



國立台灣科技大學
資訊工程系

碩士學位論文

應用強化學習於道路上交通設施的程序化重建

Procedural Reconstruction of Traffic Infrastructure Using
Reinforcement Learning

研 究 生：張立彥

學 號：M11015029

指導教授：賴祐吉博士

中華民國一百一十三年一月一十一日

中文摘要

近年來，隨著都市規劃的需求增加，工程師和規劃人員已經開始採納先進的電腦圖學技術。其中，數位雙生 (Digital Twin) 是一種在資訊化平台上建構現實世界中物體的「虛擬分身」的技術，透過三維重建技術生成對應的城市模型，並且，結合物聯網、人工智慧能即時反映現實物體的狀態和變化。特別是在道路規劃中，可以透過模擬與觀測交通號誌的狀態，藉此進行車流量的管理以及動線的規劃，進一步將智能交通納入道路規劃中。為此，需要透過模型的重建，以視覺化的方式模擬與呈現各種不同的道路情境，提供設計師與工程人員一個參考。然而，這樣的重建技術也面臨挑戰。由於重建過程完全依賴於圖片資料，雖然整體使用到的照片數量足以重建場景，但對於單一個物件而言，通常只會出現在少數幾個照片當中，造成無法在圖片之間取得足夠的特徵進行匹配，進一步導致重建後的物件出現模型不完整的情況。特定的道路物件，如紅綠燈桿和路燈桿，由於其桿狀形態的特性，在重建過程中便可能因為特徵的不足出現明顯的破碎化問題，如此便無法提供工程人員一個精準的視覺參考。此外，交通設施上附屬的標誌或條紋等元件資訊在智慧交通的技術中也同樣重要，然而，單純的三維重建無法獲得這些資訊，通常需要導入額外的圖像識別技術來獲取。但是，在智慧交通中，需要明確知道每個元件所對應的交通設施，才能避免模擬過程中出現由於號誌與標誌之間錯誤的對應關係造成誤判的情況。考慮到這些挑戰，本論文的目標是能夠同時取得交通設施物件的外型、桿子的參數以及元件 (Component) 的資訊，以參數的形式來定義道路上的交通設施，並且，透過參數來生成與實際桿子相近的模型。因此，在系統上，首先，本論文嘗試從照片中識別出特定的物件，例如交通標誌和紅綠燈號誌。這些相對容易辨識的物體成為了進一步推估整個桿子形狀的基礎。隨後，在系統上，本論文選擇導入強化學習 (Reinforcement Learning, RL) 做為一個推理的框架，根據設計的獎勵函數 (Reward) 機制，能夠學習一套策略，依據照片中的特徵訊息，以逐步的方式推估桿子與元件的參數，並且，透過參數來生成該交通設施以及對應的附屬元件。最後，本論文以現實中的交通設施以及虛擬產生的交通設施作為重建目標，前者為真實存在的交通設施，而後者為現實中不存在，但是，外型合理且較為複雜的交通設施，透過人工方式，標記出桿子與元件的位置、方位、大小、類型等資訊，並且和本系統所建構出的桿子與元件進行比較，藉此證明本系統能夠在少量的照片視角下，程序化生成合理的交通設施桿子與參數準確的元件。

關鍵字：三維重建、強化學習

ABSTRACT

In recent years, with the growing demand for urban planning, engineers and planners have started adopting advanced computer graphics technologies. Among these, Digital Twin is a technology that creates a "virtual counterpart" of real-world objects on an informational platform. By using 3D reconstruction techniques, corresponding city models can be generated. Combined with the Internet of Things (IoT) and artificial intelligence, it allows real-time reflection of the status and changes of physical objects. Specifically, in road planning, traffic signals can be simulated and observed to manage traffic flow and design movement paths, thereby integrating intelligent transportation into road planning.

To achieve this, model reconstruction is necessary to visually simulate and present various road scenarios, providing designers and engineers with a reference. However, this reconstruction process also faces challenges. Since it relies entirely on image data, while the overall number of photos may be sufficient to reconstruct a scene, individual objects often appear in only a few images. This makes it difficult to extract enough features for matching across images, leading to incomplete models of reconstructed objects. Specific road objects, such as traffic light poles and streetlight poles, are particularly prone to fragmentation during reconstruction due to their slender, pole-like shapes and lack of distinctive features. This makes it difficult to provide engineers with accurate visual references.

Additionally, information on components like signs or markings attached to traffic infrastructure is equally important in intelligent transportation systems. However, simple 3D reconstruction cannot capture such details, often requiring the integration of additional image recognition technologies. In intelligent transportation, it is crucial to accurately associate each component with its corresponding traffic infrastructure to avoid misjudgments caused by incorrect relationships between signals and signs during simulations.

Considering these challenges, this thesis aims to simultaneously obtain the shape of traffic infrastructure objects, the parameters of poles, and the details of their components. By defining traffic infrastructure on roads using parameters, the system can generate models of poles that closely resemble the actual ones.

In the proposed system, the first step involves identifying specific objects, such as traffic signs and traffic lights, from images. These relatively easy-to-recognize objects

serve as the foundation for estimating the overall shape of the pole. Subsequently, the system incorporates Reinforcement Learning (RL) as an inference framework. Using a designed reward function, the system learns a strategy to incrementally estimate the parameters of poles and components based on features extracted from images. These parameters are then used to generate models of the traffic infrastructure and their associated components.

Finally, this thesis evaluates the system by reconstructing both real-world and virtual traffic infrastructure. The former includes actual existing traffic infrastructure, while the latter consists of non-existent but realistic and more complex designs. Through manual annotation, the positions, orientations, sizes, and types of poles and components are labeled. These labels are then compared with the poles and components generated by the system to demonstrate that the proposed system can procedurally generate reasonable traffic pole models and accurately parameterized components with only a limited number of image perspectives.

Key word : 3D Reconstruction 、 Reinforcement Learning

目 錄

中文摘要	I
Abstract	II
目錄	IV
圖目錄	VI
表目錄	VII
符號說明	VIII
1 緒論	1
1.1 問題定義	2
1.2 主要貢獻	3
1.3 論文架構	4
2 相關研究	5
2.1 基於圖片的三維重建	5
2.2 逆程序化建模 (Inverse Procedural Modeling)	5
2.3 基於強化學習 (RL) 的布局 (Layout) 推估	6
3 系統總覽	8
3.1 交通設施背景	8
3.2 交通設施布局 (Layout)	9
3.3 系統流程	11
4 系統實現	14
4.1 元件 (Component) 生成	14
4.1.1 影像畸變 (Distortion)	14

4.1.2	元件 (Component) 定位	15
4.1.3	元件 (Component) 朝向與尺寸調整	17
4.2	桿子布局 (Layout) 生成	18
4.2.1	挑選起始元件 (Component)	18
4.2.2	強化學習 (RL) 與狀態 (State)	19
4.2.3	動作 (Action) 設計	21
4.2.4	獎勵函數 (Reward) 與訓練	22
4.2.5	儲存布局 (Layout) 資訊	23
4.2.6	程序化桿子模型重建	24
5	實驗結果與討論	26
5.1	誤差評估	26
5.2	消融測試 (Ablation Study)	27
5.2.1	元件 (Component) 定位參數分析	28
5.2.2	桿子推估參數分析	28
5.3	真實場景實驗	30
5.4	特殊桿子實驗	30
6	結論與展望	35
	參考文獻	36

圖 目 錄

圖 1.1	破碎模型圖例	1
圖 3.1	交通設施規格圖例	9
圖 3.2	交通設施布局 (Layout) 示意圖	10
圖 3.3	總覽	12
圖 4.1	元件定位示意圖	15
圖 4.2	元件調整示意圖	17
圖 4.3	強化學習 (RL) 示意圖	19
圖 4.4	正方形分布取樣點示意圖	21
圖 4.5	強化學習訓練流程圖	22
圖 5.1	特殊桿子虛擬場景圖	31

表 目 錄

表 5-1	不同取樣距離下各元件的姿態誤差結果。	28
表 5-2	不同邊長下熱圖的精準性比較。	29
表 5-3	不同取樣距離下桿子的誤差結果。	30
表 5-4	平拍真實場景結果圖。	32
表 5-5	空拍真實場景結果圖。	33
表 5-6	特殊桿子的元件姿態誤差與桿子誤差結果。	34
表 5-7	特殊桿子結果圖。	34

符 號 說 明

1. **自定義集合**，使用 latex 內建的數學字形 (Mathematical Fonts) 的大寫英文字母。
例如： $\mathcal{P}$ 、 $\mathcal{C}$ 。
2. **變數**，使用大小寫的英文字母或希臘字母，例如： A 、 w 、 Π 。
3. **座標點**，使用小寫英文字母，並且，上面有劃線。例如： $\bar{p}$ 。
4. **向量**，使用小寫英文字母，並且，上面有劃箭線。例如： $\vec{r}$ 。
5. **函數**，使用粗斜體英文大寫字母，例如： $\boldsymbol{R}$ 。
6. **序號**，使用下標符號，例如： S_i 、 S_j 。
7. **型態**，使用上標符號，例如： I^μ 、 I^m 。

第 1 章 緒論

爲了更有效地進行城市規劃，以及隨著電腦圖學技術的逐年進步，營建工程領域引入了三維重建技術，希望能夠在設計的階段提前瞭解特定區域內道路和建築設施的情況，以便在施工之前能夠先做好規劃。其中，道路上的交通設施影響著整個城市的動線設計，如紅綠燈、路燈和交通標誌，在道路規劃中扮演關鍵角色，合理配置能有效調節交通流量，並且，確保交通安全和流暢。爲了在三維的環境下提供設計師對於這些桿狀的交通設施能夠有合理的視覺參考，透過空拍機的應用，可以從不同視角進行道路的拍攝，並且，藉由運動推斷結構 (Structure from motion, SfM) 技術，最終產生出地景的三維模型。然而，由於重建的過程依賴於照片資料，若提供的照片數量過少，或者目標物在背景較爲複雜的城市照片中難以偵測出特徵時，就可能會導致生成的模型不完整，如圖1.1所示。道路上的紅綠燈桿和路燈桿這樣的桿狀物件中，由於其桿身細長的特性，往往在重建過程中無法正確還原其外型，出現模型斷裂的現象。爲了提供設計師一個精準的交通設施模型做爲參考，便於與工程人員進一步討論道路的設計，解決三維模型的破碎化是一個重要的課題。



圖 1.1: 重建後的不完整模型與對應照片，由左而右分別爲紅綠燈重建模型、紅綠燈照片、交通標誌重建模型、交通標誌照片。

另外，圖片中的物件偵測，往往是進行三維重建前的重要步驟，過去在解決圖片中的物件偵測問題時，經常會使用實例分割 (Instance Segmentation) 的技術來達成。然而，在偵測桿子等較細長的物體時，通常分割出來的效果不佳，原因有以下幾點。首先，因爲這些物體在圖片中所占的面積較小的關係，物體的特徵資訊容易在經過深度神經網絡 (Deep Neural Network, DNN) 中的卷積 (Convolution) 計算之後流失。此外，在偵測較細小的物件時，一旦預測位置有輕微的偏差，就可能超出物體的範圍，導致真陽性 (True

Positive) 的結果被檢測為假陽性 (False Positive)。最重要的是，本論文的應用情境是在背景較為複雜的街景照片中，細小物體的邊界更難從背景的雜訊中區分出來，造成最後無法分割出完整的物體，也進一步影響後續任務表現。為了解決上述的問題，本論文避免直接去偵測桿子本身，而是透過對桿子上的物件做偵測，例如紅綠燈的號誌、附加在桿子上的交通標誌等桿子上常見的物件。這些物件相比桿身能夠較精準地被偵測到，並且，能夠協助定位桿身在照片中的大致位置，以提供後續推估桿子參數時一個參考。

隨後，本論文透過將這些元件的位置投影到場景中，以獲得其三維位置。接著，本論文選取一個元件的位置作為起點，找出能看到該元件的所有相機，並且，根據照片中的桿子特徵資訊尋找可能有桿子的座標點，之後，在元件位置周圍進行取樣，篩選出可信度高的取樣點。接著，根據取樣點來推估下一個可能包含桿子的座標點，重複這個過程直到找到沒有足夠多的取樣點。

最後，本論文根據重建的參考對象，提出了兩種情境來驗證系統的可行性。首先，在第一種情境中，本論文會蒐集現實中道路的紅綠燈等交通設施的照片，並且，利用運動推斷結構 (SfM) 的軟體進行點雲的重建，隨後，以人工的方式標記出目標桿子的節點位置以及元件的方向、大小等資訊。而第二種情境中，則是本論文根據現實中的交通設施規則，生成更複雜的結構來做為目標，並且，同樣標記出對應的節點與元件資訊。之後，本論文會將人工標記的資訊，與本系統重建的結果進行誤差比對，來驗證系統的準確度。

1.1 問題定義

本論文旨在提供一套系統，輸入一個區域的照片 I ，與相機 C ，其中，對於每個相機 C_i 都有一張對應的照片 I_i^a 。首先，本論文會偵測區域內的元件 (Component)， O 。接著，系統會選擇一個元件 (Component) 作為起始元件 (Component)， O^s ，並且，在周圍產生取樣點，根據周圍相機的對應照片經由分割器 (Segmentor) 產生的特徵資訊來篩選取樣點，作為後續的布局 (Layout) 生成方向。之後，本系統透過強化學習 (RL)，利用照片 I^* 的特徵資訊作為參考以及目標位置周圍的取樣點，逐步調整桿子三維模型的長度，生成一個外型上接近目標的桿狀交通設施模型， M^* 。

上述的強化學習 (RL) 任務可以使用馬可夫決策過程 (Markov Decision Process，MDP) 來描述，而馬可夫決策過程 (Markov Decision Process，MDP) 是由以下四個部分所組成：

