

行政院國家科學委員會專題研究計畫 成果報告

支援 3-D 立體視訊的數位電視多媒體平台設計(3/3) 研究成果報告(完整版)

計畫類別：整合型
計畫編號：NSC 98-2220-E-009-012-
執行期間：98年11月01日至99年10月31日
執行單位：國立交通大學資訊工程學系(所)

計畫主持人：蔡淳仁
共同主持人：張添烜、范倫達、彭文孝
計畫參與人員：碩士班研究生-兼任助理人員：洪瑩蓉
碩士班研究生-兼任助理人員：許博雄
碩士班研究生-兼任助理人員：陳奕均
碩士班研究生-兼任助理人員：吳英佑
碩士班研究生-兼任助理人員：廖元歆
碩士班研究生-兼任助理人員：陳宥辰
碩士班研究生-兼任助理人員：張輔仁
碩士班研究生-兼任助理人員：鄭兆傑
碩士班研究生-兼任助理人員：董盈里
碩士班研究生-兼任助理人員：謝佳育
碩士班研究生-兼任助理人員：張家榮
碩士班研究生-兼任助理人員：劉睿峻
碩士班研究生-兼任助理人員：陳俊吉
碩士班研究生-兼任助理人員：楊復堯
碩士班研究生-兼任助理人員：白家芸
碩士班研究生-兼任助理人員：郭瀚文
碩士班研究生-兼任助理人員：黃子剛
碩士班研究生-兼任助理人員：黃雅琦
碩士班研究生-兼任助理人員：呂孟哲
博士班研究生-兼任助理人員：曾宇晟
博士班研究生-兼任助理人員：陳易群
博士班研究生-兼任助理人員：陳珮妤
博士班研究生-兼任助理人員：陳渏；紋

博士班研究生-兼任助理人員：孫域晨

報 告 附 件：出席國際會議研究心得報告及發表論文

處 理 方 式：本計畫可公開查詢

中 華 民 國 100 年 01 月 31 日

目錄

一. 前言	1
二. 研究計畫之背景及目的	2
三. 研究方法與結果	41
四. 計畫產出論文	80
五. 參考文獻	84

摘要

本計畫的目標在設計一個能支援下一代立體視訊及 3-D 人機介面的嵌入式系統多媒體應用程式平台。本平台的重點在於它不是針對任何一個殺手級應用而設計，而是依照一個開放式的 Java 執行環境 DVB-MHP 來進行設計。這個平台因為是架構在一個公開的國際標準之下，所以可以由協力商（third party）發揮創意來擴充其上的應用。這個平台的設計是透過中介軟體來包裝並整合底層的寬頻多媒體應用所需要的元件（如影音解碼、繪圖、以及 3-D 視訊）的加速功能。讓應用程式的開發者可以不用擔心底層的嵌入式系統所使用的處理器、作業系統是什麼，就能發揮出系統最豐富的功能。在這個計畫下，我們會開發出下列的關鍵元件：由 DVB-MHP 所擴充而來的 3-D 人機介面中介軟體、一個專為支援 Java 執行環境而設計的極小型 OS、3-D Video 加速器、3-D Graphics 加速器、Java 處理器、和配合 3-D Video 的 MVC 多視角合成技術。

關鍵詞：中介軟體、Java 處理器、嵌入式作業系統、3-D Video、視差估測加速器、3-D Graphics 加速器、Multti-view Video 編碼

Abstract

The goal of this project is to design an embedded platform to support next generation stereo video and 3-D man-machine user interfaces for multimedia applications. The key design focus of this platform is that it is not designed for any particular killer applications. Instead, it is designed based on an open multimedia application middleware, namely DVB-MHP. Therefore, any third party designer can develop innovative applications for this platform without having to worry about the underlying architecture, such as the type of processor or the type of OS used to create the platform. More importantly, the middleware provides user application accesses to highly efficient feature-rich multimedia components, including audio-video decoding, graphics, and 3-D video. The key technologies that will be developed in this project include: DVB-MHP middleware with extension for stereo video and man-machine user interface, a deeply-embedded minimal OS designed specifically for Java runtime support, 3-D video accelerator, 3-D graphics accelerator, Java processor, and multiview video coding technology.

Keywords: Middleware, Java Processor, Embedded OS, 3-D Video, Disparity Estimation Accelerator, 3-D Graphics Accelerator, Multiview Video Coding

一. 前言：

過去十年來，電信業者和廣播電視業者一直都在嘗試著推動新一代的數位寬頻應用來擴展市場和增加營收。但是，這些努力所收到的成果卻達不到預期的效果。例如 3G 行動網路的市場接受度十分緩慢，更重要的是轉換到 3G 網路的使用者，有很多是因為電信業者的政策促銷，而不是真正在使用寬頻增值服務。這樣的使用者對電信業者增加營收的目標並沒有幫助。在數位電視方面，也遇到同樣的困擾。把傳統的影音服務數位化，其實並不能刺激消費者的投入，更不用說能擴展傳統的電視市場。要能提供多樣化、吸引更多消費者參與的新一代數位服務，更好的影音畫質或是更高的頻寬只是其中一小部份的因素，甚至於稱不上是關鍵技術。一個很好的例子就是在遊戲機市場，低規格的 Wii（雙核心、標準畫質解析度、支援 DVD 和舊數位影音標準）市佔率遠遠超越高規格的 PS3（七核心加上圖像處理器、高畫質解析度、主打 Blu-Ray 和最新數位影音標準）。

由 3G 行動網路和數位電視推廣的遲緩經驗我們可以瞭解，一個新的數位服務網路能否成功，不單單在於頻寬的大小，更重要的是，電信業者和配套的消費電子產品系統業者不能只是像過去一樣，單純的假設幾樣“殺手級應用，”並利用封閉的系統來提供這些應用，就認為消費者會一窩蜂的投入。因為系統廠所想像的殺手級應用（如 3G 手機的影音串流應用），往往和消費者的認知有所落差。就以最近炒的十分熱門的 Apple iPhone 為例，在短短銷售兩個月之後就因為銷售不如預期而大幅降價超過 30%。主要理由之一就是因為 iPhone 是封閉的系統，惟一能擴充其應用的方式是透過內嵌在網頁內的 AJAX（Asynchronous JavaScript and XML）程式，因此功能頗為受限。而使用者對內建的一些固定應用又不覺得很吸引人，所以即使 iPhone 擁有設計花俏的使用者界面和 Apple 的完美行銷手段，還是不能讓大部份消費者心動。

基於以上的討論，本計畫的目標在設計一個能支援下一代多樣化寬頻多媒體應用的嵌入式系統應用平台。本平台除了是開放式，可以由協力商（third party）擴充其上的應用，並提供一般寬頻多媒體應用所需要的基本元件（如影音解碼、繪圖）加速、另外，我們更開發並整合現在開始慢慢流行的 3-D 視訊元件的支援。相關研究成果包含：

視差資料估測模組設計：提出一視差資料估測演算並以硬體實現，可輸出高準確度且即時處理速度的視差圖。

3D 圖學模組設計：其中前端幾何轉換子系統中的打光運算單元以及後端繪圖子系統中的三角形設定引擎與深度壓縮機制提出可重組式的演

算法與對應的硬體架構，可根據不同的輸入規格重組不同的硬體架構，達到系統的最佳效能。

3D 視訊壓縮與合成研究：在於深度圖因視訊壓縮所造成之量化效應 (Quantization Effect) 下，利用參照影像間隱藏之深度資訊，修補失真之深度圖。本子項提出了一個檢測單一像素之合成誤差的模型 (Per-pixel Synthesis Distortion Model)，並發展出合成品質導向之深度圖修補演算法，此演算法相較於現行的 MPEG FTV 標準架構下，平均多 1.2 dB，且在主觀視覺上也較接近於原始影像。另外，本計劃根據深度圖之特性，也提出低運算複雜度之一維深度圖壓縮演算法，在固定合成品質下，深度圖壓縮率較 MVC 少 4~13%，運算複雜度也較 MVC 之 Intra 預測方式降低了 5~15%。

異質雙核心 Java 處理器：我們設計並實作出一個新的異質雙核心的 Java 處理器架構，主要由一個 RISC core 和一個 Java core 所組成。有別於去國內外學界及業界所設計的 Java 處理器，我們設計的重點如下：(1) 完整支援 Java 的 object-oriented 的語言特性、(2) 對於 OS 依存度很低、(2) 易與現有硬體電路整合、(3) 不使用昂貴的電路元件，如特殊的 Java Object Cache 或者是 Java Stack Cache，以利於消費性電子產品的應用、(4) 在同時脈下，執行效能要能跟 CVM-JIT 的技術相當或更高。而我們最後設計完成的異質雙核心 Java 處理器，也實作在 Xilinx ML-507 平台上進行驗證，並符合上列各項要求。

除了上述關鍵模組的設計突破之外，本計劃已整合所有關鍵模組於同一個 Xilinx ML507 平台上，可以進行即時展示。此平台設計可供產學界未來相關研究參考。

二. 研究計畫之背景及目的：

3-D 視訊技術的發展，在過去十多年一直沒有斷過。從 90 年初期北大西洋公約組織主導的 Race - DISTIMA (Research on Advanced Communications in Europe, contract R2045, "DIgital STereoscopic IMaging and Applications) 計畫[1][2]、後來延續的 PANORAMA (Package for New Autostereoscopic Multiview Systems and Applications, ACTS project 092) 計畫、一直到最近的 ISO/IEC MPEG 主導的 MVC (Multiview Video Coding) 計畫[3]。這些計畫發展了許多立體視訊的內容製作、傳輸、編碼、及展示 (render) 技術。比方說，在 RACE-DIATIMA 計畫結案時，就展示了完

整的 3-D 即時立體視訊製作和透過 ATM 網路串流播送的系統[4]。在過去，這些技術不能實用化的主要理由是 3-D 顯示技術不夠成熟。不過最近幾年，業界在這方面有長足的進步：

3D 影像顯示設備過去幾十年一直被廣泛研究，像是全像素顯示、立體投影...等先進技術一直被科學研究...等領域採用。近年來國內外電子大廠紛紛開始研究如何將 3D 顯示技術帶入一般消費性市場，從 2003 年 Sharp 公司發表了全球第一個代表性產品，一款結合 3D 顯示設備的手機 [5]，利用 LCD 面板技術，在不同視角顯是不同的影像，將立體影像顯示效果成功帶入消費者日常的生活。Philip 公司也隨後發表 3D 液晶螢幕 [6]，並投入大量研發能量進行 3D 顯示器相關研究，至 2009 年底，3D 技術幾乎成為電視產品的避備功能。

目前立體顯示設備的關鍵在於物理技術要如何能提供使用者舒適的視覺效果，有些系統使用上會造成消費者暈眩的感覺，便不容易普及。然而近年來有不少新的顯示技術突破，可以讓使用者更舒適的取得立體影像。例如一種方法是利用偏光的原理顯示出兩個正交的影像，使用者戴上特殊的偏光鏡，便可以看到立體的結果，例如：國內廠商奇美所研發的面板[7]便是採用此類技術。另外一類的技術是利用使用者不同的視點位置，在螢幕的不同方向顯是不同的影像，Philip 公司推出 9 視點的 3D 顯示螢幕[6]，便是採用此種方式。至 2009 年底，已有很多全球電子大廠，如 Philip,Sharp,LG,Sony...等國際大廠都紛紛投入大量資源發展相關研究，而國內面板廠奇美和友達也有相關的研究計畫或產品。

目前市場的 3D 立體顯示應用大多為電影相關產業，其他的應用並不多，主因是沒有公開的系統平台可以供上下游廠商開發和整合應用。本計畫的目標並不是單單只是為了支援 3-D 立體視訊而設計一個開放式的系統平台。畢竟 3-D 立體視訊雖然很有前瞻性，但它只是多媒體寬頻服務的一項可能的應用。因此我們所設計的平台，必須是以目前廣泛被採用的多媒體嵌入式系統應用平台為基礎，並為之擴充 3-D 立體視訊的功能。因此我們鎖定的目標是希望這個平台可以方便地讓應用程式開發商自由地發展各種有趣的多媒體應用，而這些開發商在這個開放式平台上所能使用的元件，包含了影音、2-D 繪圖、以及 3-D 立體視訊。在嵌入式多媒體系統的領域，像這樣的公開平台，大部份都是以 Java 執行環境為基礎的。比方說手機所支援的 CLDC/MIDP 2.0 平台或者是數位電視所支援的 CDC/PBP/GEM 平台。Java 執行環境會成為一個嵌入式多媒體的公開應用程式平台，最主要的理由是因為 Java 本身具有跨平台的特性，不受特定處理器和作業系統的限制。這點正好符合了嵌入式系統處理器和作業系統百家爭鳴的情形。

手機因為螢幕小、記憶體少、處理器也慢，所以它的公開應用程式平台就是 MIDP 2.0，所有的影音繪圖多媒體功能都受限於 MIDP 2.0 所提供的應用程式界面。所以不算豐富。而數位電視因為上述的限制較鬆，所以他的應程式界面也可以比較豐富。目前互動式數位電視系統的整合和應用界面是由 Sun 所制定的 Java Personal Basis Profile (PBP) 和由各國數位電視聯盟分別所制定的中介軟體 (middleware) 所規範 [10]。雖然歐洲、美國、和日本各有各的中介軟體標準，但他們有一個共同的中介軟體子集合 GEM (Globally Executable Multimedia home platform)。一個應用程式如果只用到 PBP 和 GEM 所提供的影音繪圖多媒體功能，那麼就能在世界各地的數位電視機上盒執行。

圖 1 是一個 MHP 應用程式的執行範例。使用者在收看數位電視的同時，可以透過電視遙控器進行訂購餐飲外賣的服務。各家和電視公司簽約的外賣業者，會隨著廣播訊號把他們的 MHP 應用程式傳送到使用者家中的機上盒（這些程式的傳輸和起動也可以和他們的電視廣告同步）。而使用者只要透過遙控器就可以進行即時訂購，未來在台灣可能會利用數位電視內建的 GPRS 晶片把訂購的資訊上傳到電視公司，再由電視公司通知業者送貨。這樣的應用無疑可以為傳統電視增加新的商機。而這只不過是 MHP 平台的一個小小的應用範例。在圖 1 中，如果業者的圖片、影片、或動畫能以 3-D 的方式突顯於螢幕之外，想必更能吸引消費者的目光，不過目前的 MHP 平台並不支援 3-D 視訊。



圖1. 在台灣試播的 MHP 應用程式

目前歐規 DVB [11]的 MHP (Multimedia Home Platform) [12]算是比較被廣為採用的公開多媒體應用平台。採用 MHP 標準的地區以歐洲、北非、亞太地區等國家為主。另外，以 MHP 為基礎的 GEM 標準則被美國、

日本、南韓等國家採用。至於澳洲、台灣、大陸等國家則正在蓬勃發展 MHP 相關產業。因此 MHP 很可能會成為未來多媒體寬頻網路的標準應用平台。當初 MHP 是為螢幕解析度在 640x480 以上、具有寬頻網路 (IP 或 MPEG-2 TS) 連線能力的平台而設計的，但事實上，由於手機的螢幕解析度越來越高，中央處理器 (或 Java 加速器) 的功能越來越強，CLDC/MIDP 2.0 的規範早已不符合下一代手機的應用服務需求，所以未來手機也有可能走上支援 GEM 的路。本計畫的目標則是開發一個支援 3-D 視訊的高效能 MHP 平台，以供寬頻服務應用程式開發廠商在一個公開的平台標準下發揮想像力，設計各種吸引人的應用服務。當然，目前的互動式數位電視標準 (MHP 或 GEM) 並不支援 3-D 立體視訊，但是隨著製造立體面板的廠商增加 (目前國內友達及奇美都有生產樣本)，未來如果能有支援 3-D 立體視訊的 MHP 平台，更有可能開發出殺手級的應用，以提高下世代寬頻網路的普及程度，創造新的產業價值。

基於以上討論，本計畫目標是發展出支援 3-D 立體視訊的 MHP 平台，平台內含有 4 個關鍵模組 (分別以 4 個子項目討論)，包含 3-D 視訊計算加速核心之設計、可重組之 3D 繪圖加速器設計、MVC 編碼系統中介軟體和嵌入式平台的 Java 處理器設計，以下我們列出各個子項目的研究背景及目的。

子項目一：3-D 視訊計算加速核心之設計

(Design of 3D Video accelerator)

子計畫一目的於發展適合硬體實現的視差資料估測演算法與其硬體實作。視差資料估測模組可讀取已校正的兩視角影像，大小為 CIF(325x288)，輸出高正確度的視差資料，視差資料估測硬體處理速度達 5 fps 以上。配合總計劃的多媒體開放式平台，視差資料估測運算元將依據 MHP 給予的指令而運作，並且將運算結果交付給 3-D 繪圖合成運算元，進一步合成出 3-D 影像，最後經由 FPGA 平台與晶片設計驗證本計畫成果。

本子計畫的相關研究現況以下將分為三部分評述:視差資料估測演算法、影像切割化演算法、視差資料估測的硬體設計。

視差資料估測演算法(disparity map estimation algorithm)

Scharstein 與 Szeliski [63][14]建立視差資料品質的評估平台，提供統一的評估標準。評估視差資料的方式是取雷射掃描測距所得的正確視差資料(ground truth，下圖右)與各演算法計算出的視差資料比較，視差錯誤的像素點個數佔整張影像像素點個數的比例為視差錯誤比例(disparity error rate)，視差錯誤比例越小表示視差資料估測演算法的效果越佳。

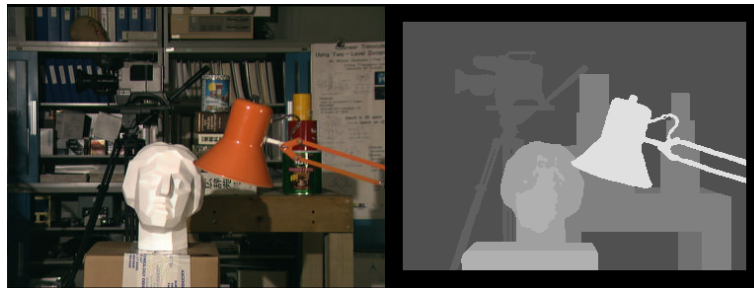


圖2. 原始影像與 ground truth

視差資料估測演算法在電腦視覺領域(computer vision)已經研究達二十年以上，各類型的演算法在[63][61]有一般化的分類。視差資料估測演算法可區分成區域方法(local approach)、半全域方法(semi-global approach)以及全域方法(global approach)三類，演算法流程如下圖。

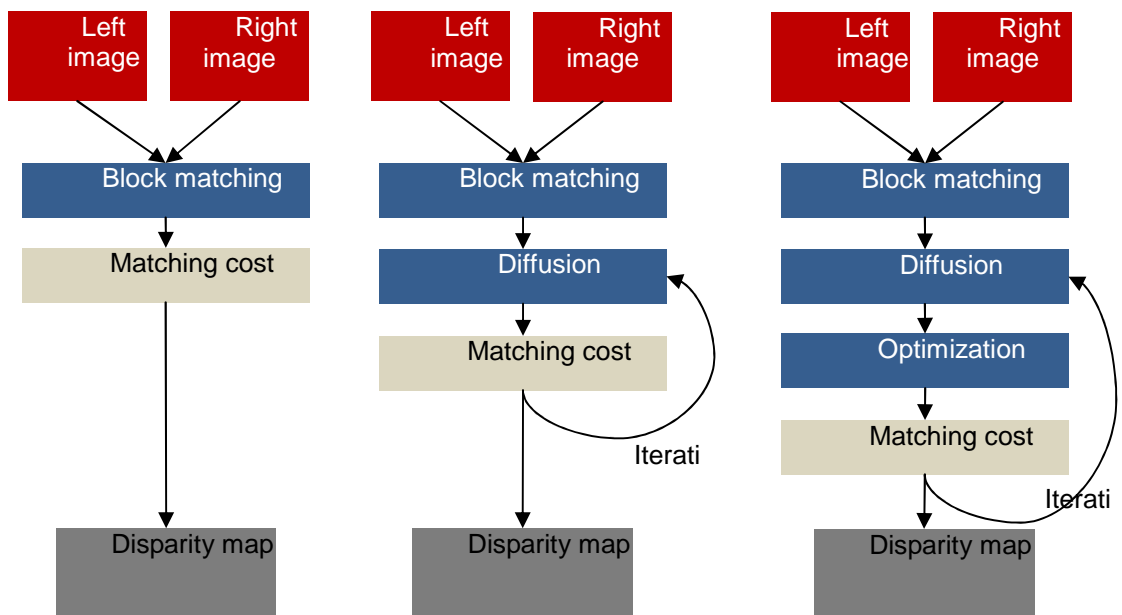


圖3. Local approach / semi-global approach / global approach 演算法流程

Local approach 主要以 block-based 的方式計算像素點在不同視差的 matching cost，其 block matching 可以是 SAD(sum of absolute differences)、ZSAD(zero-mean SAD)、CENSUS、RANK 等各種不同的方式。Semi-global approach 主要加入了擴散(diffusion)matching cost 的機制，以反覆(iteration)

的方式將每一個像素點所擁有的 matching cost 向其他像素點擴散 [64][65][18]。Global approach 對整張影像設定一個目標函數(objective function)，畫面中選擇不同的視差資料結果其目標函數將會有不同的值。Global approach 選用不同的最佳化(optimization)演算法，快速又有效的選擇到適當的視差資料，使得目標函數的值能夠最小。而常見的最佳化演算法有 dynamic programming[19]、graph cut[20]、belief propagation[67]等。

下圖為視差資料的品質比較，數值為視差資料的錯誤比例，數值越小表示視差資料品質越佳。由圖表中可得知 global approach(GC、Bay)比 local approach(SSD)的視差資料品質正確度高 0.2%~3%。而以主觀的比較，local approach 的 SSD+MF 視差資料品質明顯劣於其他四者 global approach。因此對於視差資料估測的正確度而言，global approach 優於 semi-global approach 優於 local approach。

	Tsukuba		Sawtooth		Venus		Map	
	fixed	best	fixed	best	fixed	best	fixed	best
1 SSD	5.23	5.23	2.21	1.55	3.74	2.92	0.66	0.22
2 DP	4.12	3.82	4.84	3.70	10.10	9.13	3.33	1.21
3 SO	5.08	4.66	4.06	3.47	9.44	8.31	1.84	1.04
4 GC	1.94	1.94	1.30	0.98	1.79	1.48	0.31	0.09
5 Bay	6.49	6.49	1.45	1.45	4.00	4.00	0.20	0.20

圖4. 視差資料品質比較[63]

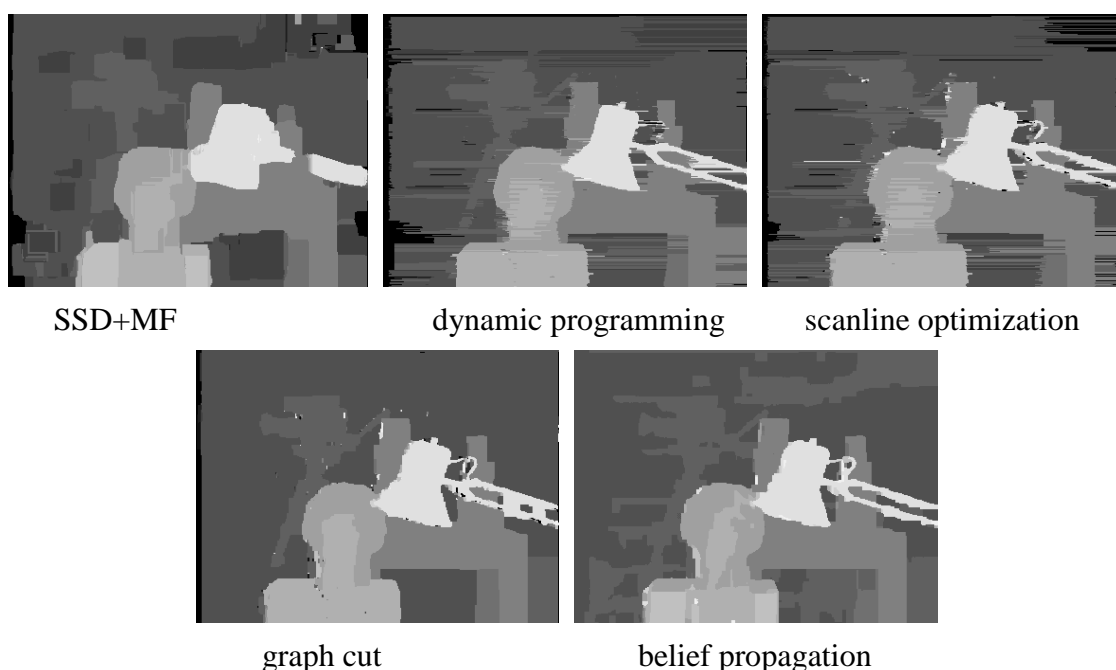


圖5. 各種演算法的視差資料結果

從 local approach 到 global approach 的演算法在運算複雜度是隨著範圍增大而增加，而視差資料的正確度則是隨範圍增大而提高。下圖為不同演算法在四種測試影像以軟體程式的執行時間，前三者屬於 local approach 的演算法，中間兩者為一維最佳化的 global approach，最後四項為二維最佳化的 global approach，其運算複雜度約是 local approach 的 20~1000 倍。因此，在本計畫選擇適當的高正確度演算法為基礎，進而轉變為適合發揮硬體運算特性的演算法，以達到高正確度且高運算效能的硬體架構設計。

	Tsukuba	Sawtooth	Venus	Map
Width	384	434	434	284
Height	288	380	383	216
Disparity levels	16	20	20	30
Time (seconds):				
14-Realtime	0.1	0.2	0.2	0.1
16-Efficient	0.2	0.3	0.3	0.2
*1-SSD + MF	1.1	1.5	1.7	0.8
*2-DP	1.0	1.8	1.9	0.8
*3-SO	1.1	2.2	2.3	1.3
10-GC	23.6	48.3	51.3	22.3
11-GC + occlusions	69.8	154.4	239.9	64.0
*4-GC	662.0	735.0	829.0	480.0
*5-Bay	1055.0	2049.0	2047.0	1236.0

圖6. 運算複雜度比較[63]

影像切割化演算法(image segmentation algorithm)

對於影像切割化演算法可大略可區分成三類[68]:image domain-based techniques、feature space-based techniques 以及 physics-based techniques。Image domain-based techniques 是以小範圍的方式，觀察空間上相鄰的像素點間特性，當像素點的特性相近時則屬同一區塊，watershed algorithm[69]即屬於此類。Feature space-based techniques 是以整張影像為範圍處理，定義各種特徵空間(feature space)，接著將所有像素點轉換到特徵空間中，以群聚(cluster)的演算法將特徵空間中的點區分成多個群，每一群則表示一個區塊，mean-shift[24]即屬於此類。Physics-based techniques 是將物體受光反射產生顏色的物理特性描述成數學模型，根據此數學模型推算出區塊[70]。

目前最常被用於視差資料估測的影像區塊演算法為 mean-shift[59][60]。由於 mean-shift 在影像切割化的效果頗佳，使得達到

高正確度的視差資料。但是 mean-shift 屬於 feature space-based techniques，在運算的過程中必須存取大量的數據在記憶體中，大量存取空間的需求問題將會成為硬體實作上的困難。而採取屬於 image domain-based techniques 的演算法其優點在於可降低存取空間的需求，而缺點則是在影像切割化的主觀評估下，其效果會劣於 mean-shift。

視差資料估測的硬體設計與發展

以硬體設計實現 stereo matching 的研究伴隨演算法發展，從 local approach 到 global approach 已有不少研究成果被發表。硬體設計所採取的方式包含 FPGA 平台和 ASIC 硬體設計，以下將概述各種演算法的硬體設計實例。

Local approach 演算法的硬體設計在 1993 年的[71]設計可處理 256x256 大小影像，處理速度 3.6fps 的硬體設計。自此之後陸續有其他的硬體設計被發表(如圖 7)，其硬體架構皆與圖 8 類似。基本架構主要包含影像擷取的介面(video interface)、影像校正、相似度(correlation 或 block matching)計算、視差資料決定四個部分，其中的 block matching 為運算複雜度最高，且需要龐大的吞吐資料量。

Real-Time System	Image Size	Frame Rate	Disp. Limit	Algorithm
INRIA	256×256	3.6		Normailized Correlation
CMU Stereo Machine	256×240	30	16	SSAD
PARTS	320×240	42	24	Census
SAZAN	320×240	20		SSAD
ACADIA	512×512		32	SAD
SRI SVS	320×240	30		SAD
FPGA	640×480	20	20	Phase based
DeepSea	512×512	30	52	Census

圖7. Local approach 硬體設計[29]

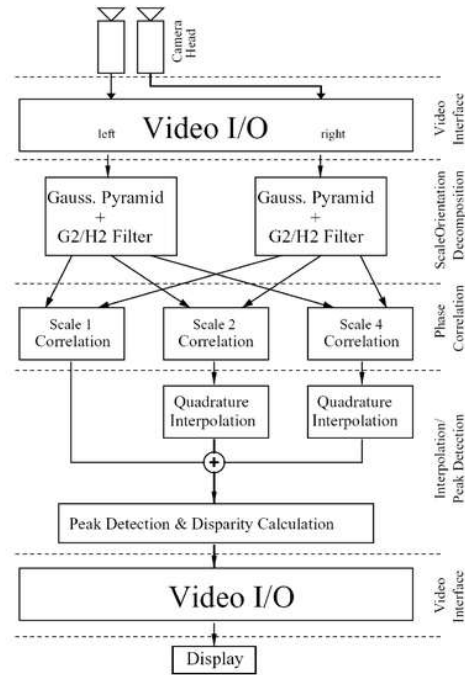


圖8. Local approach 硬體設計架構[73]

在半導體製程的進步下，單晶片中可容納的邏輯閘數量(gate count)大增，使得更多 PE 可加入硬體設計中，增加平行運算的效能，加速視差資料估測的處理速度。在下圖表示近來的 local approach 硬體設計朝向高解析度(high resolution)的目標推進。在 2007 年的[74]硬體設計中，已可處理大小 1280x960 的影像，且處理速達高達 52fps，運算能力為 1993 年[71]的 270 (18.75x14.4)倍之高。然而，如此大量 PE 的運算伴隨需要的是龐大的暫存記憶體(memory storage)，以及極高的資料頻寬(data bandwidth)。

在先前這些設計中，多半採用 FPGA 作為主要的實現方式。FPGA 實現方式有其優缺點，如大量的平行度與 on-chip RAM，但價格十分昂貴，不利廣泛使用與更高計算量的應用。先前之研究由於 FPGA 含有大量的可配置記憶體空間，故硬體設計將大部分所需之記憶體配置於 on-chip RAM，而可成功的實作於 FPGA 中。但如此之設計未考慮一旦系統整合時所面臨資料互相溝通之問題，而且先前之設計並未考慮龐大的資料頻寬的問題，多為直接實現。此外，若應用於單晶片上的硬體設計，其記憶體空間的大小會左右晶片的面積，造成可實現於 FPGA 的硬體設計架構在單晶片的硬體設計會面臨無法實現的問題。因此，周全考慮記憶體空間與資料頻寬的問題，並設計符合本計畫目標之需求演算法及硬體設計是本計畫的研究重點。

Real-time system	Image resolution	fps	Disparity range	PDS $\times 10^6$ / [PDS $\times 10^6/f_{clk}$]	Method	Processor type
Diaz et al	1280x960	52	9 15 29	585 / 9 975 / 15 1885 / 29	Phase based	Custom FPGA, Xilinx Virtex-II (65 MHz)
Gong and Yang	512x384	14.7	40	117 / --	Correlation-based with image-gradient-guided cost aggregation	Pentium 4 3GHz equipped with an ATI 9800 XT (412 MHz)
Forstmann et al.	256x256 640x480 1024x1024	30.4 7.23 2.2	100	200 / -- 222 / -- 230 / --	Dynamic programming	AMD AthlonXP 2800+ and MMX optimization
Niitsuma and Maruyama	640x480	30	27	248.8 / 3,66	Correlation. SAD	Custom FPGA, Xilinx Virtex-II (68 MHz)
Darabiha et al.	360x256	30	20	55.3 / 1,1	Correlation phase-based	Custom FPGA, Xilinx Virtex, (50MHz)
Woodfill, and Herzen	320x240	42	24	77.4 / 2,35	Census matching	Custom FPGA Xilinx XC4000 (33MHz)
T. Kanade et. al.	256x240	24,4	20	30 / --	Multi-baseline Correlation. SSAD	Custom HW & C40 DSP (2-6 cameras)

圖9. Local approach 硬體設計[74]

由於 semi-global approach 演算法並無一般性的明確界定，僅將演算法區分為 local approach 及 global approach。在本計畫中所指的 semi-global approach 為 local approach 且加入反覆的擴散機制的演算法，其亦可視為由 global approach 的簡化而成。因此在此直接進入 global approach 硬體設計的概述。

Global approach 中最常見的為 1-D 最佳化的 dynamic programming(DP) 與 scanline optimization，以及 2-D 最佳化的 graph cut 與 belief propagation(BP)。各種 global approach 演算法是選用不同的圖形模型(graph model)，並在其 model 中針對 objective function 做最佳化的運算。下圖表示不同演算法採取的 graph model，其中 DP 為一維的線型圖形，BP 為二為的格狀圖形，treeDP 是介於一維與二維之間的樹狀圖形，而 tree reweight (TRW)則是比 BP 更為複雜的二維圖形。一般而言，圖形越複雜或越不規則，運算量則越高。

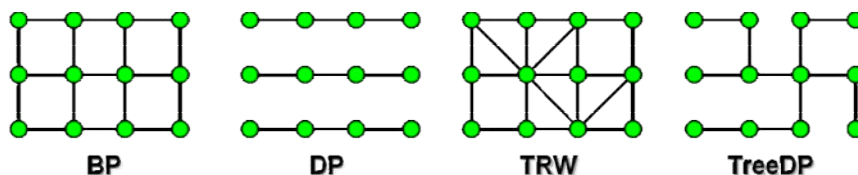


圖10. Graph model

下圖表示 dynamic programming 在 FPGA 平台上的處理效能，其處理解析度可達 512x512，處理速度 36.33fps。雖然 DP 可以達到高效能的運

算，但 DP 所產生的視差資料會出現垂直方向不連續的現象，其原因在於 1-D 的最佳化並未考慮垂直方向的擴散，scanline optimization 亦有相同之情況。

Real-Time System	Image Size	Frame Rate	Disp. Limit	Algorithm
INRIA	256×256	3.6		Normalized Correlation
CMU Stereo Machine	256×240	30	16	SSAD
PARTS	320×240	42	24	Census
SAZAN	320×240	20		SSAD
ACADIA	512×512		32	SAD
SRI SVS	320×240	30		SAD
FPGA	640×480	20	20	Phase based
DeepSea	512×512	30	52	Census

圖11. Dynamic programming 演算法的軟體及硬體設計[75]

針對視差資料在垂直方向不連續之問題，[76]-[78]提出不同的 graph model，其 graph model 採用介於 1-D 及 2-D 間的樹狀形式圖 10:TreeDP) 順利的改善 dynamic programming 的問題。然而，不規則的樹狀圖形模型卻造成運算的不規則性，成為硬體設計的困擾。

