

國立台北大學資訊工程學系專題報告

用於生成駕駛路線的最佳化車道與路徑對齊方法

Optimized Lane Route Alignment for Driving Route Generation

專題組員：陳鵬元、陳韋豪、李易宸、陳立哲

專題編號：PRJ-NTPUCSIE-113-007

執行期間：113 年 09 月 至 114 年 06 月

1. 摘要

在本次專題報告中，我們嘗試透過地圖路徑與影像中車道資訊的對齊，來生成駕駛視角下的行車路線。地圖路徑提供了俯視角度的全局資訊，能有效補足車輛前裝相機因視角限制所造成的資訊不足。此外，藉由運動感測器所提供的 GPS、車速與角速度資料，當影像中缺乏明確車道線時，仍能協助轉換地圖路徑，持續生成符合駕駛視角的行車路線。

目前的實驗結果顯示，在車輛直線行駛時，所生成的路線能穩定地位於本車道中央，並能指出前方路口的正確岔路方向。然而，在轉彎前會出現縱向誤差，轉彎後路線也需要較長的時間才能回到正確位置。

2. 簡介

相較於地圖視角的導航路徑，駕駛視角的行車路線可以帶來更直覺的導航體驗。2020 年新創公司 Phiar 推出 AR 導航系統，充分展示了此體驗。如該張圖片(圖 1)所示，導航上的路線會以駕駛人的視角像地毯一樣投影在車道線上。

直觀的想法是透過行車軌跡預測模型來實現。雖然行車軌跡與行車路線在任務上有所不同：前者預測的是未來每個時間點的车辆位置，後者則著重於預測未來一段固定距離或範圍內的路徑，但兩者都可以用一系列點的座標表示。因此，只要將軌跡預測模型的訓練資料從時間序列軌跡改為距離範圍內的路線點序列，便可應用於行車路線的預測。

在自動駕駛系統中，軌跡預測大致可分為三個主要階段：

感知 (Perception)：辨識並理解車輛周圍的環境，包括行人、其他車輛、車道線、人行道等與行駛決策密切相關的物件與標誌。

預測 (Prediction)：推估周遭物體的未來動態，例如行人的移動方向、其他車輛的行駛路徑，以及本車未來的位置變化。

規劃 (Planning)：結合上述感知與預測資訊，產生一條最安全、最平穩的行車軌跡。

目前主流的深度學習模型往往仰賴大量的感知輸入，例如高精度地圖 (HD Map)、雷達、多視角攝影機 (Multiview Cameras) 等。這些高成

本的感知系統，無論在硬體建置、資料蒐集或後續處理方面，都對系統落地造成極大挑戰。

為了解決這個問題，新創公司 comma.ai 於 2016 年推出了 openpilot——一套低成本的自駕系統，2022 年也開放了可供訓練的開源程式碼版本。Openpilot 僅使用單一前置鏡頭即可實現行車軌跡預測。然而，我們認為僅靠前裝相機的輸入仍存在侷限，特別是在遇到視線受阻（例如路口遮擋或車道線模糊）時，可能導致預測不準確。

因此，我們引入了俯視角度的輔助資訊，以補足前視角的盲點。為了兼顧成本與實用性，我們選擇使用導航地圖，而非高精地圖。導航地圖僅提供路徑資訊，形式為一系列經緯度節點。配合一組低成本的車輛感測器（提供 GPS 位置、車速與角速度），我們以此三種資訊作為模型輸入，進行駕駛視角下的行車路線生成。

3. 專題進行方式

本專題採用 GitLab 進行多人協作，並將資料集儲存在校內工作站，方便團隊進行共編與測試。部分由本地蒐集的資料，則透過 Google Colab 進行初步測試與驗證。本次實驗也與同實驗室的學長合作進行研究，雙方以每週一次的頻率進行進度回報與討論。