1. **狀態集合 (States, S)**：代表所有可能的狀態，每個狀態 S 為當前觀察位置在照片中的桿子資訊。
2. **動作集合 (Actions, A)**：代表在特定狀態下可以執行的所有可能動作。
3. **轉移機率函數 (Transition Probabilities, P)**：這是一個函數 $P(S_j|S_i, A_i)$ ，代表從狀態 S_i 通過動作 A_i 轉換到狀態 S_j 的機率。
4. **獎勵函數 (Reward, R)**：這是一個函數 $R(S_j, A_i, S_i)$ ，代表從狀態 S_i 通過動作 A_i 轉換到狀態 S_j 的獎勵。

並且，馬可夫決策過程 (Markov Decision Process, MDP) 的目標是找到一個策略，該策略為每個狀態指定一個動作，以便最大化從任何初始狀態開始的預期累積獎勵，並且，這種策略稱為最優策略 (Optimal Policy)。

根據上述，本論文的目标是訓練一個深度神經網絡 (DNN) 做為一個策略 Π ，該策略 Π 能夠為當前三維模型的狀態 S_i 指定一個最適合的動作 A_i ，並且，透過動作 A_i 來調整桿子長度，隨後將觀察位置往前，使狀態轉變為 S_j ，反覆執行後，能夠逐步接近目標三維模型， $\mathcal{M}^*$ 。

1.2 主要貢獻

整體而言，本論文的貢獻如下：

1. **實現基於強化學習 (RL) 的參數及布局 (Layout) 估計**，為了能夠重建道路上的桿子模型，以及對應的附加元件，本論文透過在強化學習 (RL) 中導入基於桿子結構相似性的獎勵函數來訓練一個代理人 (Agent)，能夠以紅綠燈、標誌等物件為起點，以桿子為單位，根據圖片中紅綠燈、標誌等物件附近特徵資訊逐步推估出一個圖 (Graph) 來描述階層式 (Hierarchical) 的桿子布局 (Layout)，隨後，將布局 (Layout) 結合偵測到的元件 (Component) 後產生現實桿子的對應模型。而在傳統的三維重建過程中，若需獲取物體附加元件的詳細資訊與其布局 (Layout)，通過分割器 (Segmentor) 對重建模型或點雲進行分析時，往往會因點雲的稀疏性而導致局部信息不足，造成精度下降或細節丟失的問題，進一步影響布局 (Layout) 的生成。

2. **透過少量照片進行桿狀交通設施的模型重建**，本系統以少量視角照片以及對應的相機參數，通過使用事先設計好的桿子、紅綠燈與標誌等模型來重建出對應的桿狀交通設施模型，以克服傳統運動推斷結構 (SfM) 方法在重建場景的三維模型時，當只有少數視角的照片可以使用時，便會無法匹配足夠的圖片特徵，特別是道路上的細長的物體，如紅綠燈桿和路燈桿，會發生模型斷裂的情況。

1.3 論文架構

本論文主要分為下列六個章節：

- **第一章緒論**，從現在都市發展的需求，說明交通設施重建的重要性，並且，從交通設施重建的困難，引導出本論文的問題目標與主要貢獻。
- **第二章相關研究**，討論既有的三維模型重建方法及基於強化學習 (RL) 重建方法的限制，同時，說明本論文如何在交通設施重建的情境下針對這些限制進行改善。
- **第三章系統總覽**，從對各國交通設施的觀察，引導出本論文對交通設施布局 (Layout) 的結構設計。隨後，說明本系統的流程與設計考量。
- **第四章系統實現**，詳細說明本系統從元件 (Component) 定位到布局 (Layout) 重建的流程中，各個步驟的技術細節與設計原由。
- **第五章實驗結果與討論**，以消融測試 (Ablation Study) 驗證參數設計的合理性，並且，透過真實場景與虛擬場景的重建結果驗證系統的可行性。
- **第六章結論與展望**，總結本系統的優缺點，並且，討論未來可能延伸或改良的方向。

第 2 章 相關研究

城市景觀的重建對於都市規劃來說是非常關鍵的。其中，尤其重要的是交通設施的重建，因為這對於道路規劃來說是一個必備的資訊，並且，能夠結合物聯網的技術，生成虛擬的城市街景，實現數位雙生的概念。本章節會介紹既有的三維模型重建方法，並且，分析現有的推估三維模型參數及布局 (Layout) 的逆程序化建模 (Inverse Procedural Modeling) 方法。

2.1 基於圖片的三維重建

運動推斷結構 (SfM) [1] 技術為三維重建提供新的可能性。該技術會從收集到的場景圖片中進行特徵的配對，接著，計算稀疏點雲來描述三維點與二維圖片中點的對應關係，並且，同時估測每個視角的相機參數。格瑟勒等人 (Goesele et al.) [2] 將運動推斷結構 (SfM) 與多視角立體 (Multi-View Stereo, MVS) 演算法的結合，利用前者產生的相機參數，接著，透過後者計算場景的深度圖，成功地在三維空間中重建場景。近年來，有部分研究開始利用神經網路將場景中以隱式輻射場表示。米爾登霍爾等人 (Mildenhall et al.) [3] 提出了神經輻射場 (Neural Radiance Fields, NeRF) 的概念，透過多層感知機 (Multilayer Perceptron, MLP) 來表示場景中每個點的顏色和密度，使得能夠從給定的圖片合成高品質的三維模型。凱博等人 (Kerbl et al.) [4] 則進一步提出以三維高斯 (3D Gaussian) 來表示場景，並且，在計算成本相對低的情況下，獲得與神經輻射場 (Neural Radiance Fields, NeRF) 方法相當的場景重建效果。然而，上述提到的這些方法僅涉及三維模型的外型重建，也就是整體上的模型資訊，但是，無法直接得到模型上的局部資訊，例如，元件本身的類型和元件之間的布局 (Layout) 等。因此，本論文透過運動推斷結構 (SfM) 產生的相機資訊作為位置資訊的參考，進一步從照片中偵測交通設施上的元件，來取得元件本身的資訊，並且，利用元件來推估桿子的布局 (Layout)。

2.2 逆程序化建模 (Inverse Procedural Modeling)

逆程序化建模 (Inverse Procedural Modeling) 提供了另一種三維模型重建的途徑，透過輸入圖片、網格等資訊，推估出目標模型的布局 (Layout) 結構，並且主要還能分成兩種研究方向。第一種是透過輸入的幾何資訊，推斷出對應的語法規則 (Grammar Rule)，

並將這些語法規則 (Grammar Rule) 經由程序化建模 (Procedural Modeling) 的方式，產生對應的模型。例如，斯塔瓦等人 (Št'ava et al.) [5] 透過分析輸入圖片中的曲線，將圖形以 L 系統 (L System) 來表示，並且，能夠透過修改 L 系統 (L System) 語法中的終端符號 (Terminal Symbol)，產生不同風格的圖形。而第二種是根據輸入的幾何資訊，以及事先定義好的規則集，推論出合理的布局 (Layout) 參數來描述目標物體。本論文也是屬於這一類的方法，因此以下會著重介紹第二種方法。凡內加斯等人 (Vanegas et al.) [6] 透過馬可夫鏈蒙地卡羅 (Markov Chain Monte Carlo, MCMC) 方法來引導程序化建模 (Procedural Modeling) 的參數可以盡量符合街景設計的標準。瑞奇等人 (Ritchie et al.) [7] 利用序列蒙地卡羅 (Sequential Monte Carlo) 方法來控制模型的生成，能夠在生成部分模型時得到即時的反饋，從而讓整個過程能夠更快收斂，並且，透過隨機的建模程式順序來避免生成的結果受到執行順序的影響而有偏差。然而，當使用者需要產生一個複雜的模型時，這類方法通常要求使用者對於其程序化建模 (Procedural Modeling) 系統具備一定的知識與經驗，並且，需要反覆地調整程式與相關的系統參數才能獲得理想的目標模型。本論文希望不需要人工調整參數的情況下，也能夠推估出模型的布局 (Layout)，因此，本論文以照片經由分割器 (Segmentor) 產生的特徵作為資訊，並利用強化學習 (RL) 來學習一套策略 (Policy) 去推估布局 (Layout)。隨著深度學習 (Deep Learning) 的出現，許多研究開始嘗試將其導入系統中，來從圖片當中推估目標物件的布局 (Layout) 結構。例如黃等人 (Huang et al.) [8] 以物件的手繪草圖作為卷積神經網絡 (Convolutional Neural Network, CNN) 的輸入，輸出對應的程序化模型參數，並且，提供多個結果讓使用者進行選擇。然而，生成的模型僅能夠確保外型上的相似，無法確切知道模型上的元件結構資訊。而郭等人 (Guo et al.) [9] 利用卷積神經網絡 (CNN) 來檢測輸入的樹狀圖片中的基本曲線元素，並從中推導出一個參數化的 L 系統 (L System) 來表示樹。然而，這個方法必須在目標物件的每個部分都能夠被有效偵測的情況下才可行。然而，實際上，並非物體的各個部分都能有效的被識別，例如，在本論文的應用情境中，需要在有複雜背景的照片中完整偵測出紅綠燈的桿身是困難的。因此，本論文避免直接偵測交通設施的桿身，而是透過偵測交通設施上的部分重要元件作為參考，並且，結合透過照片產生的特徵資訊，來估計中間的桿子布局 (Layout)，同時，確保能夠取得元件本身的資訊。

2.3 基於強化學習 (RL) 的布局 (Layout) 推估

特布爾等人 (Teboul et al.) [10] 採用了強化學習 (RL) 來處理建築物立面 (Facade) 的形狀語法解析問題，並且，利用設計好的功績 (merit) 函數，引導代理人 (Agent) 採取最佳

的策略 (Policy)，以由上而下 (Top-Down) 的方式建構出基於分割文法 (Split Grammar) 的推演樹 (Derivation Tree)，使推估出來的窗戶、屋頂等物件位置能與圖片對應，達到同時優化幾何物件與立面 (Facade) 結構的效果。然而，這個方法的限制在於分割文法 (Split Grammar) 會將物體沿著單一個軸向去做物體的分割，並且推演樹 (Derivation Tree) 中的子物件分割的好壞會受到父物件的影響。但是，本論文希望桿身的建構是根據照片中的資訊與偵測出來的元件來生成，而不受限於單一的軸向，以及布局 (Layout) 中的父子關係 (Parent-Child Relationship) 影響。因此，本論文以邊 (Edge) 為基本單位來描述交通設施桿子，利用強化學習 (RL) 來根據照片中偵測到的元件及特徵資訊建構桿身，並且，邊 (Edge) 之間不會互相影響。

第 3 章 系統總覽

本章節會從各個國家的交通設施出發，引導出可以將交通設施的布局 (Layout) 以圖 (Graph) 來描述的觀察。接著，說明本論文將交通設施與相機的過程以及它們的架構。最後，將介紹本論文以道路圖片為輸入，並且，透過基於強化學習 (RL) 的系統進行重建的運作過程。

3.1 交通設施背景

道路上的交通設施有各種不同的樣式，然而，交通設施在各個國家，例如，美國 [11] [12] 與台灣 [13]，都是遵守一定的規格標準進行設計的。以交通標誌為例，如圖3.1(a) 和 (b) 所示，整個物體是由一根垂直桿子與一個標誌物件所組成，其中，桿子的高度必須在一個規定的範圍內，而標誌物件本身的形狀大致分為圓形、矩形和三角形三種，每個種類也有各自規定的尺寸，如圖3.1(c) 所示。再以路燈為例，路燈本身由底座、垂直桿和燈桿所組成，如圖3.1(d) 和 (e) 所示，其中，與燈相連的懸臂長度與角度也有規範。並且，模型的外型也有一定的規律可循。最後，以紅綠燈為例，首先，底部會有一個底座，緊接著，由一根垂直桿與底座相連，之後，有一個交接處作為支撐點，這個交接處會撐起一個或多個橫向的懸臂，隨後，號誌的燈體會附在懸臂之上，如圖3.1(f) 和 (g) 所示。

事實上，現實中道路上的交通設施通常會出現更複雜的布局 (Layout)，尤其是在大城市中的多線道道路路口，可能會出現一個雙臂的紅綠燈桿，其桿身掛有多個標誌，並且，垂直桿頂端會有路燈附加之上。透過以上的說明，可以觀察到交通設施實際上是由桿子以及各種不同的元件 (Component) 所組成，桿子本身有不同的長度與粗度，而元件 (Component) 則通常具有較多種變化，譬如一般的圓型標誌有固定的尺寸，而號誌本身則可能隨著燈號的數量會有不同的長度，又或者桿身底部的警示用黃黑條紋會根據桿子本身的長度與粗度也一個對應的尺寸。但這些元件 (Component) 事實上都可以看成一個具有不同大小與朝向的長方體所包裹住的物體，如此一來，便能夠以更簡單且統一的方法來描述不同的交通設施，也就是說，布局 (Layout) 是由多個桿子與長方體所組合。

