

Big data analytics on point cloud - Road detection

巨量點雲分析 - 道路偵測

組號：A34 指導教授：孫民 組員：蘇詠善、林秉言

Abstract

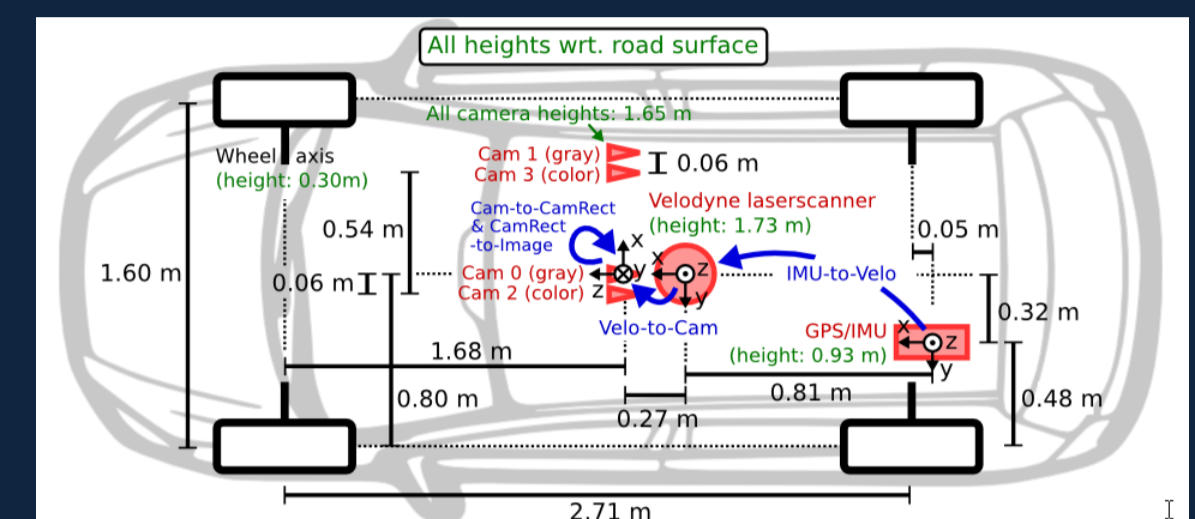
近幾年自走車盛行，Tesla與google皆已推出自走概念車，在實際道路上行駛測試，然而其研發成本十分昂貴，這起因於電腦演算法與偵測技術尚未成熟，因此需要大量高級的感測器作為輔助。若要減少開發成本，電腦的演算法優化便顯得十分重要。

本組認為每種感測器的優勢皆不同。攝影機能見到遠處景象且適合人類觀看，雷射掃描則能準確偵測近處物體們與車子之間的向量。只要能夠善用不同感測器的優勢並優化演算法，相信能大量降低自走車的感測器數量。

本組專注於對雷射掃描路況演算法的優化，資料來源於國外網站KITTI。我們使用影像處理技術將稀疏的雷射掃描點，轉為深度圖，以偵測前方即時的道路位置，未來也期待進一步以現有的演算法進行加強，偵測並標籤出車子30公尺以內的所有障礙物。

KITTI Introduction

在KITTI dataset 裡，每個行車紀錄器影片的每一個frame都有自己的相對應的3D point cloud。從下圖可以看到，車頂上的紅色圓圈是雷射掃描的中心，因此儲存的資料點都是以這個雷射的中心為基準，假設在一個frame中某一個資料點為(1,2,3)代表這個點在x方向1公尺、y方向2公尺、z方向3公尺處。



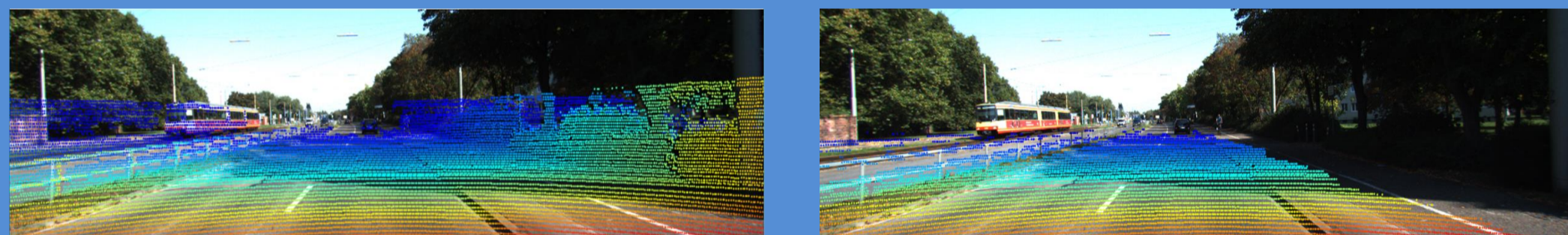
KITTI 多感測器探測車之配置

Road Detection System

本演算法一開始必須處理稀疏且分散的point cloud，將非地平面的point cloud濾除。接著進行最重要的一步「投影」，將3D離散的點雲透過矩陣，轉換成2D道路深度圖。最後透過影像處理將道路邊界偵測出來。因此我們將此演算法切割成為3個階段：3D point cloud filtering、Convert 3D point cloud to 2D road depth image、Road bounding detecting。