2-D 最佳化的 graph cut 由於運算複雜度過於龐大[79]，硬體設計雖可加速處理，但其運算量仍就過大。而 BP 在[80]研究中，採用[81]提出的 hierarchical BP 的反覆次數得以減少，並且簡化其運算複雜度，故可設計出處理 320x240 大小，處理速度 30fps 的硬體設計。然而，此架構採用兩個 FPGA，SRAM 的使用量高達 1.8M byte，其資料頻寬需求極高，並不實用，且在現今的單晶片 ASIC 硬體設計是無法實現的設計。

	Spec.(Resource usage percentage)
FPGA	Xilinx Virtex II pro-100
Clock Speed	25MHz(Max.=79MHz)
Number of FPGA	2
Number of Block RAM(18kbit)	800(90%)
Number of Multiplier	0
Number of Divider	0
Number of Slice Flip Flops	55,712(31%)
Number of 4 input LUTs	94,014(53%)

圖12. BP 在 FPGA 平台的硬體設計[80]

在考慮單晶片的 ASIC 硬體設計，以上所提及的相關研究皆存在外部記憶體存取的問題。演算法大量的記憶體使用量，使得大部分的暫存資料

必須存取於外部記憶體，而單晶片整合在完整的系統中必須與其他單晶片共用外部記憶體(DRAM)，共用的匯流(bus)將會有可使用時間長的限制，以及匯流的頻寬限制。同時考慮外部記憶體、內部記憶體、內外資料頻寬等問題，設計與發展可用於系統整合的單晶片硬體設計即為本計畫研究最重要的核心。

子項目二：可重組之 3D 繪圖加速器設計 (Reconfigurable 3D Graphics Accelerator)

近年來隨著 VLSI/SOC 技術進步與遷入式系統應用的蓬勃發展，以往只在傳統桌上型電腦出現的 3D 繪圖需求在手持式裝置與家電等嵌入式環境中的應用也逐漸流行。3D 圖學技術負責繪製總計劃中 3D 影像與 3D 視訊所需的不同視角的影像資料。與傳統 3D 視訊的不同點在於由於不同視角影像能使用 3D 繪圖技術即時繪製出來，不需要事先拍攝多重視角的影像或視訊資料，能帶來的好處就是大量的減少頻寬的需求，另一個優點是全虛擬的 3D 模型能夠提供全視角影像而不限於只能由有限的視角觀看，這給予互動式或虛擬實境的應用很大的自由度。在這個過程之中 3D 加速器扮演了相當重要的角色，主要是因為 3D 圖形成像過程中包含了複雜且大量的運算，如座標投影轉換的矩陣乘法，打光的內積、方根、倒數、次方甚至是三角形設定與計算貼圖座標所需之除法，這些運算雖然可以軟體的方式完成但是在高解析度複雜場景的即時顯示的需求下，面對龐大的 3D 運算量，其獨立的嵌入式 CPU 的架構與記憶體的頻寬並不能有效率地且即時地完成，若使用高運算能力嵌入式處理器來達到所需的效能，將導致昂貴的硬體花費與嵌入式處理器使用率不高所造成的浪費。因此在兼顧效能與成本的考量下，需將這些運算使用特定的硬體加速器完成且要大幅減少其頻寬的需求，利用其平行與管線的硬體架構來提高 3D 繪圖的效能，使處理器僅需負責一般程式的執行、介面與硬體控制上，降低處理器需求的規格之餘不但硬體成本得以有效減少，效能也能提升，因此 3D 的相關應用在嵌入式系統比傳統桌上型電腦更需要 3D 繪圖加速器的輔助。

又因 3-D 立體視訊的應用面非常廣泛且需求面與功能面非常多元，其硬體設計必須有能力根據不同解析度與不同 3D 圖形品質要求下提供具有不同壓縮能力(也就是提供不同頻寬的能力)、不同打光能力與滿足即時的需求，所以此設計必須具備可重組性(Reconfigurability)，故子項目二之重點為可重組之 3D 繪圖加速器設計，本子項目二負責進行包括 3D 繪圖軟體平台開發、程式庫與介面開發、具重組暨低頻寬需求之 3D 繪圖加速器 Soft IP/單晶片的設計與其嵌入式平台整合。在軟體部分，子項目二會提供 Mesh Reconstruction 與 Rendering 軟體模型模擬平台，使得能與子項目一之深度做軟體整合與功能上的驗證，在介面上我們除了提供 Rendering API 之外也會提供 Mesh Reconstruction 的 API 供子項目一使用。此外我們將

以 IP 的形式包裝我們的加速器讓總計劃能夠在嵌入式平台下整合，在這個階段，子項目二預計提供軟硬介面的繪圖 API 設計: OpengLES 與 Java3D，以方便程式移植與開發。在硬體部分，除分別設計可重組幾何轉換子系統 (Geometry Transformation Subsystem) 包含 Geometry Transformation，Triangle Subdivision，Lighting 與具重組性著色子系統 (Raster Subsystem) 包含 Triangle Setup，Visibility Comparison，Z buffer compressor，Shading 等；結合上述核心重組性子系統的相關研究，將各項子系統可重組化設計與減低其頻寬的需求的特色，進一步的整合成為智慧型 3D 繪圖系統加速器。

子項目二目的有二: 一為提供同時提供可重組式與低頻寬需求硬體與軟體環境，讓開發者在合理硬體成本下能針對所繪製的場景與模型特型選擇適當的硬體組態來達成繪圖品質來滿足效能的要求；二者為此可重組 3D 繪圖加速器可讓主計畫、子項目一、子項目三、子項目四能獲得足夠的系統頻寬與強大的 3D 運算支援且提供完整的軟硬體介面，其子項目二提供的軟硬介面的繪圖 API 設計使各子項目能有效且方便的使用整合與驗證，進而達到總計劃所訂定的目標。

本子項根據總計劃”支援 3D 立體視訊的數位電視多媒體平台設計”所需要的可重組 3D 圖形加速核心為設計目標，以下國內外研究設計情況將針對 3D 繪圖處理器架構、幾何轉換子系統(Geometry Transformation Subsystem)與著色子系統(Raster Subsystem)與 3D 繪圖加速器軟硬體整合介面研究等四部分之研究情形與重要參考文獻提出評述。

3D 繪圖處理器架構討論與評述

傳統 3D 繪圖管線如圖 13 包含 Geometry Subsystem 與 Raster Subsystem 兩部份[82][83]，如上圖所示: Geometry Subsystem 負責轉換三角形頂點的頂點座標至螢幕座標，而 Raster Subsystem 則將轉換後的三角形實際繪製成影像供顯示器輸出。下圖的參考的架構說明了 3D 繪圖的成像過程，實際上根據成本的考量 3D 繪圖處理器可能僅包含上面的某些單元，例如單獨的幾何子系統或是成像子系統也可能兩個都有，端看設計者的考量來決定，因此處理器可以定義為具有上述部份單元功能且能增加繪圖效能的硬體，沒有限制一定要包含全部的繪圖管線。除了下圖所呈現的傳統架構外，比較特別的架構有 M. Deering et al.[84]於 1988 年在 Proc. SIGGRAPH 提出”The triangle processor and normal vector shader: A VLSI system for high-performance graphics”的 Deferred Shading 架構，將打光與貼圖的動作移到整個 frame 繪製完成後再做，避免掉不必要的打光運算與材質存取來提高效率；以及 Microsoft 於 2006 年於 DirectX10 提出 Unified

Shader 的架構[85]，整合幾何轉換與著色子系統成為單一系統，視需求進行幾何轉換與著色的功能來避免掉兩個系統間負載不平衡所造成硬體使用率不佳的情況，然而不同架構各有其特色，孰優孰劣完全取決於成本效能與應用上的考量，因此繪圖處理器架構的選擇上必須經過深入的研究與探討。

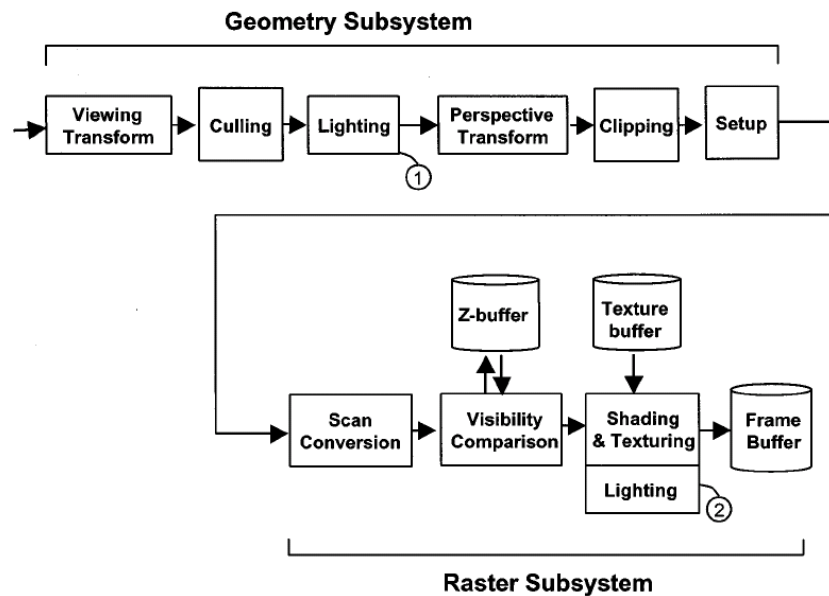


圖13. 傳統 3D 繪圖系統.

幾何轉換子系統

此幾何轉換子系統包含 Viewing Transform & Perspective Transform、Culling、Clipping、著色器(lighting & shading)，分述如下：

● *Viewing Transform & Perspective Transform*

這是 Geometry Subsystem 中最重要的部份，Viewing Transform 將 3D 模型由 world space 轉換至 view space，Perspective Transform 將 view space 中的模型投影至螢幕座標，Raster subsystem 才能依據模型的螢幕座標進行繪製的動作。三維圖形中，座標轉換乃矩陣相乘如圖 14 所示，常以多組管線乘加器平行處理[96][97]，例如有四組乘加器，各自獨立，分別產出 x、y、z、w 四元素，至於算法並無特異之處，依次完成相對應之元素乘法，同時亦連續并之，即得。

$$\begin{bmatrix} m_0 & m_4 & m_8 & m_{12} \\ m_1 & m_5 & m_9 & m_{13} \\ m_2 & m_6 & m_{10} & m_{14} \\ m_3 & m_7 & m_{11} & m_{15} \end{bmatrix} \begin{bmatrix} V.x \\ V.y \\ V.z \\ V.w \end{bmatrix} = \underbrace{\begin{bmatrix} m_0 \times V.x \\ m_1 \times V.x \\ m_2 \times V.x \\ m_3 \times V.x \end{bmatrix}}_{\text{週期一}} + \underbrace{\begin{bmatrix} m_4 \times V.y \\ m_5 \times V.y \\ m_6 \times V.y \\ m_7 \times V.y \end{bmatrix}}_{\text{週期二}} + \underbrace{\begin{bmatrix} m_8 \times V.z \\ m_9 \times V.z \\ m_{10} \times V.z \\ m_{11} \times V.z \end{bmatrix}}_{\text{週期三}} + \underbrace{\begin{bmatrix} m_{12} \times V.w \\ m_{13} \times V.w \\ m_{14} \times V.w \\ m_{15} \times V.w \end{bmatrix}}_{\text{週期四}}$$

圖14. 矩陣運算

- *Culling & Clipping*

為了減少 Raster Subsystem 的 overhead，Clipping 負責剔除不在視覺範圍內的多邊形，而 Culling 則根據多邊形的頂點順序決定其面對觀察點的方向，如果是正面則繼續處理，如果是反面則直接替除。

- 著色器(Lighting & Shading)：

光影效果是使場景中的物體能有真實感的重要關鍵，因此此步驟在整個繪圖管線中扮演相當重要的角色，首先根據頂點與光源幾何關係決定頂點所呈現的顏色(打光，lighting)，進而決定三角形內所有像素的顏色(shading)。現今主流的 shading 技術仍是 H. Gouraud 於 1971 年提出的 Gouraud Shading[86]，主要的理由是對於每個三角形他只需要對三個頂點作打光而其餘的像素使用內插的方式來計算顏色，由於計算量較少因此在 Real-time 的需求下他成為主流的標準。但是 Gouraud Shading 的缺點在於它會產生缺陷(Polygonal Defect)，也就是反射光的形狀會呈現多邊形的形狀而失去真實感。其後 Phong Bui-Tuong 提出了改善 Gouraud shading 的方式，Phong 對每個像素都內插出法向量來作打光，這個方式成功的改善 Gouraud 的缺陷故稱作 Phong Shading[97]。但 Phong 作法的缺點就是他需要大量的正規化與打光運算而難以被 real-time 或嵌入式的 graphics 系統使用。

為了避免線性內插產生的法向量所需的正規化計算，Abbas *et al.*][88] 提出使用 Spherical Linear Interpolation(SLERP)，SLERP 內插所產生的向量長度為一，因此可以省掉作 normalization 的動作，其後 Barrera *et al.* [89] 也基於 SLERP 使用不同的遞迴式來達成 Interpolation 的動作。SLERP 的演算法重點在於把 SLERP 的式子改寫成 Incremental 的遞迴形式使得 Scan Line 的計算過程僅需要執行簡單的加法運算，但相對的代價就是每條 Scan Line 需要較長的 Setup Time 來做 SLERP 的初始化動作。但目前市面上的加速硬體尚未有支援 SLERP，主要的原因在於 SLERP 的特性並非完全適

用於所有的狀況，如果場景有大量投影後的面積很小或是細長的三角形，那 SLERP 反而成為效能的負擔。

應用於嵌入式環境或低成本的 Graphics 系統可以使用定點數與 LNS (Logarithmic Number System)來執行打光所需的算術運算。LNS 最早由 Mitchell 於 1962 年在 IRE Trans. Electronic Computer 提出”Computer Multiplication and Division Using Binary Logarithms” [90]，將運算元轉為對數使用簡單的加減法與移位來取代複雜的乘除法運算，將結果轉回原系統後即可得到原系統相同的運算結果。下面是一個乘法的例子，兩數要做乘法運算時首先對兩個運算元分別取對數，做完加法再取二的冪次即可得到兩數的乘積。同樣的方法可以用在其他的運算上，下表整理了可以使用 LNS 完成的數學運算與運算方式[91]。

Table.1. Normal 與 Logarithmic 轉換表

Operation		Normal Arithmetic	Logarithmic Arithmetic
Multiplication	MUL	$z = x \cdot y$	$X + Y$
Division	DIV	$z = x / y$	$X - Y$
Reciprocal	RCP	$z = 1 / x$	$-X$
Square Root	SQRT	$z = \sqrt{x}$	$X \gg 2$
Reciprocal Square Root	RSQ	$z = 1 / \sqrt{x}$	$-X \gg 2$
Square	SQR	$z = x^2$	$X \ll 2$
Powering	POW	$z = x^y$	$Y + \log_2 X$
Addition	ADD	$z = x + y$	$X + \log_2(1+2^{Y-X})$
Subtraction	SUB	$z = x - y$	$X + \log_2(1-2^{Y-X})$

一個 LNS 運算器主要的功能單元如圖 15 所示，包括對 Logarithmic Converter、Simple Calculation Unit (SCU)與 Anti-log Converter。Log Converter 負責將運算元轉換為對數，而 SCU 則根據 opcode 執行加減法或位移的運算，裡面的主要運算單元是 Adder 與 Barrel shifter，有時根據應用的不同也可能會使用 Multiplier 來做 powering 的運算。最後一部份 Anti-log Converter 是將對數結果轉回原本的數字系統的指數運算。

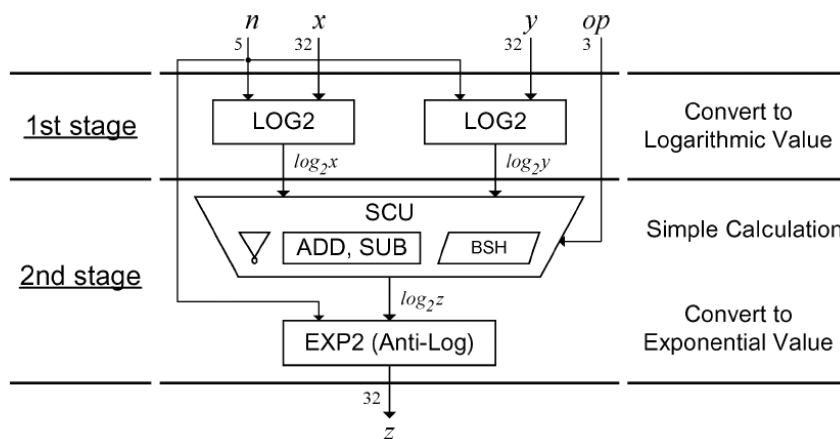


圖15. LNS 運算器主要的功能單元

Mitchell 之後 LNS 的研究重心專著在對數與反對數轉換上，這兩個單元決定了 LNS 運算器的主要面積、速度與精確度。Mitchell 所提出的對數轉換方法僅以單一直線去逼近對數曲線，如圖 16 所示誤差的範圍非常大，其後 Combet *et al.*[92], Alid *et al.*[93][94]針對 Mitchell 的做法提出改善的方式，他們將對數曲線分成數個區段使用不同的直線去逼近，並配合他們各自提出的修正方法來減少誤差。

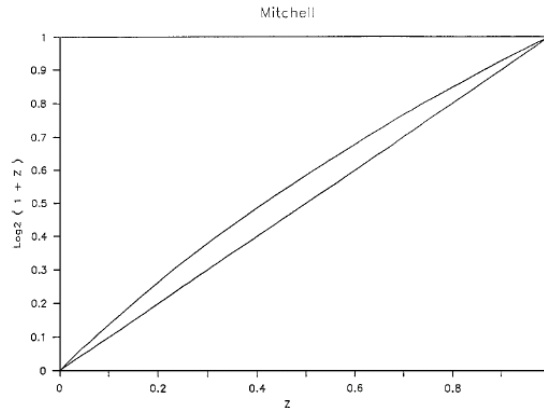


圖16. 單一直線逼近對數曲線

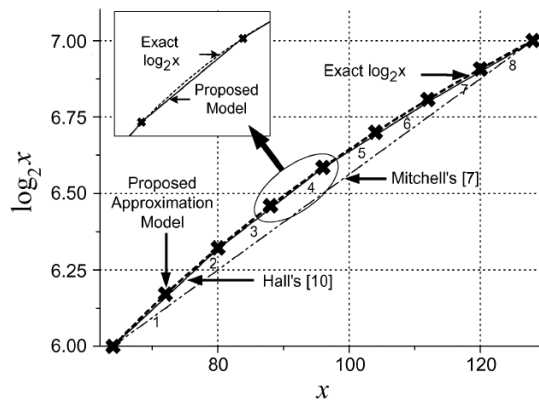


圖17. 分段直線逼近對數曲線

直到現在此種 piecewise 的直線逼近計算方法仍持續被研究與探討，如何選擇適當的分段數與分段區間的選擇以及簡單有效的修正電路設計是這類演算法的重點。圖 17 是分段逼近的示意圖。圖 18 是 Khalid 在 "CMOS VLSI Implementation of a Low-Power Logarithmic Converter" 中所提出的 piecewise 直線逼近對數轉換器的基本架構。LOD 負責偵測 binary word 中第一個 1 的位置，由於 LOD 輸出 32bits 的解碼結果因此需要將他進行編碼以給便給 Shifter 使用。Shifter 在此負責將輸入的數字 normalize 到 1 至 2 的區間以方便後續的運算。最後拿 normalize 後的 mantissa bits

做區間判斷運算與修正的動作。由此架構我們可以看到他的硬體非常簡單。

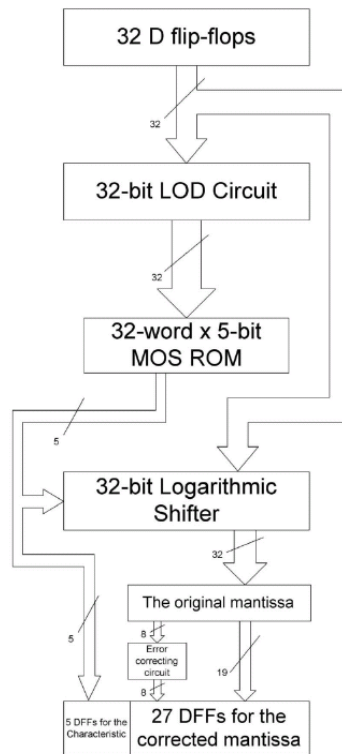


圖18. 分段直線逼近法對數轉換器

查表的也是對數轉換的一種方式，Tomas *et al.*[95]於 1971 在”Multiplication Using Logarithms Implemented with Read-Only Memory”中以直接查表的方式完成對數與指數的轉換並使用加法器進行乘法的運算，考慮硬體成本下直接查表的方式會讓需要非常大容量的 Rom 來儲存表格，所以使用查表的方式往往會配合一些減少表格 entry 的技巧來降低所需的記憶體容量，內插就是一個常見的方式，計算的流程是，先取靠近 Mantissa 的 MSB bits 查表找出其對數數值的上下界，再以剩下的 bits 在上下界之間做內插來逼近實際對數的值。

著色子系統

著色子系統包含三角形驅動引擎(Triangle Setup)、Scan Conversion、Visibility Comparison、Shading&Texturing，分述如下：

- 三角形驅動引擎(Triangle Setup Engine)

在現代 3D 繪圖管線中，三角型驅動引擎(Triangle Setup Engine)扮演連接幾何運算引擎(Geometry Engine)和繪圖引擎(Rasterization Engine)的角色。最早 TSE 是在 CPU 中以軟體實現，但受限當時 CPU 和 GPU 間的

頻寬限制而效率不如預期。接著在繪圖卡中新加一顆 DSP 晶片來實現，但是 DSP chip 並不是為了 3D 繪圖而做最佳化設計且新增 DSP 晶片會使繪圖卡的大小及成本增加，最後 1997 年 NVIDIA 在發行首兩代的繪圖晶片失利後，在第三代繪圖晶片中改採用全硬體化的三角型驅動引擎及其他相關的特殊架構，即得到市場上多數的好評與支持。由此可知三角型驅動引擎在整個 3D 管線中所佔有的運算量之大和硬體化之後所得到大幅效能提升。NVIDIA 在許多技術手冊上都明確的表示，三角形驅動引擎是種浮點術處理單元，專門接收幾何運算引擎傳來的頂點(vertex)資料，利用這些資料組成三角形並剔除畫面中看不到的三角形(Back Face Culling)，最後把剩下的三角形分解成繪圖引擎所需要的像素(pixel)資料並傳給之後的繪圖引擎做其他材質貼圖或打光等更精細的計算。

而現行大部分的 3D 繪圖晶片都把三角形驅動引擎設計再靠近繪圖引擎這端，主要是為了減少系統內單元間的頻寬負載，使經過 TSE 運算所得到的大量像素(pixel)資訊可以直接傳給之後的繪圖引擎做更精細的運算。圖 19 為 Ju-Ho Sohn 博士[97]等人 2006 年在 IEEE JOURNAL 發表的“A 155-mW 50-Mvertices/s Graphics Processor With Fixed-Point Programmable Vertex Shader”中所提到的繪圖晶片系統架構方塊。虛線框所表示的即為幾何運算引擎、三角形驅動引擎及繪圖引擎三者間的關係。

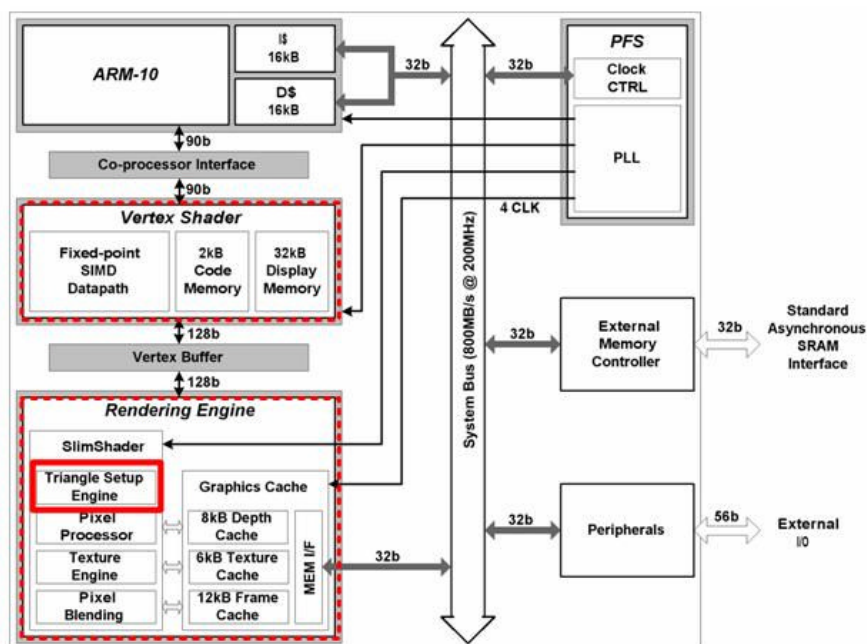


圖19. 圖形處理器方塊圖

Anders Kugler 博士[98]於 1996 年在 “The Setup for Triangle Rasterization” 提到的 3D 繪圖管線中，也特別把三角型驅動引擎獨立出來並分為兩個主要的部分(見圖 20)，分別是 edge walk 和 span interpolation。

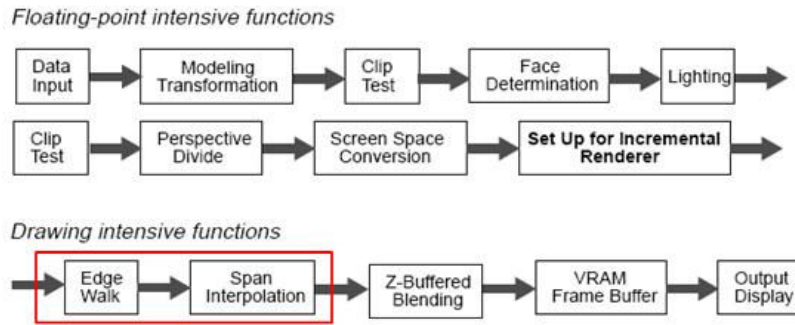


圖20. 3D 繪圖管線圖

Edge walk 的作法，Anders Kugler 博士所提到的方法是使用基本 Bresenham's algorithm，主要是將傳進來的三個頂點用密集的像素連接成三角型邊界。Span interpolation 或是一般通稱的 scan conversion 將再下一小節詳述。

同時 Anders Kugler 博士也引用了 Michael Derring 博士[99]在 1995 年在 ACM 期刊提出的 geometry compression 的概念在自己的 TSE 內。其中特別提到無失真的三角網狀圖形壓縮(triangle mesh compression)和失真的頂點參數(vertex parameter)壓縮等等。

Michael Derring 博士在幾何運算引擎及三角型驅動引擎兩端都放置一個緩衝用的堆疊(stack buffer)，使各三角形間共用的頂點存在緩衝區中，不需要重複傳輸以減少使用傳進 TSE 的頻寬，同時加快整體 3D 繪圖卡的效率。結果如下圖 21 所表示，左上角為繪圖的模型，右上角分別為壓縮前後所需要傳輸的資料點，壓縮前頂點重複傳輸率高達將近 50%，壓縮後雖然增加使用堆疊的空間，但重複傳輸率降到 5% 左右，大幅降低 TSE 的頻寬以及提高整體的效率。

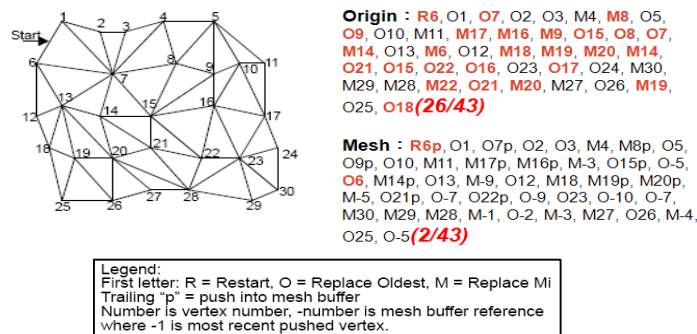


圖21. 三角形網狀圖形壓縮圖

頂點參數包含位置、顏色、法向量等。以一個三角形三個頂點的位

置和顏色不會差異太大為原則，可以套用常見的預測式失真壓縮法如 DPCM 或 ADPCM 等等，將可以獲得較好的壓縮比例和效果。

在頂點的法向量壓縮方面，Michael Derring 博士[99]從人類視覺的觀點出發，提出一個新的觀點。首先刪除人類視覺系統無法辨別的向量，把單位圓內的向量減少至 100,000 個左右，再以單位圓分為八個象限以及一個象限分為六等份來摺疊，如下圖 22 表示，使原本 100,000 個左右的向量減少至 2,000 個左右來代表 1/48 的單位圓(8*6=48)，最後將這 2,000 個向量存在表格中，可以利用 11bit 表示 index，其餘摺疊起來的部分在分別用 3bit 去編碼。解碼時只需要正負號轉換或座標交換即可得到整個單位圓 100,000 個向量中任一個向量值。

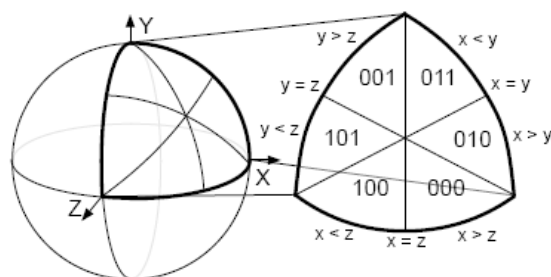


圖22. 單位圓利用象限和數值大小關係產生的編碼法則圖

- *Scan Conversion*

Scan conversion 為整個三角型驅動引擎中計算最多的部分，三角型經過 edge walk 之後得到完整的邊界像素，在每個水平掃描線上，利用兩個端點的值分別用內插法算出其中各點的位置、顏色及法向量等必要資訊。如下圖 23 中所表示， P_0 和 P_n 分別為兩個端點， δx 表示兩點的 x 差值， δz 表示兩點 z 的差值...以此類推，最後算出三角形內每個像素(pixel)的位置 (X、Y、Z)和顏色(R、G、B、 α)以及法向量(N)的值傳給之後的繪圖引擎。

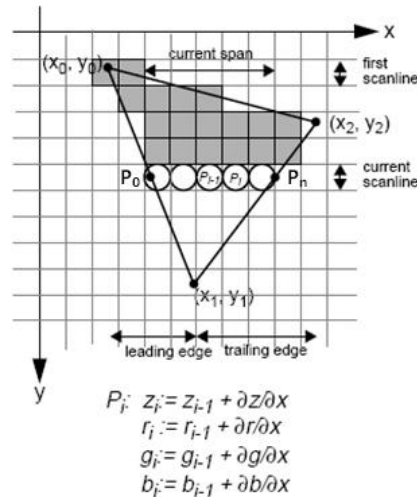


圖23. 三角形的圖形掃描轉換圖

● 深度緩衝區之壓縮(Z Buffer Compression)與 Visibility Comparison :

深度緩衝區的概念最早是由 Sutherland *et al.* 在 1974 年所提出，而深度緩衝區相關的演算法最早則是在 1975 年由 Catmull 提出；當時是以很直覺的方法來做深度緩衝區相關的研究，因此在效率上自然不會有令人太滿意的結果。

Visibility comparison 主要是為了能夠提早過濾掉確定在最後呈現的畫面上看不到的資料，減少系統需要處理的總資料量，以提高系統的效能，例如：位於相對於 viewport 而言，背對 viewport 的三角型，即是可能可以提早被過濾掉的資料；visibility comparison 相關的演算法多需要利用到 Z buffer，而且 Z buffer 的大小與顯示螢幕的大小有即密切的關係，越大的螢幕其對應的 Z buffer 就越大，此外除非是已經被過濾掉的資料，否則幾乎所有的資料到了 visibility comparison 時，一定會存取到 Z buffer。

Hierarchical Z buffer[107]，由 ATI 所發表出來，是一種用於 visibility comparison 常聽到的一種技術，利用深度緩衝區額外地新增出 hierarchical Z buffer，藉由這個 hierarchical Z buffer 可以完成具有某個效率程度的 visibility comparison 演算法；相對於 ATI 的 hierarchical Z buffer，nVidia 亦有其相對應的技術在 visibility comparison 上；此外，國內的陳良基教授 [115]、李鎮宜教授[103]及韓國的 Lee-Sup Kim 教授[104]等，皆在 visibility comparison 演算法上提出不少改進與新穎的技術。

今日已經有許多深度緩衝區改良的方法，包含：tile depth buffer，depth cache，tile tables，fast z-clears，z-min culling，z-max culling，depth buffer compression；這些方法的大致可以用下面的深度緩衝區架構概要圖如圖 24 來表示這些方法的相對位置與功能[106]。

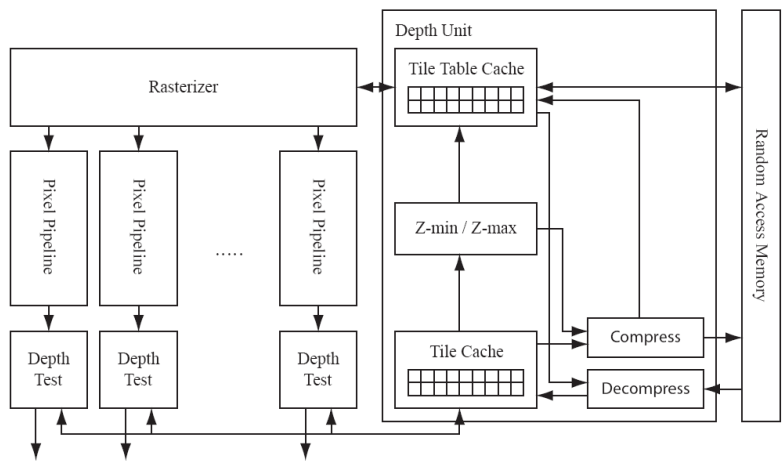


圖24. 深度緩衝區架構概要圖

其中 depth buffer compression 又有許多方法，主要可以分成 fast z-clear, offset, plane 三種壓縮方式[106]，包括: fast z-clear[108], differential differential pulse code modulation(DDPCM) [105]，Anchor encoding[113]，plane encoding[114]，depth offset compression[111]，efficient depth buffer compression[106]等方法，不過大多以專利的方式呈現。

Fast z-clear 可以讓我們避免去讀取目前已經是標記為是清除的 depth tile，而直接進行 z test 來檢查是否更新 depth tile 上的 z value，因此不僅可以使我們得到不錯的壓縮比而且也易於實做。

Differential differential pulse code modulation(DDPCM)是利用 z values 在 screen space 上為線性分佈的特性而達到壓縮的效果。圖 25 解釋 DDPCM 的計算流程。

