，最後同樣需要按鍵，才算完成，結果  如圖 13-19。

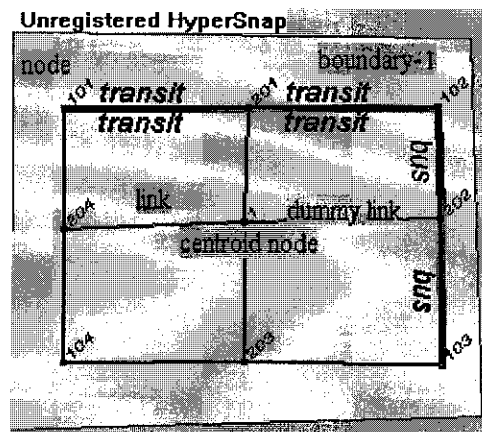


圖 13-19 公車路線的構建


範例 14.5

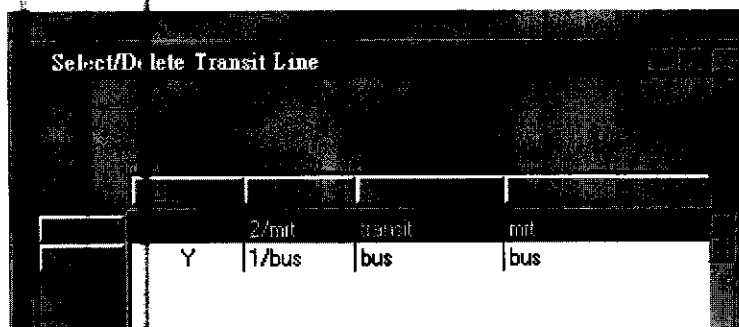
如何查詢並展示出範例 14.4 中，捷運路線資料所經過

路線？

步驟：

1. 查詢方面：

(1) 由大眾運輸路線編輯工具箱中按  快速鍵，出現如圖 13-20




對話視窗，選擇捷運路線。

圖 13-20 選擇/刪除路線視窗

(2) 接著按  快速鍵，顯示及變更捷運路線屬性資料，如圖 13-21

所示。

2. 顯示方面：

(1) 一樣由大眾運輸路線編輯工具箱中按  快速鍵，出現如圖

13-20 對話視窗，選擇捷運路線。

(3) 選擇視窗中 Style 選項，即出現大眾運輸路線型式視窗，如圖

13-22 所示，可選擇經過路網上所要展示出的線條寬度及顏

色，按 OK 以完成。

Unregistered HyperSnap

Transit Line Information

Prev Next

Transit Line Info

ID: [] Label: transit Company: 0

Headway: 3.00

Default Speed: 25.00 Mode: m/mt

Default dwell time: 1 Vehicle Type: 1/Vehicle-1

Default layover time: 5

Name: mt

Boarding Percentage: 0

Modify

First segment: 1

圖 13-21 捷運運輸路線資料

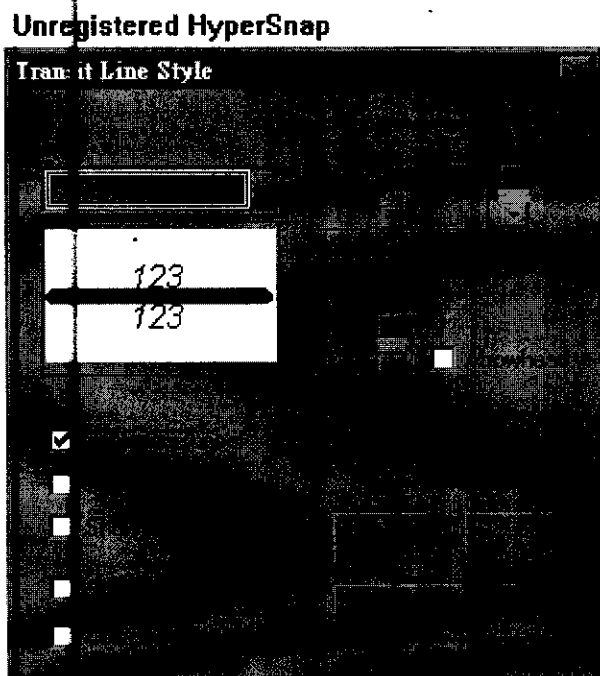
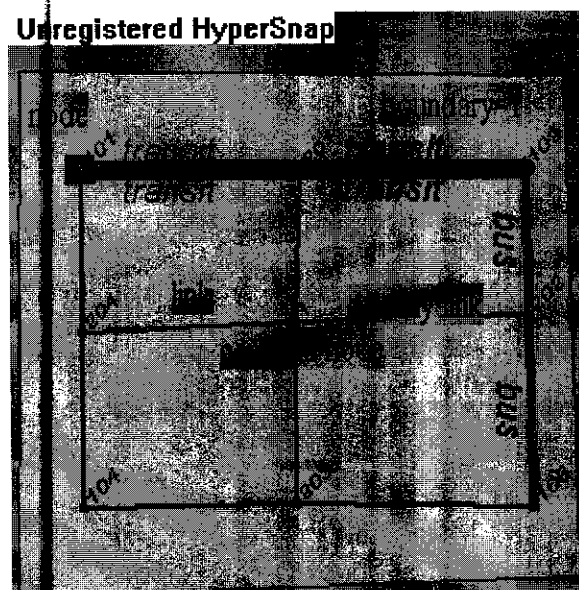


圖 13-22 大眾運輸路線型式

(3)於圖 13-20 選擇/刪除路線視窗選擇 Select 選項，捷運線經過路網將會依步驟(2)中選擇的型態展示於路網上，如圖 13-23 所示。

圖 13-23 捷運路線展示系統



第四部份

台北都會區公車旅次起迄表