附件三

國立臺北大學電機工程學系專題報告

車載乙太網路實際平台實踐環境影像系統

組員：謝允哲、黃育辰、何柏儒、曹宇慶、吳昇哲
指導老師：陳永源老師
時間:2015 年 8 月至 2016 年 6 月

1. 摘要

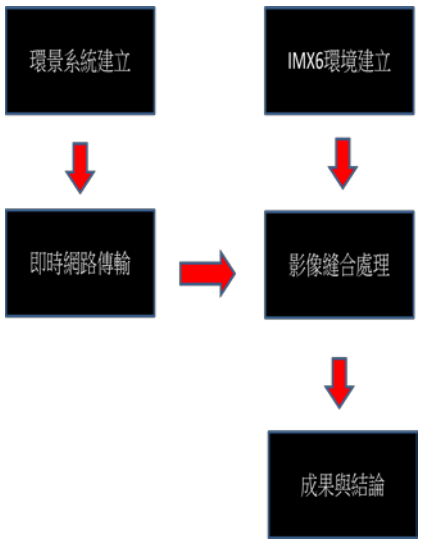
本專題報告旨在研究在電腦與電腦間透過乙太網路傳輸，此項計劃的重點在於如何使用乙太網路去達成車內與環境四周的溝通。也就是所謂的車載網路，現今科技進步越來越快，對於安全的要求也越來越高，我們希望能夠建立資訊以較快速傳輸也就是所謂即時傳輸之平台，讓駕駛者能夠透過車內的環景系統達到 360 度無死角的觀察視界，目的在於減少因死角所造成的交通意外。如:駕駛者在進行倒車入庫或路邊停車等動作時，因死角造成的意外 減至最低，環景監視系統將可發揮到最大 的輔助效能；此功能主要是利用外部 4 個 鏡頭，再搭配 OpenCV 視覺函式庫以及視點轉換技術，將同一平面上的影像對應至另一個平面上，將外部環境捕捉並進行處理與縫合， 達到使駕駛者能透過畫面得知車體與環境的相互關係。

本文使用一顆電腦攝影機透過乙太網路傳輸傳至另一台電腦，而影像縫合部分是將預錄好的影片使用縫合技術將兩部影片合成一部影片。

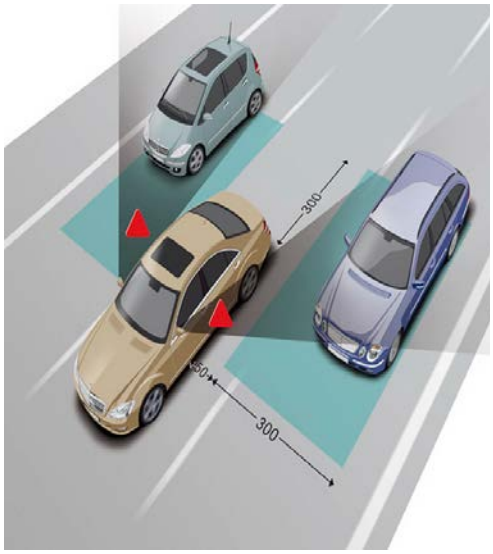
2. 簡介

本專題報告旨在研究車載視角輔助系統，透過建立乙太網路平台，將視訊即時送到車載影音系統上,結合影像縫合技術以利於用路人觀看,達到道路安全之目的。本文分為兩部份，第一部份我們透過個人電腦開啟虛擬系統 Ubuntu 並使用程式語言 C 在 Linux 系統下開發一套車用影音系統，功能有環景與車用即時網路傳輸。第二部分致力於研究即時影像縫合，在嵌入式系統 IMX6 上,將預錄好的影像透過程式語言 C,合成以利駕駛觀看的視頻 。

