自動駕駛系統感測器資料視覺化、標記與融合

By Avinash Nehemiah and Mark Corless, MathWorks

開發 ADAS 或完全自動駕駛（無人）系統感知演算法的工程師通常需要處理大量的輸入資料。然而這些資料來自於許多種不同類型的感測器，如雷達、光達(LiDAR)、攝影機和標線偵測…等等，在資料能夠真正提供問題的解答之前，往往也衍生了更多的問題，例如，到底如何理解這些資料並實現到生活中? 不同感測器資料之間的差異又該如何調解校正? 當演算法開發完成後，如何評估結果產出?

本篇文章所介紹 MATLAB®與自動駕駛系統工具箱(Automated Driving System Toolbox™)的功能將幫助你解決上述問題。我們將探討四個主要的任務：車輛感測器資料的視覺化、路面真實標記、多種感測器的資料融合、以及感測器資料合成以測試追蹤與融合的演算法。

車輛感測器資料的視覺化

資料的解讀是感知系統開發階段初期最重要的挑戰，因為感測器是以不同的格式、頻率來傳遞其輸出資料。舉例來說，攝影機提供 3D 矩陣圖片、LiDAR 提供一堆點資料、嵌入式或智慧攝影機則提供包含車輛、標線、其他物件等細節資料的物件清單。這些輸出資料特性迥然各異，實在讓人很難建構出整體畫面的概念(圖 1)。

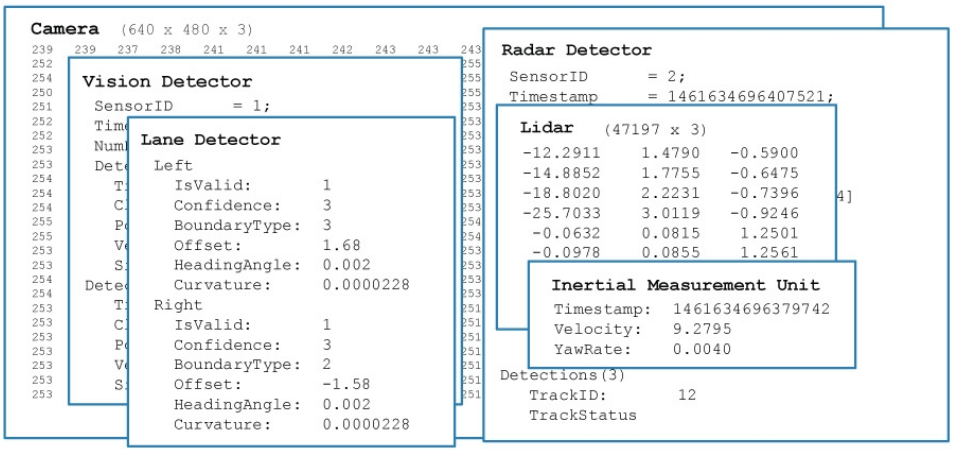

圖 1. 車輛感測器資料範例

在初期階段，我們需要很清楚地了解感測器如何描繪呈現車輛周遭的環境。這時最好的視覺化形式是鳥瞰圖，因為它可讓我們將不同感測器在同一個地方取得的資料視覺化呈現。

我們先利用 MATLAB 與自動駕駛工具箱裡的視覺化工具來建立鳥瞰圖，接著加上更多細節來查看以下這些物件：

- coverageAreaPlotter 顯示感測器涵蓋範圍
- detectionPlotter 顯示透過視覺、雷達、LiDAR 感測器偵測到的物件清單
- laneBoundaryPlotter 將車道標線偵測覆蓋到圖片上