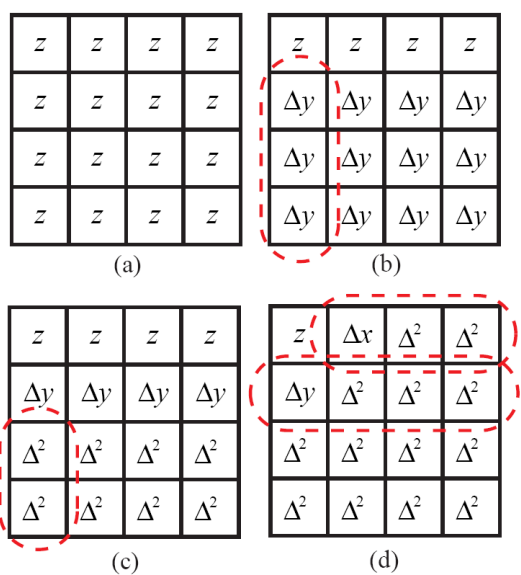


圖25. DDPCM 的計算流程。

由上圖，(a)表示最初未經過壓縮的 depth tile，(b)表示了如何計算 first order differentials，(c)表示如何計算 second order differentials，(d)則表示計算第一及第二列的 second order differentials。經過合理的假設與計算，當使用 32-bit depth buffer 時，DDPCM 大約可以提供 8:1 的壓縮比。

Anchor encoding 與上述 DDPCM 的概念極為相似，一樣是利用了 z values 在 screen space 上為線性分佈的特性。圖 26 解釋 Anchor encoding 的計算流程。經過 Anchor encoding 壓縮過後的 tile 主要由 z 、 Δx 、 Δy 、 d 所構成； z 為壓縮時的 reference point， Δx 及 Δy 分別為相對於 z 的水平方向及垂直方向 first order differentials，而 d 則為相對於 first order differentials 的 second order differentials。相較於上述的 DDPCM、Anchor encoding 的概念，plane encoding 避免儲存 differentials 的方式，而改以 index 為軸心概念的方式達到壓縮。圖 27 解釋 plane encoding 的計算流程。

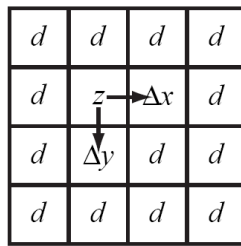


圖26. Anchor encoding of a 4x4 tile

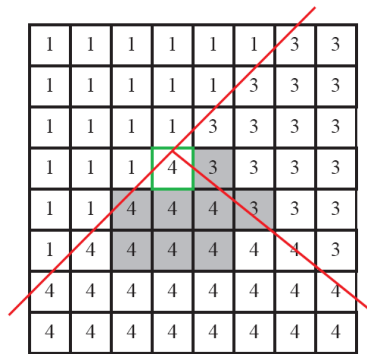


圖27. Plane encoding

由上圖 27，當 tile size 為 8x8 時，由兩條線隔出三塊區域，plane encoding 即針對不同的區域做 index 的動作，最後儲存這些 indices 達到壓縮的效果。

Depth offset compression 大致上雖然不同於 DDPCM 及 Anchor encoding，不過最後皆是儲存差值來達到壓縮的目的。圖 28 解釋 depth offset compression 的計算流程。由圖 28，介於 representable range 中的值

表示與 Z_{min} 或 Z_{max} 的差值為可儲存的，而且此方法最後就是儲存這些差值而達到壓縮的目的；若是系統設計本身沒有儲存 Z_{min} 及 Z_{max} ，則可以在每個 tile 中選擇一個或兩個 reference points，這些 reference points 可視為下圖的 Z_{min} 及 Z_{max} ，如此一來就跟 DDPCM 雷同；一般而言，depth offset compression 是比較適用於系統設計本身有儲存 Z_{min} 及 Z_{max} 。

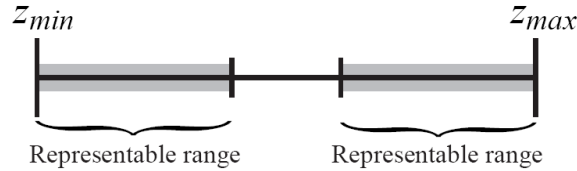


圖28. Depth offset 壓縮

Efficient depth buffer compression 與 DDPCM、Anchor encoding 以及 depth offset compression 的核心概念一樣，利用儲存差值來達到壓縮的效果，比較不同的是對差值的額外處理；下圖 29 解釋 efficient depth buffer compression 的計算流程。

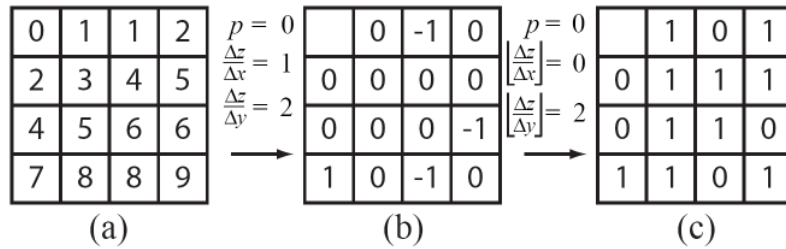


圖29. Efficient depth buffer 壓縮

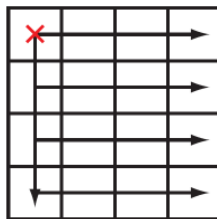


圖30. 圖 18：Traversal order

由圖 29 (a)表示最初未經過壓縮的 depth tile，(b)是經過圖 30 所描述的 traversal order 後，計算得到的差值，藉由觀察(b)可以發現，當一個 depth tile 上的 z values 夠連續時，則差值的範圍會介於-1 到 1；若所有的差值皆介於-1 到 0 時，此時我們將所有的差值都加上 1，而且對應的 $\Delta z/\Delta x$ 及 $\Delta z/\Delta y$ 都減去 1，則我們就可以用 1 bit 表是最後要被儲存的差值而非 (b)表示的 depth tile 需要 2 bits 儲存差值，如(c)所示，進而得到更好的壓縮比。

- *Texturing*

將通過 visibility comparison 的像素的顏色的相關資料與貼圖的座標計算出來，並且由 texture buffer 讀取出貼圖的材質，最後將材質貼至對應的位置，輸出至 frame buffer，呈現在螢幕。

3D 繪圖加速器軟硬體整合介面

- *軟硬介面*

軟硬介面將硬體整合進入系統，並提供應用程式存取硬體資源的能力。包含驅動程式(Driver)與低階繪圖指令介面(Rendering API)，高階應用程式介面(High level application API, optional)。驅動程式[116]負責管理硬體，控制與存取硬體以及結合 OS 進行資源與行程的管理並且提供低階的繪圖指令 API 供應用程式使用。繪圖指令 API 扮演了抽象化硬體的角色，提供應用程式一致性的繪圖指令介面。此介面的制定必須依循公認標準，主要的原因是依標準撰寫完成的程式可以順利在不同硬體平台執行，換句話說支援標準的硬體即可被現有的軟體所支援，大幅提高軟體可移植性與硬體的相容性，同時開發硬體時採用公認的介面可以避免設計發生功能上的遺漏，減少不必要的研發成本。目前嵌入式平台主流的繪圖指令介面是由 Khronos 組織公開定制的 OpenGL ES[117]，已經被 AMD、ADI、Intel、NVIDIA 等廠商支援是一個非常成熟的產業標準，不論學界或產業繪圖處理器研發都建議採用它。而 Visual Simulation, Games, Virtual Reality, Scientific Visualization 等高階複雜的應用程式，除了繪圖還需要場景物件管理的機制(Scene-graph)來處理場景間物件的互動與顯示和一些瑣碎功能如:模型/場景檔案的讀取儲存繪圖行程的管理等，高階的應用程式介面及其程式庫負責支援這些工作。基於高階應用的需求，硬體開發者應在其產品應用程式解決方案中提供自己的或使用第三方組織的高階應用程式介面例如 OpenSceneGraph[118]、Java3D[119]等。硬體本身只需支援標準的繪圖指令介面如 OPENGL 或 DirectX 即可順利的與前述之應用程式介面整合。

- *Mesh construction 與 Reconstruction*

產生三維的立體模型主要可以分成三種方式，第一，利用軟體製作三維立體模型；第二，使用三維掃描器，以雷射掃描欲建立三維模型之物體，將物體的三維資訊經過計算後以三維立體模型呈現；第三，使用攝影機，以不同角度拍攝欲建立三維立體模型之物體，得到一連串的二維影像後，將相同角度的二維影像對經由三維立體模型成像公式的計算、影像處理技術(前置處理、影像分割、特徵擷取、影像對比)，求得物體在不同角度上的深度值(depth value)，建立三維立體模型。

子項目三：MVC 編碼系統中介軟體

本項目將分析多視角視訊編解碼標準，改善現有的編碼工具並提出新的編碼工具藉以實現可任意視角觀賞之高效能多視角編解碼標準。研究方向包含(1)多視角視訊編碼演算法效率改善，(2)視角合成 (View Synthesis) 技術開發，與(3)中介軟體定義與開發。以下說明本項目研究方向之背景與目的。

MHP 平台上之互動式 TV 應用

歐美國家早在幾年前就開始利用數位電視的頻寬資源，提供互動性質的服務內容，例如直播球賽中，如果看中球員身上所穿戴的廣告商贊助衣服，可立刻利用電視下單訂購，甚至是進行場外賽局下注，而德國的電視公司 ZDF 則是推出互動式的旅遊節目來吸引消費者，觀眾可藉由虛擬的旅遊導覽來觀賞自己想看的資訊[124]。國內的數位互動電視服務業者中嘉科技結合新視波公司，推出提供即時資訊、遊戲娛樂、飲食購物、金融理財、生活休閒、文化教育等六項互動內容，稱之為「互動 TV」[125]。至於互動內容更是令人驚艷，包括電子節目選單 (EPG)、隨選視訊系統與說明設定，如同以電視操縱電腦軟體一般，利用遙控器就能與電腦進行互動。這些都已經成為相當受歡迎的電視機新娛樂功能。

MHP 平台結合自由視角電視(Free Viewpoint TV)應用

隨著 MHP 平台在各國的多方應用，倘若 MHP 平台能支援 3-D 視訊，那麼在應用方面的範圍更可以多元化；比如，上述的旅遊節目，有時會因為 2-D 的影像而顯得乏味缺，若能引進 3-D 視訊，支援多視角視訊，將帶給使用者有如身歷其境的感覺。另外，美食節目若結合自由視角電視功能，收視用戶不僅可以多方觀賞廚師製作佳餚時，不同角度的手勢、醬料分量、擺盤方式等，甚至於可立刻利用電視進行購物，購買節目中的調味料、食品原料或鍋碗瓢盆；類似的想法亦可引用於旅遊導覽系統，提供要出外旅遊的相關景點資訊，可以在螢幕點選地圖的景點，即可進入 3-D 視訊的景點介紹，除了可以讓觀賞者了解觀光景點的資訊外，更可以豐富其所要介紹的內容，來增加遊玩者的旅遊欲望；基於這出發點，亦可將此功能用於 GPS 導航系統，現階段雖有 3-D 導航，但是採用 3-D 動畫圖形的方式呈現，雖然有別於 2-D 的地圖較易於觀看，但仍缺乏真實感，且與現實生活中建築物、地面景物仍有一段差距，所以在 MVC 的應用下，皆可克服此種問題，藉由多視角的視訊傳輸，可以克服傳統上狹隘的死角畫面，搭配即時路況與導航系統，相信在未来對於駕駛而言，可以帶來諸多便利。

自由視角電視(Free Viewpoint TV, FTV)現況

自由視角電視(FTV)是一項新興的 TV 技術，允許一個人可以任意改變視訊觀賞角度，位置，與方向而獲取身歷其境之 3-D 實境視覺感受。伴隨 3-D 顯示器，擷取設備，編碼技術，與處理器能力的大幅進展，許多專家都開始相信自由視角電視(FTV)將不再是一個夢想而是一個在近期即可實現之創新 TV 技術。透過自由視角電視(FTV)，許多天馬行空之想像與商機都將變成可能例如 (1)互動式體育、展覽、活動觀賞(Interactive and Realistic Appreciations of Sports, Events, and Exhibitions)，(2)浸入式旅遊導覽、線上會議(Immersive Virtual Sightseeing and Conferencing)，(3)互動式實境 3-D 廣告(Interactive and Realistic Presentation)，(4)互動式遠距教學與醫療(Interactive Remote Education and Medicine)，(5)實境 3-D 自然景觀觀測(Realistic Bird Watching, Safari, and Undersea Park)，與(6)實境 3-D 交通監控(Traffic Monitoring at Intersection)等。

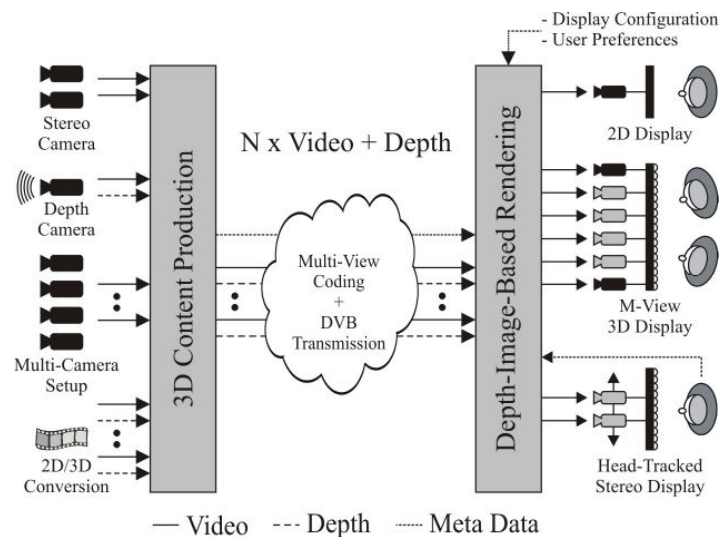


圖31. 自由視角電視系統。[120]

基於以上自由視角電視(FTV)之諸多商業需求，MPEG 標準組織在 2007 年 7 月於 Geneva 的會議由日本名古屋大學 Prof. Tanimoto 主導開始著手標準化自由視角電視(FTV)所需之前後端相關技術。圖 31 說明現階段自由視角電視之初步運作方式。Server 端將使用多組 Camera 來擷取產生多視角視訊，並透過一些校正機制來補償相機之間的色度與角度的些微差異(Misalignments)。校正過後的多角度視訊影像將進一步利用多視角視訊編碼技術(MVC)來進行壓縮與傳送。伴隨著壓縮視訊資料，相機參數與深度資訊(Depth Information)(可透過視差估測或特定深度相機所產生)則分別透過 Meta Data 與標準化之深度編碼(Depth Coding)格式來傳給 Client 端。Client 端的解碼器則可依據不同的顯示器特性進行視訊解碼並利用深度資訊適時且任意的產生各種不同的觀賞角度。初步的自由視角電視(FTV)應用需求已定義在 MPEG N9168 文件[120]，其中預定標準化目標包含(1)多視角視訊解碼器(Decoder)，(2)視訊合成技術(View Interpolation)，

與(3) FTV 傳輸資料格式(FTV Data Format)(包含視訊資料，深度資訊，與像機參數等)。

多視角視訊編碼標準(Multi-view Video Coding)現況

有鑒於三維多視角影片漸漸受到重視，Moving Picture Expert Group (MPEG, ISO/IEC SC29/WG11)會議從 2001 年即成立一個三維視訊音訊專家群(3D Audio and Video Group)開始探討三維音訊視訊相關技術。專家群創立初期，日本與韓國顯示器與消費電子大廠如 Sony、Matsushita 和 Samsung 均投入大量的研究人力積極開發。在經過多年的研究，數篇貢獻文件已於 2006 年 1 月在泰國曼谷所舉行的 Call for Proposal of Multi-View Video Coding 中證明：「相較於以不同視角個別壓縮的方法(Simulcast)，利用不同視角畫面的相關性來進行壓縮可以得到更好的壓縮效果」。因此，在 2006 年 7 月，ISO/IEC JTC 1/SC 29/WG 11 MPEG 於奧地利的第 77 次國際標準制訂研討會中正式宣佈多視角視訊壓縮編解碼(Multi-View Video Coding, MVC)成為國際電信聯盟(International Telecommunication Union, ITU)與國際標準組織(International Organization for Standardization, ISO)共同專家群(Joint Video Team, JVT)的下一個合作開發項目。

目前，多視角視訊壓縮編解碼(MVC)的開發建構在 MPEG-4 Part 10 精進視訊編碼(Advanced Video Coding)之上，並將成為暨 MPEG-4 Part 10 Amendment 3 可調視訊編碼(Scalable Video Coding)之後的另一個精進視訊編碼(Advanced Video Coding)修正案(Amendment 4)。現階段的多視角視訊編解碼標準草案(JMVM Joint Draft 4.0) [121]已針對 Sequence Parameter Set (SPS)、NAL Unit Header SVC Extension 及 Reference Picture List Reordering Syntax 提出了新的高階編碼語法(High-Level Syntax)以適應多視角視訊的編碼架構。如在 SPS 中利用 view_level 用來指定 NAL 單元視角可調性的層級(Level)，view_id 為視角的識別號碼，num_views_minus_1 用來指定視角(View)的數目。另外標準草案亦描述多視角視訊編解碼架構下的解碼程序(Decoding Process)包含參照畫面串列建立(Reference Picture List Construction)、已解碼參照畫面標記程序(Decoded Reference Picture Marking Process)以及解碼畫面暫存器的管理(Decoded Picture Buffer Management) 等。此外，現階段的 Joint Multi-view Video Model (JMVM 5.0) [122]也涵蓋多項可能標準化技術如亮度補償、Motion Skip Mode、View-first and Time-first 編碼順序等。

多視角視訊編碼技術文獻探討(Technology Survey)

多視角視訊編解碼主要問題在於如何在不同的視角視訊之間做有效的估測與補償。一般來說，多視角視訊的取像相機通常存在距離和角度的差別。同時，物體表面的傾斜度亦會造成相鄰相機在成像上的差異。因此，

當同一個三維空間的物體透過不同相機鏡頭在二維的底片呈像時，通常此一物體在兩個不同底片的呈像除了有位移(Translation)，也可能存在著較為複雜的旋轉(Rotation)與大小變化(Dilation)。而若兩畫面間的物體存在旋轉與大小變化時，傳統以固定方塊為基礎(Block-based)的補償(Compensation)與估測(Prediction)將不再有效率。甚至，視角間不同光線變化所造成的些微亮度改變對於尋找最佳預測區塊的編碼效率上也有著顯著的影響。以下我們針對目前 Joint Multi-View Video Model 5.0 [122]所考慮之相關研究分別詳述。

● A. 視角與時間軸估測結構(View-Temporal Prediction Structure)

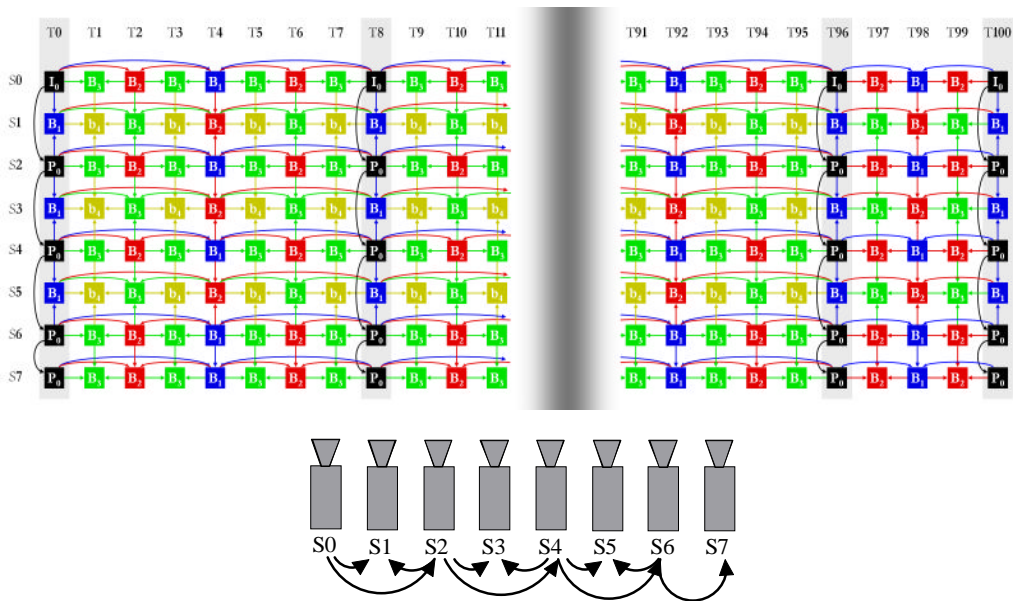


圖32. 一維相機陣列多視角編碼架構。[122]

多視角視訊編解碼標準除了支援時間上的畫面預估，同時提供不同視角畫面的預估。主要的預估架構源自於 MPEG-4 Part 10 Amd.3 可調視訊編碼之階層式 B 畫面(Hierarchical B Pictures)。圖 32 說明階層式 B 畫面(Hierarchical B Pictures)在視角和時間上的預估結構。其中橫軸代表時間而縱軸則表示不同的視角。如圖所示，個別的視角影片在時間序列上採用階層式 B 畫面(Hierarchical B Pictures)。此外，在不同視角間則採用傳統"IBPB..."-之預估結構(Prediction Structure)，也就是第偶數個視角的影片序列會利用相鄰兩個視角畫面作視角的預估。如圖中 S1, S3 及 S5 三個視角影片序列除了時間上的預估之外，還分別利用"S0, S2", "S2, S4"以及"S4, S6"作為視角間預估的預估視角。

值得一提的是，Ye-Kui Wang 等作者[123]分析了時先(Time-First)編碼架構與視角先(View-First)編碼架構在已解碼畫面儲存器的需求。圖 33 為視角先編碼架構(View-First)，目前 MVC 參照軟體的編碼架構便是採用視

角先的編碼架構。圖 34 則為時先編碼架構，同一個時間點的所有畫面在編碼的順序上是相鄰且連續的。經由分析，若在視角維度(View Dimension)上採用類似多層次的雙向估測(Hierarchical B-picture Prediction)的方式來達到視角可調性時，視角先編碼架構所需的已解碼畫面儲存器的容量需求為

$$\text{number_of_views} + \text{GOP_length} * (1 + \log_2(\text{number_of_views})) + \log_2(\text{GOP_length})$$

而時先編碼架構所需的已解碼畫面儲存器的容量需求為

$$(\log_2(\text{GOP_length}) + 1) * \text{number_of_view} + (1 + \log_2(\text{number_of_views}))$$

其中，number_of_views 為視角的數目，GOP_length 為編碼的畫面群(Group of Picture)的大小。實驗結果發現，視角先(View-First)編碼架構縮所需的儲存器容量比時先(Time-First)編碼架構來的大。以 GOP=16 且 number_of_view=9 為例，視角先編碼架構需要 77 個儲存器容量，相對的，時先編碼架構只需要 49 個單位容量。

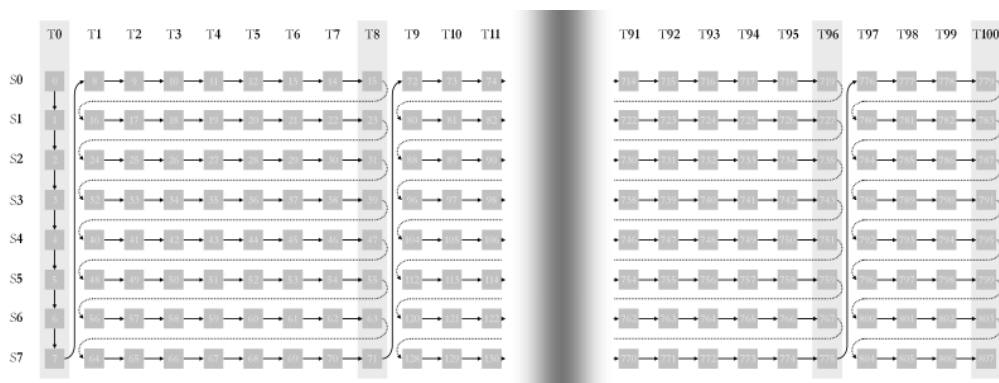


圖33. 視角先(View-First)編碼架構。[122]

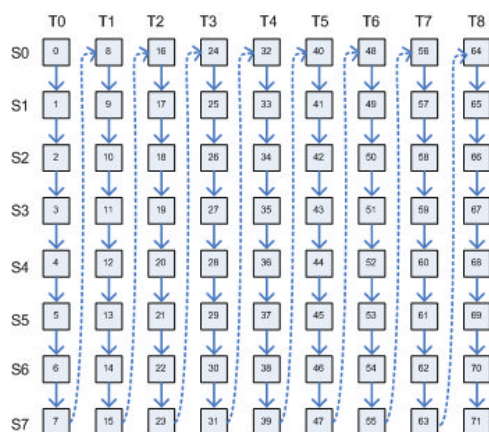


圖34. 時先(Time-First)編碼架構。[122]

● B. 亮度補償(Illumination Compensation)

不同視角的畫面因為光線條件的不同而容易導致相機測光在不同視角畫面產生亮度(Illumination)的差異。通常亮度(Illumination)的差異會進一步導致視差補償後的殘餘訊號帶有 DC 訊號進而影響編碼效率。因此，目前 JMVM 採納一亮度補償機制藉以在做視差補償的同時消除兩畫面間的 DC 亮度差異。

錯誤！物件無法用編輯功能變數代碼來建立。

圖35. JMVM 5.0 目前所採納之亮度補償架構。[122]

$$M_{cur} = \frac{1}{S \times T} \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} f(i, j)$$

$$M_{ref}(p, q) = \frac{1}{S \times T} \sum_{i=p}^{p+S-1} \sum_{j=q}^{q+T-1} r(i, j)$$

$$MR_SAD(x, y) = \sum_{i=m}^{m+S-1} \sum_{j=n}^{n+T-1} \left| \{f(i, j) - M_{cur}\} - \{r(i+x, j+y) - M_{ref}(m+x, n+y)\} \right| \quad (1)$$

$$MR_R(i, j) = \{f(i, j) - M_{cur}\} - \{r(i+x', j+y') - M_{ref}(m+x', n+y')\}$$

$$= \{f(i, j) - r(i+x', j+y')\} - \{M_{cur} - M_{ref}(m+x', n+y')\}$$

$$= \{f(i, j) - r(i+x', j+y')\} - DVIC \quad (2)$$

圖 35 說明目前 JMVM 5.0 所使用的亮度補償方法。傳統運動向量估測(Motion Estimation)利用最小化 SAD 來計算運動向量(Motion Vector)。為了補償亮度的差異值，新的 SAD 計算公式，Mean-Removed SAD (MR_SAD)，將傳統的 SAD 計算公式更改成如方程式(1)所表示，其中 f(i, j)及 r(i, j)分別表示目前畫面與參照畫面， M_{cur} 與 $M_{ref}(p, q)$ 為目前區塊與參考區塊的像素平均值，(p, q)表示參考區塊的位置， $S \times T$ 則分別表示目前區塊的長與寬。如方程式(1)，為了補償亮度的差異值，Mean-Removed SAD 在尋找運動向量(Motion Vector)的過程，即 ICA ME，會分別讓目前畫面與參考畫面減去各自的 DC 值。所產生的動態向量(x', y')會利用方程式(2)計算亮度補償過後的殘餘訊號(residual signal)。其中相關的亮度補償值 Difference Value of Illumination Change (DVIC)將以 DPCM 方式傳送給解碼端。比較特別的是目前亮度補償僅針對 Macroblock 大小為 16x16 的情況(包含 Inter16x16, B_Skip, B_Direct_16x16, 與 P_Skip)。在其他的分割(Partition)大小下亮度補償將不會被進行。

● C. Motion Skip Mode

由於相鄰視角畫面中的動態資訊(Motion Information)具有高度的相

關性，目前的 JMVM 提供了運動略過模式(Motion Skip Mode)使得 Macroblock 在編碼時可以省略動態資訊(Motion Information)藉以提高編碼效率。其中動態資訊包含 mb_type，reference_index 以及 mvd 都將改由參照視角畫面推導而得。

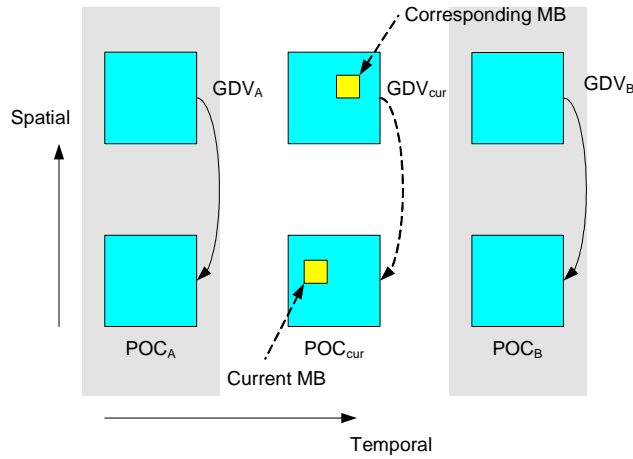


圖36. Global Disparity Vector 之推導。[122]

運動略過模式只作用在由 Sequence Parameter Set 所描述具有視角間預估的視角序列，運動略過模式分為兩個主要階段。(a)全域視差向量(Global Disparity Vector)的推導與(b)動態資訊的取得。以下分別就此兩階段詳細描述。

(1) 全域視差向量(Global Disparity Vector)的推導

全域視差向量的目的是用來指出相同時間點的參照視角中相對應 Macroblock 的位置。首先利用最小化畫面差的殘餘訊號(residual signal)來求出錨點畫面(Anchor Picture)間的全域視差向量，並將此向量編碼傳送至解碼端。非錨點畫面(Non-Anchor Picture)間的全域視差向量則利用時間順序的前一個錨點畫面與後一個錨點畫面的全域視差向量以 Bilinear 方式推導而得。如圖 9 所示，為了找出目前巨集區塊(Current MB)在參照視角中相對應的區塊(Corresponding Macroblock)，目前畫面的全域視差向量 GDV_{cur} 利用前一個與下一個錨點畫面的全域視差向量 GDV_A 及 GDV_B 推導而得，其推導公式如方程式(3)所示。 POC_{cur} ， POC_A 與 POC_B 分別表示目前畫面，前一個錨點畫面與下一個錨點畫面的 Picture Order Count (POC)。

$$GDV_{cur} = GDV_A + \left[\frac{POC_{cur} - POC_A}{POC_B - POC_A} \times (GDV_B - GDV_A) \right] \quad (3)$$

(2) 運動資訊(Motion Information)取得

利用階段(a)所推導的全域視差向量指出參照視角畫面中相對應的巨集方塊，將此相對應巨集方塊的運動資訊包含 mb_type ， $reference_index$ 以及 mvd 作為目前巨集方塊編碼所用。一旦巨集方塊選擇運動略過模式編碼，其運動資訊不須編碼而是經由推導至相對應的巨集方塊取得。

● **D. 參照畫面串列機制(Reference Picture List)**

目前最新的 MVC 工作草案[122]尚在沿用 H.264/AVC 關於參照畫面串列的機制，其流程大致可分為

1. 初始時間方向上預測畫面串列(視角內參照畫面)。
2. 根據 MVC 序列參數集(SPS)重新安排視角內參照畫面在串列上的順序。
3. 將視角間參照畫面置於原本串列的後面。
4. 依據 Reference Picture List Reordering 機制將參照頻率最高的畫面置於串列開頭，進而改善編碼效能。

視角合成文獻探討(Technology Survey on View Interpolation)

● **A. 二維圖形為基礎的方法(2-D Image-Based Approach)**

二維圖形為基礎的演算法利用兩張影像每個像素之間的視差向量(Disparity Vectors)，做線性的[126][127][128]或非線性的內插[129]並產生出兩張影像中的任意中間影像(Intermediate Picture)。以區塊為主的視差向量估計法(Block-based Disparity Vector Estimation)會因為區塊包含了物體的邊界而造成視差向量值計算上的偏差。因此，Wang[130]及 Park[126]分別提出以網狀為基礎(Mesh)與以三角區塊(Delaunay Triangulation)為基礎的視差估計法來避免區塊包含物體的邊界。然而，二維圖形為基礎的方法都會面臨到場景中物體遮蔽(Object Occlusion)所造成的問題。

● **B. View Interpolation based on Ray-Space**

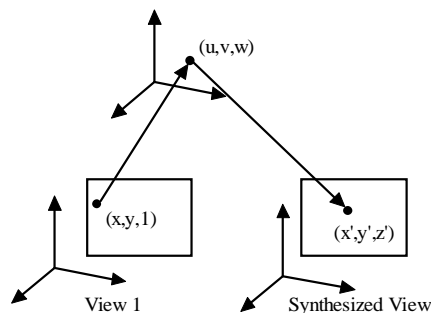


圖37. 三維資訊重建方法之視角合成概念

三維資訊重建方法其原理是利用畫面的深度圖(Depth Map)以及相機參數(Camera Parameters)經由針孔成像模型(Pinhole Model)把畫面二維空間中的像素投射到三維的真實空間。同時，再根據所得的相機參數把三維空間上的像素點重新投射到欲合成視角畫面的二維空間上即可重建出虛擬視角的畫面。如圖 37 所示，把影像中二維的像素座標利用 Pinhole 的成像投射公式投射到三維空間上，來重建三維資訊。接著再把重建後的三維資訊投射到所要合成的視角畫面上，即可完成虛擬視角的合成。三維資訊重建方式雖然可以避免物體遮蔽的問題，但是需要實際場景的深度資訊(Depth Information)。由於深度圖(Depth Map)取得不易，同時要正確估計的困難度高，且深度圖影響合成準確度甚鉅。因此，如何正確估出影像的深度圖(Depth Map)成為三維資訊重建另一個重要問題。

由於人眼之成像系統是由吸收物體反射後的光線 (Ray)，將影像呈現在視網膜上，不同於傳統紀錄圖像的格式 Pixel-based System，其紀錄單一視角的固定數量之採樣點，建立 2D 的 Pixel-space Data；於本計劃中我們採用 Ray-based System，於此系統之下，所有影像之 3D 訊息皆以“光線 (Ray)”呈現，因此可以完整保存 Sampled Ray 的所有資訊，包括其亮度、顏色、射角、位置等，因此所有影像資料不論支援幾個視角(View Point)，皆可以 Ray-based System 來保存光線之完整資訊，建立 3D 的 Ray-space Data。

Ray-space 於空間上的表示法

令 (x,y,z) 為三維空間上任何一點的空間表示法，而 θ 與 φ 分別代表不同視點間的水平轉角差與垂直仰角差，其中 $-\pi \leq \theta < \pi$ ， $-\pi/2 \leq \varphi < \pi/2$ 。以此五維表示法 (x,y,z,θ,φ) 構成 Ray-space，可以清楚表達空間上的每個點的絕對位置，與每個相異視角間的差異。Prof. Tanimoto 於 FTV (Free-view point TV) 中[131]，提供兩種不同的座標表示法來表示 Ray-space，用以在人眼視角與攝影機視角間的切換：

- **Orthogonal Ray-space**：(人眼的視角)

圖 38 中，令 R 為攝影機接收到的一個 Ray，Q 為 X-Y 平面， (x,y) 代表 R 在平面 Q 上交點，而 (θ,φ) 分別代表不同視點間的水平轉角差與垂直仰角差。