在資料調用方面，我們針對車用資料集中的 NuScenes 進行研究，從其官方網站公開的數據中擷取 GPS、yaw rate、speed 以及車道線辨識等

資料，並加以分類與評估其相互關聯性。根據官方文件說明，逐步分析其資料儲存格式，並轉換成符合工作站運算需求的格式，以便後續進行路線投影的計算與分析。

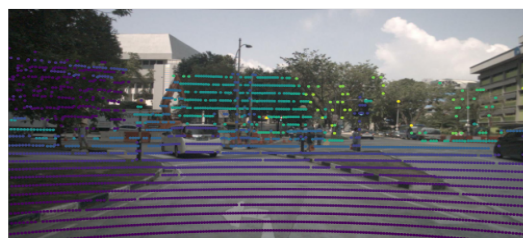


圖 2: NuScenes 資料在影像之投影

本階段的主要困難在於，初期對資料點投影的概念尚不夠具體，導致在確認資料於影像中的實際意義時，無法正確掌握其投影方式。我們在將 NuScenes 所儲存的三維座標轉換為本系統慣用的矩陣格式後，才成功完成正確的投影結果，如圖二所示。

在資料調用與座標轉換的過程中，我們利用相機的內參與外參，將世界座標系轉換至相機座標系，並進一步將其轉換為俯視圖（Bird's Eye View）座標系，以車輛中心作為原點（如圖 3 所示）。從車道線偵測模型取得的車道線資訊，在辨識出本車道後，會與地圖上的路徑進行比對與匹配，以利後續的應用。

接著我們探討兩種情境，第一種為車道線對齊的狀況。在此情境中，導航路徑與實際車道線之間的相對關係非常關鍵。若僅依賴 GPS 資料將地圖中的導航路徑直接轉換至車輛所在位置，容易因 GPS 定位誤差而導致行車路線發生偏移。因此，我們設計了額外的校正步驟來修正這種偏差。

具體作法是，擷取一定範圍內多個位置的 GPS 資料與車輛行進方向，並以這些 GPS 點為起點生成多條可能的行車路徑。接著，我們將每一條候選路徑與本車道的中線進行誤差計算，並將此誤差定義為 alignment loss。若某條路徑的 alignment loss 最小，則表示該路徑與實際車道中線的偏差最小，也最能準確對應預期的行車路徑。

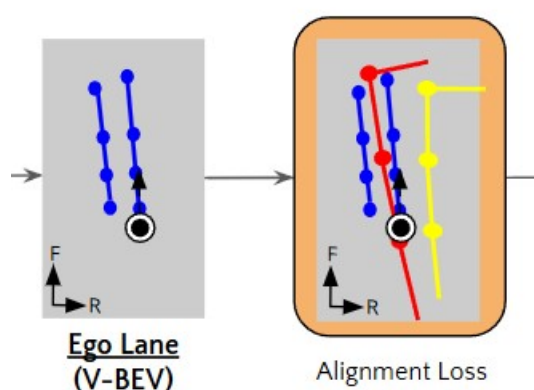


圖 3:左:俯瞰視角坐標系，上方代表車輛的正前方，黑色點則是代表車輛目前的位置。右:alignment loss 計算可能路線，紅色和黃色為可能計算路線，紅色為 loss 最小的結果

第二種情境為無車道線的道路。在這種情況下，由於缺乏可用的車道中線作為對齊基準，我們改採車輛本身的角速度(yaw rate)與速度(speed)作為參考，搭配 GPS 所提供的路徑資訊，以維持定位的連續性(continuity)與平滑程度(smoothness)。其中，「連續性」指的是資料點之間的銜接是否穩定，避免出現突兀的大幅跳躍；而「平滑程度」則代表運動軌跡的流暢性，強調變化需自然穩定，符合實際行車狀況。