我們現在已將感測器的涵蓋範圍、偵測結果與標線邊界等資料精確地視覺化出來(圖 2)。

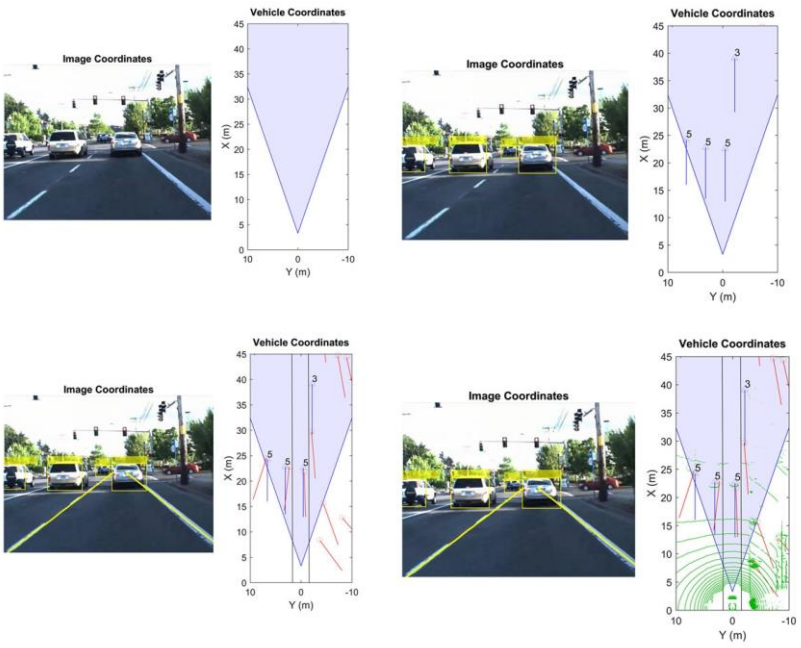

圖 2. (左上起順時針方向) 感測器涵蓋範圍圖、將車輛坐標轉換為影像坐標、標線與雷達偵測圖、LiDAR 點雲圖。

路面真實標記（Ground Truth Labeling）自動化

以機器學習或深度學習技術來訓練物件偵測器時需要有路面實況(ground truth)。而現有偵測演算法的評估也是很重要的。路面實況的建立通常是一個勞力密集的

過程，通常需要人工手動將物件標記一個一個插入到影片中的每一個畫面。自動駕駛工具箱中的 Ground Truth Labeler app 則運用並納入電腦視覺演算法來加速這個真實路面標記過程。這個 app 有幾個主要的功能(圖 3)：

- **車輛偵測(vehicle detector)**利用集合通道功能(aggregate channel feature, ACF)，可自動偵測並標記主畫面中的車輛。
- **時間插入器(temporal interpolator)**，可在所選主畫面（key frames）之間的所有畫面中，標記被偵測到的物件。
- **點追蹤器(point tracker)**，利用 Kanade-Lucas-Tomasi (KLT)演算法的其中一個版本去追蹤橫跨各畫面間欲關注的區域。
- **新增演算法(add algorithm)**，讓你可以新增客製演算法，並簡化物件偵測的迭代開發。

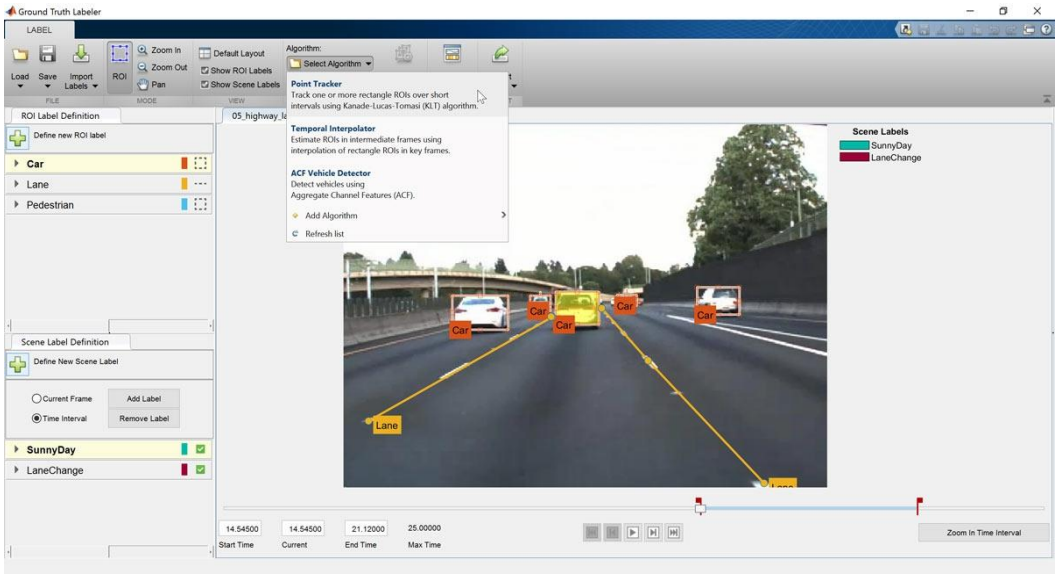

圖 3. 路面真實標記器（Ground Truth Labeler）app

多種感測器資料的融合

幾乎每一個感知系統都會使用到多個互補的感測器的輸入資料。由於每種感測器所提供的結果可能有些許的不同，要調合這些從各種感測器來的資料很具挑戰性---舉例來說，視覺偵測器可能回報車輛在某一個地方，而雷達偵測器卻顯示同一輛車位在附近一個完全不同的位置。

自動駕駛工具箱內的 multiObjectTracker（多重物件追蹤器）可協助追蹤及融合偵測結果。其中一個常見的應用是把雷達與視覺偵測器的結果融合，以改善對周圍車輛位置的估計(圖 4)。

