

嵌入式即時立體影像開發系統之研製

吳建中、翁凱文

南臺科技大學資訊工程系

wucc@stust.edu.tw

摘要

目前市面是已經商品化的立體視覺開發系統包括微軟的 Kinect、華碩的 Xtion、及 Intel 的 Realsense。然而，這些系統兩個攝影機的距離已經固定，無法依據需求彈性地調整位置。同時，需要搭配指定的作業系統及配置高等級的處理器才能使用，因此在很多需要在移動的環境上處理立體影像常常無法滿足其需求。本論文透過 Altera Cyclone V 系列 SoC FPGA 硬體平台，並透過自行開發的 IP 搭配 HPS to FPGA Bridge 將影像傳至 FPGA 進行深度影像計算，並可以將結果回傳至 HPS 繼續進行影像處理，並將結果輸出至 LCD 或 VGA 顯示。除此之外，本論文的立體影像採用雙 Web Camera 擷取影像，然後搭配半全域匹配演算法及 Sobel Filter、Census Transform、視差優化等演算法進行立體影相匹配及最佳化，建構一個即時三維深度影像開發平台。透過 HPS 與 FPGA 的軟硬體整合，將需要加速的演算法用 FPGA 硬體電路加速，並在 Embedded Linux 下透過 OpenCV 程式庫進行軟體開發，除了提供使用者更彈性的即時立體影像開發平台，同時更讓研發人員可以透過此系統依據自己的應用需求，彈性搭配自行需求的攝影機規格及攝影機距離來完成設雛型立體影像產品的設計與開發。同時在本文演算法實現結果，實現之深度資訊估算僅使用 18,023 的 LUTs 及 580,362 的 Memories，具有低複雜性特點。最後以 Middlebury Tsukuba 標準測試圖與即時影像做測試驗證本系統的準確性。

關鍵詞：立體視覺、影像開發平台、嵌入式系統、半全域演算法、硬體處理器系統、現場可程式化開陣列

The Development and Implementation of an Embedded Real-time Stereo Image Development System

Chien-Chung Wu, Kai-Wen Weng

Department of Computer Science and Information Engineering, Southern Taiwan University of Science and Technology

Abstract

There are some stereo image systems that have developed in the market, such as Microsoft's Kinect, ASUS's Xtion, and Intel's RealSense. However, the upscale CPU is more expensive and high power consumption, so it is not suitable for mobile application. Besides, the distance between two cameras and stereo matching algorithm are limited due to the camera specifications. The proposed system uses SoC and FPGA of Altera Cyclone V series for the Embedded Real-time Stereo Image Development System. Hardware implementations would be conducted by algorithms of deep computation via the intellectual property designed in this system, and applications with large elastic demand would be implemented by software. In this paper, stereo information of images would be calculated mainly through Semi-Global Matching, Sobel filter, Census transform, and Parallax optimization algorithm. Two USB cameras set up in the Embedded Linux operating system to develop Video Capturing Software with V4L2 API. The image data would be transferred to the FPGA memory through AXI Bridges, and the depth image would be obtained with SGM algorithm, then the processed depth information would be synthesized and displayed on LCD. Meanwhile, the information would be sent back to the HPS Memory for applications in Embedded Linux. The camera specifications and stereo matching algorithms can be adjusted by

Received: April 6, 2016; first revised: Aug. 17, 2016; accepted: Sep. 2016.

Corresponding author: C.-C. Wu, Department of Computer Science and Information Engineering, Southern Taiwan University of Science and Technology, Tainan, Taiwan.

designers. The stereo matching and parallax optimization algorithms consume 18,023 LUTs and 580,362 Memories in the resources of FPGA, and the system were verified by using the Tsukuba Image Pairs from Middlebury Stereo Datasets.

Keywords: Stereo Vision, Image Development Platform, Embedded System, SGM, HPS, FPGA

壹、前言

近年來，機器視覺因為其成本、不受環境限制、穩定度較高及執行速率等優點，已經廣泛應用在車輛安全、航太與工業製程監控等領域。透過單一攝影機所得的視覺資訊，如：物體輪廓、色彩及陰影等，雖然所得到的資訊能夠解決大部份的應用，但隨著需求與應用環境複雜度的提升，二維平面影像已漸漸不敷使用，然而，怎麼得到更多的影像資訊，已經成為一個相當重要的議題。因此透過雙攝影機所產生的立體視覺的三維空間資訊，逐漸廣泛研究。

一般而言，立體視覺是透過兩部不同視點的攝影機進行取像，根據所得兩張影像進行匹配，進而計算出深度資訊，並將二維影像重建為三維空間資訊。其中深度影像資訊的估算技術，為立體視覺準確與否的關鍵，該技術是從同一環境、不同位置的攝影機所得的兩張影像，找出兩張影像中每一個像素點之間的視差(Disparity)，最後透過極線幾何關係，求出具備三維空間資訊的深度影像[1]。

深度影像估算方法分為四大步驟，分別為：(1)比對代價估算(matching cost calculation)，(2)比對代價聚合(cost aggregation)，(3)視差估算(disparity selection)，及(4)視差值優化(disparity refinement) [2]。比對代價估算是估算影像像素點於不同視差下的像素灰階相關性，最常見的比對代價，如：像素灰階差的絕對值(absolute differences, AD)[3-4]，像素灰階差的平方值(squared differences, SD) [5]，比對視窗的相關係數(normalized cross correlation, NCC)[6-7]，基於 Census 特徵的漢明距離(hamming distance, HD)[8-9]。

而比對代價聚合與視差估算，可區分為區域演算法(local method)、全域演算法(global method)[10]與半全域演算法(semi-global matching, SGM)三類。本論文所使用的半全域演算法為德國航太中心(DLR) Heiko Hirschmüller 博士所提出，該方法是以多個一維方向的動態規劃法影像匹配，模擬二維影像的全域匹配最佳化，該演算法運算複雜度小，同時具備硬體實現性高與環境耐受力好的優點。並且匹配準確性相近於全域演算法[11-12]。

綜合上述，本論文以 SGM 演算法為基礎，再結合 Sobel filter、Census Transform、視差優化等演算法，提出三維深度影像估算演算流程，利用 HPS-FPGA 架構之嵌入式平台實現其演算流程。實現結果以 Middlebury 大學的 Tsukuba 標準測試圖進行三維空間資訊的深度影像進行估算正確率分析。