針對此情境，我們設計了兩個 loss 值來輔助定位：

Temporal Smoothness Loss：透過前一時刻的位置資訊，計算移動距離與方向變化量，判斷定位是否具有良好的連續性。

Sensing Consistency Loss：結合前一點位置、當前速度與角速度，利用車輛運動模型推估預期位置，再與實際 GPS 位置比較，評估其一致性。

綜合這兩項 loss 的結果，我們會選擇加總 loss 最小的位置作為目前車輛的定位點。

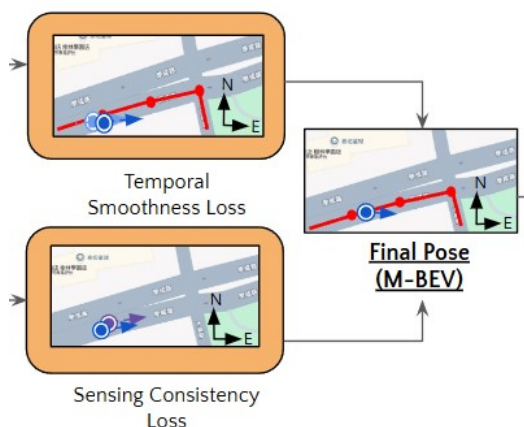


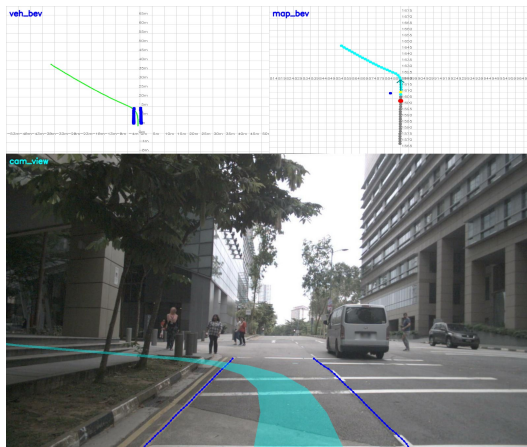
圖 5:Temporal smoothness loss 和 sensing consistency loss 加成後的 final pose。

總結而言，在有車道線的情況下，我們使用 Alignment Loss 將地圖路徑校正至車道中線；而在無車道線的情況下，Alignment Loss 為 0，因此僅有 Temporal Smoothness Loss 與 Sensing Consistency Loss 發揮作用，協助維持路徑的穩定性與準確性。

4. 主要成果與評估

最終的呈現結果包含三張圖片(如圖 6 所示)，分別展現不同視角下

的資訊整合：



左上角的圖片（veh_bev）是以車輛為原點的俯視視角圖。圖中 y 軸代表車輛的行駛方向，畫面上的兩條線為車道線偵測模型辨識出的本車道邊界；而綠色線條則代表車輛所看到的導航路徑。透過這種視角，可以清楚地觀察車道線與導航路徑之間的對齊關係。右上角的圖片（map_bev）則是以地圖為基準的俯視視角圖。圖中右側為東、上方為北，整條線即為模擬的導航路徑。當中青色區段代表尚未行駛的路段，灰色部分則表示已完成的路徑。紅色點為車輛原本的實際座標，而藍色點則是經過隨機擾動後的 GPS 座標。由於此圖為靜態地圖視角，因此導航路徑不會隨時間推移而移動，能作為真實與預測位置的對照基準。下方圖片（第一人稱視角投影圖）為將預測路徑資訊投影至駕駛人視野中的結果。畫面中間會出現一條類似地毯的圖形，表示接下來要行駛的導航路徑；而兩側由藍色點構成的線則為即時偵測到的車道邊界，協助駕駛者更直觀地理解接下來的行進方向。

這些輸出的圖片可以串接成影片，模擬車輛在特定場景中行進的過

程。

從上圖可以觀察到，導航路徑所構成的「地毯」整體上維持在車道線範圍內運行，且在多叉路口時也能正確指向應前往的方向，這些部分的表現相對穩定、效果良好。然而，我們也發現了幾個仍待改進的問題：

首先，在**轉彎即將結束時**，導航路徑偶爾會短暫偏離正確方向，之後才重新回正。這可能表示在轉彎過程中，僅依賴角速度與速度的運動推估無法精準預測車輛位置，因此我們有必要進一步探討是否需要引入其他感測資訊或強化轉彎時的動態模型。

其次，在**進入轉彎之前**，導航路徑常會出現縱向偏移的情況，使得預測軌跡看起來像是提早或延遲轉彎，進而影響定位與路徑的準確性。為了解決這個問題，可能需要提前偵測畫面中的交叉路口，並根據其位置進行校正。然而，實現此功能可能需建構一套叉路口識別模型，短期內尚不容易完成。