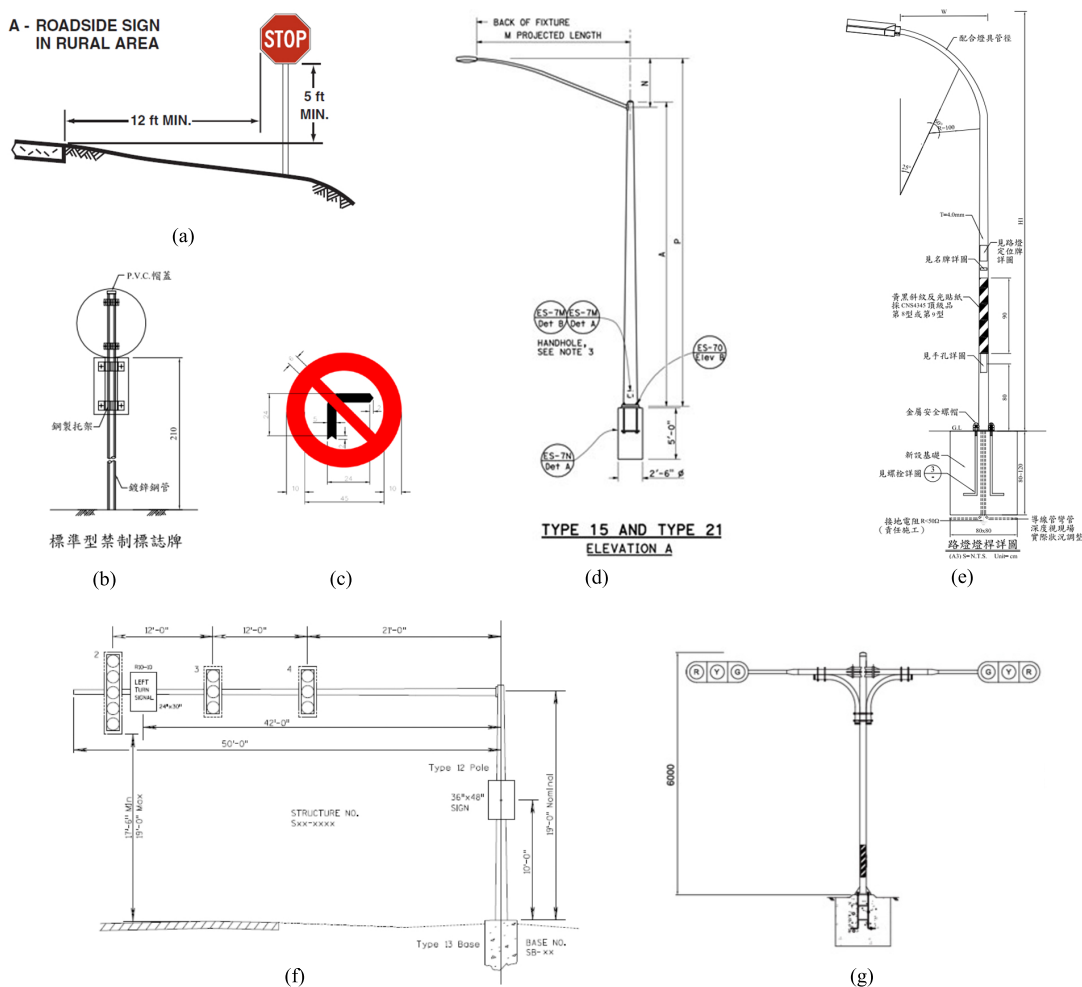


圖 3.1: 交通設施規格圖例，分別為美國交通標誌 (a)、台灣交通標誌 (b)、台灣禁止右轉標誌 (c)、美國路燈 (d)、台灣路燈 (e)、美國紅綠燈 (f) 以及台灣紅綠燈 (g)。

3.2 交通設施布局 (Layout)

延續上一段的介紹，本論文進一步發現到，無論是雙臂的紅綠燈桿或是單一的交通標誌桿，多數的交通設施桿子能夠以不同長度的垂直桿子與水平桿子所組成，並且，相連的桿子之間的方向是互相垂直的。此外，像是紅綠燈與標誌這些桿子上的元件 (Component) 都是與桿子相連接的，反過來說，桿子的方向與位置會受到元件 (Component) 的位置限制。有了以上的觀察，為了清楚描述每個元件 (Component) 與桿子的連接性與相互關係，本論文以電腦科學中常見的圖 (Graph) 結構來記錄整個交通設施的布局 (Layout)，並且，桿子以圖 (Graph) 結構中的邊 (Edge) 來記錄，每個桿子都有一個起始位置和延伸方向，以及桿子自身的長度。而圖 (Graph) 中的頂點 (Vertex) 能夠用來確定邊 (Edge) 之間的關係，也因此，在本論文中，用於紀錄桿子之間的連接關係。另外，對於每個元件 (Component)，本論文則會記錄其三維位置與所屬種類，以及該元

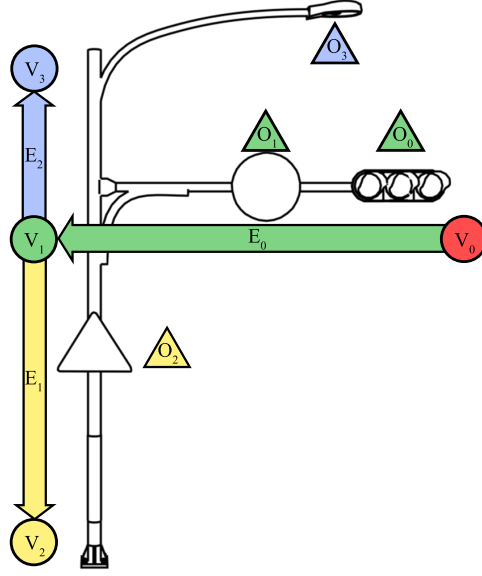


圖 3.2: 交通設施布局 (Layout) 示意圖，圖中的交通設施由一個水平桿子與一個垂直桿子所組成，並且，其上的元件 (Component) 包含一個紅綠燈號誌、一個路燈、一個圓形標誌以及一個三角形標誌。本論文透過圖 (Graph) 來描述該交通設施的布局 (Layout)。其中，每個箭頭分別表示一條邊 (Edge)，即交通設施的桿子。每個圓點則代表一個頂點 (Vertex)，用於紀錄桿子之間的連接關係。每個三角形則代表一個獨立的元件 (Component)，每個元件 (Component) 都有其所屬的邊 (Edge)，也就是與該元件 (Component) 相連接的桿子。每個頂點 (Vertex)、邊 (Edge) 與元件 (Component) 的編號代表其在布局 (Layout) 生成上的順序。

件 (Component) 所附屬的桿子。綜上所述，本論文透過圖 (Graph) 來描述桿子的布局 (Layout)，並且將其定義為：

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{O}) \quad (3-1)$$

對於每個桿子，本論文以邊 (Edge) 來描述，其中， $\bar{p}^E$ 表示起始位置， $\vec{d}^E$ 表示延伸方向， l^E 表示長度，即：

$$\bar{E} = \{\bar{p}^E, \vec{d}^E, l^E\} \quad (3-2)$$

在推估布局 (Layout) 的過程中，桿子會有不同的生成順序，以圖3.2來說，每個桿子的編號表示其生成的順序，綠色桿子是最先產生的桿子，其編號為 0，隨後，是向下的黃色桿子與向上的藍色桿子，並且，藍色桿子與黃色桿子的位置會受到綠色桿子的影響。在表示上，本論文將邊 (Edge) 再區分出父邊 (Parent Edge) 以及子邊 (Child Edge)，其中父邊 (Parent Edge) 為優先生成的邊 (Edge)。接著，以頂點 (Vertex) 來記錄相連的邊 (Edge) 資訊，每個頂點 (Vertex) 會記錄其三維位置， $\bar{p}^V$ ，和一個父邊 (Parent Edge)，

$\overline{E}^P$ ，以及一個或多個子邊 (Child Edge)， $\overline{E}^C$ ，如下所示：

$$\begin{aligned}\overline{V} &= \{\overline{p}^V, \overline{E}^P, \mathcal{E}^C\} \\ \mathcal{E}^C &= \{\overline{E}^C\}\end{aligned}\tag{3-3}$$

最後，對於每個元件 (Component)，本論文會記錄其位置， $\overline{p}^O$ ，與種類， T^O ，以及這個元件 (Component) 所屬的桿子， $\overline{E}^O$ ，即：

$$\overline{O} = \{\overline{p}^O, T^O, \overline{E}^O\}\tag{3-4}$$

有了圖 (Graph)，桿子與桿子間，以及桿子與元件之間的關係能夠被清楚的描述，這時，便能夠將其導入系統中，串起整個系統的流程，同時，整個交通設施的布局 (Layout) 推估過程也會以圖 (Graph) 具備的階層式特性，以逐步調整參數的方式，產出視覺上與目標桿狀設施相近的模型。

3.3 系統流程

基於前一章所描述，本論文以圖 (Graph) 來記錄交通設施的布局 (Layout)，並且，基於該結構提出了一個系統。本系統會輸入道路區域所拍攝的照片、以及每張照片經由重建軟體產生的相機資訊，目標是針對道路上的桿狀交通設施生成一個合理的三維重建模型，並且，這些模型的布局 (Layout) 和元件 (Component) 與照片中所呈現的是相符的。

本系統主要可以分成兩個部分，元件定位階段與布局 (Layout) 生成階段。在章節3.1中，有提到交通設施是有一定的規範，並且，道路上多數的交通設施上面都附有紅綠燈、標誌以及路燈等元件 (Component)，因此，元件 (Component) 的位置資訊對於整個布局 (Layout) 的推估提供了一個重要的參考。此外，一個交通設施通常在照片中的佔據範圍較大，在近距離的道路拍攝情境下，通常很難在一張照片中拍攝到完整的交通設施，而在遠距離的空拍情境當中，也容易因為桿子本身較細，導致分割器 (Segmentor) 直接偵測桿子時，會出現特徵不足或是複雜背景的影響，而出現分割 (Segmentation) 不完整的情況。基於以上理由，比起直接偵測桿子本身，先偵測出元件 (Component) 後再推估桿子的布局 (Layout) 會相對容易。元件 (Component) 本身在近距離拍攝的情況下較能夠完整包含整個物體，而在遠距離的情況下，元件 (Component) 通常具備較明顯的特徵，容易在照片中被辨識出來。為此，在系統的第一個階段，需要精準的定位元件 (Component) 的三維座標。

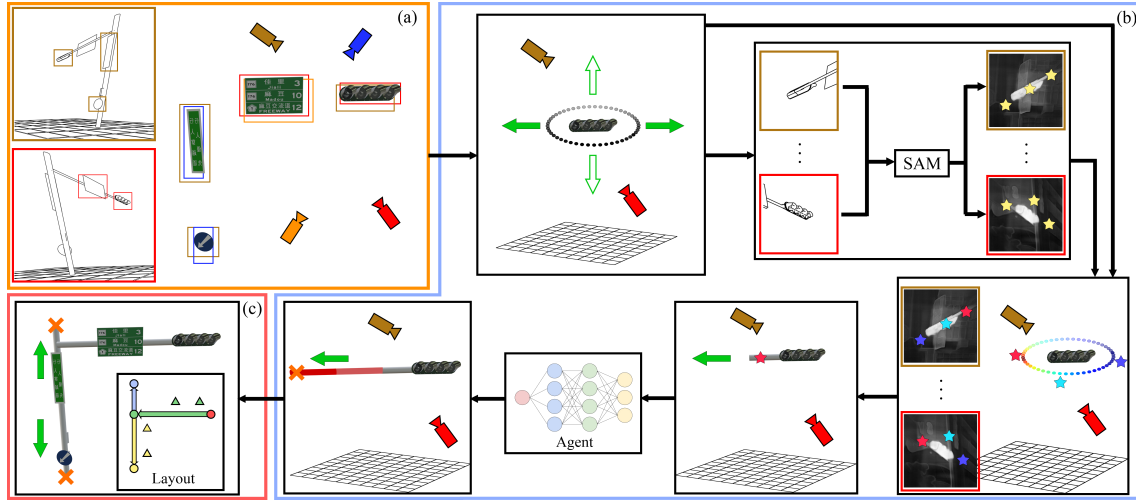


圖 3.3: 重建流程圖。首先，(a) 系統會偵測照片中的元件 (Component)，並且，整合各個視角的邊界框 (Bounding Box) 來計算每個元件 (Component) 的三維座標，隨後，(b) 挑選一個元件 (Component) 做為起點，利用分割任何事物模型 (SAM) 產生目標模型的熱圖 (Heat Map)。接著，以元件 (Component) 為起點，透過熱圖 (Heat Map) 與元件資訊，利用取樣的方式找到合適的探索方向，之後，代理人 (Agent) 會逐步生成桿子，使其能夠接近空拍圖中所呈現的姿態。最後，(c) 利用推估的布局 (Layout) 建構對應的模型。

為了從多張照片中，定位出元件 (Component) 的三維座標，首先需要對每張照片偵測場景中的元件 (Component)。然而，此時雖然每張照片中有多个元件 (Component) 的二維座標，卻無法知道哪幾張照片的哪幾個偵測到的元件 (Component) 是對應到場景中的同一個元件 (Component)。為此，本論文考慮到，同一個元件 (Component) 在不同視角下，雖然分割器 (Segmentor) 在偵測時會有些微的誤差，但其視線仍然會有一個大致的交會處，因此，本論文選擇對每個偵測到的元件 (Component)，從相機位置投射射線 (Ray)，透過該方法估算出每個元件 (Component) 的大致位置，同時也能夠確認每個元件 (Component) 在照片中對應的分割 (Segmentation) 資訊。

有了元件 (Component) 的三維資訊，在系統的第二階段，需要以元件 (Component) 為起點，推估桿子的布局 (Layout)。首先，可以確定元件 (Component) 與桿子在場景中是相連的，因此，在照片中也會是相連的。根據以上觀察，本系統導入由卡里洛夫等人 (Kirillov et al.) 提出的分割任何事物模型 (SAM) [14]，該工具原本是用於實例分割 (Instance Segmentation) 任務上的物件分割，而本論文將其作為特徵資訊的提取工具。透過分割任何事物模型 (SAM)，系統能夠取得照片中物體間的分布，當分割任何事物模型 (SAM) 辨識出元件 (Component) 時，相較於背景，與其相連的桿子也會有較高的信心 (Confidence) 會被視為與元件 (Component) 為同一個物體。因此，可以用以推估元件