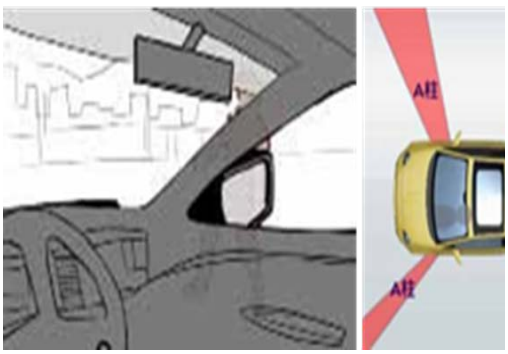
我們的專題流程如以下部分



我們製作的動機在於一般車用後視鏡只能給駕駛者有限的後視視角，為了駕駛者補足後視鏡所看不見的視野，降低事故發生率



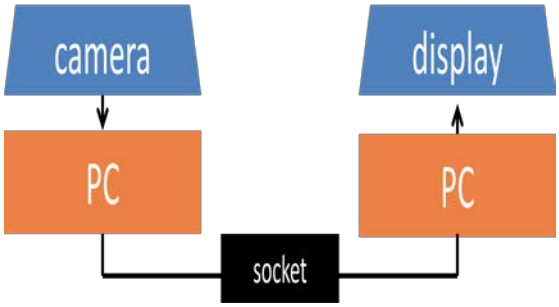
目的在減少因死角所造成的交通意外並給予駕駛者原本無法獲得的後視視角。分別在 A 柱與 C 柱四個角裝上鏡頭透過縫合影像技術便能夠得知汽車 A 柱、B 柱、C 柱死角所看不見的視 野，改善汽車死角問題，能夠有效的提供駕駛人去注意車外的狀況，並且提高安全性，降低不必要的突發事 故，減少危險，以解決因為死角而產生的情況發生。



3. 專題進行方式

本專題主要將進行方式分為兩組，一組為系統架構鏡頭組另一組為縫合組。

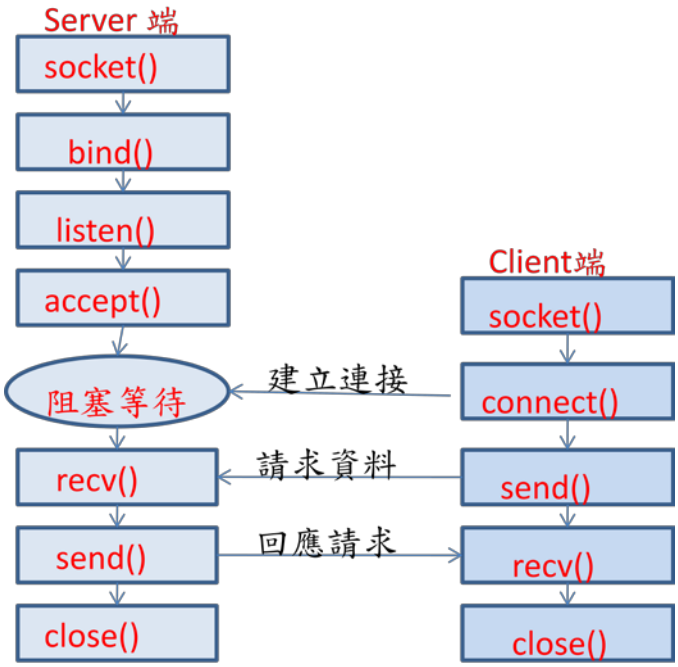
系統架構鏡頭組



單一鏡頭

透過 PC 平台下使用影像處理 OpenCV 函式庫搭配鏡頭擷取動態視訊，並以 Mat 型式存取畫面，並將畫面顯示於 PC 上

網路傳輸



Server，client 端配置

透過 socket 函式庫將操控鏡頭

的 PC 端設定為 client 端，且將欲接收畫面並做處理的 PC 設定為 server 端，再透過 socket 連線許可訪問至 server 端。連線成功後將視訊開啟，做為顯示鏡頭所擷取到的畫面，且將畫面切個成單一視訊畫格，再將每一張視訊畫格壓縮並透過 TCP/IP 協定的函式傳送至 server 端。

圖片壓縮與解碼

- 壓縮

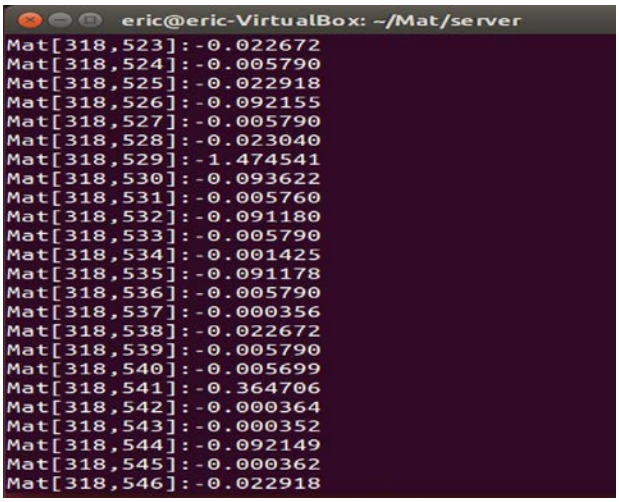
  在 TCP/IP 的協定下，傳送方式須符合 Socket 的既定形式抓取每張圖片的形式以及其內容大小並切割成多個小畫格陣列數值透過 send() 送至 server 端