隨著影像解析度的增加，處理影像資訊量也隨之大量增加，將影像進行代價估算時往往受到 CPU 運算能力不足而導致運算時間過長，為了改善運算耗時的問題，許多研發人員會透過特殊的硬體，例如：數位訊號處理器(digital signal processor, DSP)、圖形處理單元(graph process units, GPU)、現場可程式化閘陣列(field-programmable gate array, FPGA)的設備加速進行運算。DSP 雖能夠有效的加快訊號處理，但不易實現多執行緒運作，而 GPUs 運算時則會消耗過多的功率的缺點。因此，為了能夠加快訊號處理、平行化的運算及避免消耗過多的功率，本論文採用多核心處理器搭配 FPGA 實現軟硬體整合開發。

如圖 1 所示，本論文採用 Altera Cyclone V 系列 SoC FPGA 晶片進行實作，此系列晶片整合了雙核心 ARM® Cortex™-A9 MPCore™ 的硬體處理器系統 (HPS)及 FPGA，透過在 Embedded Linux OS 中，以 V4L2 API 驅動 USB video camera(UVC)攝影機，並採用 OpenCV library 將影像進行前處理，在將兩部攝影機影像資訊經由 AXI Bridges 傳送至 FPGA 的 DDR3 後，透過 semi-global matching 估算演算法進行深度影像運算，達到軟/硬體協同設計及硬體加速，同時在本系統中，可以彈性置換不同硬體演算法、也容易搭配軟體程式進行不同應用的開發。

除此之外，可以依據使用者需求彈性選購不同的攝影機與調整攝影機間的距離來完成系統需求。

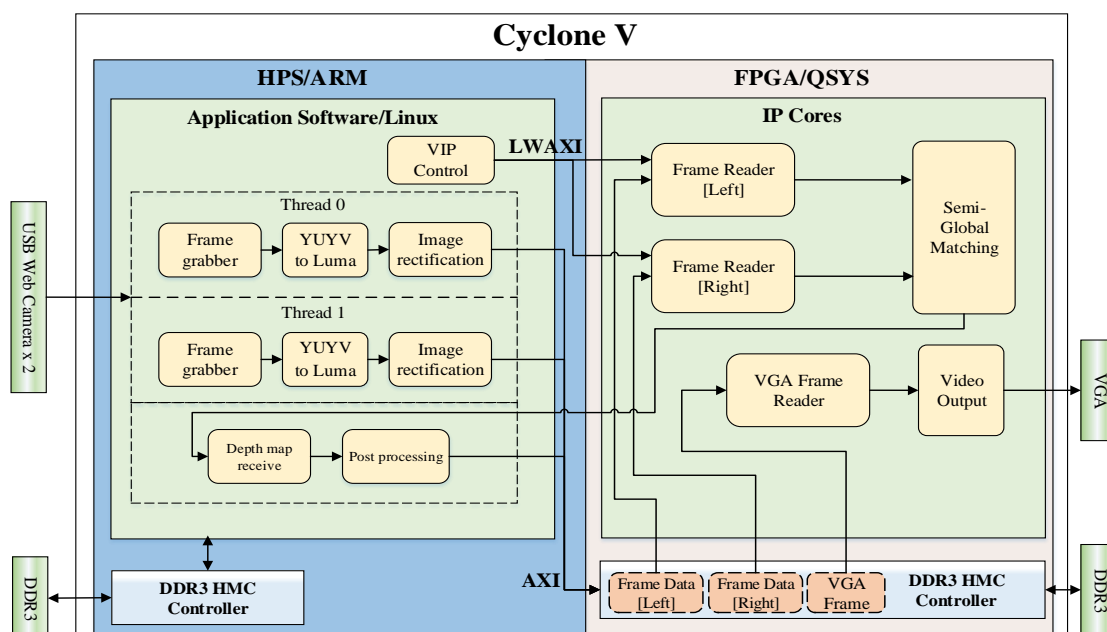


圖 1 系統架構圖

貳、方法

圖 2 為本論文系統流程圖，是由幾個子系統建構完成，包括：一、影像擷取系統，二、影像變形矯正及前處理，三、semi-global matching 硬體設計。

一、影像擷取系統

本論文為了建立輕量化的影像擷取系統，將兩部 UVC 攝影機連接至嵌入式開發平台的 USB 接頭，如圖 3 所示，為了要能夠讓 Embedded Linux 支援 UVC 裝置，需要在編譯系統核心原始碼中，在 multimedia 選項勾選，並將 UVC 驅動裝置加入編譯清單，最後儲存編譯項目後方可進行核心的編譯，當系統核心編譯完成後，透過設定系統核心，讓平台在開機啟動時，自動掛載並啟動 UVC 攝影機的驅動程式。



圖 2 系統流程圖

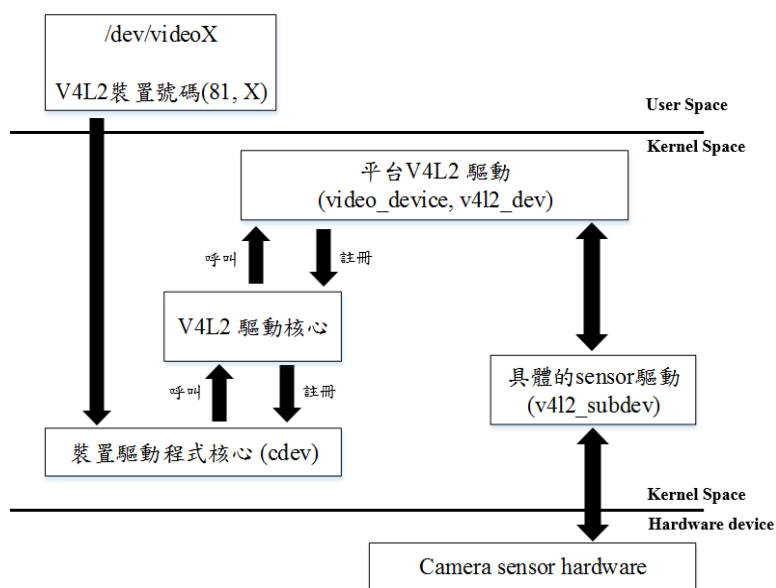


圖 3 V4L2 影像擷取原理圖

透過此種方式整合，開發者隨時可以根據不同應用的需求來替換不同功能的攝影機，只要修改部分設定即可完成，可以避免市場上如微軟的 Kinect、華碩的 Xtion、及 Intel 的 realsense 無法讓使用者依據需求來更改系統配件的缺點。

由於 UVC 攝影機事先進行高質量的壓縮，大部分攝影機採 YUV 色彩空間做為影像輸出格式，其中 YUYV 格式最常見。由於本論文主要以影像灰階影像進行處理，所以避免色彩通道的轉換，導致運算量增加，為了降低運算量，能達到即時的影像處理速度，本論文在此部份直接將原始影像的亮度值 Y 取出當作灰階處理，加速後續資料的處理。