(Component) 周圍的桿子分布。

隨後，根據之前所提到，一個交通設施在近距離拍攝的情況下，很難在一個視角中被完整捕捉到，因此，為了能夠對不同布局 (Layout) 的交通設施進行統一的處理，本系統將重建的過程拆解成對每個桿子的單獨計算。然而，一根桿子也可能因為長度導致無法在一個視角下完整觀察到，因此，本系統進一步將桿子的處理已逐步計算的方式，從一個點沿著一個方向逐漸延伸。為此，本論文選擇導入強化學習 (RL)。強化學習 (RL) 提供了一個架構，能夠讓系統逐步根據當前的資訊進行動作 (Action) 的預測，恰好適合進行桿子的布局 (Layout) 推估。然而，僅僅使用單一的視角無法提供足夠的資訊來生成桿子，因此，為了能夠在多視角，並且，視角數量不固定的情況下，將資訊提供給強化學習 (RL) 進行計算，本系統在每次推估的過程中，在需要預測的三維位置周圍分布固定數量取樣，隨後，每個取樣點各自整合每個視角下的特徵資訊，最後統一輸入給強化學習 (RL) 做處理，如此一來，本系統便能夠在任意的三維位置，根據周圍的視角提供的資訊，讓強化學習 (RL) 進行桿子生成的推估。

最後，取得用來表示桿子布局 (Layout) 的圖 (Graph) 後，本論文進一步導入 L 系統 (L System) 來表示。L 系統 (L System) 能夠更有效地描述這種層次結構的生成規則，並以參數化的方式控制各部分的屬性，例如：桿子的長度、元件 (Component) 之間的空間位置關係都可以通過文法 (Grammar) 進行精確定義與控制，並且，使得模型生成過程具有更高的靈活性和可控性，使系統能夠通過遞歸的方式逐層生成桿子及其元件 (Component)，該特性也為強化學習 (RL) 的訓練和桿子重建提供了結構上的支持。

第 4 章 系統實現

前一章中，討論了本論文對各國交通設施的觀察，並且，提出了以圖 (Graph) 來描述交通設施的布局 (Layout)。本章節會逐步討論系統的設計與實作細節，從元件 (Component) 的定位開始，接著，以圖 (Graph) 為基礎，進一步以 L 系統 (L System) 來分析布局 (Layout)，之後，以強化學習 (RL) 推估出文法 (Grammar)，最後，根據文法 (Grammar) 產生與圖片相符的交通設施。

4.1 元件 (Component) 生成

在這個章節中，本論文將詳細說明如何辨識和定位一個給定場景中的交通設施桿身，以及附屬在桿子上的元件 (Component)。由於道路上的桿狀交通設施存在各式各樣的布局 (Layout)，若直接偵測整個交通設施，可能會遇到許多例外的情況。此外，在近距離拍攝桿子的情況下，受限於視角的關係，難以在一個視角下捕捉到完整的交通設施，增加了布局 (Layout) 分析的難度。因此，本論文選擇直接辨識桿子上的各個元件 (Component)，而非整個交通設施物件，並且，利用照片中元件 (Component) 周邊的資訊，逐漸推估出整個交通設施的布局 (Layout)。

首先，本論文會將一組照片透過基於運動推斷結構 (SfM) 與多視角立體 (Multi-View Stereo, MVS) 技術的軟體來重建，得到了與每張輸入的照片相對應的相機參數，隨後，將相機參數與對應的照片作為系統的輸入。接著，為了避免不必要的相機增加計算量，本系統允許使用者在重建場景中標記一個場景點 $\bar{q}$ ，而系統會檢查所有場景中的相機 C ，並且，將所有能夠看見場景點 $\bar{q}$ 的相機定義為候選相機 (Candidate Camera)，由候選相機 (Candidate Camera) 構成的集合為 C^q ，如下所式：

$$C^q = \{ C_i \in C \mid \bar{q} \text{ is visible in } C_i \}. \quad (4-1)$$

4.1.1 影像畸變 (Distortion)

由於現實中照片的成像過程，會受到鏡頭或透鏡的影響，出現某種程度上的失真，也就是所謂的畸變 (Distortion)，造成後續在推估元件 (Component) 的三維位置時會產生誤

差，因此，在從照片中偵測元件 (Component) 之前需要先處理畸變 (Distortion) 的問題。

在一般情況下，需要透過相機校正 (Camera Calibration) 的步驟，利用多個已知的三維座標點，以及其在圖片中對應的二維座標點來估算相機的參數。所幸，多數的運動推斷結構 (SfM) 工具能夠透過圖片中的成對特徵點，利用光束平差法 (Bundle Adjustment) [15] 來同時最佳化特徵點的三維座標與相機的內外參數，藉此校正相機參數，因此，不需要額外處理參數的計算。

得到每個相機的相機參數後，便能夠根據以下公式對圖片進行畸變 (Distortion) 的修正。對於照片中的每個像素 $I(u, v)$ 都能夠計算對應的像素 $I'(u', v')$ ，其中 K_1 、 K_2 與 K_3 為相機的徑向畸變 (Radial Distortion) 參數， P_1 與 P_2 為相機的切向畸變 (Tangential Distortion) 參數。

$$\begin{aligned}
 r^2 &= u^2 + v^2 \\
 u' &= (1 + K_1 r^2 + K_2 r^4 + K_3 r^6)u + 2P_2 uv + P_1(r^2 + 2u^2) \\
 v' &= (1 + K_1 r^2 + K_2 r^4 + K_3 r^6)v + 2P_1 uv + P_2(r^2 + 2v^2)
 \end{aligned} \tag{4-2}$$

4.1.2 元件 (Component) 定位

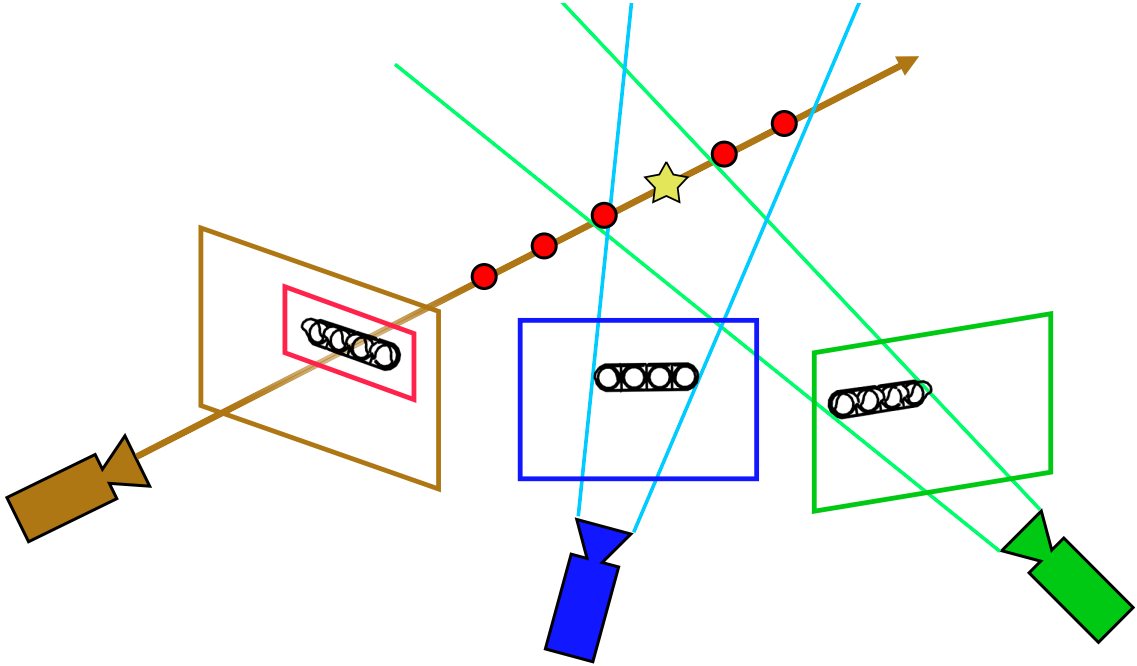


圖 4.1: 元件定位示意圖。以棕色相機位置為出發點，朝著偵測到的紅綠燈方向產生一條射線 (Ray)，同時，在線上以固定距離產生取樣點集合 $\mathcal{P}^r$ ，隨後，透過其他相機中偵測到的元件 (Component)，找到最佳的定位點 $\bar{p}^O$ 。

取得修正完的圖片後，便能夠對照片做元件 (Component) 的偵測。本論文利用實例分割網路 (Mask R-CNN) [16] 從每個候選相機 (Candidate Camera) 的對應照片中，偵測出元件 (Component) 的邊界框 (Bounding Box)， $\bar{B}$ ，和類別 (Category)， T^B ，定義如公式4-3所示，其中， x^B 為邊界框 (Bounding Box) 的四個頂點中最小 x 座標， y^B 為最小 y 座標， w^B 和 h^B 分別為邊界框 (Bounding Box) 的寬度跟高度。

$$\bar{B} = \{x^B, y^B, w^B, h^B, T^B\} \quad (4-3)$$

為了確保所有的候選相機 (Candidate Camera) 都包含有效的交通設施，本系統會進一步過濾掉沒有偵測到任何元件 (Component) 的相機，接著，對於每個照片中偵測到的元件 (Component)，本系統會從該照片所對應的相機位置 $\bar{p}^c$ ，朝著元件 (Component) 的邊界框 (Bounding Box) 中心在投影平面上的位置發射一條長度為 l^r ，方向為 $\vec{r}^c$ 的射線 (Ray)，並且，在射線 (Ray) 上每經過距離 d^r 產生一個取樣點，形成一個取樣點的集合 $\mathcal{P}^r$ ，即：

$$\mathcal{P}^r = \{\bar{p}^c + i \cdot d^r \vec{r}^c \mid i \in \mathbb{Z} \mid 0 \leq i \leq \frac{l^r}{d^r}\} \quad (4-4)$$

隨後，對於取樣點集合 $\mathcal{P}^r$ ，本系統會去檢查每個點在其他相機中，是否有偵測到同類型的元件 (Component)，並且，取出被最多視角偵測到的座標點作為該元件 (Component) 的三維座標點 $\bar{p}^O$ ，如圖4.1所示，同時，還會記錄所有偵測到該座標點 $\bar{p}^O$ ，並且，邊界框 (Bounding Box) 沒有被裁切的相機，將其定義為關聯相機 (Related Camera) 集合， $\mathcal{C}^r$ ，如公式4-6所示，其中 $\mathcal{C}^q$ 為所有候選相機 (Candidate Camera) 構成的集合。而函數 $\mathbf{F}^{in}$ 用來判斷在相機 C 中所有類型與目標類型 T^O 一致的元件 (Component) 邊界框 (Bounding Box) 中，是否存在一個邊界框 (Bounding Box)， $\bar{B}$ ，包含了座標點 $\bar{p}$ 的二維投影點，如公式4-5所示。其中，函數 $\Pi(\bar{p}, C)$ 表示計算三維座標點 $\bar{p}$ 投影到相機 C 的二維座標。之後，對於這些關聯相機 (Related Camera)，若偵測到的元件 (Component) 已經處理過，便不會再做重複計算。

$$\mathbf{F}^{in}(\bar{p}, C, T^O) = \begin{cases} 1, & \text{if } \exists \bar{B}, T^B = T^O, \\ & (x^B, y^B) \leq \Pi(\bar{p}, C) \leq (x^B + w^B, y^B + h^B), \\ 0, & \text{otherwise.} \end{cases} \quad (4-5)$$

$$\begin{aligned} \bar{p}^O &= \arg \max_{\bar{p} \in \mathcal{P}^r} \sum_{C \in \mathcal{C}^q} \mathbf{F}^{in}(\bar{p}, C, T^O) \\ \mathcal{C}^r &= \{C \in \mathcal{C}^q \mid \mathbf{F}^{in}(\bar{p}^O, C, T^O) = 1\} \end{aligned} \quad (4-6)$$

4.1.3 元件 (Component) 朝向與尺寸調整

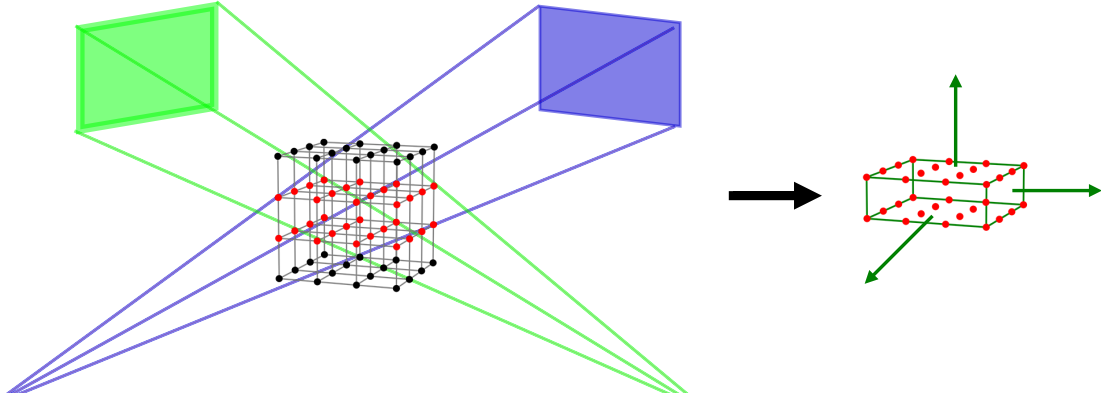


圖 4.2: 元件調整示意圖。以前一個步驟計算的座標點 $\bar{p}^O$ 為中心，在周圍立方體範圍內產生取樣點，並且，根據關聯相機 (Related Camera) 來找到合適的取樣點，最後，將取樣點擬合 (Fitting) 成一個有向邊界框 (Oriented Bounding Box)，進一步決定元件 (Component) 的方向與大小。