- 解碼

  TCP/IP 傳輸協定下，透過 recv()形式的內建函式將接收到的小陣列畫格數值及畫面形式，使用迴圈的方式將接收到的資訊一一拼回在 server 端重新宣告的陣列，重新構成原本的畫面

圖片結構原理

- 本專題使用在 OpenCV 的宣告的形式為 Mat 陣列，並宣告其解析度為 480x640 圖片大小，我們藉由找出每一張圖片的每個二維陣列像素值，來做傳送，再利用公式計算每一張圖片所需的位元頻寬為 480*640*3 經過計算結果為 921600 個 byte 所以要達到人眼所感覺得出動態要達到每秒傳輸量為 20 張，我們需要計算這個傳輸量的大小，考量在乙太網路所容許頻寬之下，否則會出現掉封包導致畫面延遲或破圖等等的情形。



SERVER

- 連線成功後 Sever 端將會做 bind 連結的動作，以防之後會有更多的 clinet 端傳輸資料所形成的 flooding 造成的資料阻塞，之後將傳來的多個資訊，如前述所說的主要包含: ImageDate、ImageSize。Sever 端會再重新宣告相同陣列，再將資料填回陣列中完成解碼的動作。解碼完成後將 Client 端視訊重現 。

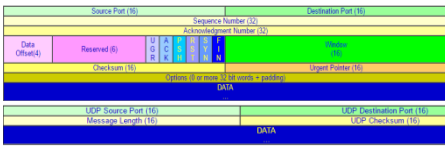
困難與解決方法

- 起初將視訊轉成影片檔並切割成每秒 30 張圖片儲存至資料夾，再將儲存的圖片傳送出去，可是由於我們經過切割影片和存檔讀檔的步驟造成速度太緩慢且儲存的動作多餘，不符合我們專題所需要的即時技術，我們之後轉向將切割畫面存放至占存記憶體中直接透過 send()與 recv()傳送之

後讓更新的資料蓋過就有的站存區，而不會造成 RAM 的壅塞

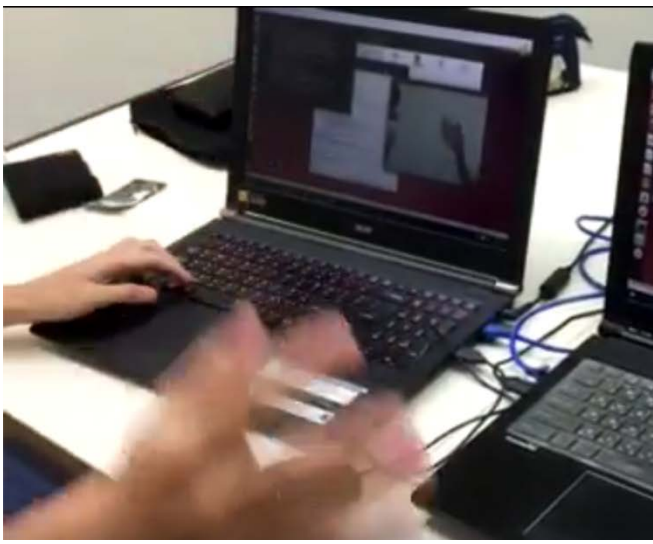
網路 SOCKET 開發過程&困難與解決方法

網路傳輸分別有 2 種Protocol 分別是 TCP/IP 和 UDP 如圖所示，分別為 TCP/IP 和 UDP 的封包表頭檔，TCP/IP 作為可靠性資料傳輸協定，在乙太完路頻寬壅塞時，可以減少傳輸封包遺失的風險，讓畫面可以完整呈現，相對的他的封包體積較大，傳輸較慢；相反的 UDP 則為快速傳輸的形式，可其在頻寬被占用情況之下，會嚴重的遺失傳輸數據，造成畫面的可讀性降低。