二、影像變形矯正及前處理

如圖 4 所示，立體視覺深度資訊的建構是以極線幾何為雙攝影機的針孔成像模型與空間中一點的幾何關係。在雙攝影機與拍攝物 P 的幾何關係中，兩攝影機置於同一水平線上，兩攝影機鏡頭之間的距離為 B，此距離稱為基線距離。而在兩攝影機的光軸與影像平面的交點分別為 C₁、C₂；P₁、P₂ 分別代表空間中一點投影到兩攝影機影像平面的投影點，又稱極點。Δ x₁ 與 Δ x₂ 代表 P₁、P₂ 與兩影像平面中心點的距離；Y 為被攝物體在兩影像的垂直位置；Z 則為被攝物體距離基線的距離。因此，深度資訊可由被攝物體的點 P 與兩攝影機中間點的三維資訊(X,Y,Z)，藉由相似三角形關係 計算而得，如公式(1)與公式(2)。

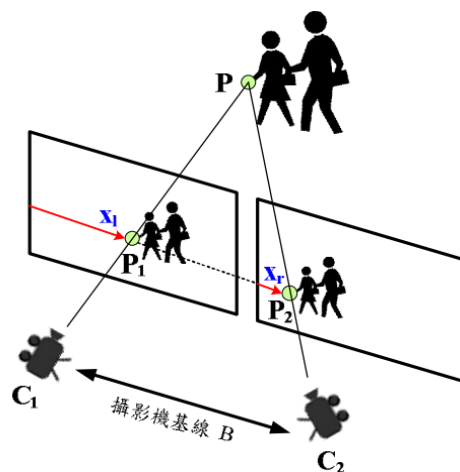


圖 4 立體視覺架構圖

$$\frac{X}{Z} = \frac{x - x_0}{f}, \frac{Y}{Z} = \frac{y - y_0}{f} \quad (1)$$

$$Z = \frac{Bf}{|\Delta x_1 - \Delta x_2|} \quad (2)$$

上兩式中， f 代表攝影機的焦距； (x, y) 與 (x_0, y_0) 分別代表點 P 投影到左影像平面中的投影點以及左影像平面的中心點； $|\Delta x_1 - \Delta x_2|$ 代表像差值。

然而兩部攝影機的鏡片及影像採集感測器參數不同，並且安裝視角及架設高度不均等，以上問題容易在影像匹配時產生些許的誤差，導致無法很好的計算出深度影像，所以當攝影機架設完成後，必需將兩部攝影機的參數計算出來並將影像矯正，方可進行影像匹配的動作。

本文以 OpenCV library 提供的 camera calibration 方法，透過拍攝西洋棋盤校正板，根據影像中角點的數量、角度及位置等資訊，將資訊進行計算後，則會得到攝影機的焦距(f)、主點(principal point)、變形值(distortion)等參數。

影像矯正流程圖如圖 5，大致上分為左/右攝影機校正包含讀入影像(load image)、提取影像角點(extract grid corner)、校正(calibration) 及儲存結果 (save results)；得到兩部攝影機參數後，則會將兩部攝影機參數傳至立體視覺校正，其中包含讀取左/右攝影機校正結果以及進行立體校正。

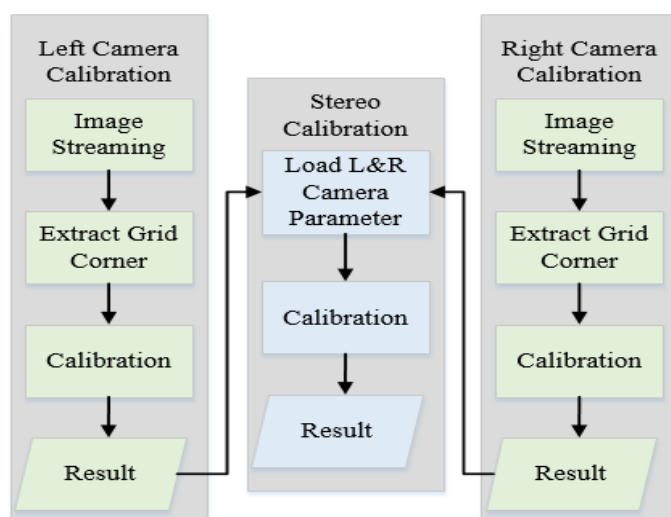


圖 5 攝影機校正流程圖

為了進行校正的動作，由已經設置好的左/右攝影機，同時拍攝含有西洋棋盤校正板的影像，並透過呼叫 OpenCV library 的函式來找尋角點的方格大小，同時自動將影像中的角點標示出來。完成上述動作後，程式便可以進行校正，計算出攝影機的內外部參數，並將其個別參數儲存起來後，再次透過呼叫立體視覺參數校正函式進行校正，經過這些步驟後，本系統能夠計算並分析左/右攝影機個別參數，最後修正西洋棋盤校正板於空間內的相對位置及調整過後攝影機的內外部參數。

如圖 6 所示，為了能減少 CPU 運算量，並得有效的特徵資訊。當影像矯正完成後，系統將透過 Sobel 濾波器來提取邊緣資訊，其作法如公式(3)~公式(5)，主要將鄰近影像與水平分量(G_x 遮罩)及垂直分量(G_y 遮罩)作旋積運算，為了有效降低運算複雜度，則將 G_x 與 G_y 取絕對值並相加求出梯度近似值。

$$G_x = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \quad (3)$$

$$G_y = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7) \quad (4)$$

$$G = \sqrt{G_x^2 + G_y^2} \cong |G_x| + |G_y| \quad (5)$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image

-1	-2	-1
0	0	0
1	2	1

G_x

-1	0	1
-2	0	2
-1	0	1

G_y

圖 6 Sobel 濾波運算子，左為鄰近影像；中為 G_x 遮罩；右為 G_y 遮罩

三、Semi-Global Matching 硬體設計

本論文透過 Verilog 設計 SGM 估算演算法，估算步驟包括：(一)影像前處理，(二)比對代價估算，(三)比對代價聚合，(四)視差估算，及(五)視差值優化，其中又以比對代價聚合與視差估算為重要步驟。圖 7 為本論文中 semi-global matching 的流程圖。