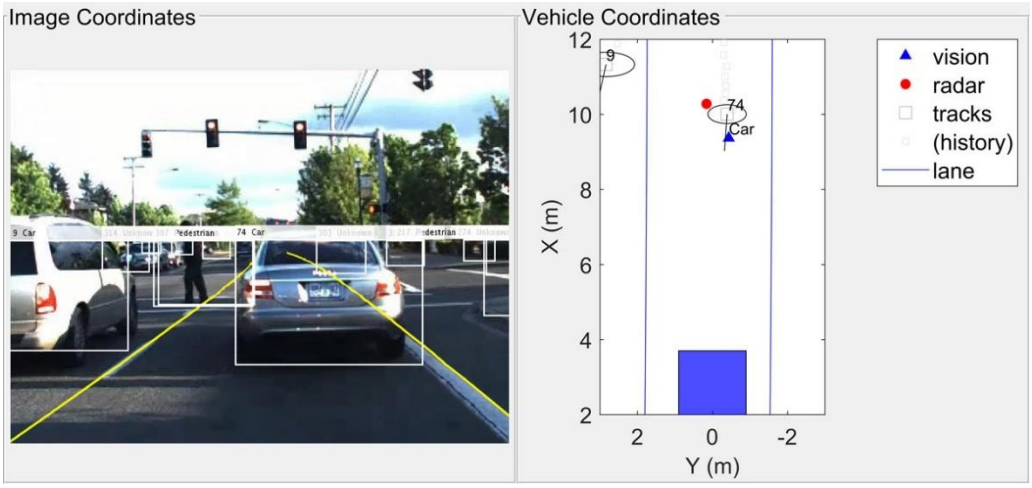

圖 4. 多重物件追蹤器（multi-object tracker），在這裡被用來融合雷達資料(紅色圓點)與視覺偵測資料(藍色三角形)，以對車輛位置(黑色橢圓)產生更精準的估計。

合成感測器資料以產生測試情境

有些測試情境太過危險而不易以真正的車輛來執行，例如即將發生碰撞的情境，而有的情境則需要特定的天候條件，比如多雲等等。針對這樣的挑戰，我們可以透過合成物件層級的感測器資料，來產生包括道路、車輛、行人等虛擬物件的情境。我們可以利用這些合成資料來測試追蹤與感測器融合演算法(圖 5)。

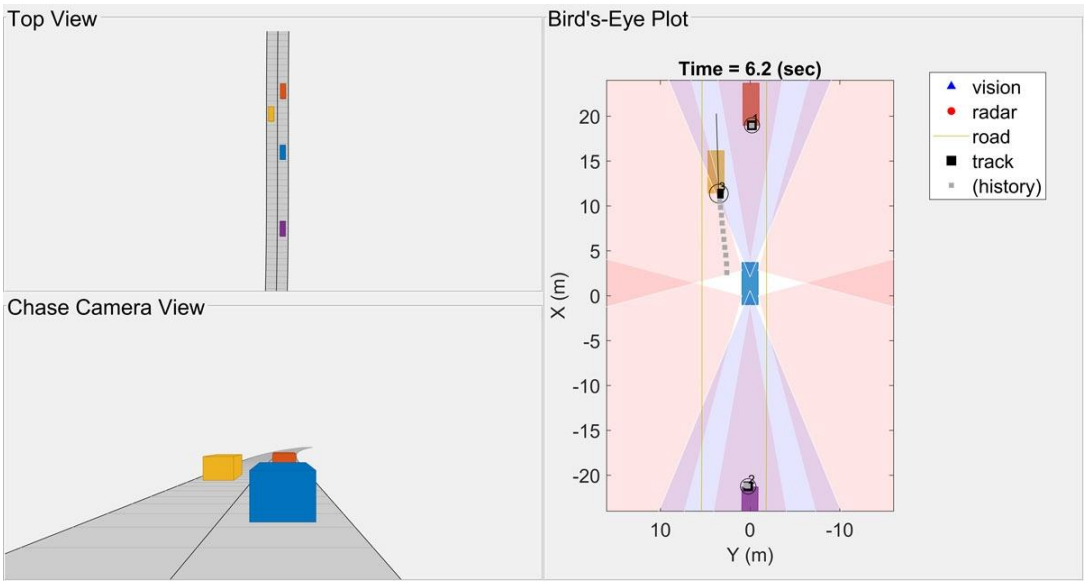

圖 5. 俯視(Top View)、追蹤攝影機視角(chase camera view)、合成測試情境的鳥瞰圖(birds-eye plot)。

將車輛資料應用於感知系統

車輛資料的視覺化、融合、與合成為物件偵測的演算法已經為自駕車的開發奠定了基礎，此時 MATLAB 演算法已經可以準備佈署到硬體上了，我們可以利用 MATLAB-C 轉碼器(MATLAB Coder™)來產生可攜帶、符合 ANSI/ISO 的 C/C++程式碼來整合到嵌入式環境之中。

若想了解更多自動駕駛系統工具箱的功能，請參考：
https://www.mathworks.com/products/automated-driving.html