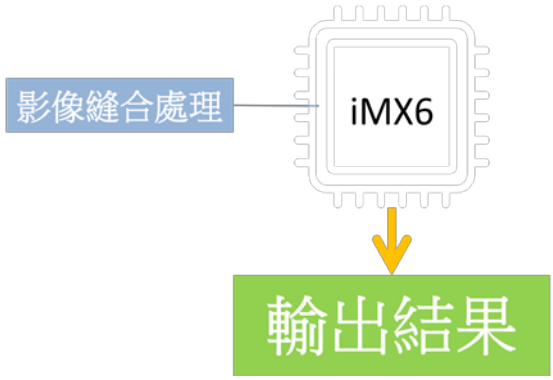


- 最初我們採納 UDP 的形式以利於傳送時間的節省，但在演算編譯中發生遇到開發困難。考量資訊可靠性，以及封包接收率所導致的破圖或無法預期的圖片處理問題，我們最終選擇可靠性高的 TCP/IP 作為我們網路傳輸的 Protocol。TCP/IP 協定封包傳輸在最終展示結果並無如預期的嚴重延遲，在能使駕駛者容許反應時間範圍內。

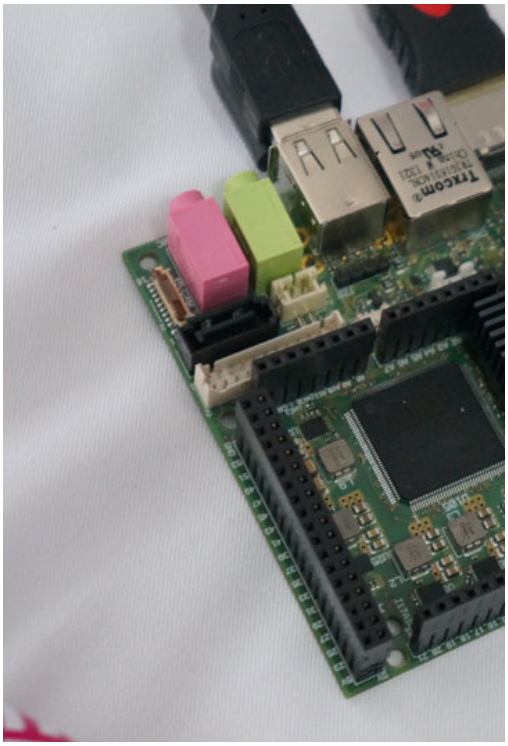
成果



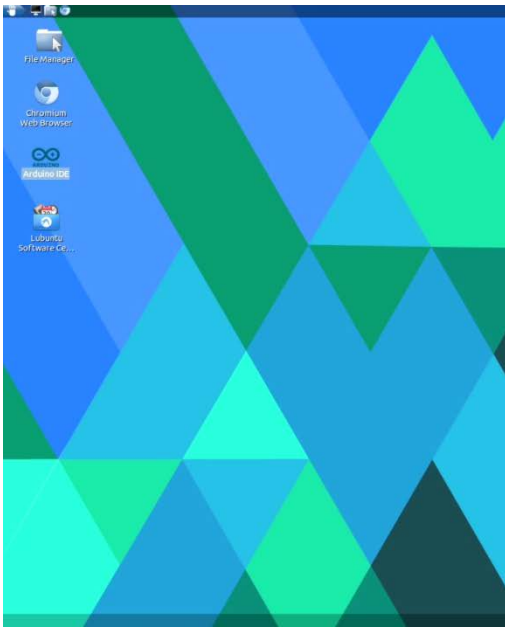
4.
5. 系統架構縫合組
6.



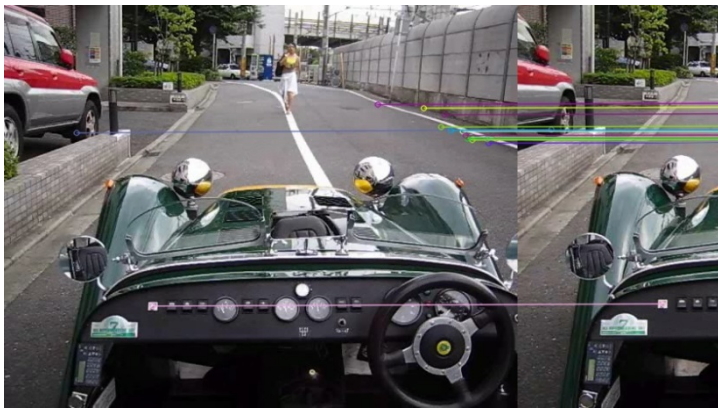
7. IMX6



- 使用系統：Udoobuntu quad v1.1



- 接收 2 個輸入影片
- 以 一組畫格 為單位，將 2 個輸入影片
  使用 ”相似點抓取” 的方法縫合



6. 結語與未來展望