取得元件的座標 $\bar{p}^O$ 後，接下來，需要進一步計算元件 (Component) 的朝向與尺寸。首先，本系統會以該點為中心，分別沿著世界座標 X 軸、Y 軸與 Z 軸的正反方向，邊長為 l^s 的立方體範圍內，根據距離 d^s 產生取樣點，形成一個立方體外型的取樣點集合， $\mathcal{P}^{cs}$ ，即：

$$\mathcal{P}^{cs} = \{\bar{p}^O + (x, y, z) \cdot d^s \mid x, y, z \in \mathbb{Z} \mid -\frac{l^s}{2} \leq x, y, z \leq \frac{l^s}{2}\} \quad (4-7)$$

之後，本系統會對取樣點集合 $\mathcal{P}^{cs}$ 中的每個座標點，根據該元件 (Component) 的所有關聯相機 (Related Camera)， $\mathcal{C}^r$ ，找出所有關聯相機 (Related Camera) 都有偵測到的取樣點， $\mathcal{P}^b$ ，即：

$$\mathcal{P}^b = \{\bar{p} \mid \bar{p} \in \mathcal{P}^{cs} \mid \sum_{C \in \mathcal{C}^r} \mathbf{F}^n(\bar{p}, C, T^O) = |\mathcal{C}^r|\} \quad (4-8)$$

最後，將取樣點集合， $\mathcal{P}^b$ ，透過周等人 (Zhou et al.) 提出的三維點雲處理函式庫 (Open3D) [17] 擬合 (Fitting) 出有向邊界框 (Oriented Bounding Box)，根據該有向邊界框 (Oriented Bounding Box) 的方向以及大小來調整元件 (Component) 的姿態與尺寸，如圖4.2所示。需注意的是，這個方法仰賴於關聯相機 (Related Camera) 的視角有較平均的分布，而非傾向於某個特定方向，如此才能夠透過取樣點有效反應出元件 (Component) 外型的大致分布，此外，本論文會透過將計算出的有向邊界框 (Oriented Bounding Box) 投影回各個關聯相機 (Related Camera) 後，利用所有對應的邊界框 (Bounding Box) 計

算出平均的有向邊界框 (Oriented Bounding Box) 的尺寸，避免有向邊界框 (Oriented Bounding Box) 的尺寸受到一些偵測結果較差的邊界框 (Bounding Box) 影響。部分元件 (Component)，如：圓形標誌，由於規範中有明確固定的尺寸，因此，不會依據有向邊界框 (Oriented Bounding Box) 來調整其大小，而紅綠燈則會因為燈號的數量不同，需要調整其寬度，此時，本論文透過使用預製模型的方式，提供多組可能的元件 (Component) 尺寸，根據有向邊界框 (Oriented Bounding Box) 從中尋找最適合的尺寸，避免關聯相機 (Related Camera) 分布不均導致尺寸出現過大的偏差。

4.2 桿子布局 (Layout) 生成

在章節 4.1 中，本論文已經說明了如何去定位元件 (Component)，並且，調整其方向與大小。在下一個階段，本論文透過建構一個強化學習 (RL) 的環境來描述桿子布局 (Layout) 生成的過程。如同章節 1.1 所說，強化學習 (RL) 能夠透過馬可夫決策過程 (Markov Decision Process, MDP) 來做描述，並且，其構成元素包含狀態 (State)、動作 (Action)、轉移機率函數 (Transition Probabilities) 與獎勵函數 (Reward)，以下將針對各項元素結合本論文的應用情境進行詳細說明，講述如何透過經由強化學習 (RL) 訓練的代理人 (Agent) 來生成對應的桿子布局 (Layout)。

4.2.1 挑選起始元件 (Component)

在本論文中，為了獲得與桿子生成有關的資訊，必須從道路的圖片中提取資訊。然而，並非整張圖片都包含有效的桿子資訊，因此，需要有一個起始的座標點來做後續的偵測，為此，本論文會挑選一個起始元件 (Component) 的位置作為推估的起始座標點。首先，經過元件 (Component) 定位的階段之後，已經有各種元件 (Component) 存在於場景中。為了加快推估速度，本論文會優先挑選場景中的紅綠燈、路燈元件 (Component) 作為起始元件 (Component)，因為，這些元件 (Component) 通常會出現在桿子布局 (Layout) 的末端，能夠進一步利用這個特性，減少一開始需要探索的方向數量。以紅綠燈為例，紅綠燈基本上只會附加在橫桿上，因此，可以直接排除掉垂直的探索方向。

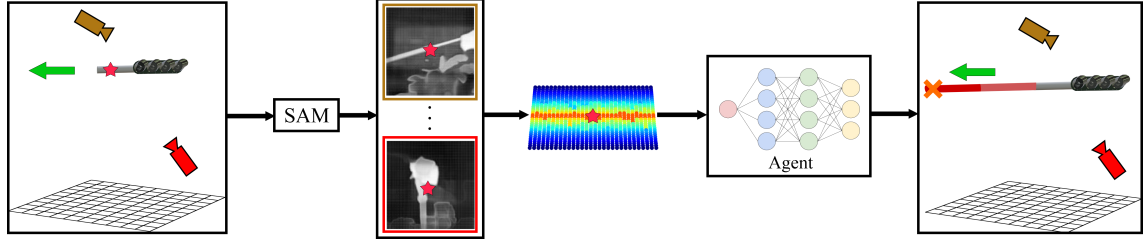


圖 4.3: 強化學習 (RL) 示意圖。圖片中的紅色星號為當前欲檢測的三維位置。將當前位置投影到各個照片後，透過分割任何事物模型 (SAM) 取得對應的熱圖 (Heat Map)。隨後，在當前位置周圍產生方形分布的取樣點，隨後根據熱圖 (Heat Map) 計算每個取樣點的平均信心 (Confidence) 分數。最後，代理人 (Agent) 將這些取樣點分數作為輸入後，決定輸出的動作 (Action) 數值，藉此決定新的檢測位置。反覆執行以上步驟直到代理人 (Agent) 輸出低於閾值的動作 (Action) 數值。

4.2.2 強化學習 (RL) 與狀態 (State)

在強化學習 (RL) 過程中，狀態 (State) 能夠幫助代理人 (Agent) 理解當前的環境變化，從而選擇合適的動作 (Action)。因此本章節將描述如何獲得狀態 (State) 作為代理人 (Agent) 的輸入。

首先，根據章節4.2.1所敘述的方式，從場景中挑選一個元件 (Component) 作為起始元件 (Component) 後，本系統會以起始元件 (Component) 的三維座標點為起點，確認每個候選相機 (Candidate Camera) 的圖片中，該座標點周圍是否有桿子的資訊。也就是說，對於第 i 個候選相機 (Candidate Camera)， C_i^q ，其對應的照片， I_i^q ，本系統將起始元件 (Component) 的三維座標點投影到照片中，定義該二維座標點為 $\bar{p}_i^p$ ，並且，以 $\bar{p}_i^p$ 為中心，從 I_i^q 裁切一張邊長為 w^H 的圖片， I_i^c ，如公式4-9所示，其中 F^{cr} 表示將相機對應的圖片進行裁切。

$$I_i^c = F^{cr}(C_i^q, \bar{p}_i^p, w^H) \quad (4-9)$$

接著，由於元件 (Component) 會與桿子相連，因此，能夠透過提取圖片中元件 (Component) 周圍的特徵資訊，來獲得桿子的資訊。為此，本論文將裁切後的圖片 I_i^c 輸入由卡里洛夫等人 (Kirillov et al.) 提出的分割任何事物模型 (SAM) 工具 [14] 來做進一步處理。該工具能夠提取圖片的特徵，並且，結合給定的邊界框 (Bounding Box) 做為輸入的提示 (Prompt)，能夠分割出物體的分布區域。透過分割任何事物模型 (SAM) 工具探索元件 (Component) 周圍的特徵資訊，本論文能夠得到圖片 I_i^c 的對應熱圖 (Heat Map)， I_i^h ，該圖片能夠表示點 $\bar{p}_i^p$ 周圍每個像素可能為桿子的信心 (Confidence) 分數，

如公式4-10所示，其中 $\bar{b}^h$ 表示輸入分割任何事物模型 (SAM) 的邊界框 (Bounding Box) 提示 (Prompt)，而 F^{sam} 表示利用分割任何事物模型 (SAM) 取得圖片的特徵資訊。

$$I_i^h = F^{sam}(I_i^c, \bar{b}^h) \quad (4-10)$$

此時，結合公式4-9與4-10後，可以得到，對於一個任意給定的三維座標點 $\bar{p}^s$ ，投影到每個候選相機 (Candidate Camera) 後，得到的桿子信心 (Confidence) 資訊熱圖 (Heat Map) 所組成的集合 $\mathcal{I}^h$ 如公式4-11所示，其中，二維座標點 $\bar{p}_i^p$ 為給定的三維座標點投影到第 i 個相機的座標點。

$$\mathcal{I}^h = \{I_i^h \mid I_i^h = F^{sam}(F^{cr}(C_i^q, \bar{p}_i^p, w^H), \bar{b}^h), i \in 1, 2, \dots, |C^q|\} \quad (4-11)$$

有了上述的方法，本論文能夠獲得圖片中，可能為桿子的信心 (Confidence) 分布。因此，在找到起始元件 (Component) 後，為了確認桿子的起始推估方向，本論文會在起始元件 (Component) 的三維座標點 $\bar{p}^O$ 周圍，以該點為圓心產生一圈在世界座標 X-Y 平面上的取樣點集合， $\mathcal{P}^d$ ，接著，本論文會根據每個候選相機 (Candidate Camera) 對應的熱圖 (Heat Map)，計算每個取樣點 $\bar{p}_i^d$ 的信心 (Confidence) 分數總和平均，並且，將其定義為 F^A ，如公式4-12所示，其中， $\bar{p}_i^p$ 表示取樣點投影到熱圖 (Heat Map) 的二維座標。隨後，從中找到信心 (Confidence) 分數總和平均最高分的其中一個座標點作為桿子的水平延伸方向的依據，並且，將其定義為， $\bar{u}^H$ ，如公式4-13所示。

$$F^A(\bar{p}_i^d) = \frac{1}{|\mathcal{I}^h|} \sum_i I_i^h(\bar{p}_i^p), i \in 1, 2, \dots, |C^q| \quad (4-12)$$

$$\bar{u}^H = \arg \max_{\bar{p} \in \mathcal{P}^d} F^A(\bar{p}) - \bar{p}^O \quad (4-13)$$

而垂直延伸方向則直接採用世界座標的 Z 軸，將其定義為 $\bar{u}^V$ ，最後，第三個軸向則是與前兩個方向垂直的方向，即 $\bar{u}^H \times \bar{u}^V$ 。結合上述的三個軸向以及其各自的反方向後，至此，得到了桿子可以延伸的方向。

為了能夠在任意數量的候選相機 (Candidate Camera) 下確保強化學習 (RL) 的訓練過程中，代理人 (Agent) 得到的狀態 (State) 資訊大小是固定的，需要在推估的過程中將所有桿子的熱圖 (Heat Map) 資訊做整合，因此，本論文在推估時，同樣透過產生取樣點的方式，以任意給定的三維座標點 $\bar{p}^s$ 為中心，根據當前桿子所在的平面，結合前面所計

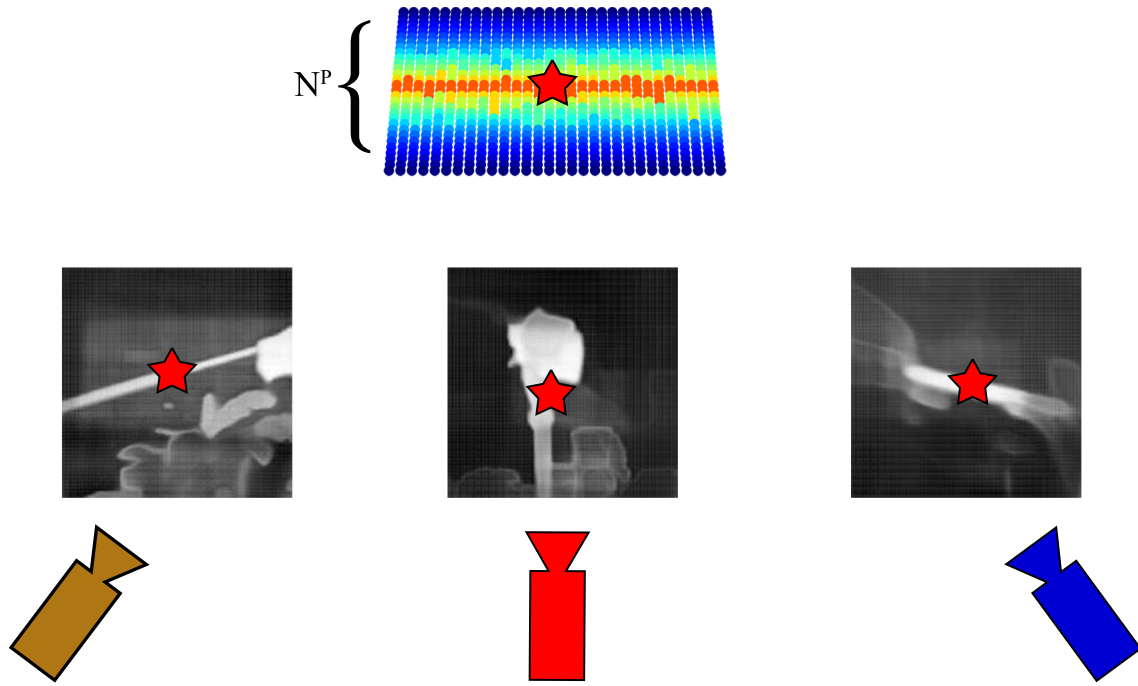


圖 4.4: 正方形分布取樣點示意圖。該取樣點集合會被轉換成一張邊長為 N^P 的熱圖 (Heat Map)，隨後，做為代理人 (Agent) 的輸入

算的兩個軸的正反方向，加上取樣點間距 d^p ，以及單邊的取樣數量 N^p ，形成一個數量為 $N^P \times N^P$ 的正方形三維取樣點集合，如圖4.5所示。隨後，每個取樣點 $\bar{p}^d$ 根據其投影到集合 $\mathcal{I}^h$ 中每個熱圖 (Heat Map) 中的二維座標，能夠根據公式4-12計算總和平均做為該取樣點的分數。