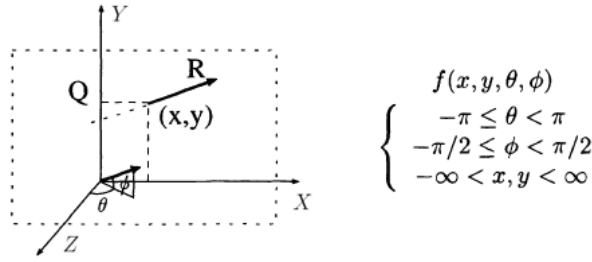


圖38. Orthogonal Ray-space [132]

如此，透過以下轉換式可將 5D 的 Ray-space 轉換為 4D 的 Ray-space。
[133]

$$\begin{pmatrix} x \\ y \\ \theta \\ \phi \end{pmatrix} = \begin{pmatrix} 1 & 0 & -\tan \theta & 0 & 0 \\ 0 & 1 & -\tan \phi / \cos \theta & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \theta \\ \phi \end{pmatrix}$$

• Spherical Ray-space：(攝影機的視角)

圖 39 中，令 R 為攝影機接收到的一個 Ray，Q 為 X-Y 平面經轉角 θ 仰角 ϕ 的轉後，得到與 R 垂直的平面，而 (ξ, η) 代表 R 在 Q 平面上的投影向量， (θ, ϕ) 分別代表不同視點間的水平轉角差與垂直仰角差。

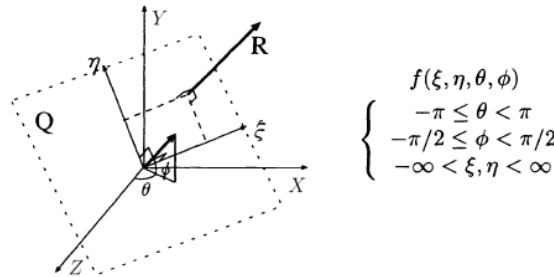


圖39. Spherical Ray-space [132]

如此，透過以下轉換式可將 5D 的 Ray-space 轉換為 4D 的 Ray-space。[3]

$$\begin{pmatrix} \xi \\ \eta \\ \theta \\ \phi \end{pmatrix} = \begin{pmatrix} \cos \theta & 0 & -\sin \theta & 0 & 0 \\ -\sin \theta \sin \phi & \cos \phi & -\cos \theta \sin \phi & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ \theta \\ \phi \end{pmatrix}$$

Ray-space 空間上的視角內插

圖 40(左)為內差演算法的流程圖，analyze 為計算每個 Orthogonal Ray-space 的 $X \cdot \tan \theta$ 截平面[131]的 Variance，Selector 待 Variance 計算完

後，執行 Filter Selection 演算法找出最佳的 Filter，而 $H_1 \sim H_n$ 即代表各種不同的 Filter。

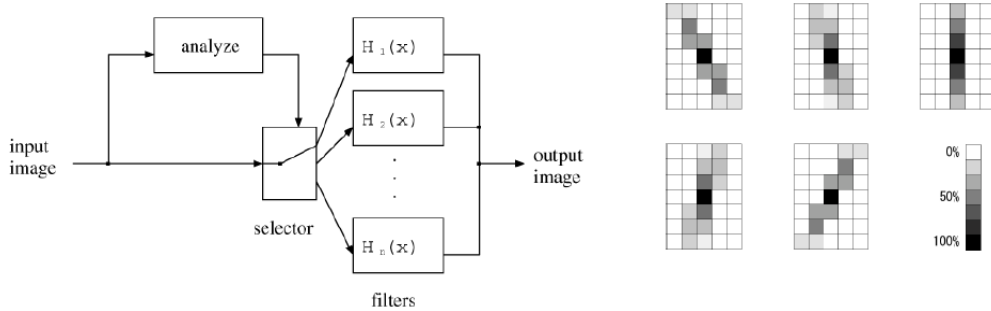


圖40. (左) Ray-Space 上的 Adaptive Filter、(右)數種 Filters [134]

內差前必須先對 Orthogonal Ray-space 的 $X-\tan\theta$ 截平面做 Upsampling，如圖 41(左)。而右圖則為經過 Interpolation 之後的結果。



圖41. (左)內差前的 Ray-space $X-\tan\theta$ 截平面、(右) 內插後的 Ray-space $X-\tan\theta$ 截平面[134]

Filter Selection 演算法，利用 Variance 值來選擇適當的 Filter(取 Variance 總和之最小值)，

Step 1：Row process - Variance Filter 分別對每個 Row-data 計算水平方向的 Variance。

Step 2：Line process - 沿著“目前正要內差的點”，依點中心之座標旋轉，計算每個角度(虛線)的 Variances 之和，並取最小值之方向線為依據，選擇適當的 Filter。

Improved Filter Selection 演算法，如圖 42 所示，P 為鄰近已完成內差的點，C 為目前正要內差的點，P 點上的實線代表 P 點選用實現兩端點當線性內插的參考點(圖 42 的灰色區域)，C 點上的虛線代表正在選出最佳之 Filter。因此 C 點在預測其 Filter 時(虛線)，虛線沿 C 點中心作旋轉，其選用之參考點的連線不允許超過 P 點的實線，因為在 Ray-space $X-\tan\theta$ 截平面(見圖 42) 上所看見的線段，並未直接越過前一條線段。

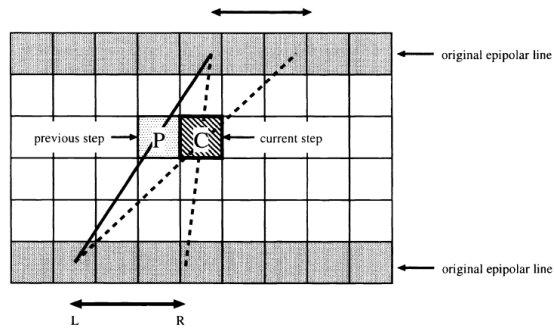


圖42. Selector 選擇參考點示意圖[134]

子項目四：嵌入式平台的 Java 處理器設計 (Java Processor Design for Embedded Systems)

在嵌入式系統中，最令應用程式開發者困擾的一個問題是，即使是為同一類應用所設計的平台（如多媒體手機），不同業者卻會採用不同的處理器和作業系統。因此所有的應用程式都必須客製化，也因此 PC 市場常見的小規模，但是卻充滿創意的應用程式開發商，很難進入寬頻多媒體應用開發的領域。也相對限制了嵌入式寬頻多媒體應用市場的發展。為了解決這個問題，各個不同的國際標準組織（包括數位電視的 DVB、ATSC、OCAP 和電信業的 3GPP、3GPP2）都不約而同的採用了 Java 執行環境 (Java Runtime Environment, JRE) 來作為新一代寬頻多媒體應用程式的執行環境。其中最主要的理由是 Sun 所主導的 Java，在經營跨平台的應用式執行環境已經有多年的基礎。

為了能支援不同的嵌入式應用，Sun Microsystems 定義了 Java 2 Micro Edition (J2ME) 的規格。在這個規格中，定義了不同的 profiles 和 configurations，來滿足各種不同應用及設備的需求[135]。對於手機一類的手持式裝置，3GPP 等單位採用 J2ME 定義的 Connected Limited Device Configuration (CLDC) 以及 Mobile Information Device Profile (MIDP) 來做為手機的開放式應用程式平台。而支援 CLDC/MIDP 的 Java 虛擬機器則是一個較簡單的 VM，稱做 KVM。而對數位電視而言，Connected Device Configuration (CDC) 和 Personal Basis Profile (PBP) 則被採用為開放式應用程式平台[136]。而在 CDC/PBP 之下的 VM 則是一個全功能的 VM。不過 Sun 針對嵌入式系統的需求，特別設計了一個方便移植的 VM，稱做 CVM。因為 CVM 的設計主要是針對不同平台的處理器和 OS 移植做考量，所以並沒有針對平台做最佳化。所以有些機上盒因為直接採用 CVM 及 CDC/PBP 的參考程式碼來做為 Java VM，所以效能並不理想。也因此不能執行太複雜的應用程式。

而在手機方面，雖然在發展多年後，Java 執行環境的效能都很不錯，

但是早期手機規格中所選用的 CLDC/MIDP 所能提供的使用者界面和功能太過陽春，而進年來，多媒體手機的功能和螢幕大小的發展，早已遠超過當年 CLDC/MIDP 所定義的規範，所以在高功能的多媒體手機方面，未來遲早也會走向支援 CDC/PBP 規格的開放式應用程式平台。

最初 Java 語言及虛擬機器在定義的時後並沒有想到有一天會成為嵌入式系統的主流語言，所以它的設計造成在嵌入系統的實作有很多效能上的問題要解決。首先，物件導向 (object-oriented) 的程式需要依靠許多動態記憶體配置的功能，而對嵌入式系統而言，這是相當沒有效率的。其次，Java VM 基本上是一個堆疊機器 (stack machine) [137]。因此在進行任何運算時，會有不斷利用 Stack 來進行計算的中間質存取的动作，這也是非常沒有效率的。一般而言，高效能處理器是會用 registers 來記錄運算過程的中間結果，以提升效能。最後，大部份的嵌入式系統使用了一個頻率低於 300MHz 的 RISC 處理器做為主處理器，這樣的架構對於執行虛擬機器而言是很沒有效率的[138][139]。

過去，也有不少研究試圖加快 Java VM 的效能的[138]。對於嵌入式系統而言，以軟體為主的加速技術如 Just in Time (JIT) [144][145] 編譯的方法是比較不適當的。因為 JIT 編譯器往往需要較大的額外記憶體，而且在應用程式執行前會有額外編譯的 overhead，由於嵌入式系統的處理器較慢，所以這些額外的時間延遲對使用者而言會比較明顯。在[141]，一些 object-oriented 相關的動態解參考 (dynamic resolution) 等指令，都是用硬體加速計算，這邊較困難的部份是要把一些 runtime 的資料結構設計的比較適合用硬體讀寫。有些系統 (如 ARM 所開發的 Jazella [143]) 則是使用 co-processor 的方式，把一些指令直接用硬體解碼及執行。也有一些是直接設計一個硬體直譯器來進行虛擬機器碼的線上翻譯[142][153][154]。也有一些是直接設計 Java 處理器[139][140]。JOP (意為 Java Optimized Processor)是由 Schoberl [139]所開發的一個公開 RTL 碼的 Java 處理器。不過 JOP 定義了自己的 Java profile/configuration，所以它提供的執行環境並不支援 Sun 所定義的任何標準。不過基本上，JOP 執行環境的規格是比較接近 CLDC。

在子項目四中，我們會設計一個 Java 處理器來支援 CDC/PBP 的完整 Java 執行環境。事實上，子計畫四團隊已經在執行過去的一項國科會建置計畫時開發了第一代的 double-issue Java 處理器，並且在 FPGA 上驗證成功。不過，要能支援完整的 CDC/PBP Java 執行環境，還有很多地方需要改進。

三. 研究方法與結果：

前面提到，新的寬頻多媒體的標準及應用一直在演進中（如視訊標準從早期的 MPEG-1/2 到現在的 MPEG-4, H.264, 及 WMV 9，傳輸協定也從早期的 MPEG-2 transport 往 all-IP 的方向走，而 Java profiles 及 presentation scripting language 如 SMIL、Flash、AJAX 等標準也一直在增訂中）。在過去採用硬體佈線的方式設計應用平台的年代（例如機上盒或 DVD 播放器），想要支援新標準往往只能透過購買全新的硬體設備來達成。在未來服務導向的時代，這樣的設計只會增加使用者的負擔而阻礙新服務的推出。這也是為什麼有彈性的高效能中介軟體平台在未來的寬頻網路的推廣會具有十分重要的地位，因為消費者只要透過新應用程式的安裝就可以享受新的服務。

總計畫整合了一個開放式多媒體應用程式平台(圖 43)，並且支援 3-D 視訊的功能，它的特點如下：

1. 以主流的 MHP Java 多媒體中介軟體為基礎進行設計，以提供開放的應用程式平台。
2. 系統核心使用 PowerPC 處理器，時脈頻率在 300 MHz 以下（這是根據 DVB-MHP 的規範 [12]）。
3. 系統包含其它三個加速核心：Java 處理器、3-D video 計算加速核心、以及 3-D Graphics/Video 成像引擎。這三個核心可以大幅提昇標準 MHP 應用程式的執行能力，並提供本計畫所設計的 3-D 視訊擴充的功能。
4. 系統整合一個具有 VGA (640x480) 解析度的 3-D 螢幕（利用紅藍鏡產生立體效果）。
5. 系統整合是以 Java 執行環境為中心，我們所設計的系統軟體包含了一個以 interrupt-driven service routines 為核心的 native-code RISC kernel，系統要執行的工作分成四類：Java services、2-D graphics services、3-D graphics services、和 3-D video synthesis services。除了 Java services 是利用 interrupt-driven (由 Java core 發 interrupt requests 給 RISC core) 之外，其它三類的 services 是用簡單的 round-robin 方式排程。
6. 所有互動式多媒體和 3-D 視訊的應用程式只要用 Java 編碼就可以，如此可以增加程式的可移植性。
7. 系統可以把一些 3-D Video 的應用程式執行的畫面編碼存下來

整個系統的完整架構如圖 43 所示。其中的關鍵元件如下：

1. 3-D Video 計算加速核心及相關程式庫－由子項目一開發
2. 3-D Graphics/Video 成像引擎及相關程式庫－由子項目二開發
3. 3-D Video 編碼技術及相關程式庫－由子項目三開發
4. 高效率 Java 處理器及 Java 執行環境－由子項目四開發
5. 系統中介軟體及 native-code 排程核心－由總計畫開發

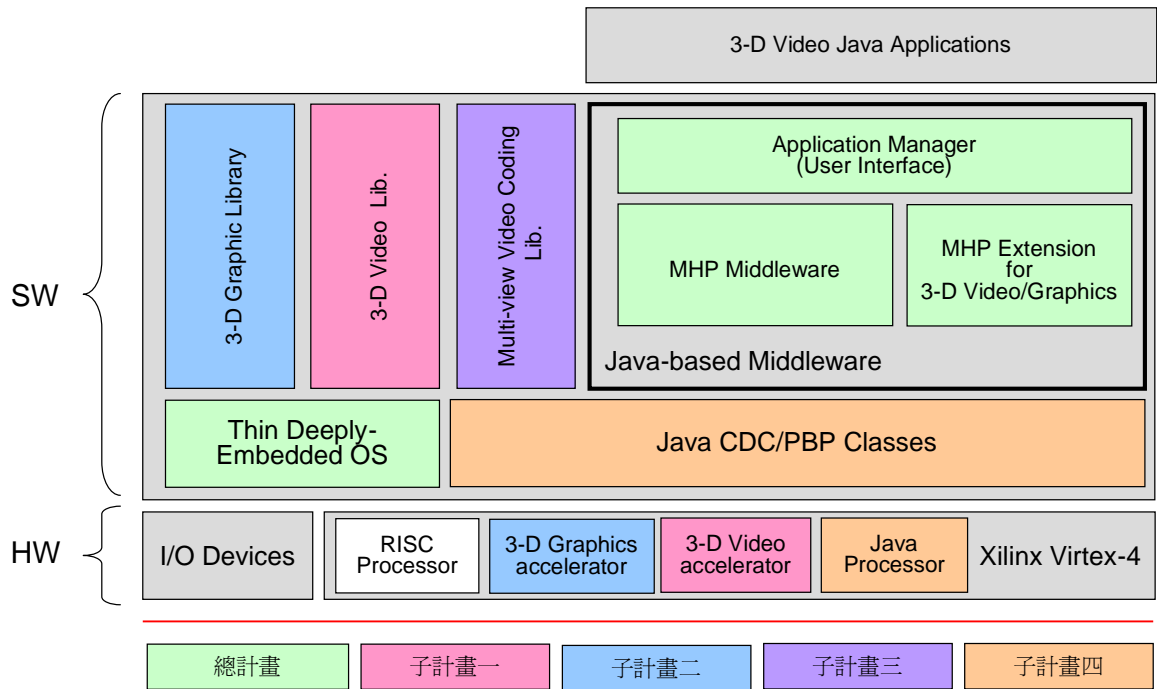


圖43. 計畫預計完成的開放式多媒體平台

以下針對上述系統的每一個子項目的研究方法及成果、以及總計畫系統整合的幾項要點進行進一步分析和說明：

總計畫：支援 3-D 立體視訊的數位電視多媒體平台設計

(Design of a DVB-MHP Platform with 3-D Video Support)

總計畫已整合各子項目所開發的元件成為一個完整的系統。在圖 43，我們已經描述了四個子項目分別的工作，而總計畫這三年中負責溝通協調各子項目的進度及介面，另外也會負責開發系統整合用的作業系統和中介軟體。總計畫選用的整合平台是 Xilinx ML-507。這個硬體平台如圖 44。這個平台可以驗證極為複雜的軟硬體整合設計。其中的核心晶片是 Xilinx Virtex-5 的 Platform FPGA。內含兩顆 PowerPC 處理器的 hard IP cores、56,880 個 FPGA 的 Logic cells(大約相當於 3M system gates)、4Mbits 的 on-chip block RAM。這個平台的一個優點是它在 Virtex-4 晶片內所採用的 on-chip bus 是 IBM 的 CoreConnect，而不是 AMBA，所以我們在設計 IP 時是透過一個 bus wrapper 來和其它的 IP 溝通，只要把這個 bus wrapper 移植到其它的 bus protocol 之上，我們就可以把設計的 IP 移植到別的系統(如 AMBA)。這可以讓我們在設計 IP 時，不至於被單一 BUS protocol 綁死。

目前總計畫以整合所發展各個關件模組，提出一個參考設計平台，並在已在 ML507 上驗證完成。不過受限於發展版的限制(FPGA 大小和處理器速度)我們將一些模組抽出關鍵部分整合。圖 45 是目前在 ML507 上整合的平台架構，藍色為硬體部分，綠色是軟體部分，系統的運行時脈為 100MHz。發展版中的 PowerPC 會和 Java Processor 構成 Java 執行環境，Java 程式可透過開發的關鍵模組支援 3-D 電視系統中各種應用(包含了 3-D 視訊和 3-D 圖學顯示...等)。

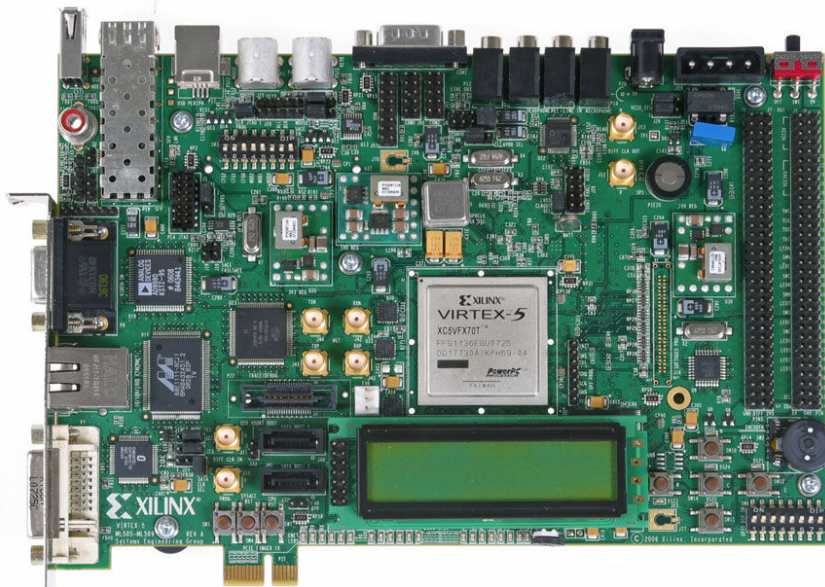


圖44. 總計畫預計採用的整合平台 Xilinx ML-507

在發展版上驗證的過程中，受限發展版資源有限(FPGA 大小、BUS 架構和處理器速度)，一些關鍵模組將部分功能化簡，例如：3-D Graphics 只能將 geometry subsystem 以電路方式整合燒入 FPGA，另一半 rendering subsystem 則以軟體方式呈現在 PowerPC 上執行；另外 3-D video accelerator 也折衷採用化簡後的演算法電路。但這些化簡並不會影像系統整合驗證，並且可以輕易移植到其他設計方法上(例：ASIC)。另外，雖然在整合上有部分化簡，但是相關的模組亦可另外獨立驗證並呈現。

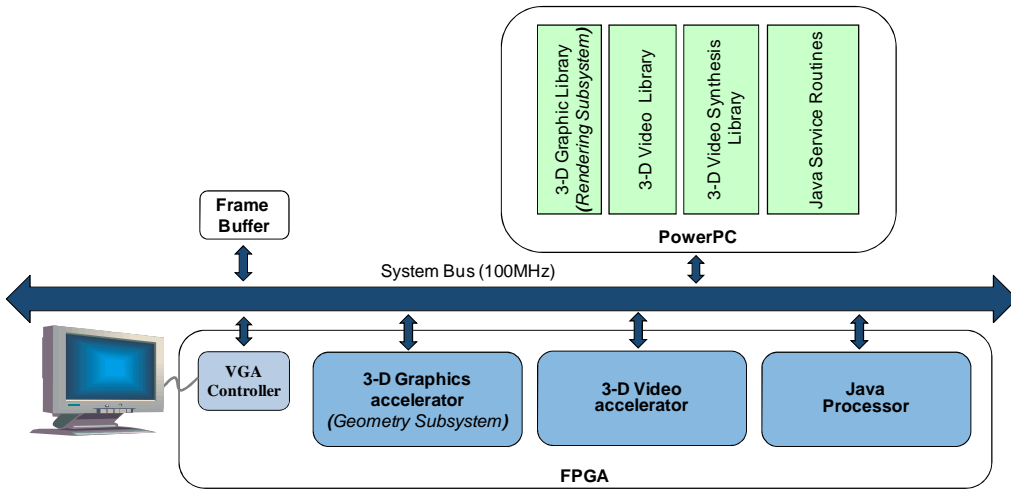


圖45. 總計劃整合平台架構

整合平台的執行範例程式是一隻 JAVA 3-D 人機介面程式，他透過計畫發展的 3-D 函式庫可以畫出 3-D 顯示畫面，如圖 46。圖中下方發展版是 ML507，我們在 Java 平台上(子項目四)執行一個 3-D UI 應用程式，可輸出 3-D 應用畫面如螢幕所示。而畫面中 3-D 效果以紅藍眼鏡呈現。應用中整合了 3-D 視訊應用(子項目一和子項目三)、3-D 圖學應用(子項目二)以及一些 3-D 立體介面按鈕。

這個範例程式展示了一個標準 3-D 電視系統的雛型，所提出的平台提供了一個很好的設計參考給未來產學界相關研究使用。接下來篇幅繼續分述各個關鍵模組的研究方法和成果。

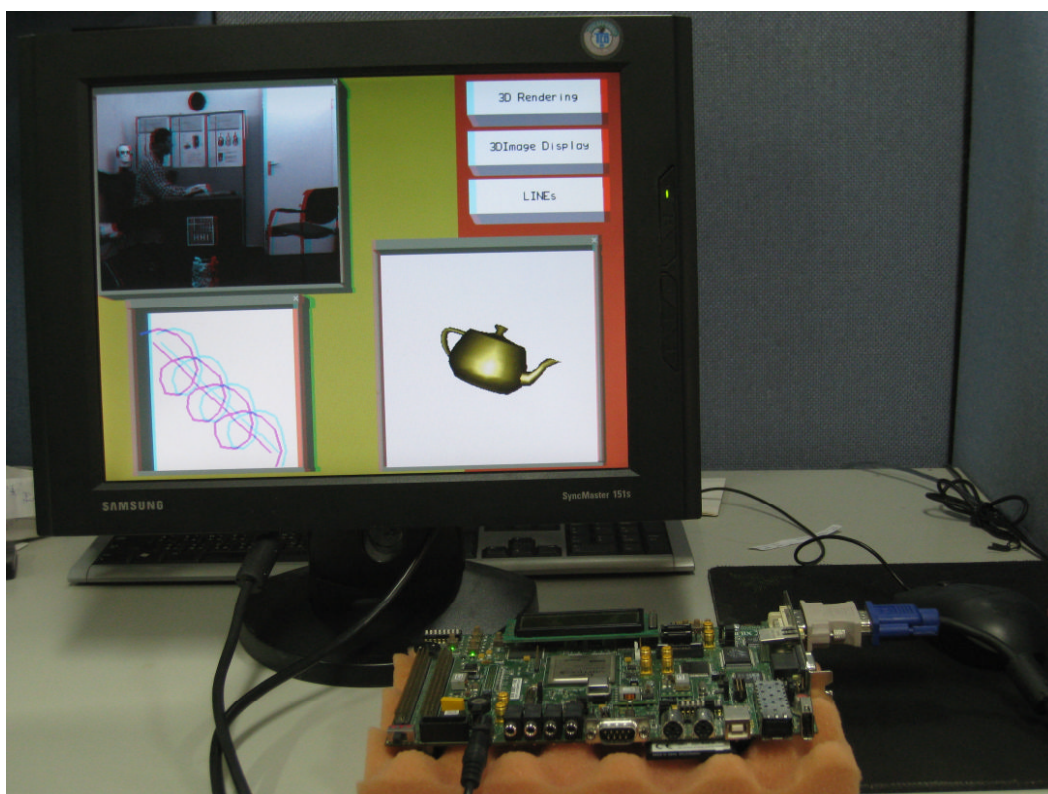


圖46. 總計劃整合 Demo

Table.2. FPGA 使用資源列表

Selected Device : 5vfx70tff1136-1	Integrated System		
Number of Slices:	21139	out of 44800	47%
Number of Slice 6 input LUTs:	31194	out of 44800	69%
Number used as logic:	30803	out of 44800	68%
Number of IOs:	232		
Number of bonded IOBs:	140	out of 640	21%
Number of Block RAM/FIFO	25	out of 148	57%
Number using Block RAM only	57		
Number of PPC440:	1	out of 1	100%
Minimum period:	8.224ns(Max Freq: 121.590MHz)		

子項目一：3-D 視訊計算加速核心之設計 (Design of 3D Video accelerator)

子計畫一已發展出適合硬體實現的半全域視差資料估測演算法。此演算法基於適應性權重計算 (Adaptive Weight) 演算法結合微型普查 (Mini-Census) 的比對方式，以及量子化指數曼哈頓色彩距離 (Quantized Manhattan Color Distance) 等技巧。演算法如圖 47 所示。

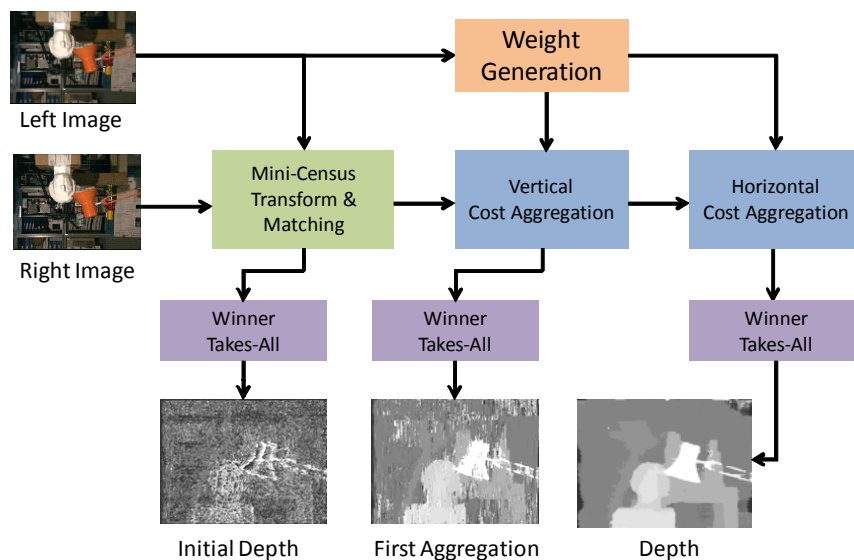


圖47. 視差資料估測演算法

首先，由 Mini-Census 計算兩像素的比對值(matching cost)可減少運算量，從原始的一個方型視窗的運算量轉換為六個點的運算量。其運算如圖 48 所示。Mini-Census 優點在於較原始演算法更能承受影像受光線影響的問題。

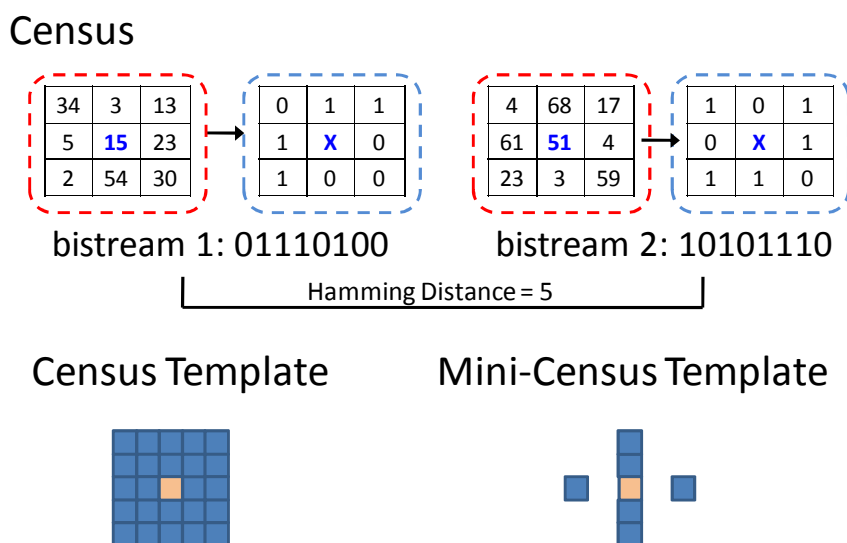


圖48. Mini-Census

接著，以 Weight Generation 以量子化指數曼哈頓色彩距離產生 Weight，此 Weight 用於加權相鄰像素 matching cost 的匯聚。Weight 在影像上的分布如圖 49 所示，Proximity 表示距中心越近的像素點比重越高，Color Similarity 表示與中心顏色越接近得像素點比重越高。此兩項相乘成為 matching cost 匯聚的比重。

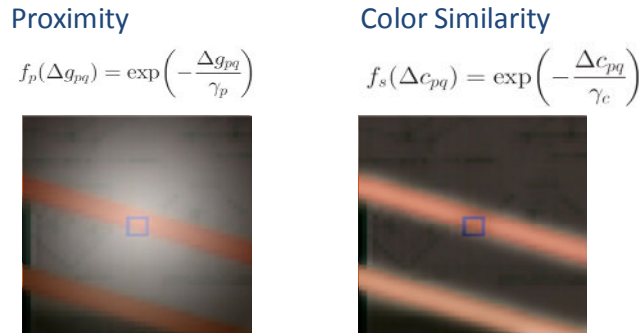


圖49. Cost Aggregation

為減少運算量，我們分析 Weight 有無 Proximity 在視差圖準確度上的變化。圖 50 顯示對於有 Proximity 的視差圖而言，在 Mask size 大於 39 時較無改變。並且，對於 Mask size 在 39 以內，無 Proximity 與有 Proximity 的視差準確度是相同的。因此我們可以省略 Proximity 於 Weight 中，並設定 Mask size 於 39 以內，即可達到相同於包含 Proximity 的效果。

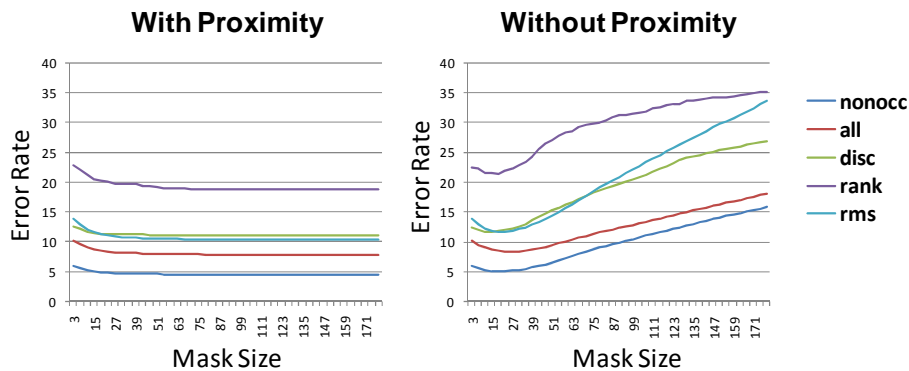


圖50. 比較有 Proximity 與無 Proximity 在視差圖準確度的差別

為提升 Weight Generation 的運算速度，我們在匯聚 matching cost 時分離原始運算成垂直匯聚 (Vertical Cost Aggregation) 與水平匯聚 (Horizontal Cost Aggregation) 兩步驟。利用以上所提之方法，可減少原始演算法運算量的 64.2%。另外，我們化簡 Color Similarity 的 exp 運算為 2 幕次的查表運算，各種運算化簡的方式如圖 51 所示。最後我們選擇圖中 x64, Quantize P1-bit 的簡化方式。

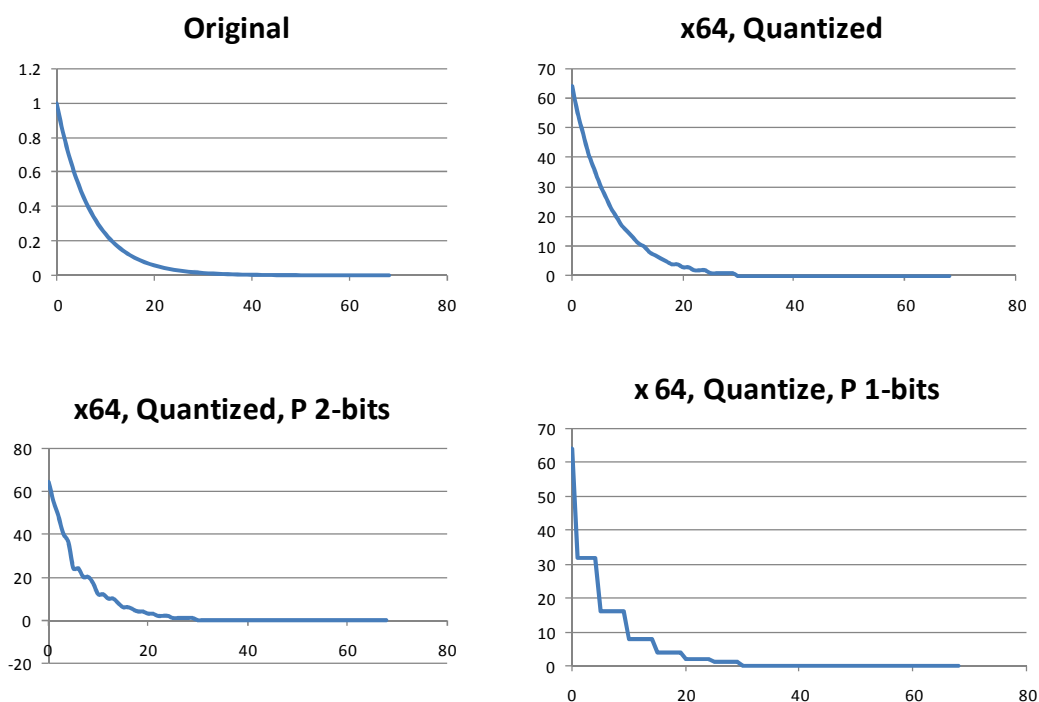


圖51. 比較 Color Similarity 的運算化簡

總和以上改變演算法以利於硬體設計的技巧，在個人電腦運算時間與視差圖準確度如下表所示。對於準確度約降低 1.2% 以內，而運算時間大幅降低至原始運算的 1.9%。如此的硬體導向演算法將有利於進一步的硬體架構設計。