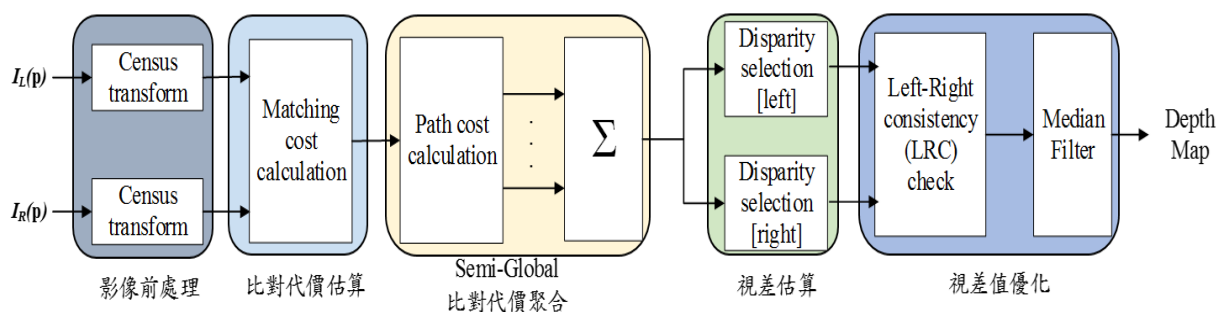
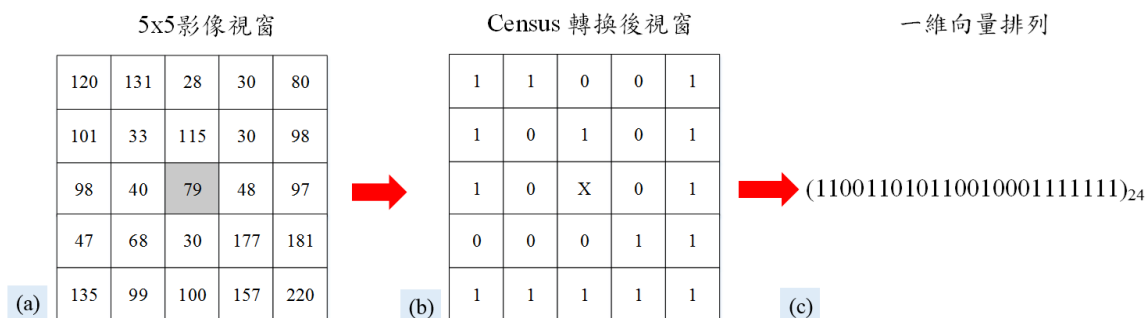


圖 7 Semi-Global Matching 流程圖

(一)影像前處理

在本論文的 semi-global matching 流程中，第一個部分為影像的前處理，由於架設雙攝影機時，因兩部攝影機的鏡片、成像感測器及架攝角度不同等因素，容易導致左/右影像的資訊產生嚴重的亮度差而部分區塊匹配結果不滿意，因此本論文捨棄過去對像素點灰階值直接運算，改採用 census transform 轉換作為第一個部分是影像的前處理的主要方法。Census 特徵需定義一個特定尺寸的視窗，將視窗內各點與視窗的中心點進行灰階值比較，若灰階值大於中心點，則該點標記為 1；反之，若小於則標記為 0。比較完畢後，將結果轉換為一維的向量排列。

如圖 8(a)所示，本論文所應用的視窗大小為 5x5，假設視窗中心的像素值是 79，如圖 8(b) 透過與中心像素進行比較後，則可以得到 Census 特徵值。



(a)目前像素座標的鄰近像素視窗 (b)與中值進行比較後得到的結果 (c)將轉換結果以一維陣列排列示意

圖 8 Census 特徵計算示意圖

(二) 比對代價估算

當左/右影像完成 census transform 轉換後，將根據編碼值進行匹配點搜尋，而搜尋過程採用漢明距離(hamming distance)的概念做為估算演算法。如圖 9 所示，漢明距離的定義是給予兩筆資訊後，根據兩訊息中找出不同的位元，並將其數量統計。在此例中，有 6 個位元不同。

如圖 10 所示，本論文硬體設計以 5x5 的視窗來進行實作，把左/右影像的 Census 特徵輸入，再將兩個不同的訊息進行 XOR 運算，根據 XOR 運算的特性，若兩輸入值為不同的位元時，得到的輸出值為 1；反之，則會輸出 0，就能夠找出兩資訊中不同的位元，最後再將每一個運算結果加起來即可。

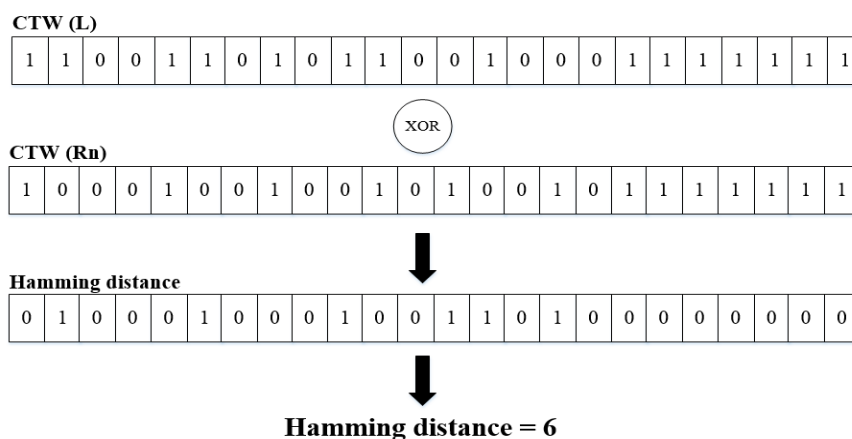


圖 9 Hamming distance 運算流程

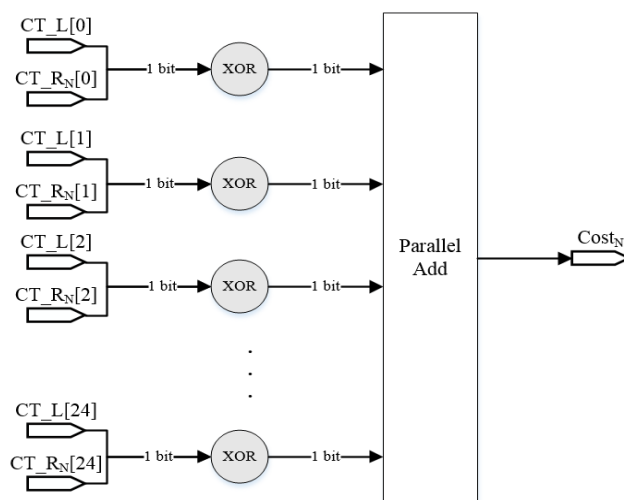
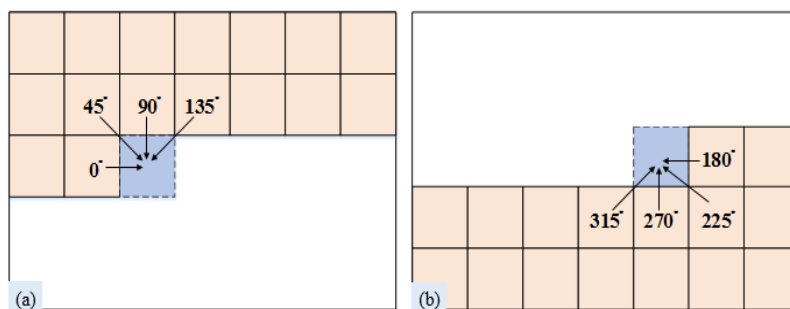