最後，綜上所述，對於任意給定的三維座標點 $\bar{p}^s$ ，本論文將其周圍產生的正方形取樣點集合轉換成一張邊長為 N^p 的熱圖 (Heat Map) 來表示每個取樣點的分數，並且，以此作為狀態 (State) 來讓代理人 (Agent) 獲得桿子的資訊，如圖4.3所示。

4.2.3 動作 (Action) 設計

在強化學習 (RL) 中，代理人 (Agent) 透過採取動作 (Action) 與環境互動，藉此改變環境來獲得下一次的狀態 (State)。然而在設計動作 (Action) 時，需要盡量避免代理人 (Agent) 能夠採用的動作 (Action) 數量過於龐大，否則容易導致代理人 (Agent) 無法學習一套有效的策略 (Policy)。對於一個桿子的生成來說，需要有三個參數來決定生成的結果。第一，需要有一個位置作為生長的起點，第二，需要一個向量作為其生長的方向，第三，需要一個長度來決定其生長的終點座標。在本系統中，由於第一根桿子的起點是由元件 (Component) 所決定，而後續的桿子又是根據前一個桿子的終點座標來決定，因

此，代理人 (Agent) 不需要控制桿子的生成起始位置。再者，桿子的生長方向需要在布局 (Layout) 推估之前決定，並且，本論文根據對現實中的交通設施桿子的觀察，假設桿子的可能生長方向為兩個互相垂直的水平軸向以及一個與世界座標 Z 軸相同的垂直軸向，以及這些軸的反方向，共六個方向，因此，也不需要由代理人 (Agent) 來控制。也就是說，代理人 (Agent) 只需要決定桿子生長的長度。

然而，由於桿子本身的長度是不固定的，並且，對於較靠近桿子的視角，其桿子本身可能會在照片中被裁切，因此，本論文不是直接讓代理人 (Agent) 決定整個桿子的長度，而是透過逐步延伸的方式，從起始位置開始，代理人 (Agent) 根據當前位置的桿子資訊，決定目前桿子所要向前延伸的長度。本論文將代理人 (Agent) 所能決定的延伸長度設為一個動作 (Action) 空間， $\mathcal{A}$ ，其定義如下：

$$\mathcal{A} = \{a \mid a \in \mathbb{R} \mid 0 \leq a \leq 1\} \quad (4-14)$$

本論文會將代理人 (Agent) 輸出的動作 (Action) 數值 a 四捨五入到小數點後第一位，同時乘上一個移動的步長作為這次的延伸長度，並且，若數值小於 0.1，則會視為代理人 (Agent) 決定停止延伸，此時，會將當前位置設置為目前桿子的終點位置，同時，會在該位置重新探索其他方向是否存在桿子。

4.2.4 獎勵函數 (Reward) 與訓練

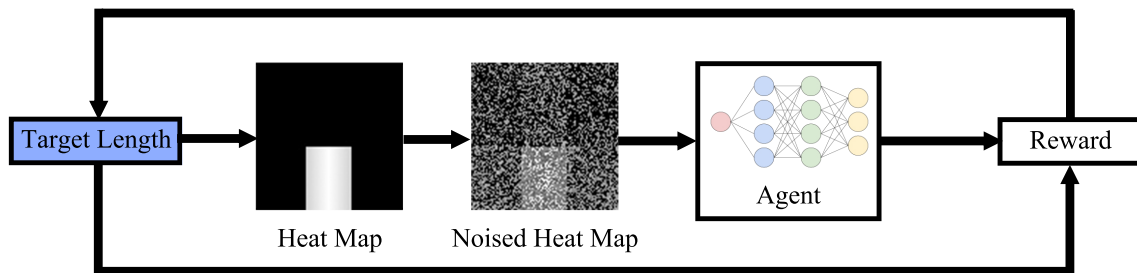


圖 4.5: 強化學習訓練流程圖。每次系統會指定一個目標長度 (Target Length)，而代理人 (Agent) 必須在規定步數內完成該桿子長度的推估，若能夠在指定步數內完成，則給予適當的獎勵。

由於本論文的目的是希望產生的布局 (Layout) 能夠與目標布局 (Layout) 越接近越好，因此，在強化學習 (RL) 的訓練過程中也會以這個目標來設定代理人 (Agent) 的獎勵機制。原本，依照應用的情境，在訓練的過程中，需要在三維建模軟體 (Blender 3D-Modeling Software) 內每次隨機建構一個桿子布局 (Layout)，並且，隨機產生相機

來模擬空拍圖的相機後，讓代理人 (Agent) 逐步利用熱圖 (Heat Map) 學習推估的方法。然而，爲了增加訓練的效率，本論文簡化了訓練的方式。首先，在章節4.2.2中有提到，代理人 (Agent) 所得到的狀態 (State) 資訊是由每個相機視角下的信心 (Confidence) 資訊進一步整合後的熱圖 (Heat Map) 所構成，因此，在訓練時，本論文會直接產生一張熱圖 (Heat Map)， I^g ，並且，利用這張圖來模擬由取樣點構成的狀態 (State)。本論文在每次產生該熱圖 (Heat Map) 時，會隨機決定一個目標長度的桿子，並且，將其以矩形的方方式添加進該圖中，隨後，添加一些噪點來調整該圖片。之後，透過讓代理人 (Agent) 讀取該熱圖 (Heat Map)，決定要延伸桿子或停止延伸。接著，本論文會設定一個目標桿子長度，作爲代理人 (Agent) 在每個回合 (Episode) 中需要推估的長度。透過記錄代理人 (Agent) 採取動作 (Action) 後所決定的長度與每個熱圖 (Heat Map) 所指定的目標桿子長度，能夠監控採取的動作 (Action) 的結果。若兩者相差小於一個閾值時，便賦予代理人 (Agent) 正面獎勵，並且，若能在指定動作 (Action) 次數內完成回合 (Episode)，能夠獲得額外正面獎勵。本論文使用的訓練策略爲舒爾曼等人 (Schulman et al.) 提出的近端策略優化法 (Proximal Policy Optimization, PPO) [18]，因爲近端策略優化法 (PPO) 限制了策略 (Policy) 的優化，以防止由於巨大的策略 (Policy) 更新而導致的不穩定訓練，從而降低了訓練策略 (Policy) 深度神經網絡 (DNN) 的難度。爲了使訓練的策略 (Policy) 能夠推估各種布局 (Layout)，本論文準備了多種桿子布局 (Layout) 來訓練策略 (Policy)。此外，在訓練中，本論文設計了終止條件以確保優化的收斂。一個軌跡 (Trajectory) 的訓練在滿足以下兩種條件的其中一種時便會終止：一種是代理人 (Agent) 成功達到目標桿子的長度時，另一種是代理人 (Agent) 的行動次數達到本論文設定的次數限制。

4.2.5 儲存布局 (Layout) 資訊

爲了重建出最後的模型，需要將系統所推估出的資訊轉換成章節3.2中所描述的布局 (Layout) 結構。首先，在確認了起始元件 (Component) 後，會同時將其紀錄爲第 0 號頂點 (Vertex)， $\overline{V}_0$ ，之後在重建模型時，會以該頂點 (Vertex) 開始處理。之後，在強化學習 (RL) 推估桿子的過程中，每當代理人 (Agent) 停止了當前的桿子推估後，便會在該桿子的終點建構一個新的頂點 (Vertex)， $\overline{V}_i$ ，同時，會產生一個新的邊 (Edge)， $\overline{E}_{i-1}$ 來紀錄桿子的長度、方向等資訊。過程中，若有任何元件 (Component) 需要附屬在桿子上，便會將指定的邊 (Edge) 儲存在元件 (Component)， $\overline{O}$ ，的資料中。透過上述方法，本系統能夠在推估布局 (Layout) 的過程中，同步將資訊紀錄成一個圖 (Graph) 格式，便於後續模型的生成。

4.2.6 程序化桿子模型重建

前面幾個章節說明了強化學習 (RL) 的架構設計，最後一步是將交通設施模型以程序化的方式進行重建，本章節將說明如何將模型進行重建，並且，將重建過程以文法 (Grammar) 系統進行表示。在章節3.2中，本論文已經透過圖 (Graph) 來描述桿子與桿子之間以及桿子與元件 (Component) 之間的關係，同時，在章節4.2.5中，本論文也說明了如何將過程中產生的資訊轉換成布局 (Layout) 結構。為了進一步清楚表示布局 (Layout) 與模型重建的關係，本論文透過導入 L 系統 (L System) 來參數化桿子的重建流程。L 系統 (L System) 是一種根據推演文法 (Production Grammar) 改寫字串的系統，起初是用來描述植物細胞的成長模式，隨後，在計算機圖學、分形幾何學和其他自然系統的模擬中得到了廣泛應用。L 系統 (L System) 的規則推演特性經常用來模擬階層式 (Hierarchical) 結構植被的生長過程，例如：樹幹之上會有樹枝，而樹枝上又會生出更細的樹枝。而事實上，交通桿子也是一個階層式 (Hierarchical) 結構，在推估以及建構的過程中，桿子與桿子之間，是有一定順序的。因此，交通桿子能夠透過 L 系統 (L System) 來表示模型生成的過程。此外，在章節4.2.3中有提到，為了讓強化學習 (RL) 較容易訓練，代理人 (Agent) 不需要考慮桿子的起點位置，其位置是透過前一個桿子的終點所決定的，透過 L 系統 (L System) 的特性，本論文能夠進一步描述在生長後面桿子的時候，能夠將前面桿子的位置與方向考慮進去，而不需要去一次考慮整個桿子的參數。鑒於以上的原因，本論文決定以 L 系統 (L System) 來做表示。

L 系統 (L System) 總共包含三種元素，符號系統 (Alphabet)、公理 (Axiom) 以及推演文法 (Production Grammar)。其中符號系統 (Alphabet) 由多種符號所組成，每個符號可以用來表示物體的某個狀態。而公理 (Axiom) 則是表示整個系統的初始狀態。推演文法 (Production Grammar) 則是用來表示字串的替換規則。首先，本論文用 Ω 表示公理 (Axiom)，並且，以符號 G_0^V 表示整個桿子的重建是從第 0 號頂點 (Vertex) 開始，其推演文法 (Production Grammar) 如下所示：

$$\Omega : G_0^V \quad (4-15)$$

接著，每個頂點 (Vertex) 會去產生與其相連的桿子，如下所示：

$$G_i^V : G_{(i,i+1)}^E \quad (4-16)$$

對於桿子來說，原本需要考慮起始位置、方向以及長度，為了簡化表示方式，本論文進一步改成用桿子的兩端點來做表示，也就是說，透過桿子的兩個頂點 (Vertex) 的三維位置，能夠同時描述桿子的位置、方向以及長度的數值。因此，在表示上，以 $\overline{E}_{(i,j)}$

來表示以第 i 個頂點 (Vertex) 與第 j 個頂點 (Vertex) 相連的邊 (Edge)，而 $M^E(\bar{V}_i, \bar{V}_j)$ 則表示在這兩個頂點 (Vertex) 的位置之間產生一個桿子模型。產生完桿子模型後，會同時產生桿子上的附屬元件 (Component)，以 $G_{(i,j)}^C$ 表示，之後會從新的頂點 (Vertex)， $\bar{V}_j$ ，來做後續的推演，完整的推演文法 (Production Grammar) 如下所示：

$$G_{(i,j)}^E = \begin{cases} G_{(i,j+1)}^E M^E(\bar{V}_i, \bar{V}_j) G_{(i,j)}^C(0) G_j^V, & \text{if } \bar{E}_{(i,j)} \in \mathcal{E} \\ G_{(i,j+1)}^E, & \text{otherwise} \end{cases} \quad (4-17)$$

最後，對於每個元件 (Component)，本論文以事先建模好的模型來做使用，並且，以 $M^C(\bar{O}_k)$ 結合公式3-4來表示在位置 $\bar{p}^O$ 產生類型為 T^O 的元件 (Component) 模型，而， $\bar{E}_k^O$ 表示元件 (Component)， $\bar{O}_k$ ，所屬的桿子， k 表示該元件 (Component) 是布局 (Layout) 中的第 k 個元件 (Component)，其推演文法 (Production Grammar) 如下所示：

$$G_{(i,j)}^C(k) = \begin{cases} M^C(\bar{O}_k) G_{(i,j)}^C(k+1), & \text{if } \bar{E}_k^O == \bar{E}_{(i,j)} \\ G_{(i,j)}^C(k+1), & \text{otherwise} \end{cases} \quad (4-18)$$

第 5 章 實驗結果與討論

本章節將進一步透過實驗分析系統的有效性，首先，本論文設計了一套誤差計算方法來進行驗證，並且，透過消融測試 (Ablation Study) 分析系統的參數合理性，隨後，本論文分別導入真實場景和虛擬場景兩種情境來驗證生成結果的準確性。其中，真實場景用來確保整個框架在真實世界中的可行性，而虛擬場景是為了驗證在更複雜的布局 (Layout) 下強化學習 (RL) 的生成有效性與穩定性。