3D point cloud filtering :

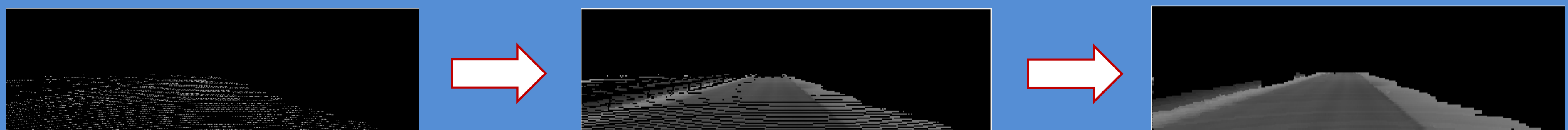
此部分我們做幾個基本假設：(a) 道路為一個平面 (b) 車子與車子間皆有保持安全距離。因此我們使用車前幾平方公尺範圍作為參考平面，將範圍內點雲fit成最佳平面，將與此最佳平面距離超過門檻的點雲濾除，剩餘點雲會涵蓋道路、人行道、分隔島、及路邊鑲邊石。



將左圖中大部分非地平面點雲濾除後，在右圖能輕易觀察到只餘人行道、道路及分隔島區段之點雲。

Convert 3D point cloud to 2D road depth image :

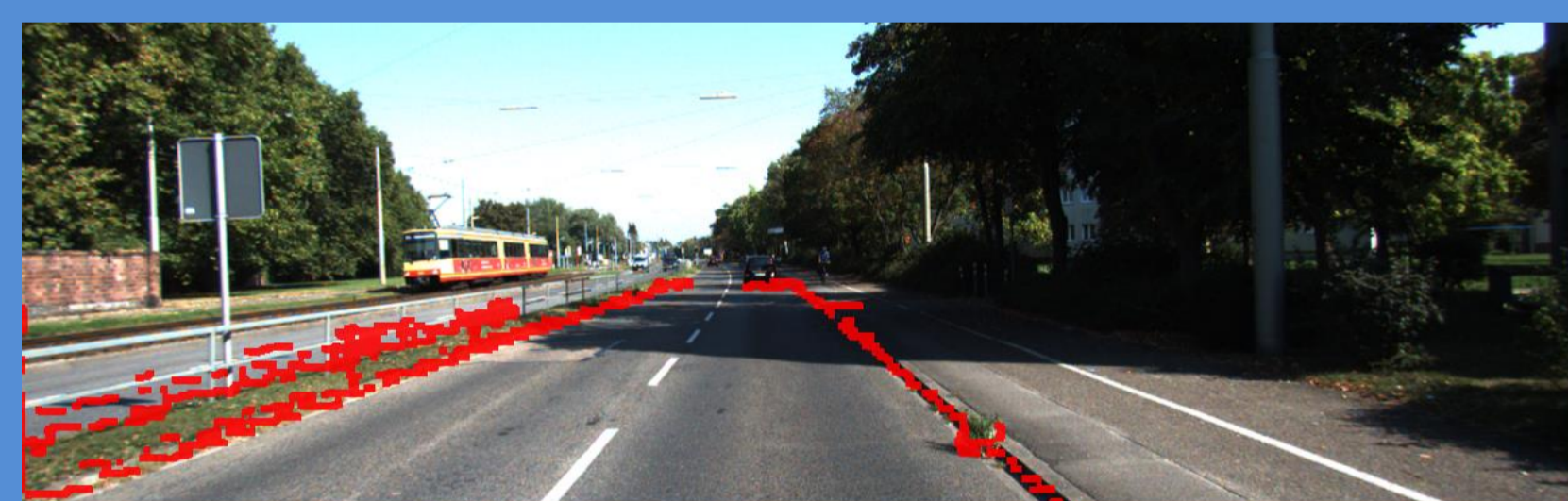
進行完第一步驟後，我們使用KITTI網站提供的轉換矩陣，將點雲投影至相對應的攝影機影像中進行處理，投影後我們將高度做為此影像的亮度值，部分區塊愈亮，代表地勢愈高，部分區塊愈暗，代表地勢愈低。由於非連續點雲的投影，因此投影後的圖形難免會有空缺，此空缺我們以Interpolation的方式進行填補，直到全部空缺皆被填滿。



投影後的地面高度圖十分離散，藉由多次interpolation的處理後，道路高度圖可以達到連續的圖形

Road bounding detection :

第二步驟完成後，我們得到一張道路高度圖。回到一開始的路況觀察，我們將發現地平面可分成 (a) 障礙物如車子、電線桿、行人 (b) 馬路 (c) 路邊鑲邊石 (d) 分隔島 (e) 人行道，而障礙物已在第一步驟被濾除，而我們想得到的道路會夾在路邊鑲邊石及分隔島之間，而它們與道路高度有落差。因此我們自製Canny filter，將此道路深度圖進行微分，並經由實驗調整無數種門檻的值，進行濾除，最後得到道路與分隔島、路邊鑲邊石之間的邊界。



Experiment Result