圖 10 漢明距離估算硬體架構

(三) 比對代價聚合

然而本論文因考慮 FPGA 設計的複雜度與匹配效果，在比對比價聚合演算法採用 SGM 半全域比對演算法，SGM 是透過代價函數及動態規劃法的觀念，以左/右影像中的水平像素序列作基準，進行水平位置的像素序列進行比對，再透過多個一維方向的動態規劃法影像匹配，模擬二維影像的全域匹配最佳化。

立體視覺匹配演算法中的全域匹配(global matching)是給定一個能量函數，再搭配其優化方法以求出能量函數的最小值，雖然此演算法有著良好的匹配準確度，但相當困難且耗時，則本論文採用 SGM 演算法實作，根據多個一維方向的代價聚合進行影像匹配，以此方式模擬二維影像的全域匹配最佳化，半全域的比對代價聚合的方向，如圖 11 所示，圖 11(a)為中為四個一維方向(0°、45°、90°、135°)動態規劃法影像匹配；圖 11(b)為另四個一維方向(180°、215°、270°、315°)動態規劃法影像匹配掃描方式。透過八方向的聚合，即可模擬二維影像的全域匹配最佳化，得到匹配效果相當佳的深度影像。



(a) 四個一維方向(0°、45°、90°、135°) (b) 四個一維方向(180°、215°、270°、315°)

圖 11 半全域的比對代價聚合(8方向)

然而，八方向的代價聚合，雖其背景平滑效果不錯，但是此設計方式不但複雜，也會增加系統運算負荷，且匹配後的影像邊界會過度模糊，若要進行後續處理會相當的困難，故在半全域比對代價聚合方向的選擇，本論文將考量到系統運算負荷與匹配效果，選擇四方向(0°、45°、90°、135°)聚合，並結合後續的視差優化方法，以求得高精確的深度影像，SGM 硬體架構圖如圖 12 所示。

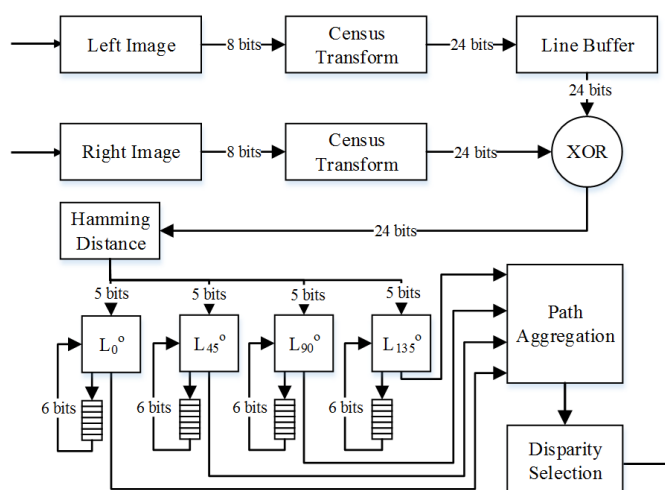


圖 12 Semi-Global Matching 硬體架構圖

本文使用的 SGM 半全域比對代價聚合，每一個路徑(path)的代價聚合公式是參考一維的動態規劃法影像匹配的平滑度限制，如公式(6)與公式(7)所示：

$$C_s(p-r, d) = \min \begin{pmatrix} L_r(p-r, d), \\ L_r(p-r, d-1) + P1, \\ L_r(p-r, d+1) + P1, \\ \min L_r(p-r, i) + P2 \end{pmatrix} \quad (6)$$

$$L_r(p, d) = C(p, d) + C_s(p-r, d) - \min L_r(p-r, i) \quad (7)$$

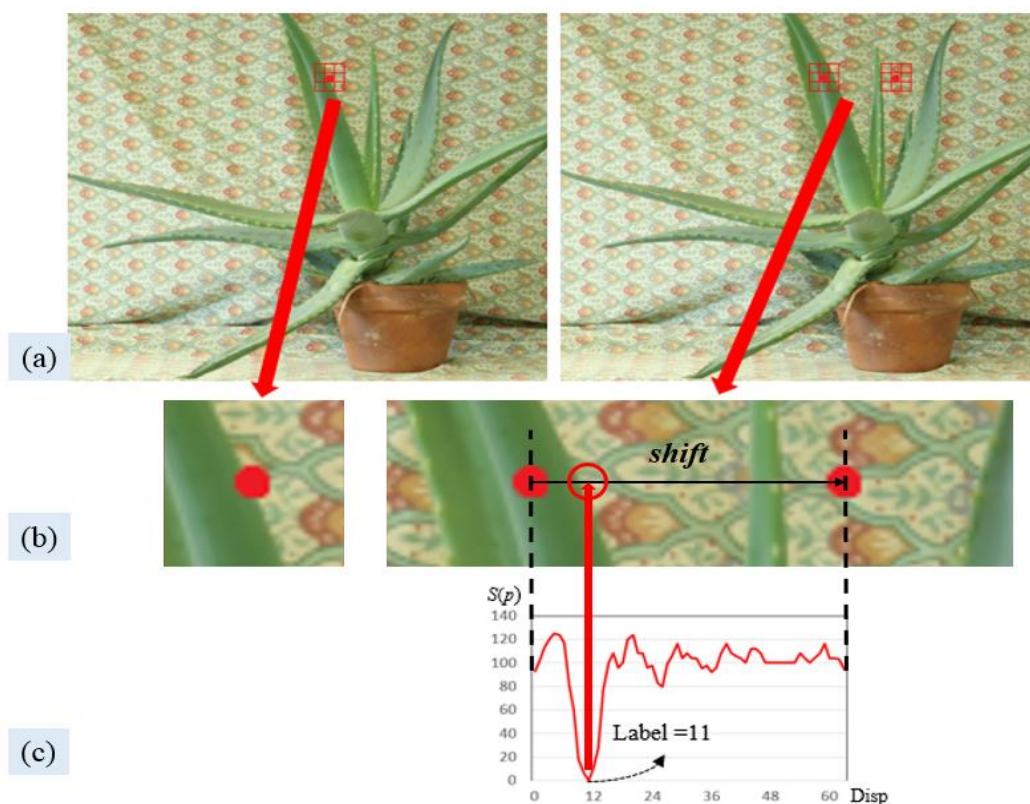
其中 $C(p, d)$ 為影像像素點 p 所計算深度 d 的匹配代價， $C_s(p-r, d)$ 則是目前聚合路徑 r 上的上一像素點 $p-r$ 影像像素點，其中包含懲罰常數的最小匹配代價，然而為了防止 L_r 過大的問題，則透過第三項來進行數值大小的限制。最後再將各個方向的路徑匹配代價聚合進行加總，如公式(8)所示：