5.1 誤差評估

為了準確評估重建模型與基準真相 (Ground Truth) 的匹配程度，本論文設計了一套針對桿子和元件 (Component) 的誤差計算方法。這些方法考慮了桿子端點座標以及元件 (Component) 的空間位置、朝向與尺寸，並且，以總誤差作為衡量模型整體準確性的指標。以下將詳細描述桿子誤差、元件 (Component) 誤差以及總誤差的計算方式。

首先，桿子的誤差主要針對桿子的基準真相 (Ground Truth) 與重建結果進行比較，由於桿子模型本身可以視為一條線段，可以用桿子的兩端點來決定其位置、長度與方向，因此，本論文以端點的座標誤差來計算桿子的誤差。如公式5-1所示，對於每個基準真相 (Ground Truth) 端點座標 $\bar{p}_i^{GV}$ ，找到位置最接近的重建端點座標 $\bar{p}_j^V$ ，隨後，計算二者之間的歐幾里得距離作為該端點的誤差 E_i^V 。

$$E_i^V = \min_j \|\bar{p}_i^{GV} - \bar{p}_j^V\| \quad (5-1)$$

有了上述的公式，能夠得到整個交通設施桿子的平均誤差 E^P ，其中， N^V 表示端點的總數量，如下所示：

$$E^P = \frac{1}{N^V} \sum_i^{N^V} E_i^V \quad (5-2)$$

接著，對於元件 (Component) 誤差計算，為了全面評估重建元件 (Component) 的準確性，需要考慮元件 (Component) 的中心座標、朝向及尺寸的誤差。首先，對於中心座標誤差，本論文直接根據每個基準真相 (Ground Truth) 元件 (Component) 位置 $\bar{p}_i^{GO}$ ，以及與其最接近的重建元件 (Component) 位置 $\bar{p}_j^V$ ，計算兩者之間的歐幾里得距離，如公式5-3所示，並且，這個編號為 j 的元件 (Component) 也會拿來計算後續的朝向誤差

與尺寸誤差。

$$E_i^{OC} = \min_j \|\bar{p}_i^{GO} - \bar{p}_j^V\| \quad (5-3)$$

對於元件 (Component) 的朝向，本論文會根據其物體空間座標系 (Object Space Coordinate) 底下的 X 軸， $\vec{d}_j^O$ ，以及基準真相 (Ground Truth) 元件 (Component) 的軸向， $\vec{d}_i^{GO}$ ，計算兩軸之間的夾角作為元件 (Component) 的朝向誤差，如公式5-4所示。

$$E_i^{OD} = \arccos \left(\frac{\vec{d}_i^{GO} \cdot \vec{d}_j^O}{\|\vec{d}_i^{GO}\| \|\vec{d}_j^O\|} \right) \quad (5-4)$$

對於元件 (Component) 的尺寸，本論文根據其有向邊界框 (Oriented Bounding Box) 在三個軸向的尺寸 h_j^O 、 w_j^O 與 d_j^O ，分別與基準真相 (Ground Truth) 元件 (Component) 的有向邊界框 (Oriented Bounding Box) 尺寸 h_j^{GO} 、 w_j^{GO} 與 d_j^{GO} 作誤差計算，如公式5-5所示。

$$E_i^{OS} = \min_j \|h_i^{GO} - h_j^O\| + \|w_i^{GO} - w_j^O\| + \|d_i^{GO} - d_j^O\| \quad (5-5)$$

最後，有了上述的公式，能夠得到所有元件的平均位置誤差 E^C 、平均朝向誤差 E^D 以及平均尺寸誤差 E^S ，其中， N^O 表示元件 (Component) 的總數量，如下所示：

$$\begin{aligned} E^C &= \frac{1}{N^O} \sum_i^{N^O} E_i^{OC} \\ E^D &= \frac{1}{N^O} \sum_i^{N^O} E_i^{OD} \\ E^S &= \frac{1}{N^O} \sum_i^{N^O} E_i^{OS} \end{aligned} \quad (5-6)$$

5.2 消融測試 (Ablation Study)

在設計與優化複雜系統的過程中，理解各參數與模塊對系統整體性能的影響是至關重要的。為此，本論文透過消融測試 (Ablation Study) 控制系統參數，並且，導入誤差計算來分析系統中關鍵參數對生成結果的影響。

5.2.1 元件 (Component) 定位參數分析

首先，針對元件 (Component) 定位的階段，本論文針對射線 (Ray) 取樣距離， d^r ，來做調整，以探討其對元件 (Component) 姿態準確性的影響。如表5-1所示，本論文將 d^r 設定為多個不同的值來進行實驗，隨後，本論文觀察到較小的 d^r 值提升了取樣點的密度，從而提高元件 (Component) 姿態的準確性，但也增加了計算負擔與冗餘點的干擾，而較大的 d^r 值雖然減少了取樣點數量，降低了計算成本，卻也導致部分準確性降低。實驗結果顯示，當 d^r 設定為 20 公分時，系統在準確性與計算效率之間達到了最佳平衡。因此，過大或過小的 d^r 均對最終的元件 (Component) 姿態準確性造成負面影響，說明 d^r 是影響結果的關鍵參數。

元件種類	d^r (cm)	E^C (cm)	E^D (degree)	E^S (cm)	時間 (s)
標誌	40	6.30	0.96	6.80	85
	20	3.15	0.79	5.21	158
	10	2.46	0.78	5.41	307
紅綠燈	40	5.81	1.62	2.59	102
	20	4.91	1.61	2.41	198
	10	4.51	1.58	2.36	378
路燈	40	6.19	1.06	1.86	56
	20	6.17	0.85	1.74	107
	10	6.12	0.81	1.64	212

表 5-1: 不同取樣距離下各元件的姿態誤差結果。

5.2.2 桿子推估參數分析

針對桿子布局 (Layout) 的推估階段，本論文針對在布局 (Layout) 推估的過程中，各個視角下的裁切邊長參數 w^H 進行分析，觀察其對布局 (Layout) 準確性的影響。從表5-2中邊長為 80 像素長的圖片可以發現，在檢測範圍中，當背景所佔的比例較高時，容易發生背景被分割任何事物模型 (SAM) 視為主目標的情況，反之，當整張照片能夠較完整包含背景物件時，分割任何事物模型 (SAM) 能夠較清楚區分出前景的桿子，本論文推測，可能是因為在當目標物體在圖片中佔據的範圍較少時，分割任何事物模型 (SAM) 模型內的自注意力 (Self-Attention) 機制的作用範圍有限，導致獲取的全局特徵不足，從而無法正確辨識前景物件。因此，較高的前景物件占比能夠提高整體系統的穩定度，使代理人

(Agent) 能夠得到更好的桿子資訊，也代表著參數 w^H 也是影響結果的重要參數。

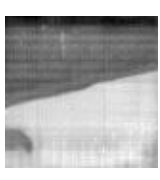
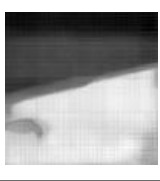
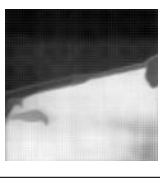
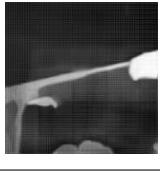
尺寸 (pixel)	原始照片	熱圖
80x80		
100x100		
120x120		
140x140		
160x160		

表 5-2: 不同邊長下熱圖的精準性比較。

確定好合適的邊長參數 w^H 後，接下來本論文會固定使用邊長為 160 像素長的圖片，並且，針對用來作為代理人 (Agent) 輸入的取樣點機制中，決定取樣距離的參數， d^p ，來做調整，探討其對預測的桿子布局 (Layout) 準確性的影響。從表5-3中可以觀察到，較大的 d^p 雖然能夠一次探索較大的範圍，使計算時間降低，然而，也由於降低二值圖片 (Binary Image) 的解析度，進而影響代理人 (Agent) 對圖片中桿子的感知精準度。相反地，較小的 d^p 縮小了探索空間，有助於代理人 (Agent) 更精準判斷桿子分布，但也提高了計算成本。因此，當 d^p 設定為最佳值時，桿子布局 (Layout) 的精準度與計算效率之間實現了理想的平衡。而過大或過小的 d^p 都會對代理人 (Agent) 感知的桿子狀態準確性產生負面影響，顯示出參數 d^p 同樣是決定系統性能的重要參數。

$d^p(\text{cm})$	$E^p(\text{cm})$	時間 (s)
2	0.430	121
1	0.315	206
0.5	0.246	403

表 5-3: 不同取樣距離下桿子的誤差結果。

5.3 真實場景實驗

在真實場景的實驗中，本論文分別使用了以道路平拍照片結合運動推斷結構 (SfM) 工具重建的點雲場景，以及透過空拍圖經由運動推斷結構 (SfM) 重建的場景，重建結果如表5-4和5-5所示。這兩種情境分別代表了近距離拍攝與遠距離拍攝的情況。其中，在道路平拍的場景中，導入了實時動態技術 (Real-Time Kinematic, RTK) 技術 [19] 來進行場景座標的校正。實時動態技術 (RTK) 技術是一種基於全球導航衛星系統的高精度定位技術，主要通過消除和修正衛星信號中的各種誤差，實現厘米級甚至亞厘米級的定位精度。

5.4 特殊桿子實驗

為了進一步展示本論文的有效性 (Validity)，本論文準備了幾個不同的布局 (Layout) 作為特殊的 (Ad hoc) 測試案例，並且，透過虛擬場景來進行模擬，如圖5.1所示。隨後，利用本論文的框架來推估是對應的布局 (Layout)。為了模擬從真實世界照片中推估交通設施桿子布局 (Layout) 的情境，本論文會隨機選取相機的姿態 (Pose)，確保其翻滾角 (Roll Angle) 為零，同時，確保每個視角會包含部分的基準真相 (Ground Truth) 桿子。這些抽樣的相機姿態 (Pose) 和基準真相 (Ground Truth) 桿子會進一步被輸入到三維建模軟體 (Blender 3D-Modeling Software) [20]，結合預先部署好的虛擬城市場景，渲染 (Render) 出場景圖來模擬真實世界的照片，重建的結果如表5-7所示。本論文將測試資料分成三種難度，分別為簡單、普通和困難，難度會根據以下的標準來設計：

1. **元件 (Component) 數量**：基準真相 (Ground Truth) 布局 (Layout) 中的元件 (Component) 數量。
2. **結構複雜度**：基準真相 (Ground Truth) 布局 (Layout) 中的頂點 (Vertex) 數量以及是否存在對稱的結構。

3. 元件 (Component) 類別 (Category) 的多樣性：基準真相 (Ground Truth) 布局 (Layout) 中出現的元件 (Component) 類別 (Category) 多寡。