Method	Error Rate %				Exec. Time(sec)
	TSUKUBA	VENUS	TEDDY	CONES	
Original	1.85	1.19	13.3	9.79	95.65
+MC+2P	3.47	0.91	14.3	11.2	4.75
+MC+2P+ Manhattan	3.08	0.59	14	10.1	3.12
+MC+2P+ Manhattan +Truc(64,2)	3.03	0.61	14	10.1	2.52
+MC+2P+ Manhattan+Truc(64,1)	3.06	0.66	13.9	10.1	1.84

在硬體架構設計方面，我們針對 Mini-Census 及 Cost Aggregation 的運算分析各種資料對記憶體存取的方法。在 Mini-Census 有視差優先再利用 (Disparity-Order Reuse) 以及像素優先再利用 (Pixel-Order Reuse) 之方法。在 Cost Aggregation 有列局部再利用 (Partial Column Reuse) 以及行垂直延伸再利用 (Vertically Expanded Row Reuse)。考慮運算量、記憶體用量以及頻寬用量，最後採用 Disparity-Order Reuse、Pixel-Order Reuse 與

Vertically Expanded Row Reuse 方法於硬體實現。硬體架構如圖 52 所示。此架構所有運算皆以握手機制(Handshaking)讀寫資料，以達到各運算處理時間的彈性。

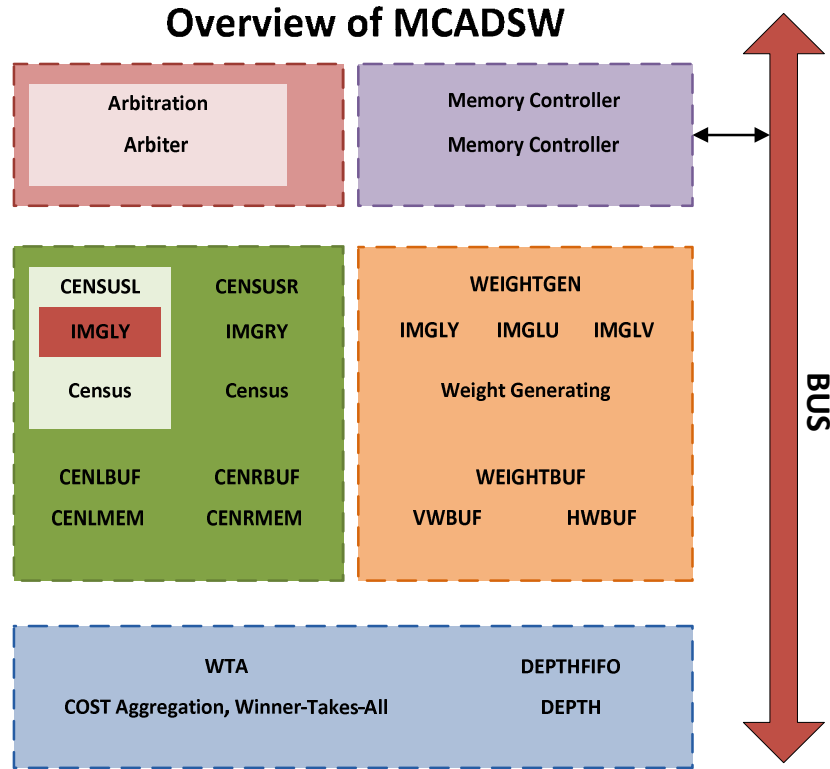


圖52. 視差資料估測架構圖

本硬體架構在聯華電子 90 奈米製程可工作於 100MHz 且 bus 寬度為 32-bit，可處理 CIF(352x288)影像大小且 64 視差範圍(disparity range)達到 42 fps。下表為合成的記憶體用量與組合電路的邏輯閘數量。總內部記憶體用量約 21k 位元組，總組合電路用量約 562k 邏輯閘。如此硬體效能已達到子計畫一的目標。

Performance under UMC 90nm Technology	
Clock rate	100 MHz
External bus width	32 bit
Image size	352x288
Disparity level	64
Logic	562k equivalent gate counts
Internal memory	21k bytes
FPS	42

對於演算法之效能，以下比較本演算法與其他文獻視差圖的錯誤率以及運算速度。我們的演算法可達高準確度且即時運算速度。

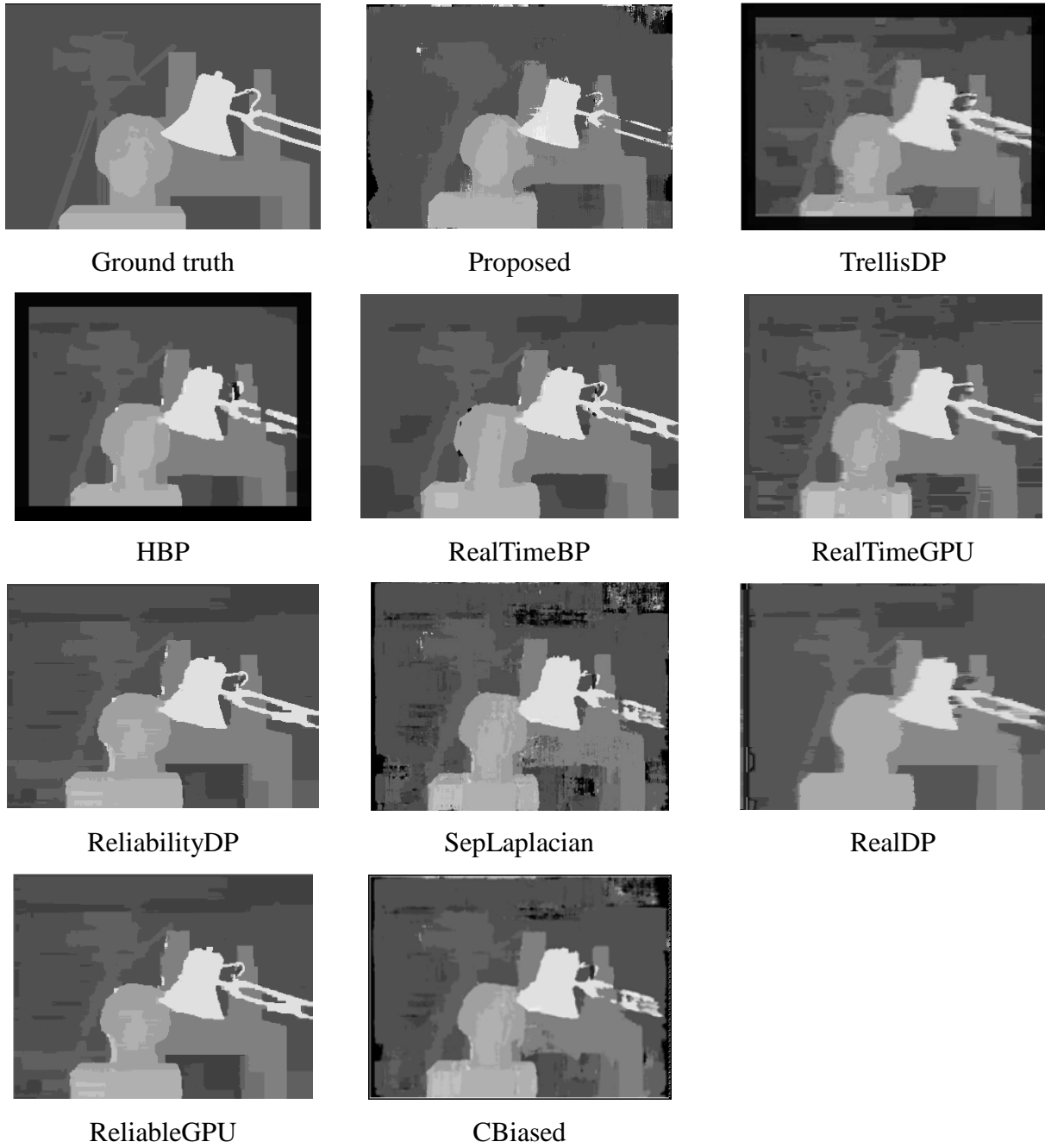


圖53. 視差圖比較

Design	Category	MDE/s	TSU	VEN	TED	CON	SAW	MAP
Proposed	Hardware	272.5	2.80	0.64	13.7	10.1	2.11	3.21
TrellisDP [39]	Hardware	294	2.63	3.44	-	-	1.88	0.91
HBP [80]	Hardware	73.7	2.85	1.92	-	-	6.25	6.45
EffectAggr [40]	CPU	18.9	2.96	3.53	10.7	4.92	-	-
RealDP [41]	CPU	209	2.85	6.42	-	-	6.25	6.45
Cbiased [42]	CPU+GPU	605	4.77	10.2	-	-	0.82	0.65
SepLaplacian [43]	CPU+GPU	679	13.0	-	-	-	-	-
RealTimeBP [44]	CPU+GPU	19.6	3.40	1.90	13.2	11.6	-	-
RealTimeGPU [45]	CPU+GPU	19.6	4.22	2.98	14.4	13.7	-	-
ReliableGPU [46]	CPU+GPU	-	1.36	1.09	-	-	2.35	0.55
GradientGuided [47]	CPU+GPU	117	2.48	3.91	-	-	1.63	0.73

針對全域估測演算法，本子計畫分別開發了適於硬體設計 BP 演算法及 DP 演算法。針對 BP 演算法，本子計畫提出低記憶體用量的運算元(PE)架構，其架構如圖 54 所示。另外，本子計畫提出低記憶體用量的資料配置方法，其方法如圖 55 所示。

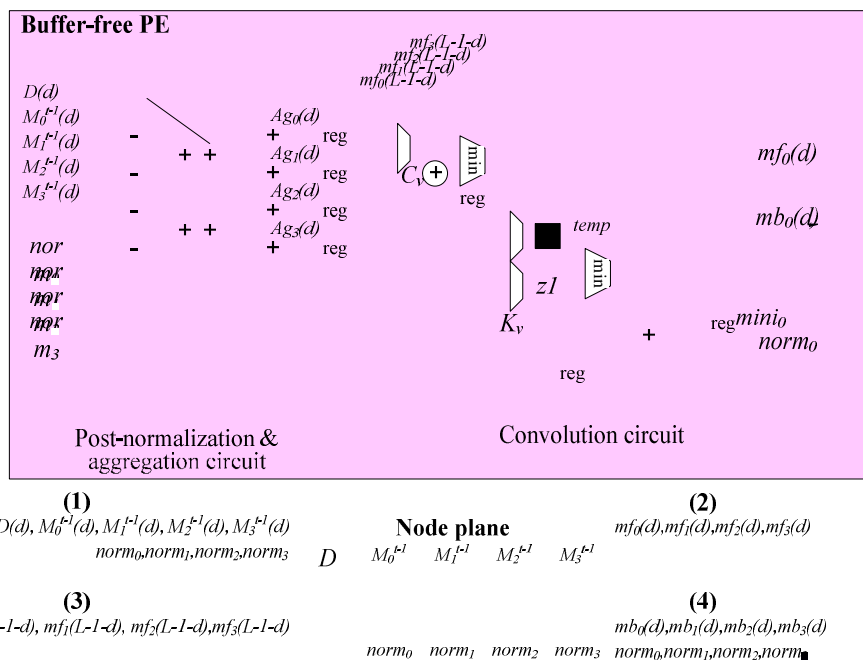


圖 54. Buffer-free PE 架構

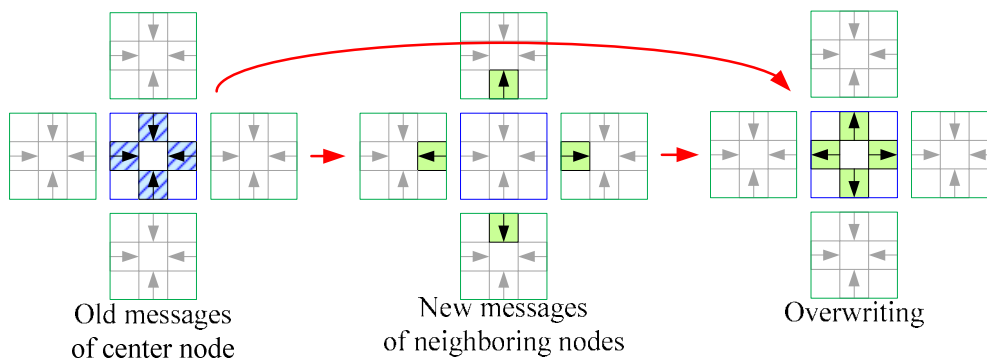


圖 55. Spinning message 記憶體配置方法

所提出之硬體架構可適用於現存的各種 BP 演算法，並且達到低記憶體用量與即時處理的運算速度，其硬體電路效能如下表所示。然而，該硬體設計的整體成本仍過高，導致無法整合於總計畫之目標平台，故最終整合平台的演算法並非採用 BP。

	Baseline BP [67]	Hierarchical BP [81]	Block-based BP [39]	Tile-based BP [49]
Iteration T	30	5, 5, 10, 5	30	inner=8, outer=2
Required Throughput (Node/Frame)	4,608,000	1,212,000	4,608,000	4,915,200
Operating Frequency (MHz)	285	285	285	285
Number of PE	33	9	32	32
Gate Count (K)	273.9	74.7	265.6	265.6
Size of Sliding-Bipartite Node Plane	30x480 (image-scale)	5x480 (image-scale)	30x32 (block-scale)	8x32 (block-scale)
Memory Cost of Messages and Data costs (KB)	2,793	465	186	49
FPS	30.01	31.12	29.11	27.29

本子計畫亦針對 DP 演算法改善其高記憶體用量的問題，使得 global approach 演算法可實現於總計畫的 FPGA 整合平台。原始的 DP 演算法包含兩主要步驟:forward cost accumulation 與 backward path tracing。由於兩步驟的運算流程，必須存取一整列的資料，導致記憶體用量過高。因此，本子計畫提出單步驟的演算法，將原始的兩步驟合併，直接在第一步驟中就決定 path，其流程圖如圖 56 所示。。由於大幅降低記憶體用量，DP 演算法可順利地實現於總計畫 FPGA 的整合平台。圖 57 及圖 58 分別為本子計畫所實現的 DP 演算法之 ASIC 與 FPGA 的硬體效能。

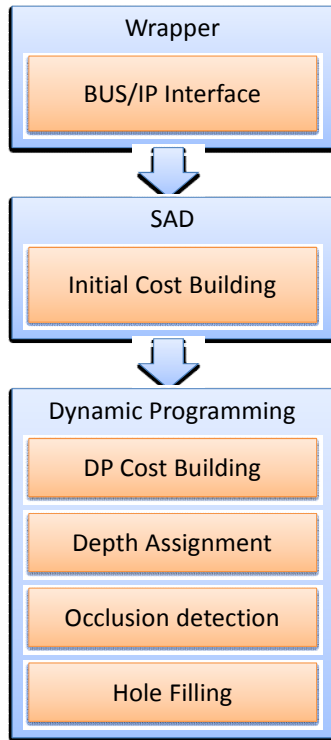


圖56. DP 演算法

Performance under UMC 90nm Technology	
Clock rate	100 MHz
External bus width	32 bit
Image size	352x288
Disparity level	32
Logic	91k equivalent gate counts
FPS	46

圖57. DP 演算法之 ASIC 硬體設計效能

Item	Sub1		System	
Number of Slice Registers	4644 out of 44800	10%	21139 out of 44800	47%
Number of Slice LUTs	15703 out of 44800	35%	31194 out of 44800	69%
Number of LUT Flip Flop pairs used	15830		41322	
Number of IOs	795		232	
Number of BUFG/BUFGCTRLs	1 out of 32	3%	10 out of 32	31%
Frequency(max)	154MHZ		100MHZ	

圖58. DP 演算法之 FPGA 硬體設計效能

子項目二：可重組之 3D 繪圖加速器設計 (Reconfigurable 3D Graphics Accelerator)

本子項目二之研究重點為可重組之三維繪圖加速器設計如圖 59 所示。因此我們分別在前端幾何轉換子系統中的打光運算單元以及後端繪圖子系統中的三角形設定引擎與深度壓縮機制提出可重組式的演算法與對應的硬體架構，根據不同的輸入規格重組不同的硬體架構，達到系統的最佳效能。同時也已各別完成前後兩級的基本的硬體電路設計與 FPGA 模擬驗證平台，另外也成功將前後兩級整合在一起，並已在 FPGA Xilinx ML507 平台模擬驗證完成。

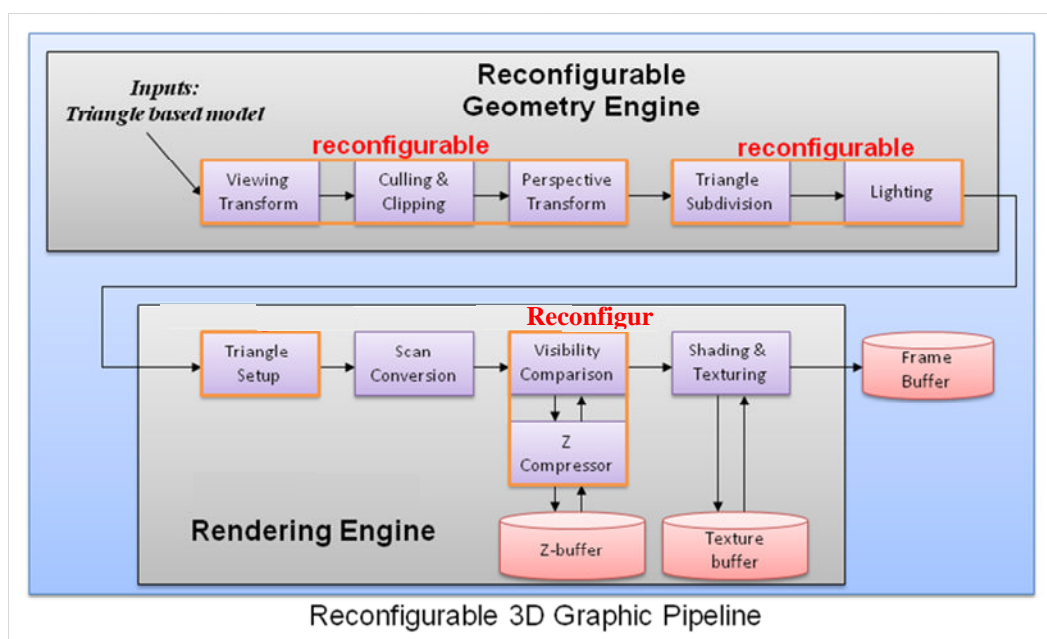


圖59. 重組式 3D 繪圖系統架構圖

本子計畫我們將各子區塊的實作情形以及研究成果分項列出，主要分為幾何轉換子系統與著色子系統的設計與硬體實現等，最後總結整個子計畫的實作成果。

幾何轉換子系統(Geometry Engine)

在幾何轉換子系統部分，主要是打光器的演算法與硬體實現，以下分別深入探討其實際設計的理念。

● *Approximating Phong Shading*

本子計畫在打光器設計上主要有三項突破點以及實現結果，分別是：三角型切割之演算法與其硬體設計與實現、頂點變數共用機制、邊函數修正機制。以下分別討論之。

三角形切割演算法：圖 60 和圖 61 分別是兩種主流打光演算法及 Subdivision Based Shading 的結果，由這三張圖得比較後可知 Subdivision Based Shading 與 Phong Shading 的繪圖結果相當接近，而兩者間運算量的比較可參考圖 62，圖中黑色的部分表示須作打光運算的像素，由圖可知在打光效果相近的情形下，切割式著色演算法大約可以省下近 50% 的打光運算。

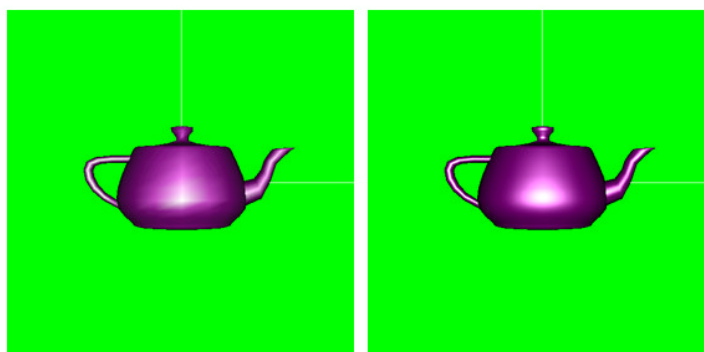


圖60. 左)Gouraud shading 的茶壺打光 右)Phong shading 的茶壺打光

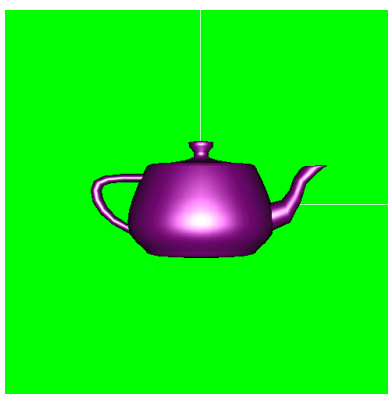


圖61. Subdivision Based Shading 的茶壺打光

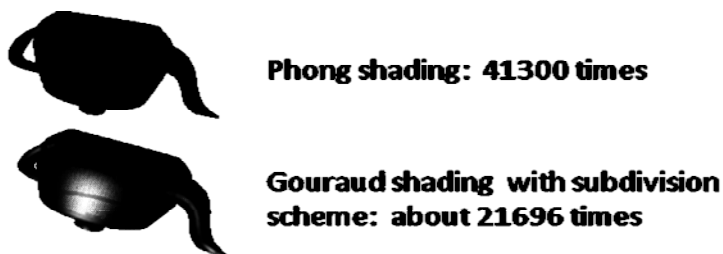


圖62. Phong shading 和 Subdivision Based Shading 運算次數比較

切割器是切割式著色法的核心，而切割器不適合在軟體中實作的其中一個原因是切割時會產生新的頂點，若切割的動作是在處理器中完成，則切割出來的新頂點需要花費額外的頻寬將它們送到繪圖硬體上；反之若切

割的動作是在繪圖硬體上完成則新頂點就不需要在匯流排上傳送，如此就可省下許多的頻寬而使繪圖效能提升。傳統的切割演算法使用遞迴的方式，但遞迴演算法不適合在硬體上實作，主要的原因在於堆疊的支援與切割出頂點的管理。由於遞迴式演算法會在共用邊上切割出相同的頂點，管理這些頂點所需的資料結構在硬體上實現相當複雜。但如果不使用管理頂點的機制而直接將產生的任何頂點直接送到打光單元中，就會對相同的頂點重複打光。打光的運算是相當耗時的，所以我們希望盡量避免重複的打光動作，為了兼顧硬體複雜度與效能我們使用了漸增式(Incremental)切割的方式來實作切割器。圖 63 說明漸進式切割的概念，要將一個三角形切成四個小三角形需要計算出在各邊上中點的頂點，而 dy 與 dx 可以事先計算，透過這種逐一加上一個向量的方式就可以循序地產生所有的頂點。由於頂點是依序被產生因此可以直接送到打光單元進行打光而不需要擔心重覆頂點的問題，因此也不需要額外的頂點管理機制。漸增式切割可以很容易的進行任意數量的切割運算，如圖 64 所示，只要計算出需要的 dx 與 dy 向量並透過漸增的方式即可產生所有需要的頂點。

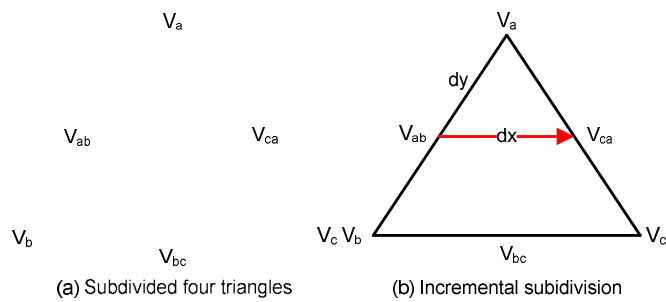


圖 63. 漸增式切割演算法

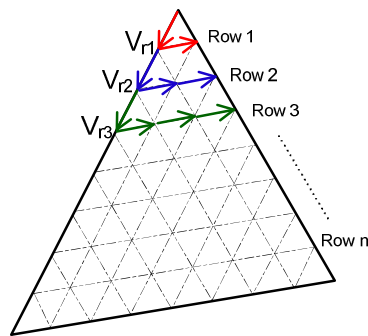


圖 64. 任意數量的漸增式切割

頂點變數共用機制:而經由切割器切割後產生的許多小三角形無疑的會對效能造成衝擊，頂點數量增加意味著需要更多的幾何轉換，三角形的數量增加也代表三角形準備的運算和後端著色子系統所有的效能負擔的增加，因此前後端如何互相配合才能不影響系統效能也成為我們的研究重

點。在後端我們透過了係數共用的機制便可以減少一半以上的三角形準備運算，如圖 65 所示，共用的係數一旦被計算出來便可以提供給所有切割出來的三角形使用而不需要重新計算這些係數。Table.3 整理了變數共用機制前後的運算量比較，可以看出使用這套方法確實能有效的降低運算複雜度進而減少切割式渲染演算法對系統效能的影響。

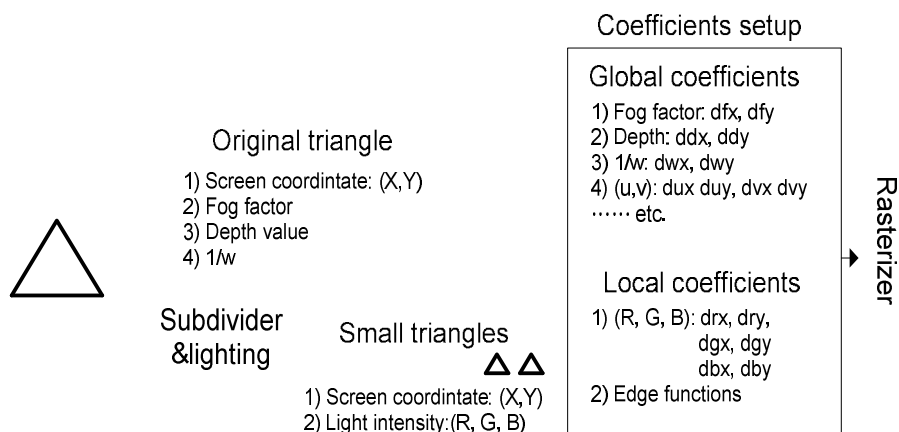


圖65. 係數共用機制圖

Table.3. 變數共用機制前後運算量之比較表

	Conventional subdivision algorithm	Proposed subdivision algorithm	Complexity reduction in percent
Number of the lighting operations	24	15	37.5%
Number of the 4x4 matrix multiplications for perspective transformation	24	3	87.5%
Number of the clipping/culling test operations	16	1	93.75%
Number of the 3x3 matrix multiplications for setup operation for rasterization	80	27	66.25%
Rasterization anomalies	Sometimes occur	Completely eliminated	

邊函數修正機制：第二個問題來自於切割過程中產生的誤差，如圖 66 所示，原本在共用邊上的頂點應該要在相同的位置上，但是切割時的誤差使得最後的位置產生落差，因而造成像素化異常的現象，如圖 66 之上所示，我們在畫面上可以看到物體上因為像素化異常而產生破洞。為了處理這個問題，我們也提出以原本三角型的頂點取代部分切割後的頂點的方法，成功的解決了這個問題，如圖 67 之下所示所有的破洞已經完全消失。

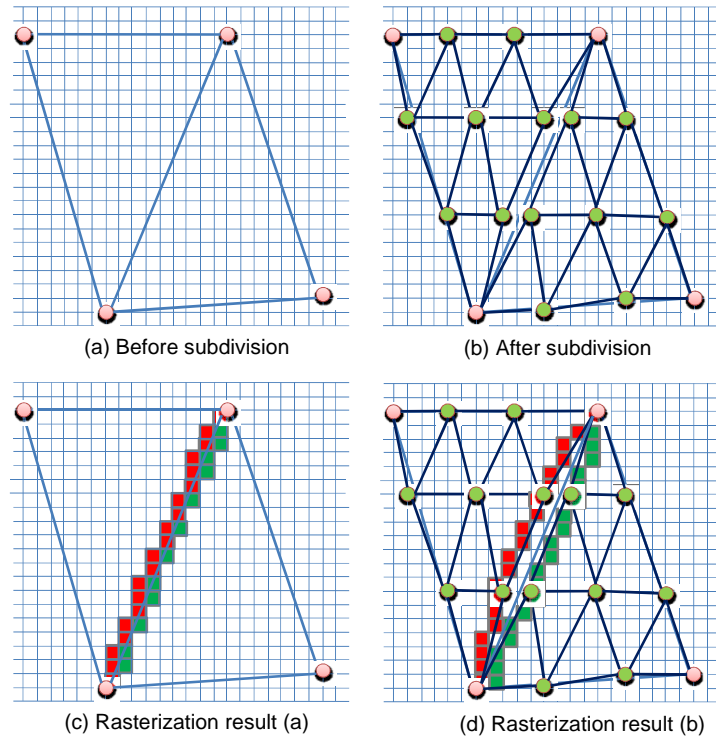


圖66. 破洞圖解說明

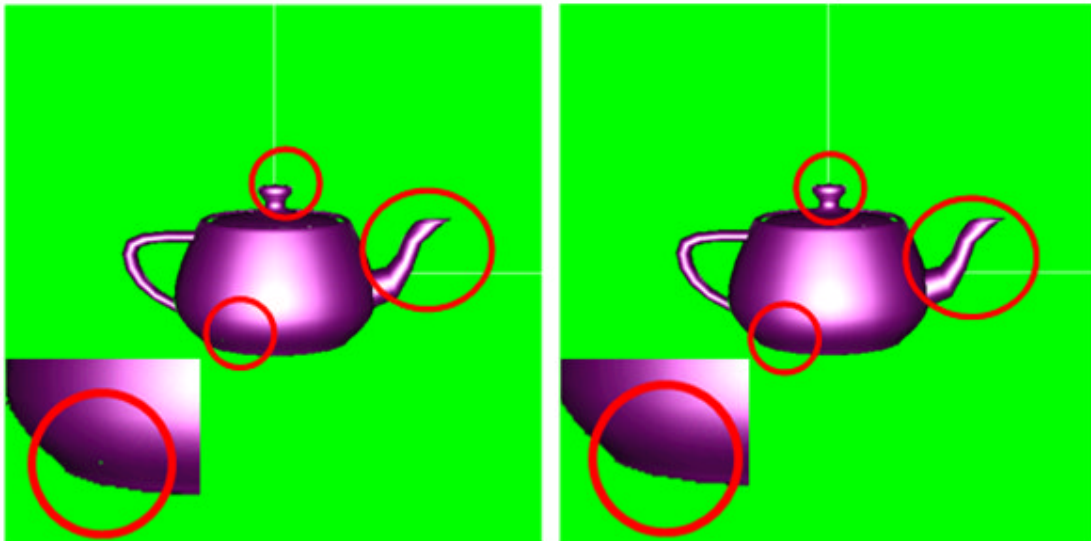


圖67. 消除破洞模擬結果

著色子系統(Rendering Engine)

在著色子系統部分，其系統流程圖如下圖 68，總共可以區分成三塊主要子系統，包含 Triangle Setup Engine 負責將幾何轉換子系統的 Triangle-level 轉換成後級的 Pixel-level；Depth Testing Unit 負責測試每個像素(pixel)的深度值，決定該像素是否需要顯示在螢幕上；Texture Mapping Unit 負責替每個像素套上該場景下的色彩。接下來會分別深入探討三個子

系統的實際設計理念。

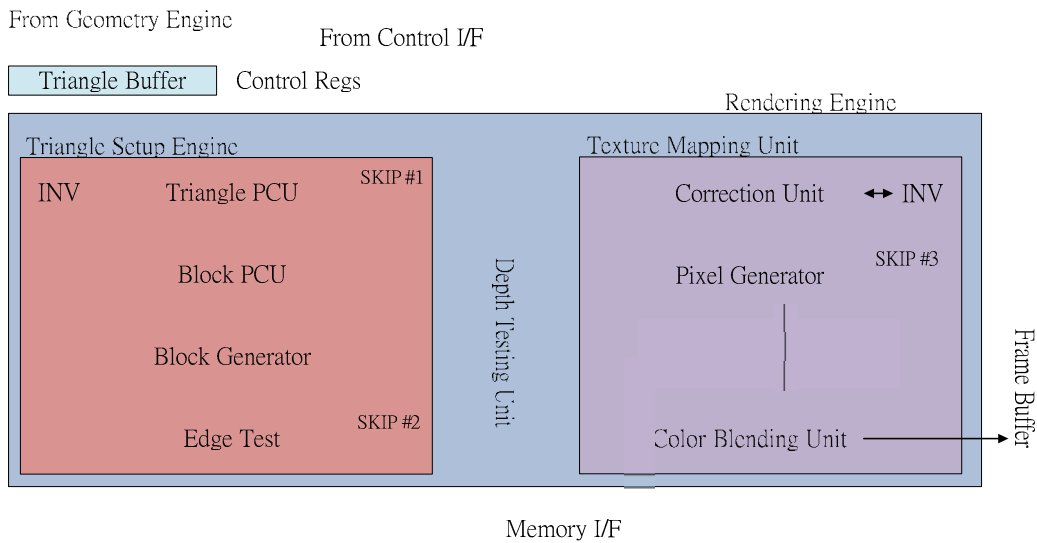


圖 68. 圖 11：著色子系統的細部流程圖

Triangle Setup Engine (TSE)：主要工作在於將前級的 Triangle-level 轉變成後級的 Pixel-level。首先 TSE 會接收來自前級傳送過來的三角形三個頂點(Vertex)資訊及控制訊號，然後透過 Triangle PCU 利用內插法的方式，內插出三角形內部所有點的資料，接著利用 Block PCU 及 Block Generator 的技巧，對整個 Frame Buffer 進行掃描工作，最後再經由 Edge Test 檢測 Block 中是否存在任何物件的 Pixel 資訊，以減少後半部原件多餘的資料量，整個流程結束後，會將每個像素資料傳給深度測試原件(Depth Testing Unit)。另外，TSE 中的 SKIP #1 與 SKIP#2 分別將太小的三角形清除與空的 Block 清除。

可重組深度緩衝區壓縮器：目前此子項的硬體設計所提供使用者的壓縮演算法共有 Modified HA, Modified DDPCM, General DDPCM, one/two plane mode... 等等數種主流的壓縮法，並可自動判斷輸入的 3D 場景適合何種壓縮法。經由模擬的驗證已經確定我們所提出的演算法能夠有效提升平均壓縮率，並改善約 38.7% 至 76.9%，目前已經完成硬體實作與驗證。而圖 69 (a)藍色曲線為適應性壓縮演算法的壓縮比，紅色曲線則為目前被廣泛運用之 DDPCM 深度壓縮演算法之壓縮比；(b)同樣場景下適應性壓縮演算法(藍色)和另一種廣泛運用之 HA 深度壓縮演算法之壓縮比比較。三種演算法(proposed：DDPCM：HA)壓縮比比較為 1.91：1.37：1.09。