$$S(p, d) = \sum L_r(p, d) \quad (8)$$

其中 $S(p, d)$ 為半全域的比對代價聚合，將四方向的路徑(path)代價加總。

(四) 視差估算

如圖 13 所示，在視差估算中，為選取灰階相關性最大的匹配點，將座標 p 的所有比對代價聚合數值做最小值比較，並將其所對應的標記點作為代價比對的結果。



(a)立體匹配測試圖 (b)視差估算範圍示意圖 (c)當聚合值越低，對應的 Label 將填進深度圖

圖 13 視差估算示意圖

(五) 視差值優化

由於左/右深度影像各別有匹配錯誤的部份，為了將其去除，本論文採用 left-right consistency (LRC) check 方法，如公式(9)及公式(10)所示，通過 D_R 深度值與 D_L 以 D_R 深度值所對應坐標之深度值相減後，計算的結果將進行閥值判斷，若其相減結果小於閥值，則將 D_R 的深度值做為新深度圖的深度資訊；反之則為 0。其中 D_R 為以圖 13(a)右圖做為參考所得深度影像； D_L 為以圖 13(a)左圖做為參考所得深度影像；LRC 則為將，左/右深度影像交互驗證， T_c 通常設定為 1。

$$LRC = |D_R(x, y) - D_L(x + D_R(x, y), y)| \quad (9)$$

$$D(x, y) = \begin{cases} LRC < T_c, & D_R(x, y) \\ other, & 0 \end{cases} \quad (10)$$

左/右影像通過交互驗證後，能夠有效的將錯誤部份去除，然而會去除的部份則會使深度影像變成空洞，其像素值會變成 0，這些空洞值會造成影像失真，本論文採用的方式是透過找尋附近最接近的影像進行填補。系統會以此空洞的座標做為起點，向右搜尋最近的深度值，搜尋附近深度值為非 0 值，則以此數值填補。最後再以 5x5 中間值濾波器將雜訊濾除及影像平滑後，則將此深度影像輸出。

參、結果

本論文以 Arrow SoCKit Evaluation Board 作為實驗平台，此平台採用 Altera Cyclone V 系列晶片作為主要核心，其中 CPU 規格為 Dual-Core ARM Cortex™-A9 MPCore™ processor，處理器時脈為 800MHz。

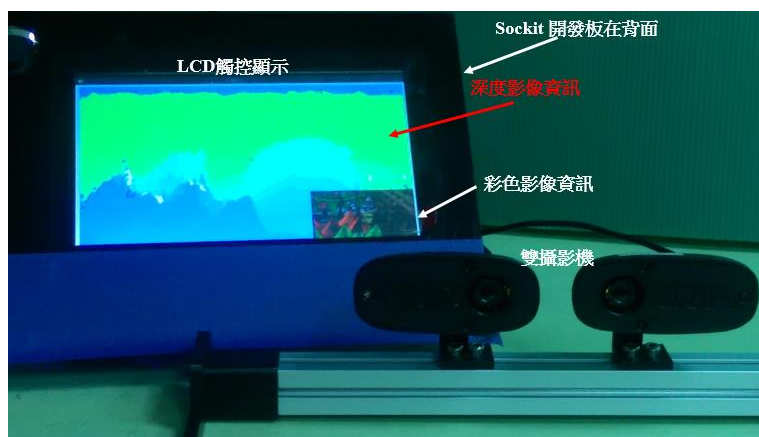


圖 14 系統架設照片

三組實驗透過電路設計完成後，則是使用 Qsys 與 Quartus II ver.15.0 web edition 進行編譯及硬體規劃，編譯及配置完成所產生的資源報告如表 1 所示，CT+SGM 因為演算法較簡單，硬體電路總共使用 6,219 個暫存器，邏輯閘使用率則是 16,780，CT+SGM+LR 硬體電路總共使用 6,912 個暫存器，邏輯閘使用率則是 17,532，CT+SGM+LR+MF_5 硬體電路總共使用 7,081 個暫存器，邏輯閘使用率則是 18,023。

將編譯好的硬體檔案燒錄至平台後，將透過進行靜態影像進行演算法驗證。本論文採用 Middlebury Stereo 資料庫中 Tsukuba、Baby、Teddy 及 Cones 進行驗證。

將測試圖檔儲存到 Micro SD 裝置且讀取圖檔資料後，進行本論文所述之影像濾波，再將測試影像傳至 FPGA 中。當 FPGA 接收影像後，經由本文設計之硬體進行 SGM 匹配且得到運算結果輸出，如圖 15 所示，分別是圖 15(a) Middlebury Stereo 資料庫圖片，圖 15(b) 則是理想深度圖，圖 15(c) 則是本文產生的結果圖。

表 1 本論文硬體資源報告

Unit	LUTs	Register	Memory
CT+SGM	16,780	6,219	568,128
CT+SGM+L/R	17,532	6,912	570,186
CT+SGM+L/R+MF_5	18,023	7,081	580,362

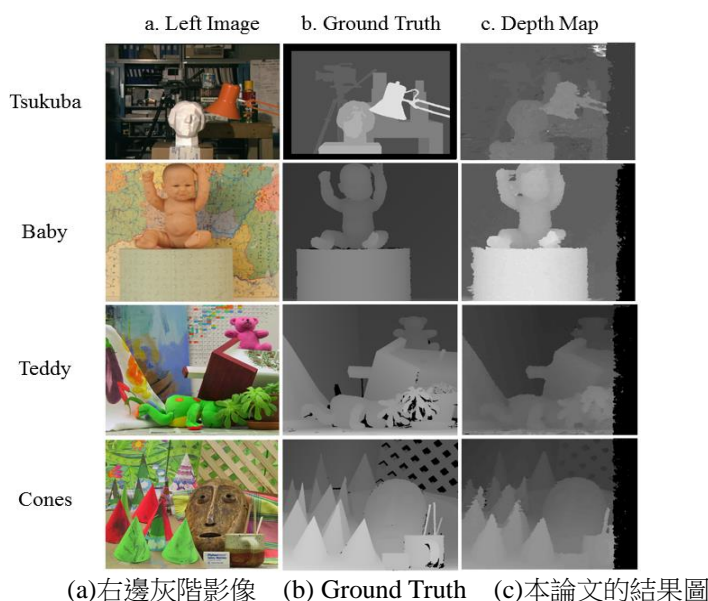


圖 15 Middlebury Stereo 資料庫-測試圖檔

深度影像估算正確率分析上，本論文採用錯誤匹配像素百分率(percentage of bad matching, PBM)，定義如公式(11)所示：

$$\%BadMatching = \frac{1}{N} \sum_{(x,y)} (|d_c(x,y) - d_t(x,y)| > \delta_{thresh}) \quad (11)$$