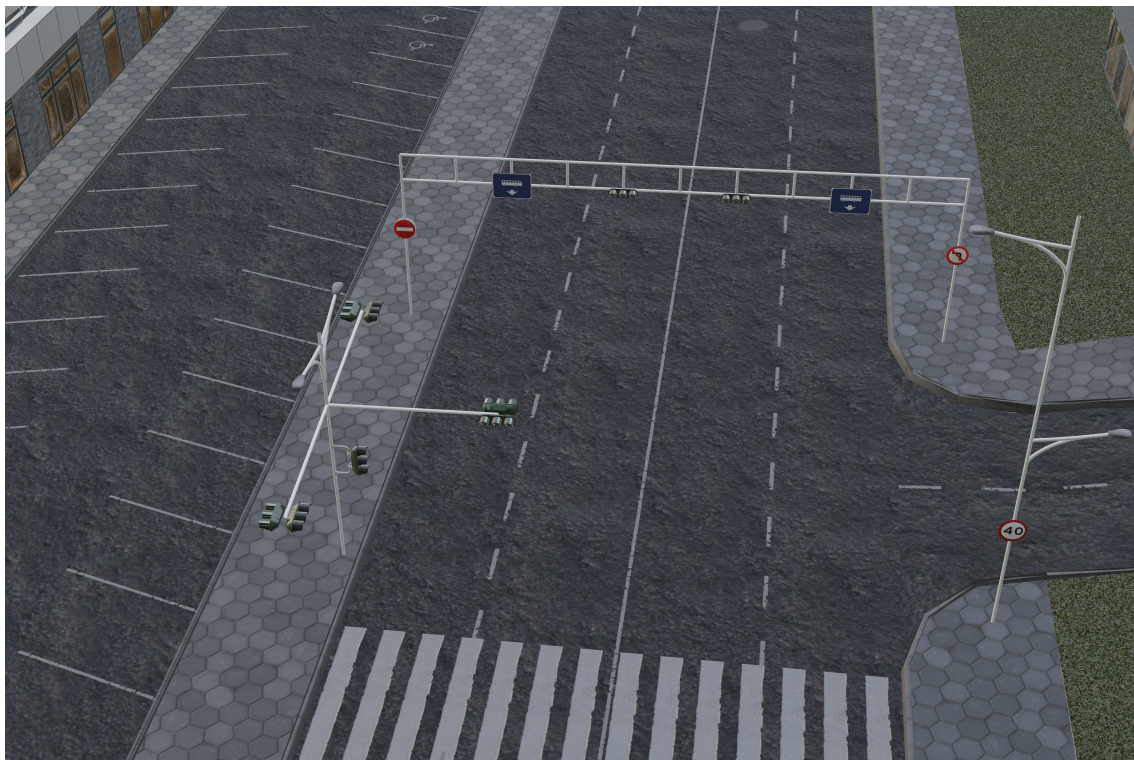


圖 5.1: 虛擬場景圖。從左到右分別為三臂紅綠燈、鐵架與高低路燈

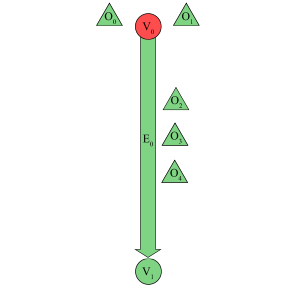
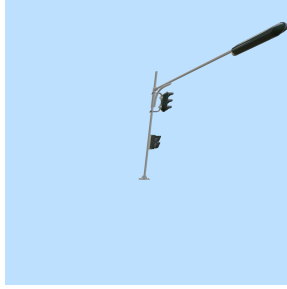
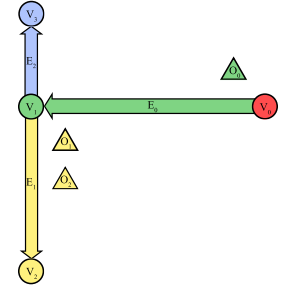
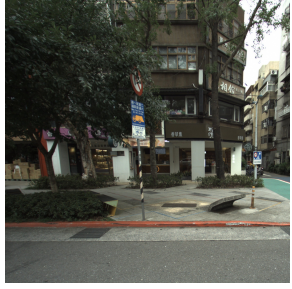
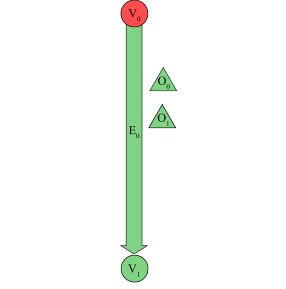
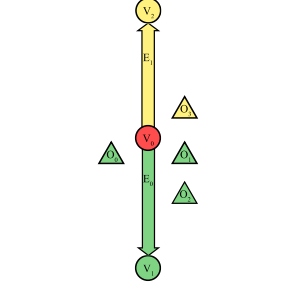
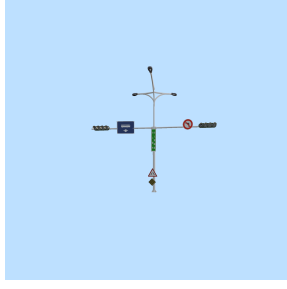
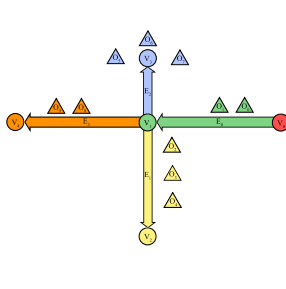
原始圖片	投影圖	結果圖	布局
			
			
			
			
			

表 5-4: 平拍真實場景結果圖。

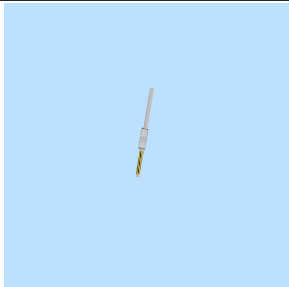
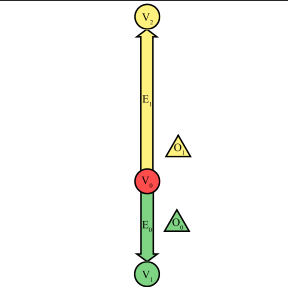
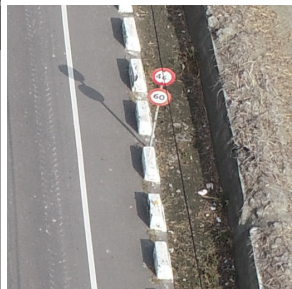
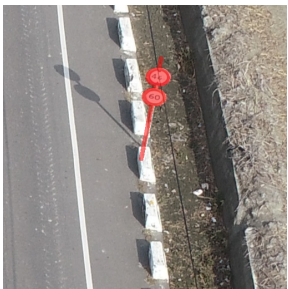
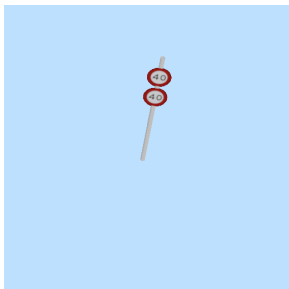
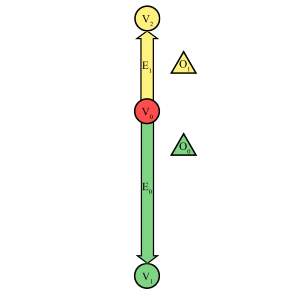
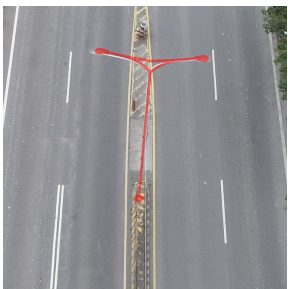
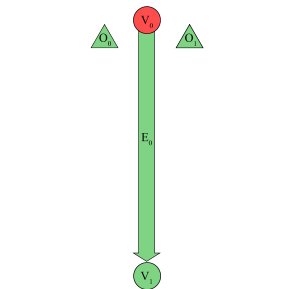
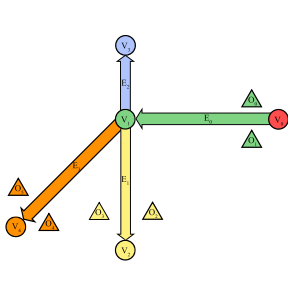
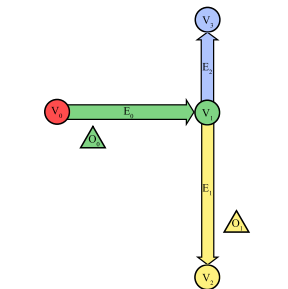
原始圖片	投影圖	結果圖	布局
			
			
			
			
			

表 5-5: 空拍真實場景結果圖。

目標	$E^C(\text{cm})$	$E^D(\text{degree})$	$E^S(\text{cm})$	$E^P(\text{cm})$
三臂紅綠燈	1.30	1.46	1.52	2.52
鐵架	1.71	1.7	1.39	4.41
高低路燈	1.29	1.42	1.66	1.71

表 5-6: 特殊桿子的元件姿態誤差與桿子誤差結果。

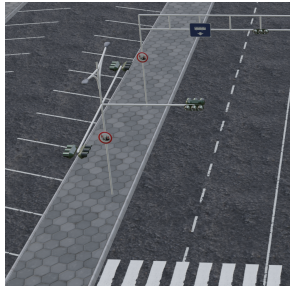
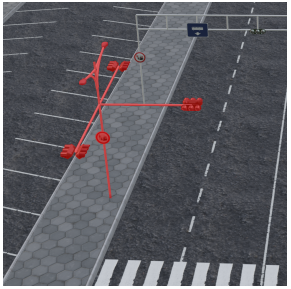
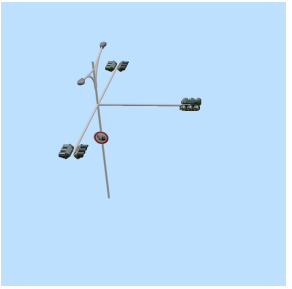
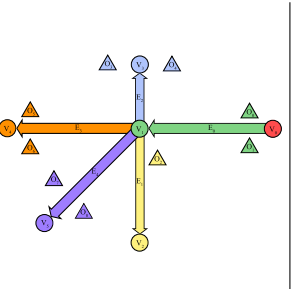
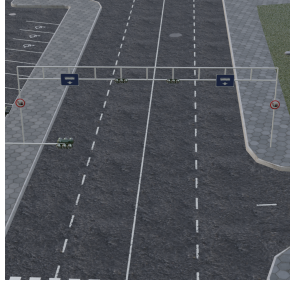
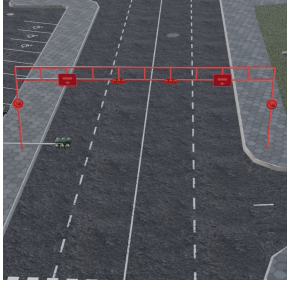
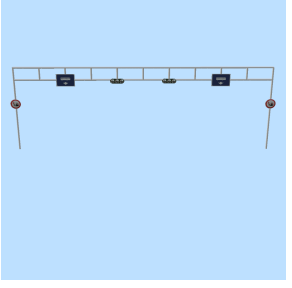
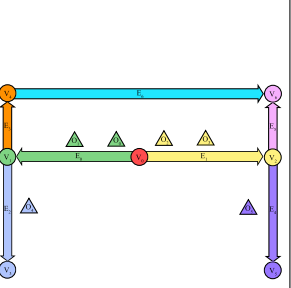
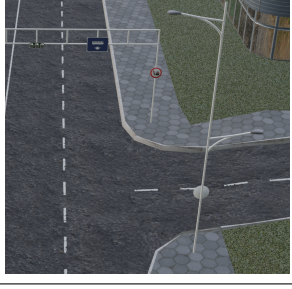
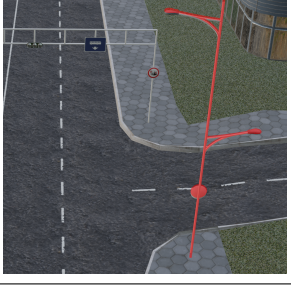
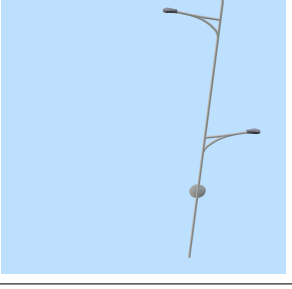
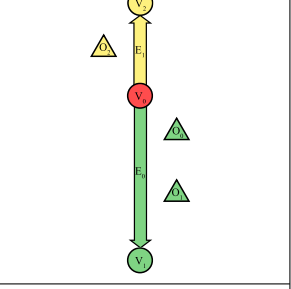
原始圖片	投影圖	結果圖	布局
			
			
			

表 5-7: 特殊桿子結果圖。

第 6 章 結論與展望

本論文提出了一個框架，用於自動生成空拍圖中的交通設施桿子。本論文從空拍圖中檢測元件 (Component)，用於定位桿子的可能分布。爲了使生成的桿子外觀更接近目標信號桿，本論文利用強化學習 (RL) 學習了一種策略，根據空拍圖中目標的熱圖 (Heat Map) 訊息，調整生成的桿子布局 (Layout)。隨後，在平拍與空拍的真實場景中進行的評估證明了本論文框架的有效性 (Validity) 和穩健性 (Robustness)，並且，證實本論文的框架可以應用於廣泛的場景。本論文還通過設計的實驗展示了建構的桿子具有相當的多樣性，並且，在交通信號桿子的生成上展示了卓越的成果。

然而，本論文在桿子與元件 (Component) 的生成上都是使用預先產生的三維模型來組合，僅能針對每個模型進行簡單的幾何轉換 (Transformation)。然而，在現實場景中，許多的標誌或桿子本身由於受到外力的影響，可能出現變形的情況。因此，本論文未來可以針對每個物件去進一步偵測其變形狀況，從而更精細地調整每個物件的外型，使其更符合現實環境中所呈現的模樣。

参 考 文 献

- [1] N. Snavely, S. M. Seitz, and R. Szeliski, “Photo tourism: Exploring photo collections in 3d,” *ACM Trans. Graph.*, vol. 25, p. 835–846, July 2006.
- [2] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. M. Seitz, “Multi-view stereo for community photo collections,” in *2007 IEEE 11th International Conference on Computer Vision*, pp. 1–8, 2007.
- [3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “Nerf: Representing scenes as neural radiance fields for view synthesis,” 2020.
- [4] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, “3d gaussian splatting for real-time radiance field rendering,” 2023.
- [5] O. Št’ava, B. Beneš, R. Měch, D. G. Aliaga, and P. Krištof, “Inverse procedural modeling by automatic generation of l-systems,” *Computer Graphics Forum*, vol. 29, no. 2, pp. 665–674, 2010.
- [6] C. A. Vanegas, I. Garcia-Dorado, D. G. Aliaga, B. Benes, and P. Waddell, “Inverse design of urban procedural models,” *ACM Trans. Graph.*, vol. 31, Nov. 2012.
- [7] D. Ritchie, B. Mildenhall, N. D. Goodman, and P. Hanrahan, “Controlling procedural modeling programs with stochastically-ordered sequential monte carlo,” *ACM Trans. Graph.*, vol. 34, July 2015.
- [8] H. Huang, E. Kalogerakis, E. Yumer, and R. Mech, “Shape synthesis from sketches via procedural models and convolutional networks,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 8, pp. 2003–2013, 2017.
- [9] J. Guo, H. Jiang, B. Benes, O. Deussen, X. Zhang, D. Lischinski, and H. Huang, “Inverse procedural modeling of branching structures by inferring l-systems,” *ACM Trans. Graph.*, vol. 39, June 2020.
- [10] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, and N. Paragios, “Shape grammar parsing via reinforcement learning,” in *CVPR 2011*, (Colorado Springs, CO, USA), pp. 2273–2280, IEEE, June 2011.

- [11] U. Transportation and F. Administration, *Manual on Uniform Traffic Control Devices for Streets and Highways: 2009 Edition with 2012 Revisions*. Colchis Books, 2009.
- [12] D. o. T. State of California, “I-5 north coast corridor pwp/trep,” 2010.
- [13] 新北市政府新建工程處, “新北市民間興闢道路標準圖,” 2019.
- [14] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, “Segment anything,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4015–4026, 2023.
- [15] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle adjustment — a modern synthesis,” in *Vision Algorithms: Theory and Practice* (B. Triggs, A. Zisserman, and R. Szeliski, eds.), 2000.
- [16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” 2018.
- [17] Q.-Y. Zhou, J. Park, and V. Koltun, “Open3d: A modern library for 3d data processing,” *arXiv:1801.09847*, 2018.
- [18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017.
- [19] Y. Feng, J. Wang, *et al.*, “Gps rtk performance characteristics and analysis,” *Positioning*, vol. 1, no. 13, 2008.
- [20] B. O. Community, *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam, 2021.