我們也將本系統的結果與 OpenPilot 所生成的路徑進行比較（如圖 7 所示）。從圖中可以發現，OpenPilot 所產生的導航路徑主要呈現直線延伸，缺乏對於彎道路段的適應能力。而我們所開發的系統則能依據 ground truth 路徑準確地進行轉彎，顯示出在動態環境下具備更高的路徑適應性與靈活性。

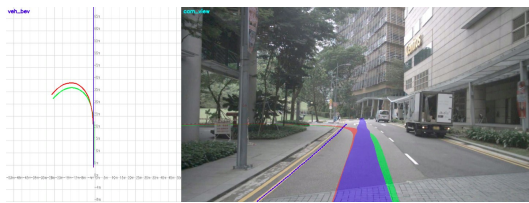


圖 7:與 openpilot 的比較。紅色：ground truth。綠色：本演算法的預測結果。藍色：openpilot 預測的結果。

5. 結語與展望

原先我們預期能利用完整的 NuScenes 資料集進行更大規模的測試，或是嘗試套用實際的導航路線進行驗證與實驗。然而，受到時間限制影響，無法進一步嚴謹測試其效能。但從初步成果得知將地圖路徑與車道線進行對齊以生成駕駛視角的行車路線是有可行性的。期望此次的專題能在後續開啟一個有價值的研究方向。

這一年的專題研究讓我們收穫良多，不僅實際體驗到在實驗室中與他人合作的過程，也學會了如何在組內有效溝通、分工協作，並共同克服研究中遇到的各種挑戰。未來，我們也期望能將這次累積的經驗應用於後續的研究工作與團隊討論中，持續提升自己的實作與合作能力。

6. 銘謝

感謝我們專題教授的指導與實驗室學長的協助，協助我們完成專題且教導我們如何與小組溝通並完成目標。

7. 參考文獻

[1] Ali Almagbile, Jinling Wang, and Weidong Ding. Evaluating the performances of adaptive kalman

filter methods in gps/ins integration. *Journal of Global Positioning Systems*, 9(1):33 – 40, 2010.

[2] David Andersson and Johan Fjellström. Vehicle positioning with map matching using integration of a dead reckoning system and gps, 2004.

[3] Mohamed M. Atia and Steven L. Waslander. Map-aided adaptive gnss/imu sensor fusion scheme for robust urban navigation. *Measurement*, 131:615 – 627, 2019.

[4] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194 – 220, 2017.

[5] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621 – 11631, 2020.

[6] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified

model to map, perceive, predict and plan. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 14398 – 14407. IEEE Computer Society, 2021.

[7] Li Chen, Tutian Tang, Zhitian Cai, Yang Li, Penghao Wu, Hongyang Li, Jianping Shi, Junchi Yan, and Yu Qiao. Level 2 autonomous driving on a single device: Diving into the devils of openpilot, 2022.

[8] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In International Conference on Computer Vision (ICCV), 2021.

[9] comma.ai. openpilot. <https://github.com/commaai/openpilot>. Accessed: 2024-04-18.

[10] comma.ai. openpilot 0.94. <https://blog.comma.ai/094release/>. Accessed: 2024-04-18.

[11] Friedrich Fraundorfer and Davide Scaramuzza. Visual odometry: Part ii: Matching, robustness, optimization, and applications. IEEE Robotics & Automation Magazine, 19(2):78 – 90, 2012.

[12] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc

Proesmans, and Luc Van Gool. End-to-end lane detection through differentiable least-squares fitting. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), pages 905 – 913, 2019.

[13] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. The International Journal of Robotics Research, 32(11):1231 – 1237, 2013.

[14] Mitchell Goff. Navigate on openpilot. <https://www.youtube.com/watch?v=ZEMPQjK1ZPw&t=143s>. Accessed: 2024-04-18.

[15] Fredrik Gustafsson, Fredrik Gunnarsson, Niclas Bergman, Urban Forssell, Jonas Jansson, Rickard Karlsson, and Per-Johan Nordlund. Particle filters for positioning, navigation, and tracking. IEEE Transactions on Signal Processing, 50(2):425 – 437, 2010.