其中 $d_c(x,y)$ 為匹配視差圖， $d_t(x,y)$ 為真實視差圖。錯誤匹配像素百分率代表視差值誤差大於某一閾值的像素在整張影像中所佔之百分比，閾值 δ_{thresh} 通常設為 1，計算過程中只針對非遮蔽區域的像素點進行統計。

所示，為不同視差優化方法的比對精確率，由表可知使用中間值濾波器有效降低視差影像的高頻雜訊，越大的中間值濾波器，平滑效果越明顯，但中間值濾波器過大會增加系統運算負荷，並且造成邊界模糊，故本論文以 5x5 中間值濾波器來將雜訊濾除，並得到不錯的效果，在不同圖形 Tsukuba，Baby，Teddy 與 Cones 的測試結果的錯誤匹配像素百分率分別是 5.128，1.877%，3.998% 及 3.525%。

即時影像測試則是將 Embedded OS 啟動後，以 V4L2 API 編寫影像擷取應用程式，取得兩部 UVC 攝影機影像串流後，透過 H2F bridges 將左/右攝影機影像，傳輸至 FPGA 的 DDR3 記憶體裡，透過 Semi-global matching IP 中進行運算，最後將結果透過 VGA 輸出顯示。

由於擷取的原始影像未經過變形矯正，計算的深度影像則會呈現大量的匹配錯誤。本論文透過個人電腦進行攝影機參數計算的動作，兩部攝影機拍攝一定數量與不同角度的西洋棋盤版畫面，以 OpenCV library 進行角點偵測，再將每個角點連起來並繪製，用以確保棋盤板準確的出現在攝影機畫面裡。如圖 16 為棋盤板角點偵測圖。

表 2 不同視差優化的比對精準度

Percentage of Bad Matching Pixels	Tsukuba	Baby	Teddy	Cones
CT+SGM+L/R+MF_3	5.847%	2.403%	4.451%	3.984%
<i>Our method</i> + MF_3	5.372%	2.09%	4.246%	3.699%
CT+SGM+L/R+MF_5	5.652%	2.2%	4.164%	3.819%
<i>Our method</i> + MF_5	5.128%	1.877%	3.998%	3.525%

(註)：MF_n 為 n*n 中間值濾波，L/R 為交叉驗證，CT 為 CT 法比對代價。

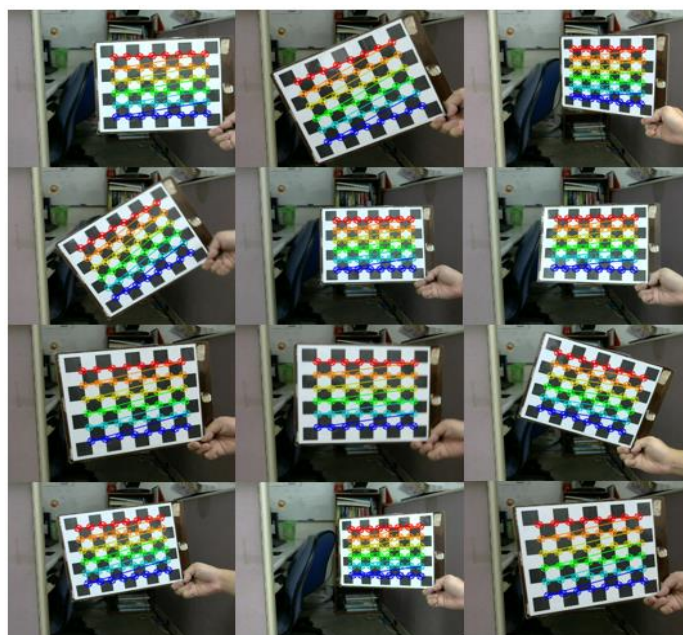


圖 16 攝影機校正影像

拍攝到有效的棋盤影像後，通過 stereoCalibrate 函式將攝影機的內外參數計算出來，並以 stereoRectify 函式求得世界座標的轉換矩陣，再將參數儲存至檔案後傳輸至開發平台儲存空間中。

當系統重新安置攝影機或調整兩攝影機位置時，上述校正步驟需要重新執行一次，當攝影機未正確校正時，產生的畫面會如圖 17(a)所示，是未矯正影像；其中畫面會呈現模糊及歪斜，其程度會跟雙攝影機架設的情形有關，當完成校正程序後，會有較好的影像匹配結果，如圖 17(b)所示。

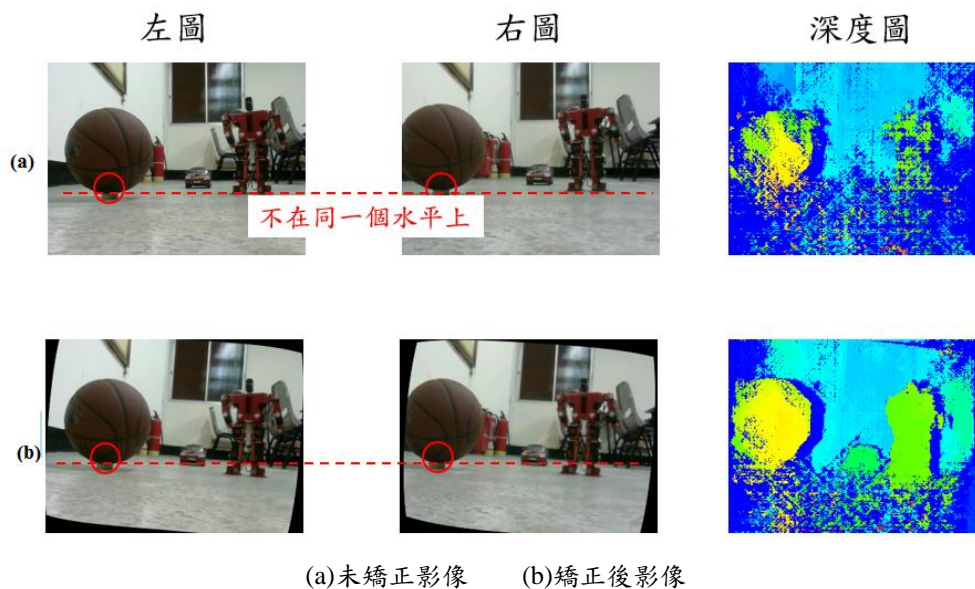


圖 17 進行影像矯正之深度結果比較圖

本論文將 SoCKit 開發平台、立體攝影機以及拍攝物件放置於同一水平面的步道上，並且每一個拍攝物件分別放置於攝影機的 1 公尺、2 公尺、3 公尺等距離，進行實際拍攝且以計算的深度影像進行距離評估，本論文對即時影像進行深度計算後，透過公式(12)將深度影像轉換成距離資訊，其中，計算需要的資訊如： Z 為距離、 B 為雙攝影機鏡頭距離、 f 為攝影機焦距及 d 為視差值。如圖 18 為距離評估之實驗環境架設圖