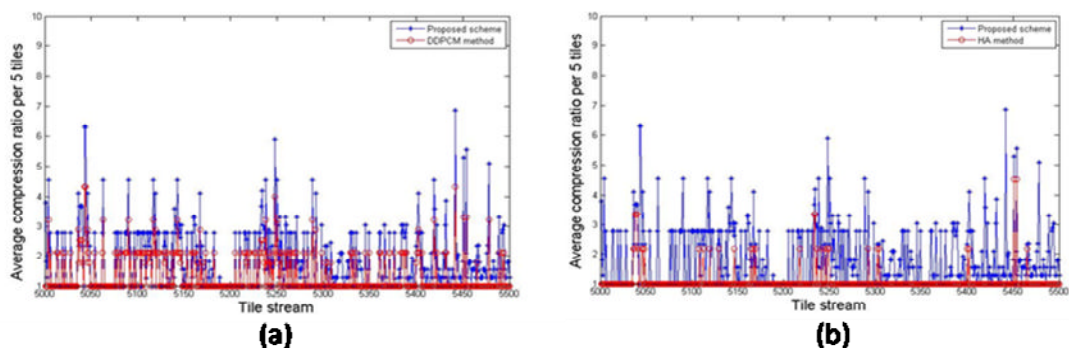


圖69. 常用演算法和 proposed 演算法之壓縮比之比較

Texture Mapping Unit : Texture Mapping Unit 就是傳統的 Pixel Processor，主要在於計算每個像素的色彩資訊，包含了校正機制(Correction Unit)、像素產生器(Pixel Generator)和色彩混和器(Color Blending Unit)。校正機制接收來自 TSE 的資料，進行校正的資料處理，而像素產生器則接收來自深度測試原件的資料，產生相對應的像素資料，最後再利用色彩混和器將色彩做混和的動作，並把結果傳送到 Frame Buffer。

Xilinx ML507 驗證

Table. 4 為 3D 管線前後兩個子系統硬體化與整合後所使用的 FPGA 資源。

Table.4. 3D 繪圖系統所占 FPGA(ML507)資源統計表

Device : vertex-5 5vfx70tff1136-1		
Resource	3D Graphic Pipeline(GE+RE)	%
Slice Logic Utilization		
Number of Slices Register	18536	41%
Number of Slices LUTs	13622	30%
Number used as Logic	13574	30%
I/O Utilization		
Number of IOs	2515	-
Number of bonded IOBs	0	-
Specific Feature Utilization		
Number of BUFG/BUFGCTRLs	2	6%
Number of DSP48E	11	8%

而目前我們使用的平台和硬體架構為 Xilinx ML507 開發平台，其系統架構如圖 70。由 FPGA 中的 PowerPC processor 啟動 FPGA 中的三維繪圖加速器，其中的幾何轉換子系統抓取記憶體中的頂點和顏色資料運算完後，會把處理後的三角形資料直接傳至著色子系統，而著色子系統掃描產

生像素與深度測試之後，將處理後的像素資料寫入畫面緩衝區，在完成所有資料的色彩處理之後，通知 PowerPC 藉由 VGA Controller 執行影像顯示之作業。

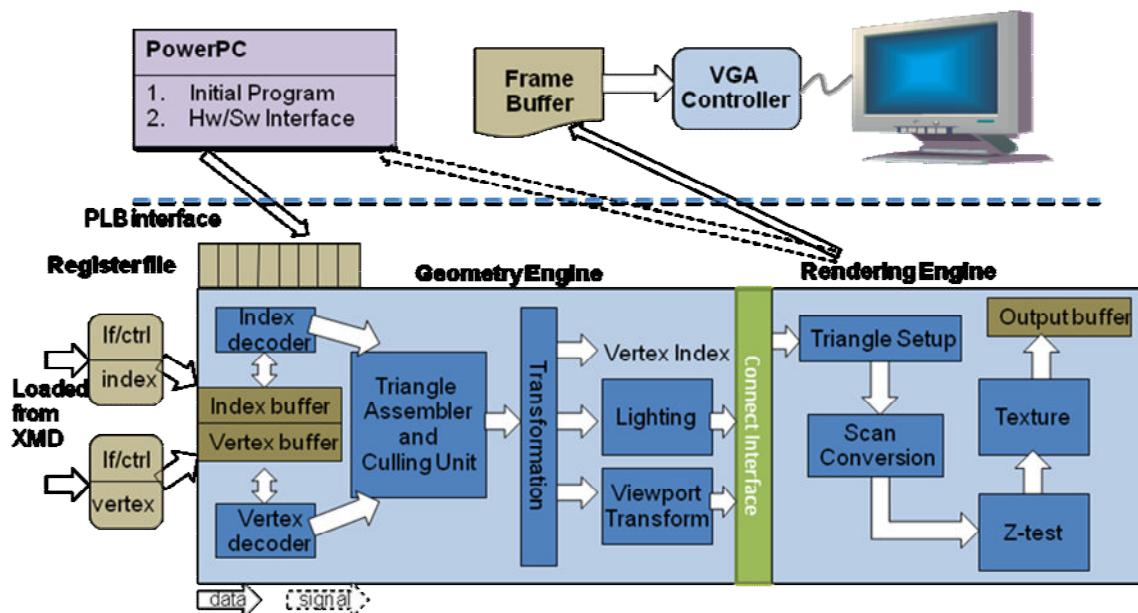


圖70. 圖 12：3D 繪圖管線系統架構流程圖

子項目三：MVC 編碼系統中介軟體

本計畫將分析多視角視訊編解碼標準，改善現有的編碼工具並提出新的編碼工具藉以實現可任意視角觀賞之高效能多視角編解碼標準。研究方向包含(1)多視角視訊編碼演算法效率改善，(2)視角合成 (View Synthesis) 技術開發，與(3)中介軟體定義與開發。以下說明本計畫研究方向之背景與目的。以下列出詳細的研究方法與步驟。

深度圖(Depth map)經過現行視訊壓縮標準的編解碼後，將造成解碼重建後深度圖有不同程度的失真。這些失真可能會影響到最終虛擬影像的合成品質，例如在物體邊界的深度值發生錯誤時，顯然將直接造成主、客觀影像品質大幅度的降低。採用現行最新的視訊壓縮標準 MVC 對原始深度圖 (圖 71) 進行壓縮後，其重建畫面(圖 72)於主觀視覺上出現明顯的落差，例如圖 71 中，人背上的深度影像產生明顯的區塊效應(Blocking Effect)。再者，利用 MVC 壓縮具單純內容的深度圖，相較於簡單的 All Intra 模式，需要花費鉅額的運算代價來換取微薄的深度圖壓縮效能。如圖 73 顯示，在固定合成品質下，開啟 Inter 模式雖可為深度壓縮帶來 48%

的 Bitrate Saving，但對整體的 Bitrate 來說，僅只省下 4%，然而複雜度卻因此上升近 16 倍。

目前 MPEG 標準制定小組已出現兩篇於接收端修正深度圖的演算法：第一篇由 Tanimoto 提出，其利用合成相鄰參照視角時的合成誤差，以 Linear Prediction 的方式，推測出虛擬視角可能的合成誤差，最後將此預測出來的合成誤差補回虛擬視角之合成影像；第二篇由 Sung 提出，其利用合成虛擬視角時，來自不同參照畫面的亮度值與深度值的差值，若此差值越過某個門檻值，則判定此深度值需要被修正；其次，此方法為每個 Connected Component 找出一個最佳的深度值偏量，使得合成誤差可以降到最低。然而，上述方法未考慮深度值壓縮的情況，兩者皆無法根據不同的壓縮品質進行調整，造成修改的效能嚴重受到壓縮之品質所影響。



圖71. 原始深度圖.

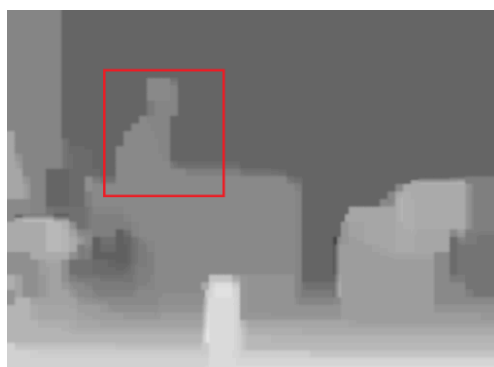


圖72. 解碼後之深度圖.

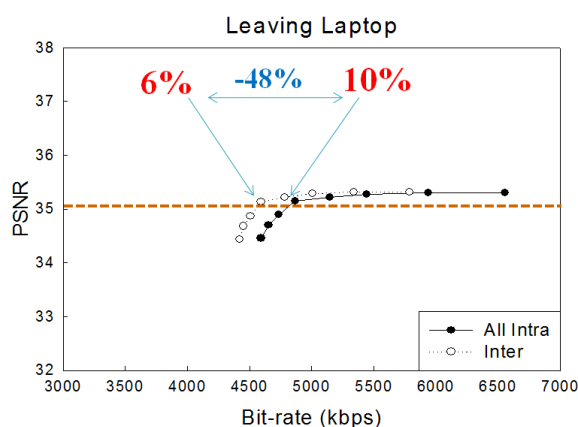


圖73. Current approach applies complex coding scheme for intensity to depth, while it only forms a small portion of the entire bit-stream.

因此，本子項針對重建後的深度圖對虛擬影像合成的影響，提出一個單一像素之合成誤差的檢測模型。透過此模型之分析，可有效預測出每個像素可能造成的合成誤差。利用不同參照視角(Reference Views)畫面之間對應點關係，找出隱藏的深度資訊，並依此修正不可靠的深度像素

(Unreliable Depth Pixel)，使其在虛擬影像合成上可得到較好的品質。其次，本子項提出低演算複雜度之一維深度圖 CODEC，其根據深度圖之特性，利用眾數挑選演算法擷取出重要之深度資訊，並使用線性權重過濾器修補重建之深度圖。

單一像素之合成誤差的檢測模型

為修正深度圖經影像壓縮而產生的失真誤差，本子項設計一個單一像素之合成誤差的檢測模型，於理論上分析虛擬影像合成時，所有可能影合成品質的因素。根據 Depth-Image-Based Rendering(DIBR)描述參照視角和虛擬視角的對應點關係，如公式(1)：

$$\Psi: \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \mapsto \begin{bmatrix} \mathbf{p}' \\ 1 \end{bmatrix} = \mathbf{A}'\mathbf{R}\mathbf{A}^{-1} \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} + \frac{1}{Z_p} \mathbf{A}'\mathbf{T} \quad (1)$$

其中 \mathbf{p} 和 \mathbf{p}' 分別為參照視角和虛擬視角的對應點座標， \mathbf{R} 和 \mathbf{T} 為虛擬相機的旋轉矩陣與位移向量， \mathbf{A} 和 \mathbf{A}' 分別為參照相機和虛擬相機的相機內部參數，本文以 $\Psi(\mathbf{p}; Z_p)$ 簡化表示公式(1)之運算子。在深度圖沒有失真的

情況下，可確保 $\mathbf{p}' = \Psi(\mathbf{p}; Z_p)$ ，且亮度值相同 $I_T(\mathbf{p}') = I_R(\mathbf{p})$ ；若深度圖有

失真(Z_p 加上雜訊， $\bar{Z}_p = Z_p + n_p$)，以 $\mathbf{q}' = \Psi(\mathbf{p}; \bar{Z}_p)$ 表示，則合成誤差 ξ_p 可由泰勒展開式逼近：

$$\begin{aligned} \xi_p &= (I_R(\mathbf{p}) - I_T(\mathbf{q}'))^2 = (I_R(\mathbf{p}) - I_R(\mathbf{q}))^2 \\ &\approx (I_R(\mathbf{p}) - I_R(\mathbf{p}) - \nabla I_R(\mathbf{p}) \cdot (\mathbf{q} - \mathbf{p}))^2 \\ &= (-\nabla I_R(\mathbf{p}) \cdot (\mathbf{q} - \mathbf{p}))^2 \end{aligned} \quad (2)$$

其中 $\mathbf{q} = \Psi^{-1}(\mathbf{q}'; Z_q)$ ， $\nabla I_R(\mathbf{p})$ 為 \mathbf{p} 的 Gradient。利用 \mathbf{p} 、 \mathbf{q} 和 \mathbf{q}' 三點的關係

$\mathbf{q}' = \Psi(\mathbf{q}; Z_q) = \Psi(\mathbf{p}; Z_p + n_p)$ ，可將 $(\mathbf{q} - \mathbf{p})$ 可表示為公式(3)：

$$(\mathbf{q} - \mathbf{p}) = \frac{-n_p}{Z_q(Z_p + n_p)} \mathbf{c} \quad (3)$$

其中 $\mathbf{c} = [\mathbf{I}_2 \quad \mathbf{0}_{2 \times 1}] \mathbf{A}\mathbf{R}^{-1}\mathbf{T}$ 。將 $(\mathbf{q} - \mathbf{p})$ 帶入公式(2)，並導出下式：

$$\xi_p \approx \left(\frac{n_p}{Z_q(Z_p + n_p)} \nabla I_R(\mathbf{p}) \cdot \mathbf{c} \right)^2 \quad (4)$$

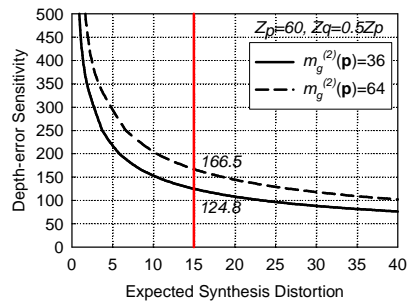
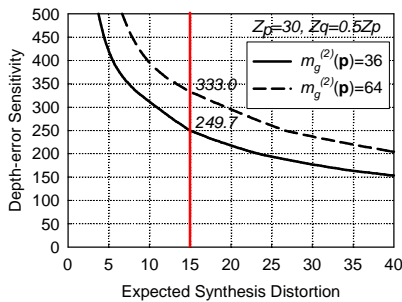
在 MPEG FTV 的水平相機設置的限定下，位移矩陣只含水平方向之分量，因此可將公式(4)簡化為下式：

$$\xi_p \approx \left(\frac{n_p}{Z_q(Z_p + n_p)} \right)^2 \times g_x^2(\mathbf{p}) \times c^2 \quad (5)$$

(1)其中 $g_x^2(\mathbf{p})$ 為 $\nabla I_R(\mathbf{p})$ 的水平分量。因此，假設有真實(Ground-truth)深度資訊的情況下，計算出合成影像誤差的條件期望值：

$$\begin{aligned} & E\{\xi_p | Z_p, Z_q\} \\ & \approx E\left\{ \left(\frac{n_p}{Z_q(Z_p + n_p)} \right)^2 | Z_p, Z_q \right\} \times m_g^{(2)}(\mathbf{p}) \times c^2 \\ & = \frac{1}{Z_q^2} \times \left(\frac{E\{n_p^2\}}{Z_p^2} - 2 \frac{E\{n_p^3\}}{Z_p^3} + 3 \frac{E\{n_p^4\}}{Z_p^4} - \dots \right) \times m_g^{(2)}(\mathbf{p}) \times c^2 \quad (6) \\ & = \frac{1}{Z_q^2} \times \left(\frac{\sigma_n^2(\mathbf{p})}{Z_p^2} + 9 \frac{\sigma_n^4(\mathbf{p})}{Z_p^3} + 75 \frac{\sigma_n^6(\mathbf{p})}{Z_p^4} - \dots \right) \times m_g^{(2)}(\mathbf{p}) \times c^2 \\ & \approx \frac{1}{Z_q^2} \times \frac{\sigma_n^2(\mathbf{p})}{Z_p^2} \times m_g^{(2)}(\mathbf{p}) \times c^2 \end{aligned}$$

其中 $m_g^{(2)}(\mathbf{p}) = E\{g_x^2(\mathbf{p})\}$ ， $\sigma_n^2(\mathbf{p})$ 為 n_p 之變異數，並假設 n_p 之數值呈現常態分布 $n_p \sim N(0, \sigma_n^2(\mathbf{p}))$ 。透過公式(6)的推導結果得知，單一像素之合成品質係由四個因素共同影響：Depth-error Variance、Intensity Variation、



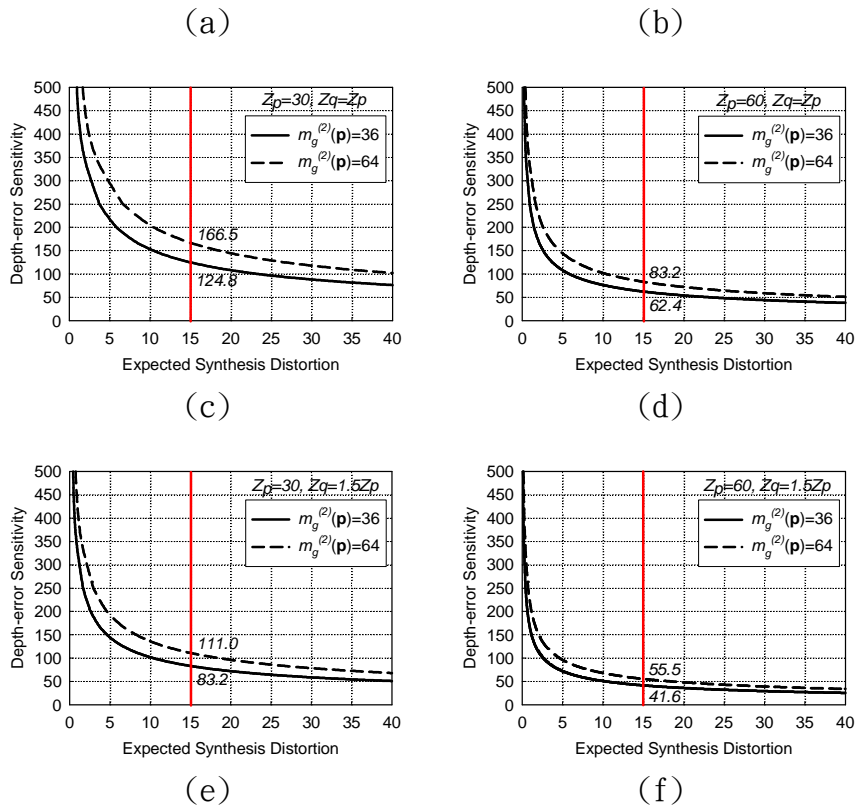


圖74. Measuring the depth-error sensitivity under various settings of Z_p , Z_q

and $g_x^2(\mathbf{p})$.

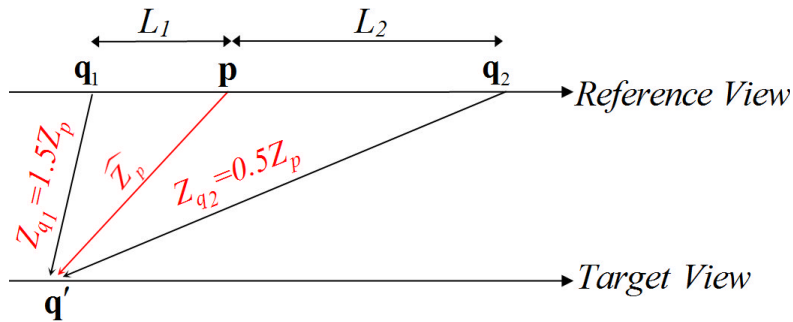


圖75. A geometrical interpretation of the effect of Z_q on depth-error sensitivity.

Ground-truth Depth Value、Virtual Camera Location。本文給定公式(6)

各種不同的參數設定，並將結果描繪於圖 74 中，並提出四項詳細之觀察：

(2) 當 $m_g^{(2)}(\mathbf{p})$ 越大，則 $\text{Depth-error Sensitivity}(Z_p / \sigma_n(\mathbf{p}))$ 越高，可歸納出物體邊界或複雜的圖像內容，這些區域之深度值較不可靠。

(3) 比較圖(a)(c)(e)與(b)(d)(f)，當景深越深，其造成之合成誤差相對越小。

(4)比較圖(e)與(a)(c) (或是圖(f)與(b)(d)),若 $Z_q > Z_p$, 其 Depth-error Sensitivity

會比 $Z_q < Z_p$ 來的小。由圖 75 得知, 當 $Z_{q1} \square Z_p \square Z_{q2}$, \mathbf{p} 相對於 \mathbf{q}_2 為一個背景的点, 使用背景的点覆盖前景的点, 將產生較嚴重的合成誤差; 反之, \mathbf{p} 相對於 \mathbf{q}_1 為一個前景的点, 此覆盖情形較不會出現明顯的合成誤差。

由公式(6)的 $\frac{\sigma_n^2(\mathbf{p})}{Z_p^2}$ 和 c^2 可知, 若虛擬視角與參照視角的距離越遠,

其 Depth-error sensitivity 也會越高。

低演算複雜度之一維深度圖壓縮

本子項壓縮演算法與 MVC Intra 模式差異處主要有三:

- (1)深度值預測: MVC 採用相鄰區塊的像素深度值估測目前區塊的像素深度值; 本子項在每個區塊採用眾數演算法, 挑選代表該區塊之深度值, 且不再以 MSE 作為預測模式的決策條件。
- (2)資料量化器: MVC 採用純量量化器(Scalar Quantizer)精簡資料輸出值的種類數至一個較小的集合裡; 本子項則以門檻值(Threshold)設計一個簡易的量化器, 將不同於眾數之像素深度值, 取代為眾數深度值。
- (3)資料型態: MVC 將預測誤差值(Residual)經過離散餘弦轉換(DCT)後成為頻率域(Frequency Domain)表示之資料型態; 本子項不需任何 Transform Coding, 直接在空間域(Spatial Domain)中進行幾近無損壓縮的 Residual 編碼。

預估區塊以 16x16 為單位分割每張深度圖, 並取出每個區塊深度值之眾數代表該區塊之深度值, 稱為眾數決議預測(Mode Determination Prediction), 各個區塊的眾數將直接經過 Entropy Coder, 以無損的方式壓縮。本子項的深度值預測方式為眾數決議(圖 76), 一般壓縮演算法採用的是最小均方根預測該像素值。若以平均數預測深度值, 雖可滿足最小 Residual 具最小 MSE, 但每個深度值都將產生些許的預估誤差, 造成合成虛擬影像不可預期之合成誤差。

次之, 本子項採取門檻值量化器量化區塊中所有深度值, 量化方式如下:

$$D_i = \begin{cases} M, & \text{if } |D_i - M| < TH_1; \\ D_i & \text{otherwise.} \end{cases} \quad (7)$$

其中 D_i 為該區塊某一點的深度值、 M 為該區塊之眾數值、 TH_1 為使用者設定門檻之常數，若該像素深度值與眾數之絕對值差小於給定的門檻值，則該像素深度值會被量化為眾數(圖 76)。 TH_1 為控制深度圖品質之用，其效果類似於 AVC 中的 QP 設定， TH_1 越大，雖然可以增加深度圖之壓縮效率，但是也降低深度圖之品質。

眾數決議之預估畫面與門檻值量化畫面之相異像素，定義為 Residual 畫面(圖 76)。Residual 畫面先以水平方式掃描為一維訊號，並以(深度值,位置,長度)的方式將 Residual 畫面分割為各個大小不等的群組(圖 77(a))。由於分割結果存在一些只有一個點的分割(isolated segment)，如圖 77(a)中的(3,3,1)分割。這種分割容易造成壓縮效能的降低且並不會帶來太多合成上的好處，因此本子項將其視為雜訊而與鄰近分割合併，圖 77(b)為合併後的結果。合併 isolated segment 雖可有效提升壓縮率，但也因而引入額外的深度值誤差。最後，各個 Segmentation 經由 Table.5 設定其 Segment Coding Mode，以利進一步降低資料量。

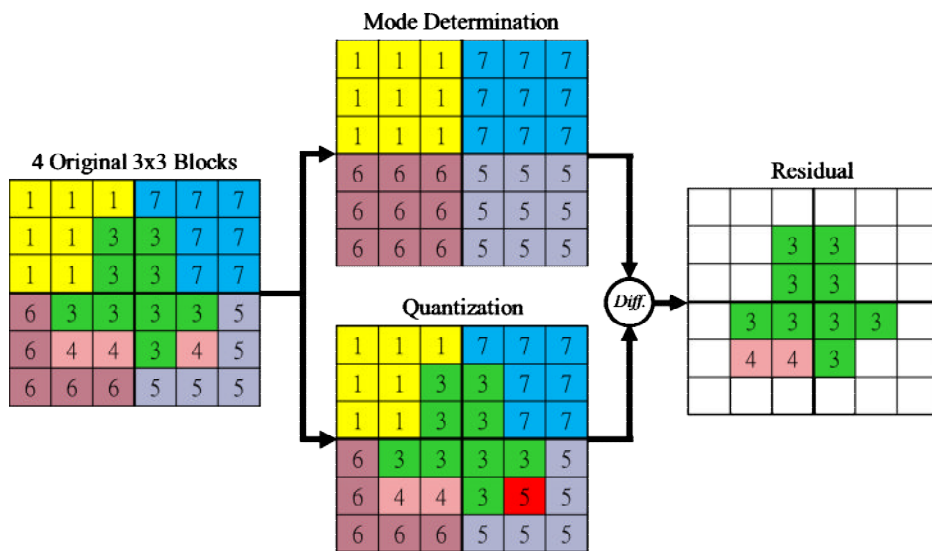


圖 76. Residual 由眾數決議預測區塊與門檻值量化區塊之相異像素所構成。其中區塊以 3x3 大小為例，畫面大小為 6x6， TH_1 設為 2。

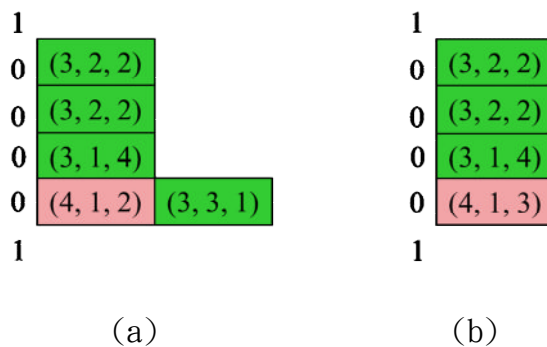


圖77. (a) Line-based residual segmentation; (b) removal of isolated segments.其中每列開頭的 0 與 1 代表此列"有/沒有"Residual Signals。

Table.5. 各種 segment coding mode 的 coding rule。

<i>Segment Coding Mode</i>	<i>Coding Rule</i>
0	All the segment parameters are the same.
1	Only depth values are the same.
2	Otherwise.

中介軟體定義與開發

本項目將依據多視角視訊編解碼(MVC)之需求以及各種不同的視角內插機制，擴展目前 MHP 之 API 並提供 3-D Video/Graphics 中介軟體。同時我們將以 C/C++ 提供針對 Power PC 與其他子項硬體設計所優化之 MVC 編解碼與視角合成程式庫(Libraries)。並將以 back-door Java native interface 形式提供給所定義之 Java API 呼叫。

子項目四：嵌入式平台的 Java 處理器設計 (Java Processor Design for Embedded Systems)

本子項目的目標在於針對嵌入式多媒體的應用，設計一個全新的異質雙核心 Java 處理器，而我們設計的考量重點包含了以下重點：

(1)完整支援 Java 的 object-oriented 的語言特性。絕大部份已發表的 Java 處理器論文，都只是針對 Java 語言的部份特性提出新架構，有很多甚至只是以模擬的方式來分析所提出架構的效能，並沒有真的實作出一個完整的處理器。特別針對給嵌入式應用的 Java 處理器，多半沒有實作出 dynamic class loading 的機制。我們所實作的 Java 處理器，則包含了這些機制。

(2)對於 OS 依存度很低。很多嵌入式 Java 執行環境，都是針對某一個作業系統（如 Linux）設計的。我們在設計子項四的 Java 執行環境時，特別在不影響效能的前題下，把 Java 子系統包裝成一個獨立的單元。只要現有的作業系統可以提供 interrupt service routines 的安裝，就可以和我們的 Java 子系統整合。

(2)易與現有硬體電路整合。我們是依照 platform-based design 原則進行系統設計。Java 核心是一個 reusable IP，透過 on-chip bus protocol 和其它 IP（如 RISC 核心、外部記憶體控制器、I/O 設備）溝通。雖然系統實作時是採用 IBM CoreConnect 的 bus，但我們透過一個適當的 bus wrapper 來包裝 Java Core IP，因此可以輕易移植到其它平台（如 AMBA）。

(3)不使用昂貴的電路元件，有許多的 Java 處理器學術論文在設計時引進了昂貴的電路設計來達到最高效能。例如複雜的 instruction folding 控制電路來減少 stack operations 的數量、並採用一個超大的 register file 來當作 Java Stack Cache，以減少 stack data dependency、或是利用特殊的 Java Object Cache 來加速 object data accesses 的速度。這些設計雖然可以增進效能，但在實用上未必切合消費性電子產品的應用。我們的設計則是著重在使用最基本的電路元件來達到最大效能。而我們提出的創新電路部份包括下列幾點：double-issue 的 stack machine architecture、flexible Java heap memory location、method area controller design for dynamic class loading and caching。

(4) 在同時脈下，執行效能要能跟 CVM-JIT 的技術相當或更高。根據我們最後設計完成的異質雙核心 Java 處理器，實作在 Xilinx ML-507 平台上進行驗證測試的結果。如果是純計算為主的程式，或是難以用 JIT 簡化的控制部份(如較複雜的 method call structure)，我們的 Java 處理器都會超越 CVM-JIT 的效能。不過如果是單純的 control loop 或 method call structure, CVM-JIT 則會執行的較快。但整體而言，我們設計的處理器是比較適合嵌入式的應用的。

Java 核心的微架構

圖 78 是我們所設計的異質雙核心的 Java 系統架構圖。其中 Java Core 的部份完全是由我們在這個計畫中全新開發的核心。而系統的開機啟動流程如圖 79 所示。RISC 核心和 Java Core 溝通的機制有兩種，一個是透過 shared memory，另一個則是透過 interrupt-driven 的 mailbox 模組。後者是完全根據 Java 語言的特性而設計的特殊 mailbox 電路，跟傳統的 JNI 相較，這個電路的最大優點是大幅降低 Java-to-C 的呼叫的參數傳遞的 overhead，不過，另一方面，因為這是 inter-processor 的呼叫，因此相對地也會付出較高的 communication cost。

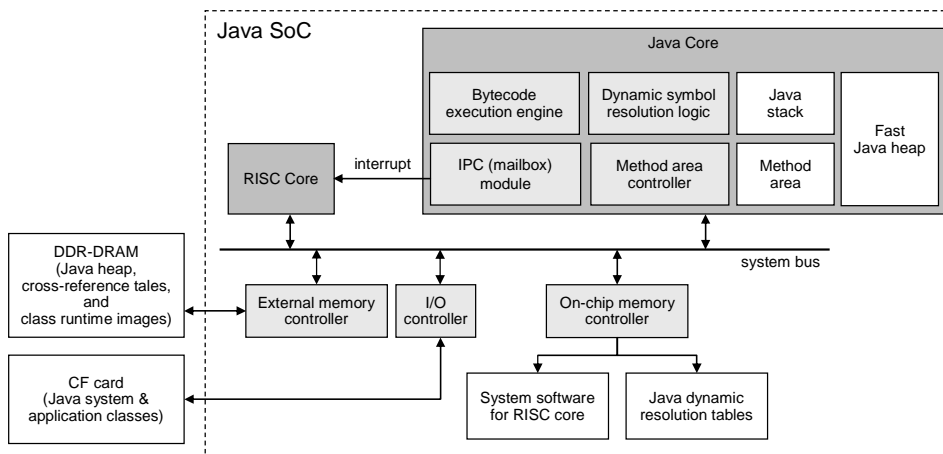


圖78. 異質雙核 Java 處理器架構。

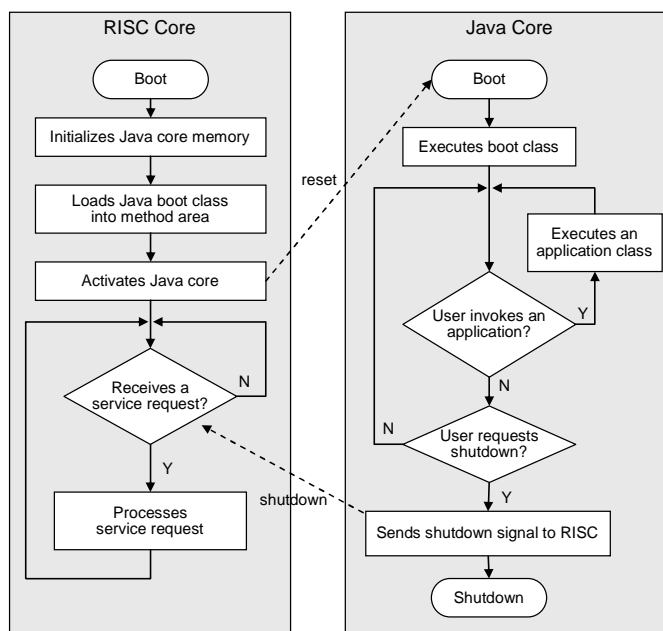


圖79. Java 系統啟動流程。

Java 核心微架構如圖 80 所示，我們採用了一個 four-stage pipeline architecture。其中，translate stage 會把 Java bytecode 轉換成我們定義的微指令。由於 Java 指令是長度可變的，為了方便讀取足夠的 instruction bytes 來提供 translate stage 和 decode stage 進行指令解碼，我們設計了一個特別的 instruction buffer 電路。

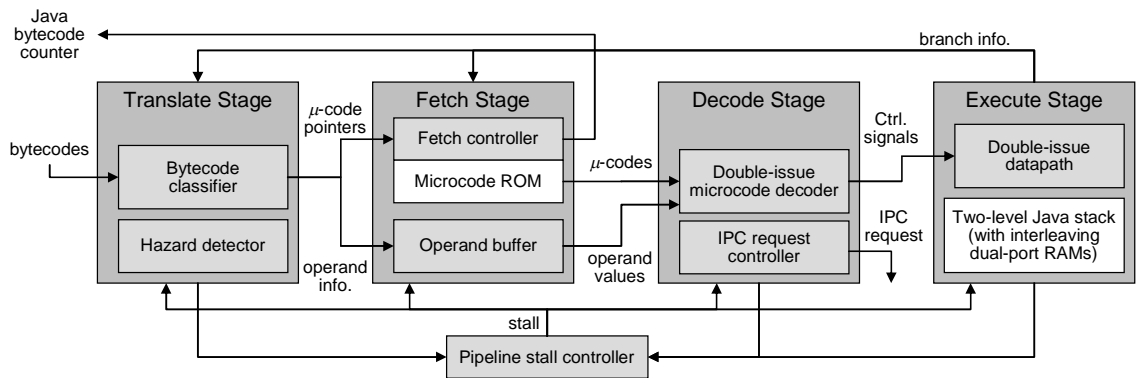


圖80. Java Microarchitecture。

Instruction buffer

instruction buffer 長度為 48bits，分為 3 個 buffer cell，各 16bits。目的是為了因應 Java bytecode 的 variable length instruction。以至少能裝下最長 instruction 為原則，所以總長度為 48bits，而因為系統為 double issue，所以一個 cell 為兩個短 instruction 長。也利用此特性使得 instruction buffer 亦可當作 operand buffer 使用。buffer controller 在每個 cycle 會從 method area 中抓取 16bit 的 bytecode 去更新 buffer，並且以 cell 為單位 shift 整個 buffer (圖 81)。

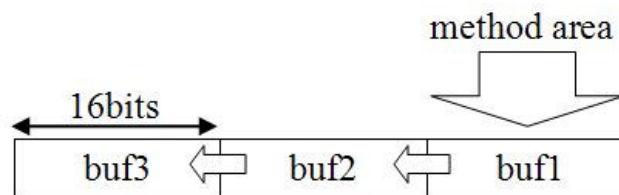


圖81. Instruction buffer controller.

Translate stage

為了實作上的簡化，我們自定義了一套 ISA 以執行 Java bytecode。並將所有的 Java bytecode 型態分類成 simple、complex、operand 三種。大部分 Java bytecode 的指令都是一對一的轉換，我們稱之為 simple instruction。少部分的指令(ex: invoke)則轉換成 code sequence，我們稱之為 complex instruction。附帶在 instruction 後面的參數則稱之為 operand。

Translate stage 的任務很簡單，僅只是按照 Fetch stage 的訊號來選定 instruction buffer 內幾個 bytecode，並將其 map 到 ISA。Simple instruction 直接 map 成 ISA。若為 complex instruction 則會 map 成 u-code address，Fetch stage 便根據此 address 讀出一連串預先寫好的 instruction pair。Operand 的取用是由 decode stage 與 execute stage 依狀況由 instruction buffer 內取出。所以理論上並不用轉換 Operand，但是 Translate stage 無法在這些 bytecode 被 translate 之前得知更多信息，所以 Operand 也會被 map。而到 Fetch stage 才有足夠訊息去判斷是否為 Operand，若是則忽略之。

instruction buffer 的解讀方式，如圖 82 所示，有 translate buf1 only、translate

buf2 only 以及 translate [upper buf1 & lower buf2]。Translate stage 則將此轉換完成的 16-bit instruction package 以及相關資訊 pass 到 Fetch stage。

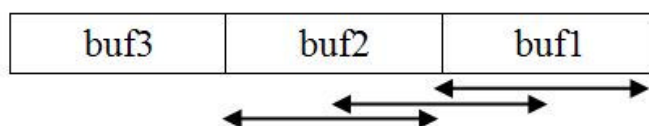


圖82. Translation form.

Fetch stage

Fetch stage 的工作主要是從傳送真正要執行的指令給 decode stage，以及控制 Translate stage 的 instruction buffer 解讀方式。從 Translate stage pass 過來的 16bits instruction package 以及相關資訊會經由判斷邏輯分析前後 instruction 各屬於 simple、complex、operand 哪一種，再進一步判斷是否為 ordinary case。圖 83 標示了所有 ordinary case，在這種情況下只需把 operand 代換為 nop，就能繼續 pass 到 Decode stage。若有 complex instruction，則 Fetch stage 進入 complex mode。若為 separated case 需要拆開處理。其情況如圖 84 所示，大致可分為兩種。當兩個 Simple instruction 發生 hazard 時或 instruction package 都不為 operand 卻含有 complex instruction。

1 st Instr.	2 nd Instr.	Output instr. combination
S	S	1 st Instr. + 2 nd Instr.
S	O	1 st Instr. + nop
O	S	nop + 2 nd Instr.
O	O	nop + nop.
C	O	complex mode
O	C	

圖83. The ordinary cases.

1 st Instr.	2 nd Instr.	Output instr. combination	
		Current	Next
S	S	1 st Instr. + nop	2 nd Instr. + 3 rd Instr.
S	C		
C	S	complex	2 nd Instr. +
C	C	mode	3 rd Instr.

圖84. The separated cases.

其處理方式是把 lower 8bits(2ndInstr.)則與下一個 8bits bytecode 結合成新的 instruction package。稍後 Translate 過，再次進入 Fetch stage。而 upper 8bits(也就是 1stInstr.)若為 simple，是則搭配一個 nop 後 pass 到 Decode stage。若是不附帶 operand 的 complex instruction，則 Fetch stage 進入 complex mode。Fetch stage 有兩個 mode，complex mode 和 normal mode。平時是 normal mode，在 1stInstr.為 complex instruction 或 1stInstr.為 operand 而 2ndInstr. 為 complex instruction 會進入 complex mode。此時會依此 instruction 附帶的 operand 數量來 stall jpc 和 instruction buffer。此操作是為了能完整把附帶的 operand 讀到 instruction buffer 所設計。也因為是 complex，所以 pass 過來的 data 是 u-code address。隨後則依此 address 從 instruction code sequence ROM 裡讀出 instruction pair 給 decode stage，直到 sequence 結束。拆開處理實作方式則是直接去控制 Translate stage 的 instruction buffer 解讀方式。那三種解讀方式組合剛好各相差 8bits，此設計即可把 2ndInstr. 與下一個 8bits bytecode 做結合，以達到 delay 效果。

Decode stage

在這個 stage，我們會在每個 cycle 從前級電路讀取兩個指令進行解碼。如果這兩個指令可以同時 double-issue，decode stage 則會產生適當的 control signal 來設定 double-issue data path，否則，第二個指令會被 buffer 起來，而第一個指令則會和 NOP 指令一起執行。

Execution stage

為了達到 double-issue 的目的，我們在 execution stage 的 datapath 採用了一個 two-level stack architecture，在已發表的文獻中，我們尚未看到其他人採用這種架構。第一個 level 的 stack 是由三個 register A, B, C 所組成，而第二個 level 的 stack 則是由兩個 interleaving 的 dual-port memory 組成。假設我們要執行 LOAD-LOAD-ALU-STORE 等四個 bytecode instructions，在連續的兩個 cycles 中，datapath 的設定是如圖 85 所示。在第一個 cycle 時，second level stack 的兩個 read ports 和兩個 write ports 會打開，以把兩個 local variables 的值 push 到 top of stacks (first-level registers)中，而在第二個 cycle 時，A 和 B 會成為 ALU 的 input，而且 ALU 的 output 會寫回 local variables 並把 top-of-stack registers 做適當的 update。

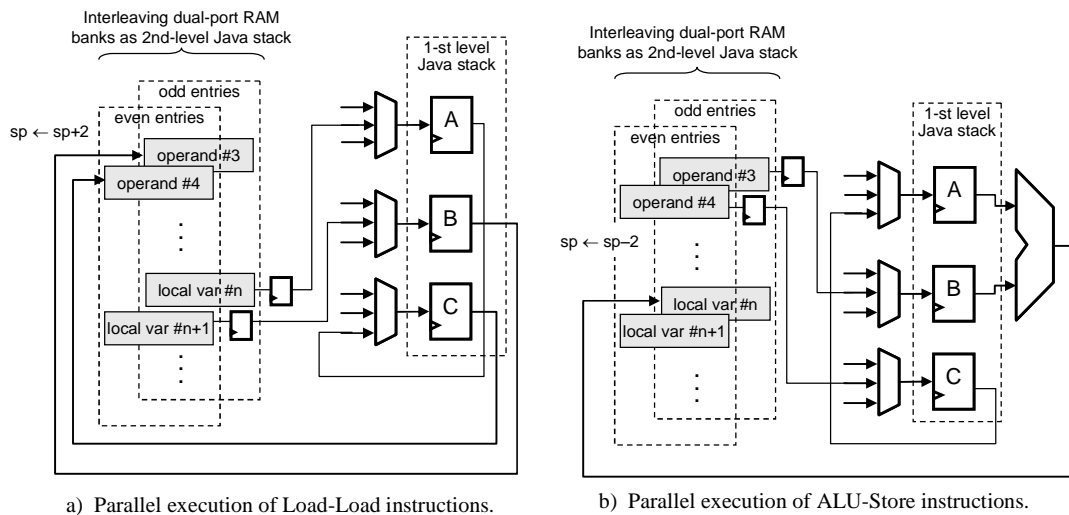
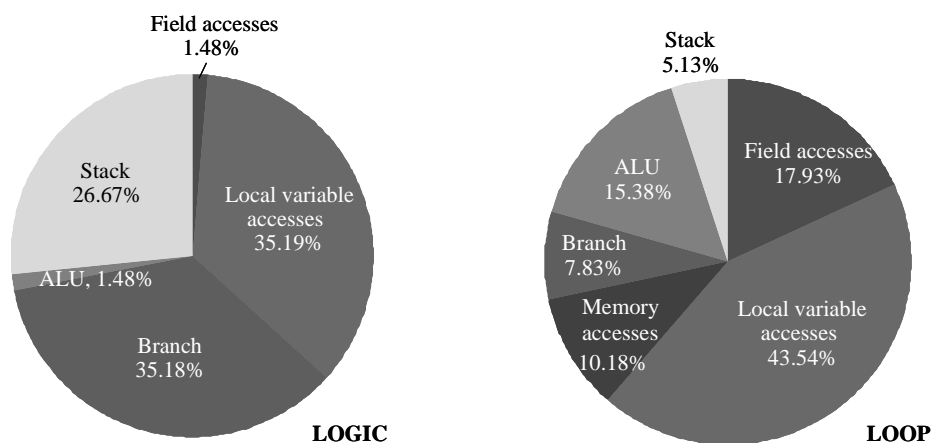


圖85. Double-issue datapath 範例。

異質雙核心的 Java 處理器整體效能分析

因為我們所設計的 Java 處理器是為嵌入式系統應用而設計的，因此當我們要評估 JavaCore 系統時，主要是利用嵌入式 Java 最廣為採用的 Caffeine Mark 測試 Benchmark 來對 JavaCore 做效能上與 CVM 及 CVM-JIT 的比較，但是因為使用 benchmark 測試跟真實 Application 的應用其實是會有些出入，所以我們藉由分析 caffeinemark 的指令分布，可以瞭解我們所使用的 benchmark 是否夠具代表性，我們分析的類別主要參考自 (Java runtime system: characterization an architectural implications)，分析結果如圖 86，分析類別如圖 87 所示。



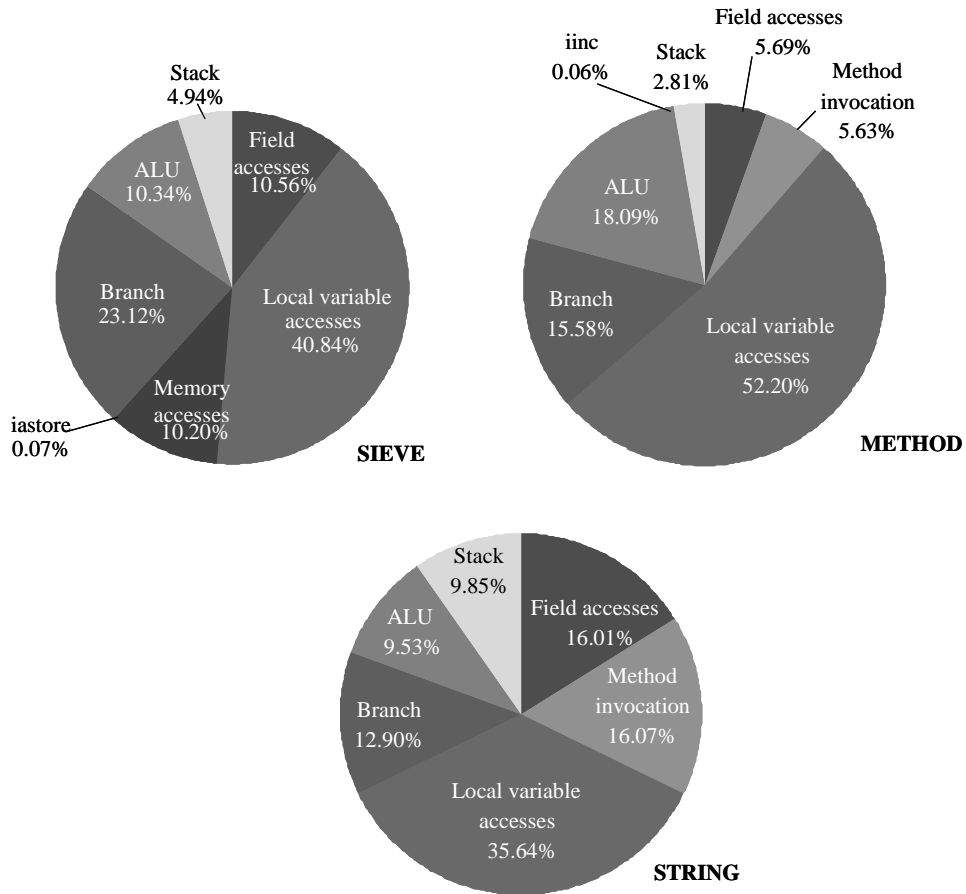


圖86. 在 Caffeine Mark 中的 Java bytecode 指令分佈。

Bytecode classification	Description	Examples
Local variable accesses	transfer data between local variables and stack	iload_1, astore_0
Array accesses	loads or store data from array (heap)	iaload, iastore
Field accesses	loads or store data from field (heap) (dynamic resolution)	getfield, putfield
Stack	allows for push or pop data into the stack	iconst_1, pop
ALU	arithmetic or logic instruction	iadd, iinc
Method invocation	method call (dynamic resolution)	invokevirtual
Branches	condition branch or jump, and return	Ifne, goto, Return
Invoke RISC service	store service ID in a register and generate an interrupt	new, ldc

圖87. Classification of bytecodes.

經過以上分析評估後我們開始對 JavaCore、CVM、及 CVM-JIT 進行測試。因為 CVM 和 CVM-JIT 在執行時，有 data cache 和 instruction cache，而我們的 JavaCore 並沒有 data cache，因此，我們將 JavaCore 的 heap space 放在 on-chip BRAM 上，測試結果如圖 88 所示。

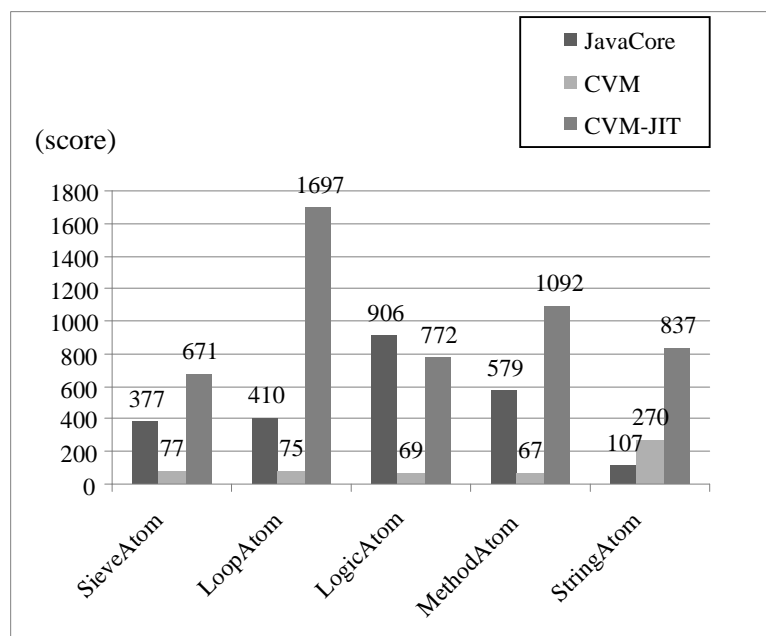


圖88. Caffeine Mark Score for JavaCore, CVM, and CVM-JIT.

其中，Loop 測試因為有很簡單的迴圈結構，因此 CVM-JIT 可能透過最佳化，而把反覆的計算簡化了。另外，Method 測試只是在單純的遞迴呼叫中進行簡單的算術計算。因此 CVM-JIT 可能把它轉成簡單的循序執行。為了驗證我們猜測 just in time compiler 的機制，會將 recursive function call 展開成 loop 執行的 Machine Code，進而使得在作 function call 時的操作上會突顯出不公平的現象，所以我們設計了一連串根據 input 循環呼叫的程式，使得 JIT 在 compile 階段無法決定何時會結束，進而無法展開成迴圈執行的 machine code 來避開 function call 的行為，結果如圖 89 所示，用電路執行 Java method invocation 的效能遠遠超過 CVM-JIT。

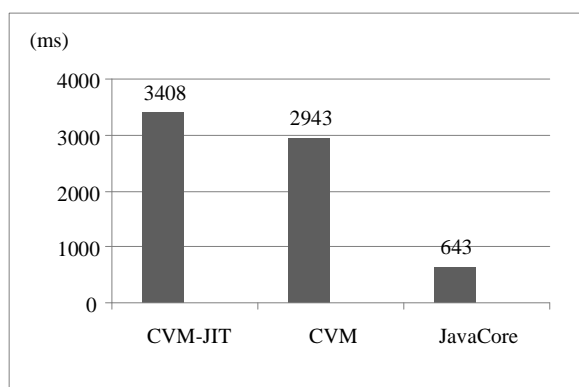


圖89. Cyclic method invocation benchmarks.

另外，JavaCore 在執行 String 的測試時效能不彰是因為目前 Java string 的 operations 我們完全是透過 IPC 呼叫 RISC core 來進行處理，因此有很高的 overhead 是花在 inter-process communication。

在我們所使用的開發平台(Xilinx ML507)上，一共有三塊空間可以存放 Java heap，除了原先有預設掛在 CoreConnect 系統 PLB Bus 上的 BRAM 之外，還有 DDR2 的 external memory，以及我們自行設計，直接連結到 JavaCore 內部的 BRAM。如果我們將 heap space 存放在不同記憶體區塊，以軟體的方式對 Heap Space 存取測試得到實際透過 PLB Bus 存取 BRAM 的 over head 約 7~8cycles(結果如圖 90)，故我們最後所使用的 heap space 為手動建立在 JavaCore 內部的 BRAM 電路而非經由 PLB BUS 所存取的 BRAM。

Memory Time	DDR2	BRAM (thru PLB)	BRAM (direct wires)
Cycles	18.87	8.39	1

圖90. Memory access time from within Java logic.

而圖 91 則分析了當我們把 Java heap 空間放在不同的記憶體時，對 Caffeine Mark 效能的影響。

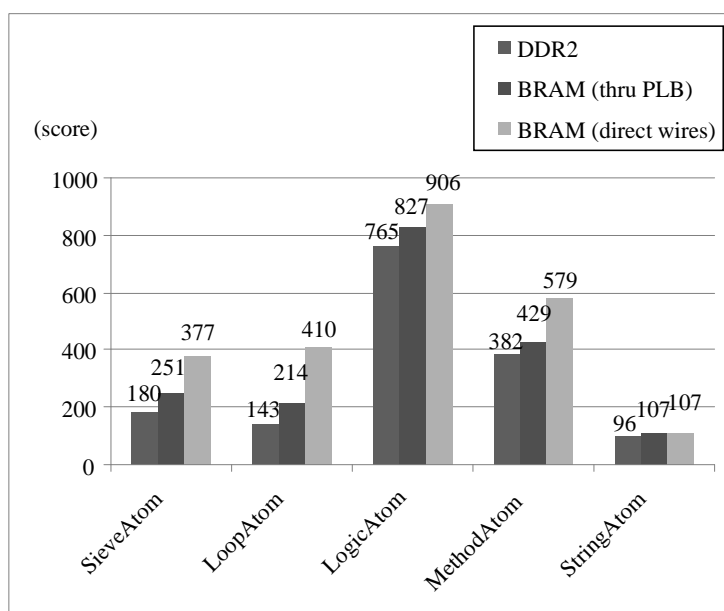


圖91. Caffeine Mark with Java heap allocated from different memory space.

Dynamic Class Loading 機制的效能分析

對於 dynamic class loading 的效能評估我們分為兩個部分，一個為 parsing 的時間(跟原始 class 大小有關)，另一個為 class loading 的時間(跟 runtime image 的大小有關)，所以我們先建立了不同 class 大小以及 parse 完的 runtime image 大小

在 dynamic class loading 上所需的時間，分析結果如圖 92 及圖 93 所示。

Run Time Image Size (byte) \ Class size (byte)	500	1000	2000	4000
1000	3766			
2000	3783	6739		
4000	3760	7350	13160	
8000	4937	7700	14836	26518

圖92. Class loading time.

Run Time Image Size (byte) \ Class size (byte)	500	1000	2000	4000
1000	8214			
2000	8782	8271		
4000	14664	10631	83894	
8000	36042	31013	17466	8639

圖93. Class parsing time.

當我們建立完 class vs. runtime image size 相對關係在 dynamic class loading 機制上的影響之後，我們希望可以進一步得到一般 application 的 class File 是落在表上的那個區間內，而這資訊未來可以作為我們在制定 JavaCore 內部的 method area 的大小，這邊我們分析的方式是使用我們 PC 版本的 Parser 去分析 J2ME 的 System Class File，去統計這些 class 與 Run time image 的大小來作為代表，分析結果如圖 94。

Item \ Size(byte)	Class	Run time image
0 - 1000	62.3 %	86.7 %
1000 - 2000	18.4 %	8.2 %
2000 - 3000	5.1 %	3.1 %
3000 - 4000	6.1 %	1.0 %
4000 - 5000	2.0 %	0.0 %
5000 - 6000	3.1 %	0.0 %
6000 - 7000	2.0 %	0.0 %

7000 - 8000	0.0 %	0.0 %
8000 - 9000	0.0 %	1.0 %
>10000	1.0 %	0.0 %

圖94. J2ME system classes and run time image size statistics (average class size: 1366 bytes, average runtime image size: 509 bytes).

最後我們的設計參數是讓 method area 由 32 塊 2KB 的 BRAM 所組成。

Double-issue 的效能分析

由於我們所設計的 JavaCore 是採用了 double-issue 的架構，因此我們接下來針對 JavaCore 進行 double issue 的執行比例的分析，來觀察 double-issue 的設計是否有實質上的義意。在這邊，除了 Caffeine Mark 的試程式之外，我們也引進計算 pi 的程式來進行分析。主要理由是 Caffeine Mark 是以控制的測試為主，計算較不密集，因此我們引進了額外的程式，來驗證 double issue 對效能的幫助，結果如圖 95 所示。如果 Java 程式是以計算為主的話，double-issue 的指令比例可以達到 38%，不過如果是以控制為主的程式，只有不到 10%的指令是 double-issued。由於我們只採用了簡單的電路架構就達到 double-issue 的效果，因此這個結果算是相當不錯的。

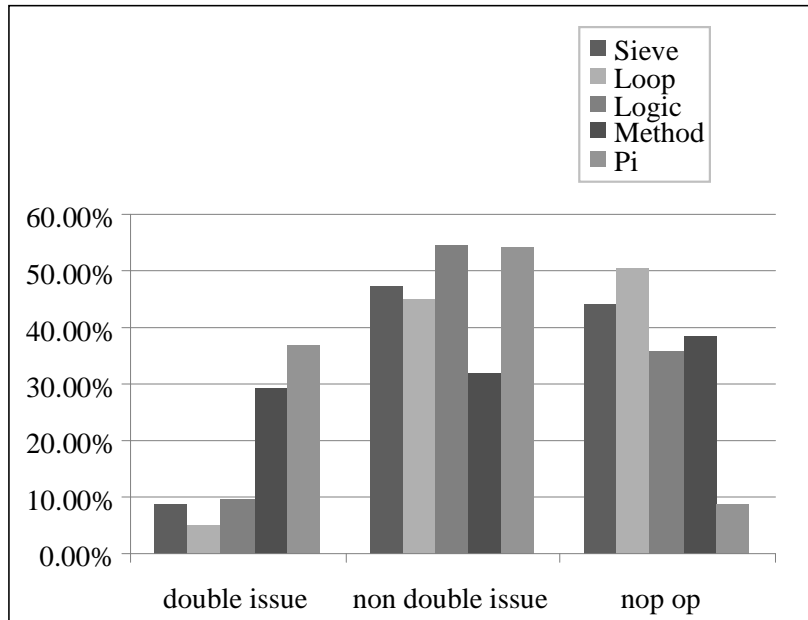


圖95. Percentage of double-issue for different Java programs.

四. 計畫產出論文

已發表之論文

- [1] J.-C. Chan, N. T.-C. Chang, and T.-S. Chang, "ISID: In-order scan and indexed diffusion segmentation algorithm for Stereo Vision," in *Proc. of IEEE Int. Symposium on Circuits and Systems*, Seattle, U.S., 2008.
- [2] T.-H. Tsai, N. Y.-C. Chang, and T.-S. Chang, "Data reuse analysis of local stereo matching," in *Proc. of IEEE Int. Symposium on Circuits and Systems*, Seattle, U.S., 2008.
- [3] L.-Y. Ku, S.-H. Wen, N. Y.-C. Chang, and T.-S. Chang, "A low-Cost real-time command control system based on stereo-vision and hand motion," in *Proc. of Conf. on Computer Vision, Graphics, and Image Processing*, Taiwan, 2008.
- [4] J.-C. Chan, N. Y.-C. Chang, Y.-C. Tseng, and T.-S. Chang, "Local belief aggregation for MRF-based color image segmentation," in *Proc. of Conf. on Computer Vision, Graphics, and Image Processing*, Taiwan, 2008.
- [5] Y.-C. Tseng, N. Y.-C. Chang, and T.-S. Chang, "Block-based belief propagation with in-place message updating for stereo video," in *Proc. of IEEE Asia Pacific Conf. on Circuits and Systems*, China, Macao, 2008.
- [6] N. Y.-C. Chang, Y.-C. Tseng, and T.-S. Chang, "Analysis of color space and similarity measure impact on stereo block matching," in *Proc. of IEEE Asia Pacific Conf. on Circuits and Systems*, China, Macao, 2008.
- [7] Y.-C. Tseng, N. Y.-C. Chang, and T.-S. Chang, "Low-memory cost belief propagation architecture for disparity estimation," in *Proc. of IEEE Int. Symposium on Circuits and Systems*, Taipei, Taiwan, 2009.
- [8] P.-H. Hsu, Y.-C. Tseng, and T.-S. Chang, "Low memory cost bilateral filtering using stripe-based sliding integral histogram," in *Proc. of IEEE Int'l Symposium on Circuits and Systems*, Paris, France, 2010.
- [9] Y.-R. Horng, Y.-C. Tseng, and T.-S. Chang, "Stereoscopic images generation with directional Gaussian filter," in *Proc. of IEEE Int'l Symposium on Circuits and Systems*, Paris, France, 2010.
- [10] N. Y.-C. Chang, T.-H. Tsai, B.-H. Hsu, Y.-C. Chen, and T.-S. Chang, "Algorithm and architecture of disparity estimation with mini-census adaptive support weight," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 6, pp. 792-805, June 2010,
- [11] Y.-C. Tseng and T.-S. Chang, "Architecture design of belief propagation for real-time disparity estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 20, no. 11, pp. 1555-1564, Nov. 2010.

- [12] J. H. Tu and Lan-Da Van, "Power-efficient pipelined reconfigurable fixed-width Baugh-Wooley multipliers," *IEEE Trans. Computers*, vol. 58, no. 10, pp. 1346-1355, Oct. 2009. (SCI & EI, Full Paper)
- [13] C. T. Lin, Y. C. Yu, and Lan-Da Van, "Cost-effective triple-mode reconfigurable pipeline FFT/IFFT/2-D DCT processor," *IEEE Trans. VLSI Syst.*, vol. 16, no. 8, pp. 1058-1071, Aug. 2008. (SCI & EI, Full Paper)
- [14] T. Y. Sheu, L. D. Van, T. R. Jung, C. W. Lin, and T. W. Chang, "Low complexity subdivision algorithm to approximate Phong shading using forward difference," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May. 2009, pp. 2373-2376, Taipei, Taiwan.
- [15] P. Y. Chen, L. D. Van, H. C. Reddy, and C. T. Lin, "A new VLSI 2-D fourfold-rotational-symmetry filter architecture design," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May. 2009, pp. 93-96, Taipei, Taiwan.
- [16] I. H. Khoo, H. C. Reddy, L. D. Van, and C. T. Lin, "2-D digital filter architectures without global broadcast and some symmetry applications," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May. 2009, pp. 952-955, Taipei, Taiwan.
- [17] L. Y. Lin, H. K. Lin, C. Y. Wang, L. D. Van, and J. Y. Jou, "Hierarchical architecture for network-on-chip platform," in *Proc. VLSI-DAT*, Apr. 2009, pp. 343-346, Hsinchu, Taiwan.
- [18] P. Y. Chen, L. D. Van, H. C. Reddy, and C. T. Lin, "A new VLSI 2-D diagonal-symmetry filter architecture design," in *Proc. IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, Nov. 2008, pp. 320-323, Macao, China.
- [19] T. R. Jung, L. D. Van, T. Y. Sheu, C. W. Lin, W. C. Fang, "Design of multi-mode depth buffer compression for 3D graphics system," in *Proc. IEEE Int. Conf. Multimedia and Expo. (ICME)*, July 2008, pp. 789-792, Hannover, Germany.
- [20] T. R. Jung, L. D. Van, W. C. Fang, T. Y. Sheu, "Reconfigurable depth buffer compression design for 3D graphics system," in *Proc. Int. Conf. MUE.*, Apr. 2008, pp. 470-474, Busan, Korea.
- [21] Y.-C. Tseng, C.-H. Wu, Y.-W. Chen, T.-W. Wang, and W.-H. Peng, "On the Analysis and Design of Motion Sampling Structure for Advanced Motion-Compensated Prediction," *IEEE Int'l Conf. on Image Processing*, 2010.
- [22] T. W. Wang, Y. W. Chen, and W. H. Peng, "Analysis of Template Matching Prediction and Its Application to Parametric Overlapped Block Motion Compensation," *IEEE Int'l Symposium on Circuits and Systems*, 2010.

- [23] C. C. Chen, Y. W. Chen, F. Y. Yang, and W. H. Peng, "A Synthesis-Quality-Oriented Depth Refinement Scheme for MPEG Free Viewpoint Television (FTV)," *IEEE Int'l Symposium on Multimedia*, 2009.
- [24] Y. W. Chen, T. W. Wang, Y. C. Tseng, W. H. Peng, and S. Y. Lee, "A Parametric Window Design for Overlapped Block Motion Compensation with Variable Block-size Motion Estimates," *IEEE Int'l Workshop on Multimedia Signal Processing*, 2009.
- [25] Y. W. Chen, C. H. Wu, C. L. Lee, T. W. Wang and W. H. Peng, " MB Mode with Joint Application of Template and Block Motion Compensations," ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 2nd meeting, JCTVC-B072, Geneva, CH, July 2010.
- [26] Y. W. Chen, T. W. Wang, C. H. Chan, C. L. Lee, C. H. Wu, Y. C. Tseng, W. H. Peng, C. J. Tsai, and H. M. Hang, " Description of video coding technology proposal by NCTU ", ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 1st meeting, JCT-VC –A123, Dresden, DE, April 2010.
- [27] T.-F. Shen and C.-J. Tsai, "Dynamic Task Partitioning for Video Decoding on Heterogeneous Dual-Core Platforms," *Proc. of VLSI Design/CAD*, Ken-Ting, Taiwan, Aug., 2008.
- [28] K.-N. Su, H.-J. Ko, and C.-J. Tsai, "Java Runtime Environment Design for Embedded Multimedia Services," *Proc. of VLSI Design/CAD*, Ken-Ting, Taiwan, Aug., 2008.
- [29] M.-J. Wu, Y.-T. Chen, and C.-J. Tsai, "Hardware-assisted Syntax Decoding Model for Software AVC/H.264 Decoders," *Proc. of IEEE Int. Symposium on Circuit and System*, Taipei, May 2009.
- [30] K.-N. Su and C.-J. Tsai, "Fast Host Service Interface Design for Embedded Java Application Processor," *Proc. of IEEE Int. Symposium on Circuit and System*, Taipei, May 2009.
- [31] C.-N. Huwang, C.-Y. Bai, K.-N. Su, and C.-J. Tsai, "Dual-Core Java RE SoC with Embedded GUI Middleware," *Proc. of VLSI Design/CAD*, Hua-Liang, Taiwan, Aug., 2009.
- [32] C.-F. Hwang, K.-N. Su, and C.-J. Tsai, "Low-Cost Class Caching Mechanism for Java SoC," *Proc. of IEEE Int. Symposium on Circuit and System*, Paris, May 2010.

已接受之論文

- [1] C.-J. Tsai, T.-F. Shen, P.-C. Liao, "Dynamic Task Partition for Video Decoding on Heterogeneous Dual-core Platforms," *ACM Transactions on Embedded Computing Systems*, Accepted Jan. 2011.