圖 18 距離評估之實驗環境架設圖

$$Z = \frac{B * f}{d} \quad (12)$$

如圖 19 所示，本實驗以立體攝影機對擺設物進行物件拍攝，將靜態物件的深度資訊及距離資訊呈現於畫面中。此外，本論文將拍攝物件放置 100、150、200、250 及 300 公分等距離，將每一距離所拍攝的影像進行深度計算後，再根據深度資訊帶入公式(12)轉換為距離值，根據表 3 數據得知，當攝影機在進行影像取樣時，容易因光線而影響深度資訊的匹配結果，導致計算的距離結果與真實距離有些微的誤差。

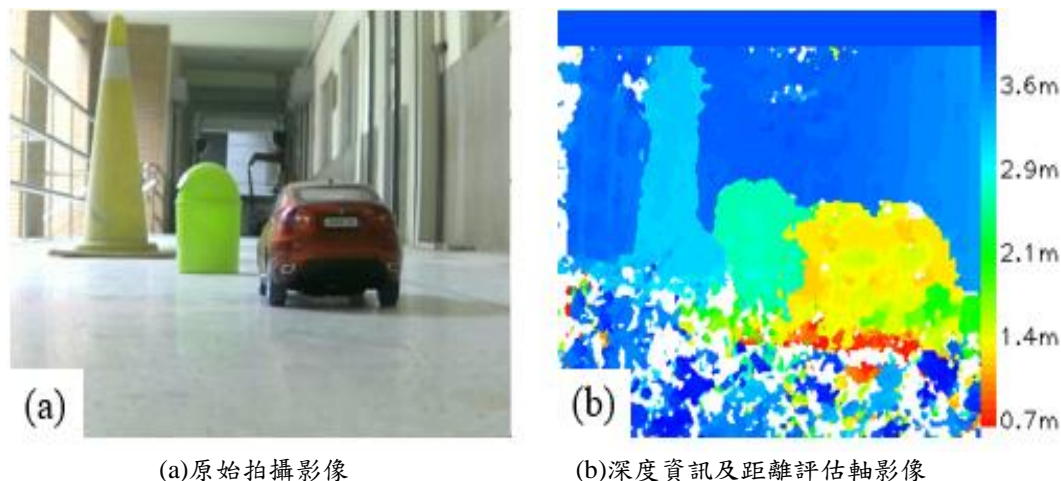


圖 19 即時影像距離測試

表 3 即時影像之深度資訊影像及物件距離評估結果

編號	真實距離(cm)	量測最大	量測最小	平均
1	100	124.4	105	114.7
2	150	160	146.087	153.0435
3	200	224	176.8421	200.4211
4	250	258.4615	210	234.2308
5	300	336	305.454	320.727

肆、結論

本論文以 HPS-FPGA 架構實現立體視覺系統，以半全域匹配 SGM 演算法為基礎，並以 Middlebury 標準測試圖與實際拍攝影像，分析所提出演算方法的比對效果。利用 FPGA 來實現其演算法運算，演算法是由像素灰階編碼的比對代價，並結合 SGM 比對代價聚合計算，最後再以交叉驗證、中間值濾波與空洞填補。未來將以該演算法為主架構，並深度影像優化與系統加速處理等議題，進行其演算法的優化，開發出精確且適用性高的深度影像估算方法。

除此之外，由於深度資訊跟攝影機的聚焦，及雙攝影機架設的位置與距離相關，為了讓使用者可以自行替換攝影機鏡頭及彈性調整其架設位置，本論文提出的平台，可以讓使用者自行更換攝影機、調整兩攝影機的距離，可以讓使用者自行依據需求客製化即時立體影像開發系統的規格，此種設計比市售的系統如微軟的 Kinect、華碩的 Xtion、及 Intel 的 Realsense 更具彈性。

參考文獻

- [1] Q. Yang and N. Ahuja. (2012). Stereo matching using epipolar distance transform, in *IEEE Transactions on Image Processing*, 21(10), 4410-4419.
- [2] D. Scharstein and R. Szeliski. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *International Journal of Computer Vision*, 47(1), 7-42.
- [3] R. A. Hamzah, R. A. Rahim, Z. M. Noh. (2010). *Sum of absolute differences algorithm in stereo correspondence problem for stereo matching in computer vision application*. in *Computer Science and Information Technology (ICCSIT)*, 652-657.
- [4] S. K. Dong, S. L. Sang and H. C. Byeong. (2010). A real-time stereo depth extraction hardware for intelligent home assistant robot, in *IEEE Transactions on Consumer Electronics*, 56(3), 1782-1788.
- [5] J. Zhao, J. Li Chai and G. Z. Men. (2009). *A fast quasi-dense matching method*. in *International Asia Symposium on Intelligent Interaction and Affective Computing*, 100-103.
- [6] X. T. Wang and X. Bo. Wang. (2009). *FPGA based parallel architectures for normalized cross-correlation*, The 1st International Conference on Information Science and Engineering (ICISE2009), 225-229.
- [7] G. J. Li. (2012). *Stereo matching using normalized cross-correlation in LogRGB space*, in 2012 International Conference on Computer Vision in Remote Sensing (CVRS), 19-23.
- [8] H. Hirschmuller and D. Scharstein. (2007). *Evaluation of Cost Functions for Stereo Matching*, in *IEEE Conference of Computer Vision and Pattern Recognition*, 1-8.
- [9] R. Zabih and J. Woodfill. (1994). *Non-parametric local transforms for computing visual correspondence*, in *Proc. ECCV*, 151-158.
- [10] 陳育崧、廖育昇與徐子建(2011)。立體視覺特徵點比對演算法分析與實現。中華民國第十六屆車輛工程學術研討會論文集，B2-007。台北，國立臺北科技大學車輛工程系。
- [11] H. Hirschmuller. (2005). Accurate and efficient stereoprocessing by semi-global matching and mutual information, in *IEEE Conference of Computer Vision and Pattern Recognition*, 2, 807-814.
- [12] H. Hirschmuller. (2008). Stereo processing by semi-global matching and mutual information, in *IEEE Transactions of Pattern Analysis and Machine Intelligence*, 30, 328-341.