審稿中之論文

- [1] Y.-R. Horng, Y.-C. Tseng, and T.-S. Chang, "VLSI architecture of real time HD1080p view synthesis engine," *IEEE Trans. on Circuits and Systems for Video Technology*. (Under reviewing)
- [2] Y.-C. Tseng, B.-H. Hsu, and T.-S. Chang, "A 124 Mpixels/sec VLSI design for histogram-based joint bilateral filtering," *IEEE Trans. on Image Processing*. (Under reviewing)

五. 參考文獻

- [1]. M. Ziegler, "Digital stereoscopic imaging and application, a way toward new dimensions, the RACE II project DISTIMA," presented at *IEE Colloq. Stereoscopic Television*, London, 1992.
- [2]. M. Ziegler, "Digital stereoscopic television—State of the art of the European project DISTIMA," in *Proc. 4th Euro. Workshop on 3DTV*, Rome, 1993.
- [3]. A. Vetro, P. Pandit, H. Kimata, and A. Smolic, *Working Draft 3.0 of Multiview Video Coding, MPEG Document N8966*, San Jose, April, 2007.
- [4]. D. Tzovaras, N. Grammalidis, and M. G.Strintzis, "Object-based Coding of Stereo Image Sequences using Joint 3-D Motion/Disparity Compensation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 2, pp. 312-327, April 1997.
- [5]. A. Jacobs, J. Mather, R. Winlow, D. Montgomery, G. Jones, M. Willis, M. Tillin, L. Hill, M. Khazova, H. Stevenson, G. Bourhill, "2D/3D Switchable Displays," *Sharp Technical Journal*, No.17, p15-18, 2003.4.
- [6]. Ellen de Vries, "Philips showcases 3D display technology at SID 2004," *PhilipsResearch Press Release*, May 24, 2004.
- [7]. O. Tishutin and T. Striegler, "Stereoscopic 3D and iZ3D Perception," *iZ3D Whitepaper*, March 2007.
- [8]. Neovisionlabs, "iFusion," <http://www.neovisionlabs.com/>
- [9]. <http://www.iz3d.com.tw/home.html>
- [10]. 交通部電信總局, 「數位廣播電視系統整合之研究」, 2004 .
- [11]. ETSI TS101 192, "Digital Video Broadcasting (DVB); Specification for Data Broadcasting", 2000.
- [12]. TS 102 812, "DVB Multimedia Home Platform (MHP) Specification 1.1", Nov. 2001.
- [13]. D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int'l Journal of Computer Vision*, vol. 47(1/2/3), pp. 7-42, 2002.
- [14]. Middlebury Stereo Vision Page, Available: <http://vision.middlebury.edu/stereo/>
- [15]. M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp.993-1008, August, 2003.
- [16]. H. Hirschmuller, "Stereo vision in structured environments by consistent semi-global matching," in *Proc. of IEEE Conf. on Computer Vision and*

- Pattern Recognition*, 2006.
- [17]. M. Gerrits and P. Bekaert, "Local stereo matching using segmentation-based outlier rejection," in *Proc. of Conf. on Computer and Robot Vision*, 2006.
 - [18]. K.-J. Yoon and I.-S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650-656, April 2006.
 - [19]. I. J. Cox, L. Hingorani, and S. B. Rao, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 542-567, 1996.
 - [20]. V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. Eighth IEEE International Conference on Computer Vision*, 2001, ICCV 2001, vol.2, pp.508 - 515, July 2001.
 - [21]. J. Sun, N. N. Zheng, H. Y. Shum, "Stereo Matching using Belief Propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.7 , pp. 787-800, July 2003.
 - [22]. L. Lucchese and S. K. Mitra, "Color image segmentation: a state-of-the-art survey," in *Proc. Indian National Science Academy(INSA-A)*, vol. 67, A, New Delhi, India, pp.207-221, Mar. 2001.
 - [23]. L. Shafarenko, M. Petrou, and J. Kittler, "Automatic watershed segmentation of randomly textured color images," *IEEE Trans. on Image Processing*, vol. IP-6, no. 11, pp. 1530-1544, Nov. 1997.
 - [24]. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603-619, May 2002.
 - [25]. S. A. Shafer, "Using color to separate reflection components," *Color Research and Application*, vol. 10, no. 4, pp. 210-218, 1985.
 - [26]. A. Klaus, M. Sormann and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proc. Int'l Conf. of Pattern Recognition*, 2006.
 - [27]. Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.
 - [28]. O. Faugeras, B. Hotz, H. Matthieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, "Real time correlation-based stereo: algorithm, implementations and application," *INRIA, Tech. Rep. 2013*, 1993.
 - [29]. C. Cuadrado, A. Zuloaga, J. L. Martin, J. Lazaro, and J. Jimenez, "Real-time

- stereo vision processing system in a FPGA,” in *Proc. of IEEE Industrial Electronics, IECON 2006*.
- [30]. A. Darabiha, W. J. MacLean, and J. Rose, “Reconfigurable hardware implementation of a phase-correlation stereo algorithm,” *Machine Vision Application*, vol. 17, no. 2, pp. 116-132, 2006.
- [31]. J. Diaz, E. Ros, R. Carrillo, and A. Prieto, “Real-time system for high-image resolution disparity estimation,” *IEEE Trans. on Image Processing*, vol. 16, no. 1, pp.280-285, Jan. 2007.
- [32]. S. Sabihuddin, and W. J. MacLean, “Maximum-likelihood stereo correspondence using field programmable gate array,” in *Proc. of Int’l Conf. on Computer Vision Systems(ICCV)*, 2007.
- [33]. O. Veksler, “Stereo correspondence by dynamic programming on a tree,” in *Proc. of IEEE Computer Vision and Pattern Recognition(CVPR)*, 2005.
- [34]. Y. Deng and X. Lin, “A fast line segment based dense stereo algorithm using tree dynamic programming,” in *Proc. of European Conference on Computer Vision (ECCV)*, 2006.
- [35]. C. Lei, J. Selzer, and Y. H. Yang, “Region-tree based stereo using dynamic programming optimization,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [36]. N. Y. C. Chang and T. S. Chang, “A scalable graph-cut engine architecture for stereo vision,” in *Proc. VLSI Design/CAD Symposium*, 2007.
- [37]. S. Park and H. Jeong, “VLSI architecture for MRF based stereo matching,” in *Proc. of Inter’l Symposium on Systems, Architectures, Modeling and Simulation (SAMOS)*, 2007.
- [38]. P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient belief propagation for early vision,” in *Proc. of IEEE Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [39]. S. Park, H. Jeong, K. Pohang, and S. Korea, "Real-time stereo vision FPGA chip with low error rate," *Proc. of Int’l Conf. on Multimedia and Ubiquitous Engineering*, pp. 751-756, 2007.
- [40]. F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda. “Near real-time stereo based on effective cost aggregation,” in *Proc. of IEEE Int’l Conf. on Computer Vision and Pattern Recognition*, 2008.
- [41]. S. Forstmann, Y. Kanou, O. Jun, S. Thuring, and A. Schmitt, "Real-Time Stereo by using Dynamic Programming," in *Proc. of Computer Vision and Pattern Recognition Workshop on Real-Time 3D Sensor and Their Use*, , 2004, pp. 29-29, 2004.
- [42]. J. Lu, G. Lafruit, and F. Catthoor, "Fast variable center-biased windowing for

- high-speed stereo on programmable graphics hardware," in *Proc. of IEEE Int'l Conf. on Image Processing*, pp. 568- 571, 2007.
- [43]. J. Lu, S. Rogmans, G. Lafruit, and F. Catthoor, "Real-time stereo correspondence using a truncated separable Laplacian kernel approximation on graphics hardware," in *Proc. of IEEE Int'l Conf. on Multimedia and Expo*, pp. 1946-1949, 2007.
- [44]. Q. Yang, L. Wang, R. Yang, S. Wang, M. Liao, and D. Nister, "Real-time global stereo matching using hierarchical belief propagation," in *Proc. of The British Machine Vision Conference*, 2006.
- [45]. L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, "High-quality real-time stereo using adaptive cost aggregation and dynamic programming," in *Proc. of the 3rd Int'l Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, pp. 798-805, 2006.
- [46]. G. Minglun and Y. Yee-Hong, "Near real-time reliable stereo matching using programmable graphics hardware," in *Proc. of IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol.1, pp. 924-931, 2005.
- [47]. M. Gong and R. Yang, "Image-gradient-guided real-time stereo on graphics hardware," in *Proc. of 5th Int'l Conf. on 3-D Digital Imaging and Modeling*, pp. 548-555, 2005.
- [48]. Y.-C. Tseng, N. Chang, and T.-S. Chang, "Low memory cost block-based belief propagation for stereo correspondence," in *Proc. of IEEE Int'l Conf. of Multimedia & Expo (ICME)*, Beijing, China, July 2007.
- [49]. C.-C. Cheng, C.-K. Liang, Y.-C. Lai, H. H. Chen, and L.-G. Chen, "Analysis of belief propagation for hardware realization," in *Proc. of IEEE Workshop on Signal Processing Systems (SiPS)*, Washington DC, USA, October 2008.
- [50]. A. Smolic and P. Kau, "Interactive 3D video representation and coding technologies," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 98-110, Jan. 2005.
- [51]. E. Martinian, A. Behrens, J. Xin, A. Vetro, and H. Sun, "Extensions of H.264/AVC for Multiview Video Compression", *IEEE International Conference on Image Processing*, pp. 2981-2984, October, 2006.
- [52]. C.-J. Tsai, P. Eisert, B. Girod, and A.K. Katsaggelos, Oct., "Model-based Synthetic View Generation from a Monocular Video Sequence," *Proc. IEEE Int. Conf. on Image Processing*, Vol I, pp.444-447, Santa Barbara, USA, 1997.
- [53]. C.-J. Tsai and A. K. Katsaggelos, "Dense Disparity Map Estimation with a Divide-and-Conquer Disparity-Space Image Technique," *IEEE Trans. Multimedia*, Vol. 1 No. 1, March 1999, pp. 18-29.
- [54]. C.-J. Tsai and A. K. Katsaggelos, "Sequential Construction of 3D-Based Scene Description," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 10

No. 4, June 2000, pp. 576-584.

- [55]. M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. SIGGRAPH*, 1996.
- [56]. J. J. Labresse, *MicroC/OS-II: The Real-Time Kernel*, CMP Book, ISBN: 1-57820-103-9, 2002.
- [57]. A. J. Massa, *Embedded Software Development with eCos*, Prentice Hall, ISBN 0-13-035473-2, 2002.
- [58]. D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. On Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, May 2002.
- [59]. A. Klaus, M. Sormann and K. Karner, "Segment-based Stereo Matching Using Belief Propagation and a Self-adapting Dissimilarity Measure," in *Proc. International Conf. of Pattern Recognition*, 2006.
- [60]. Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo Matching with Color-weighted Correlation, Hierarchical Belief Propagation and Occlusion Handling," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.
- [61]. <http://vision.middlebury.edu/stereo/>
- [62]. M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in Computational Stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp.993-1008, August, 2003.
- [63]. D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal of Computer Vision*, vol. 47(1/2/3), pp. 7-42, 2002.
- [64]. H. Hirschmuller, "Stereo Vision in Structured Environments by Consistent Semi-global Matching," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.
- [65]. M. Gerrits and P. Bekaert, "Local Stereo Matching using Segmentation-based Outlier Rejection," *proc. Conf. on Computer and Robot Vision*, 2006.
- [66]. V. Kolmogorov,; R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Proc. Eighth IEEE International Conference on Computer Vision*, 2001, ICCV 2001, vol.2, pp.508 - 515, July 2001.
- [67]. J. Sun, N. N. Zheng, H. Y. Shum, "Stereo Matching using Belief Propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.25, no.7 , pp. 787-800, July 2003.
- [68]. L. Lucchese and S. K. Mitra, "Color Image Segmentation: A State-of-the-art Survey," in *Proc. Indian National Science Academy(INSA-A)*, vol. 67, A, New Delhi, India, pp.207-221, Mar. 2001.
- [69]. L. Shafarenko, M. Petrou, and J. Kittler, "Automatic Watershed Segmentation

- of Randomly Textured Color Images,” *IEEE Trans. on Image Processing*, vol. IP-6, no. 11, pp. 1530-1544, Nov. 1997.
- [70]. S. A. Shafer, “Using Color to Separate Reflection Components,” *Color Research and Application*, vol. 10, no. 4, pp. 210-218, 1985.
- [71]. O. Faugeras, B. Hotz, H. Matthieu, T. Vieville, Z. Zhang, P. Fua, E. Theron, L. Moll, G. Berry, J. Vuillemin, P. Bertin, and C. Proy, “Real Time Correlation-Based Stereo: Algorithm, Implementations and Application,” *INRIA, Tech. Rep. 2013*, 1993.
- [72]. C. Cuadrado, A. Zuloaga, J. L. Martin, J. Lazaro, and J. Jimenez, “Real-Time Stereo Vision Processing System in a FPGA,” in *Proc. IEEE Industrial Electronics, IECON 2006*.
- [73]. A. Darabiha, W. J. MacLean, and J. Rose, “Reconfigurable hardware implementation of a phase-correlation stereo algorithm,” *Machine Vision Application*, vol. 17, no. 2, pp. 116-132, 2006.
- [74]. J. Diaz, E. Ros, R. Carrillo, and A. Prieto, “Real-Time System for High-Image Resolution Disparity Estimation,” *IEEE Trans. on Image Processing*, vol. 16, no. 1, pp.280-285, Jan. 2007.
- [75]. S. Sabihuddin, and W. J. MacLean, “Maximum-Likelihood Stereo Correspondence using Field Programmable Gate Array,” in *Proc. International Conference on Computer Vision Systems (ICCV)*, 2007.
- [76]. O. Veksler, “Stereo Correspondence by Dynamic Programming on a Tree,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005.
- [77]. Y. Deng and X. Lin, “A Fast Line Segment Based Dense Stereo Algorithm Using Tree Dynamic Programming,” in *Proc. European Conference on Computer Vision (ECCV)*, 2006.
- [78]. C. Lei, J. Selzer, and Y. H. Yang, “Region-Tree based Stereo Using Dynamic Programming Optimization,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2006.
- [79]. N. Y. C. Chang and T. S. Chang, “A Scalable Graph-cut Engine Architecture for Stereo Vision,” in *Proc. VLSI Design/CAD Symposium*, 2007.
- [80]. S. Park and H. Jeong, “VLSI Architecture for MRF Based Stereo Matching,” in *Proc. International Symposium on Systems, Architectures, Modeling and Simulation (SAMOS)*, 2007.
- [81]. P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient Belief Propagation for Early Vision,” in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, 2004.
- [82]. Bor-Sung Liang et al., “Index Rendering: Hardware-Efficient Architecture for 3-D Graphics in Multimedia System,” *IEEE Trans. on Multimedia*, vol.4, no.2,

Jun. 2002.

- [83]. D. Kim et al., "An SoC With 1.3 Gtexels/s 3-D Graphics Full Pipeline for Consumer Applications", IEEE JSSC, vol.41, no.1, Jan. 2006.
- [84]. M.Deering et al., "The triangle processor and normal vector shader: A VLSI system for high-performance graphics," in Proc. SIGGRAPH, pp21-30, 1998.
- [85]. David Blythe, Microsoft Corporation, "The Direct3D 10 System," 2006.
- [86]. H. Gouraud, "Continuous Shading of Curved Surfaces," Comm. ACM, vol. 18, 1971.
- [87]. Bui Tuong Phong, "Illumination for computer generated pictures," Comm. ACM, vol.18, Jun. 1975.
- [88]. A.M. Abbas, L. Szirmay-Kalos, and T. Horvath, "Hardware Implementation of Phong Shading Using Spherical Interpolation," Periodica Polytechnica, vol.44, 2000.
- [89]. A. Hast, T. Barrera, and E. Bengtsson, "Faster Shading by Equal Angle Interpolation of Vectors," IEEE Trans. on Visualization and Comp. Graphics, vol.10, no.2, Mar. 2004.
- [90]. J.N. Mitchell Jr., "Computer Multiplication and Division Using Binary Logarithms," IRE Trans. Electronic Computers, vol.11, pp.512-517, Aug. 1962.
- [91]. H. Kim, B.G. Nam, J.H Sohn and H.J. Yoo, "A 231MHz, 2.18mW 32-bit Logarithmic Arithmetic Unit for Fixed-Point 3D Graphics System," IEEE JSSC, vol.41, no.11, Nov. 2006.
- [92]. M. Combet, H. Zonneveld, and L. Verbeek, "Computation of the Base Two Logarithm of Binary Numbers," IEEE trans. Electronic Computers, vol. 14, Dec. 1965.
- [93]. K.H. Abed and R.E. Siferd, "CMOS VLSI Implementation of a Low-Power Logarithmic Converter," IEEE Trans. on Computers, vol.52, no.11, Sep. 2003.
- [94]. K.H. Abed and R.E. Siferd, "CMOS VLSI Implementation of a Low-Power Antilogarithmic Converter," IEEE Trans. on Computers, vol.52, no.11, Sep. 2003.
- [95]. Thomas A. Brubaker and John C. Becker, "Multiplication Using Logarithms Implemented with Read-Only Memory," IEEE Trans. on Computers, vol.24 no.8, Aug. 1975.
- [96]. F. Bensaali, A. Amira and A. Bouridane, "Accelerating matrix product on reconfigurable hardware for image processing applications," IEE Proc.-Circuits Devices Syst., vol. 152, no. 3, pp. 236-246, Jun. 2005.
- [97]. J.-H. Sohn, J.-H. Woo, M.-W. Lee, H.-J. Kim, R. Woo, and H.-J. Yoo, "A 155-mW 50-Mvertices/s Graphics Processor With Fixed-Point Programmable Vertex Shader for

- Mobile Applications,” IEEE Journal of Solid-State Circuits, vol. 41, no. 5, pp. 1081-1091, May 2006.
- [98]. A. Kugler, “The Setup for Triangle Rasterization,” Proc. 11th Eurographics Workshop Computer Graphics Hardware, pp. 49-58, Aug. 1996.
- [99]. M. Deering, “Geometry Compression,” ACM SIGGRAPH Proceedings, 1995.
- [100]. Aaftab Munshi et al., “OpenGL ES Common/Common-Lite Profile Specification Version 1.1.10 (Full Specification),” The Khronos Group Inc. , Apr. 2007.
- [101]. J. Foley, A. van Dam, S. Feiner, J. Hughes, “Computer Graphics: Principles and Practice,” Addison-Wesley, 1990.
- [102]. AKENINE-MÖLLER T., STRÖM J, “Graphics for the Masses: A Hardware Rasterization Architecture for Mobile Phones,” ACM Transactions on Graphics, vol. 22 , pp.801-808, Issue 3, Jul. 2003.
- [103]. Cheng-Hsien Chen and Chen-Yi Lee, “Two-level Hierarchical Z-buffer with Compression Technique for 3D Graphics Hardware,” The Visual Computer, Springer, vol. 19, no. 7-8, pp. 467-479, Dec. 2003.
- [104]. Chang-Hyo Yu and Lee-Sup Kim, “A Hierarchical Depth Buffer for Minimizing Memory Bandwidth in 3D Rendering Engine: Depth Filter,” ISCAS '03, vol.2, pp.II-724- II-727, May 2003.
- [105]. DEROO J., MOREIN S., FAVELA B., WRIGHT M., “Method and Apparatus for Compressing Parameter Values for Pixels in a Display Frame,” In US Patent 6,476,811.
- [106]. Jon Hasselgren and Tomas AkenJu-ine-Möller, “Efficient Depth Buffer Compression,” In Graphics Hardware, pp. 103-110, 2006.
- [107]. MOREIN S., “ATI Radeon HyperZ Technology,” in Hot3D Proc. ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware, Aug. 2000.
- [108]. MOREIN S., “Method and Apparatus for Efficient Clearing of Memory,” In US Patent 6,421,764.
- [109]. MOREIN S., WRIGHT M., YEE K., “Method and Apparatus for Controlling Compressed Z Information in a Video Graphics System,” US Patent 6,636,226.
- [110]. Ned Greene, Michael Kass, and Gavin Miller, “Hierarchical Z-Buffer Visibility,” Proceedings of SIGGRAPH 93, pp.231 – 238, 1993.
- [111]. Orenstein et al., “Z-Compression Mechanism,” In US Patent 6580427.
- [112]. Van Hook et al., “Compression and Decompression of Data Using Plane Equations,” In US Patent 7242400.
- [113]. Van Dyke et al., “Method and Apparatus for Managing and Accessing Depth Data in a Computer Graphics System,” In US Patent 6961057.
- [114]. Van Hook, “Method and Apparatus for Compression and Decompression of Z Data,” In US Patent 6630933.

- [115]. You-Ming Tsao, Chi-Ling Wu, Shao-Yi Chien, and Liang-Gee Chen, "Adaptive Tile Depth Filter for the Depth Buffer Bandwidth Minimization in the Low Power Graphics Systems," ISCAS 2006, pp. 5023-5026, May 2006.
- [116]. Jonathan Corbet et al, "Linux Device Drivers, 3rd," .
- [117]. Keronos Group, "OpenGL ES 1.1.10 Specification," 2007.
- [118]. OpenSceneGraph, <http://www.openscenegraph.org/projects/osg>, 2007.
- [119]. Sun Microsystems, "The Java 3D API Specification Version 1.2," Apr. 2000.
- [120]. M. Tanimoto, "Preliminary FTV Model and Requirements", *MPEG Document N9168*, Geneva, July, 2007.
- [121]. A. Vetro, P. Pandit, H. Kimata, A. Smolic, "Joint Multiview Video Model (JMVM) 5.0", *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-X207*, 2007.
- [122]. A. Vetro, P. Pandit, H. Kimata, A. Smolic, "Joint Draft 4.0 on Multiview Video Coding", *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-X209.doc*, 2007.
- [123]. Ye-Kui Wang, Ying Chen, and Miska M. Hannuksela, "Time-first coding for multi-view video coding," *ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, JVT-U104.doc*, 2006.
- [124]. IFA: Report on IFA – 2001 Consumer Electronics Show to the Members of Digital Broadcasting Australia by Tim O'Keefe, Director of Digital Business Consulting Pty Limited, Sept. 2001.
- [125]. http://www.cns.net.tw/company_info.php
- [126]. J. Oh and R. Park, "Reconstruction of Intermediate Views from Stereoscopic Images Using Disparity Vectors Estimated by the Geometrical Constraint," *IEEE Trans. on Circuits Syst. for Video Technol.*, no. 5, 2006.
- [127]. M. S. J. McVeigh and A. Jordan, "Intermediate View Synthesis Considering Occluded and Ambiguously Referenced Image Regions," *Signal Processing-Image Communications*, pp. 21--28, 1996.
- [128]. G. Sharma, A. Kumar, and S. Kamal, "Novel View Synthesis Using a Translating Camera," *Pattern Recognition Letters*, pp. 483-492, 2005.
- [129]. H. Bao, L. Chen, and J. Ying, "Non-linear View Interpolation," *Journal OF Visualization and Computer Animation*, pp. 233-241, 1999.
- [130]. R. Wang and Y. Wang, "Multiview Video Sequence Analysis, Compression, and Virtual Viewpoint Synthesis," *IEEE Trans. on Circuits Syst. for Video Technol.*, no. 3, 2000.
- [131]. M. Tanimoto, "Overview of Free Viewpoint Television," *Signal Processing: Image Communication*, vol.21, no.6, 2006, pp.454-461.
- [132]. Fujii, Toshiaki; Kimoto, Tadahiko; Tanimoto, Masayuki, "Ray space

- representation for 3D image processing", Proc. SPIE Vol. 3012, p. 330-336, May 1997.
- [133]. T. Fujii, M. Tanimoto, "Free viewpoint TV system based on ray-space representation," Proc. SPIE ITCOM 2002, pp.175-189, August 2002.
- [134]. T. Kobayashi, T. Fujii, T. Kimoto, and M. Tanimoto, "Interpolation of ray-space data by adaptive filtering," Proc. SPIE, Electron. Imag., vol.3958, pp.252-259, Jan. 2000.
- [135]. Q. H. Mahmoud, J2ME for Home Appliances and Consumer Electronics Devices, Sun Microsystems White Paper, Jan. 2003.
- [136]. Digital Video Broadcasting (DVB), Multimedia Home Platform (MHP) Specification 1.0.2, ETSI TS 101 812, June, 2002.
- [137]. T. Lindholm and F. Yelling, The Java Virtual Machine Specification, Addison-Wesley, 1996.
- [138]. A. Krall, K. Ertl, and M. Gschwind, Java VM Implementation: Compilers versus Hardware, *John Morris (ed.), Computer Architecture (ACAC '98)*, Perth, pp. 101-110, 1998.
- [139]. Martin Schoberl, JOP: A Java Optimized Processor for Embedded Real-Time Systems, Ph.D. Thesis, Tech. Universitaet Wien, Jan 2005.
- [140]. A. Kim and M. Chang, "Designing a Java Microprocessor Core Using FPGA Technology," *Computing & Control Engineering Journal*, June 2000, pp.135-141.
- [141]. Tan, Y.Y. Yau, C.H. Lo, K.M. Yu, W.S. Mok, P.L. Fong, A.S. "Design and implementation of a Java processor," *IEE Proceedings*, Volume: 153 , On page(s): 20 – 30, 2006 .
- [142]. Ramesh Radhakrishnan, Ravi Bhargava, Lizy K. John. Improving Java performance using hardware translation. *ACM Press*. June 2001 Pages: 427 – 439.
- [143]. ARM. Jazelle – ARM Architecture Extensions for Java Applications. White paper.
- [144]. Andreas Krall. Efficient JavaVM Just-in-Time Compilation. In Proceedings of the 1998 International Conference on Parallel Architectures and Compilation Techniques (PACT '98), pages 205-212, Paris, October 12-18, 1998. IEEE Computer Society Press.
- [145]. Georg Acher. JIFFY — Ein FPGA-basierter Java Just-in-Time Compiler für eingebettete Anwendungen. PhD thesis, Technische Universität at München, 2003.
- [146]. Sun. *picoJava-II Microarchitecture Guide*. Sun Microsystems, March 1999.
- [147]. D.S. Hardin. Real-Time Objects on the Bare Metal: An Efficient Hardware

- Realization of the Java™ Virtual Machine. In *Proceedings of the Fourth International Symposium on Object-Oriented Real-Time Distributed Computing*, page 53. IEEE Computer Society, 2001.
- [148]. PTSC. IGNITE Processor Brochure, Rev 1.0. Available at <http://www.ptsc.com>.
- [149]. R. Zulauf. Entwurf eines Java-Mikrocontrollers und prototypische Implementierung auf einem FPGA. Master's thesis, University of Karlsruhe, 2000.
- [150]. S.A. Ito, L. Carro, and R.P. Jacobi. Making Java Work for Microcontroller Applications. *IEEE Design & Test of Computers*, 18(5):100–110, 2001.
- [151]. Ramesh Radhakrishnan, Deependra Talla and Lizy Kurian John, “Allowing for ILP in an Embedded Java Processor,” *ACM SIGARCH Computer Architecture News*, pp. 294-305, 2000.
- [152]. Zhilei Chai, Wenke Zhao, Wenbo Xu. System On Chip Design And Software Supports (SODSS): Real-Time Java Processor Optimized for RTSJ, *ACM Press*. March 2007
- [153]. R. Radhakrishnan, *Microarchitectural Techniques to Enable Efficient Java Execution*, PhD thesis, University of Texas at Austin, 2000.
- [154]. C. J. Glossner. *The DEFLT-JAVA Engine*, Ph.D. thesis, Delft University of Technology, 2001.

國科會補助專題研究計畫項下出席國際學術會議心得報告

日期：99 年 12 月 16 日

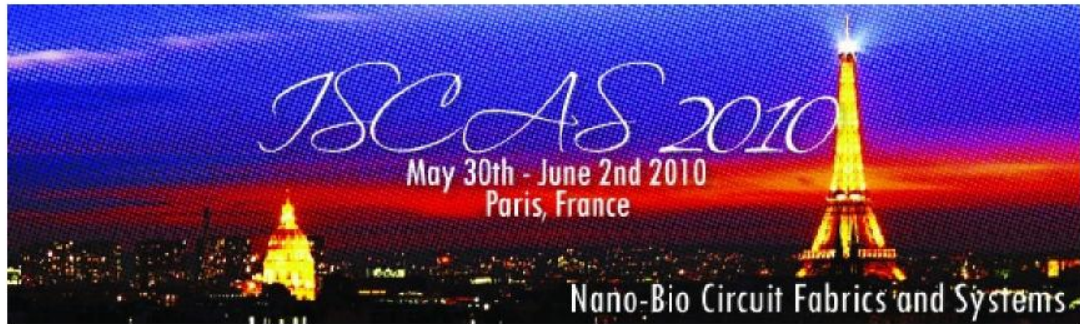
計畫編號	NSC 98-2220-E-009-012-		
計畫名稱	支援 3-D 立體視訊的數位電視多媒體平台設計(3/3)		
出國人員姓名	范倫達	服務機構及職稱	國立交通大學資訊工程系助理教授
會議時間	99 年 5 月 30 日至 99 年 6 月 2 日	會議地點	法國巴黎
會議名稱	(中文) IEEE 電路與系統國際研討會 (英文) IEEE International Symposium on Circuits and Systems (ISCAS)		

一、參加會議經過

個人乘坐由台北出發之長榮航班抵達法國的機場，個人前往會議地點向大會報到領取議程如圖一所示，確認個人要主持會議的時間與聆聽的論文發表議程，大會邀請主持函如圖二所示。此行目的有三：第一、成為 ISCAS CAS-COM Track 之 Technical Committee 的成員；第二、擔任主持 “C2L-E: Simulation & Modeling of Nano-electronics & Giga-scale Systems” Session 會議；第三、由於個人目前擔任此計畫之共同主持人，因此將聆聽重點放在多媒體與通訊與 Gigascal 系統的論文，進而了解研究的趨勢，獲益良多。在議程結束後，本人搭機返台，結束此次參訪 ISCAS 大會之行程。



圖一、2010 IEEE International Symposium on Circuits and Systems (ISCAS)之大會入口。



8 March 2010

Lan-Da Van
National Chiao Tung University
Department of Computer Science, Rm. EC419
1001 Ta Hsueh Road, Hsinchu 300
TAIWAN, R.O.C.

Dear Lan-Da Van:

On behalf of the 2010 IEEE International Symposium on Circuits and Systems (ISCAS), we are pleased to invite you to attend ISCAS 2010, to be held in Paris, France from May 30th – June 2nd, 2010, at the Hotel New York in Disneyland Paris.

The Symposium will focus on circuits and systems employing nano-devices and circuit fabrics, and their implementation cost, switching speed, energy efficiency, and reliability. The ISCAS 2010 will include oral and poster sessions; tutorials given by experts in state-of-the-art topics; and special sessions, with the aim of complementing the regular program with topics of particular interest to the circuits and systems community that cut across and beyond disciplines traditionally represented at ISCAS. Your participation as a Session Chair of "C2L-E: Simulation & Modeling of Nano-electronics & Giga-scale Systems" will indeed contribute to the success of this meeting.

We understand that you will bear all travel expenses and will have all necessary insurances during this period. We look forward to seeing you at ISCAS 2010 in Paris.

Sincerely,

Prof. Amara Amara
General Chairman, 2010 IEEE Int'l Symposium on Circuits & Systems
<http://www.iscas2010.org>
Deputy Managing Director, Research and International Cooperation
ISEP, 21 Rue d'Assas 75270
Paris cedex 06, FRANCE
Tel: +33 1 49545273
Fax: +33 1 49 54 52 51
Email: amara.amara@isep.fr

圖二、ISCAS 大會邀請主持 Session 之電子信函。

二、與會心得

在大會之 Keynote talk 方面，個人聆聽大師 Prof. Giovanni De Micheli (at EPFL Lauzanne) 之演講，主題為：Nanosystems: devices, circuits, architectures and applications。參與聆聽人

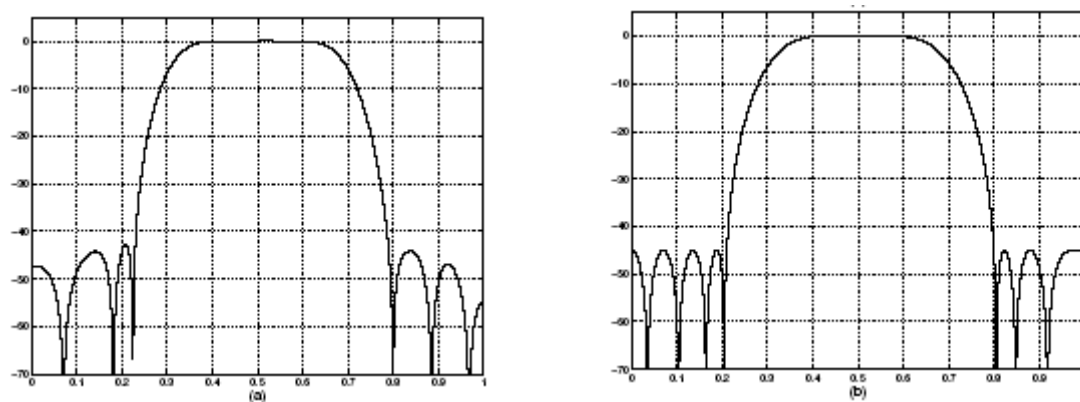
數相當眾多，其演講內容摘要如下。

基於未來我們的生活會大受奈米的技術影響，所以3D技術奈米元件之整合就相對重要且是未來主流的技術，也因為此技術的誕生，使得新式且具有較高複雜度的運算架構，如3D NOC架構也得以實現，其中NOC架構是一種提供on-chip溝通的方式，但如何Modeling以及如何讓此架構具有相當的彈性也是一大挑戰。最後Prof. Giovanni De Micheli亦指出abstraction of computation, the nanosystem architecture, the technological feasibility envelope與the multivariate design optimization problems均是未來3D技術奈米元件之整合需要面對的挑戰與研究契機。

在6/2日，個人與另外一位教授(Prof. Rose)共同完成主持“C2L-E: Simulation & Modeling of Nano-electronics & Giga-scale Systems” Session會議。同天，個人經由Prof. Sobelman (at University of Minnesota)推薦且成員的投票，順利成為核心成員，日後要負責大會相關的論文審查與第一層決策事務。

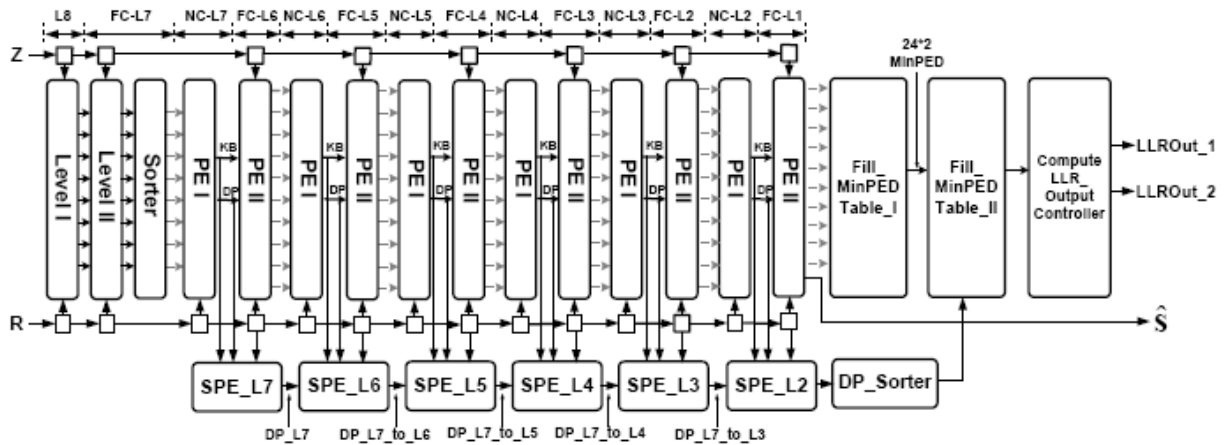
最後與會的過程之中深感多媒體與通訊與SOC設計乃目前工程最熱門之研究主題，故聆聽許多場次，在此僅列舉分享兩篇相關之技術報告，分別敘述如下。

第一篇為 Prof. Wu-Sheng Lu 等人所發表之” Digital Filters with Sparse Coefficients” 論文。FIR 濾波器不論在多媒體或是通訊端均是重要且不可或缺的設計，若要將系統設計成低功耗第一步就要考慮如何將此濾波器設計成低功耗，此篇提出一個論點就是將濾波器之係數設計成 Sparse Coefficients，其頻譜特性如圖三所示，此方法好處是在硬體設計上會比較有效率。雖然此作者並非從事硬體/晶片之研究，但從個人的眼光這方式應該未來可以應用在晶片設計上，成為有效降低功耗的方式。



圖三: Bandpass filter using (a) sparse coefficients (b) non-sparse coefficients. [1]

第二篇為 Dimpesh Patel 等作者所發表之” VLSI Implementation of a WiMAX/LTE Compliant Low-Complexity High-Throughput Soft-Output K-Best MIMO Detector” 論文。MIMO 偵測也是多媒體通訊系統不可或缺之技術，目前有許多偵測技術一直被探討如:K-Best, MMSE, Parallel, V-BLAST 等偵測方式，其探討之重點包含:1)運算複雜度降低, 2)改善 BER 性能, 3)達到 Energy Efficiency。此篇就是針對 K-Best 提出改善的 Pipeline 架構如圖四所示，因其架構特性關係與採用下列三個想法: 1) relevant discarded paths selection, 2) last stage on-demand expansion, 3)relaxed LLR computation，故其吞吐量(Throughput)可以達到 2GBPS。其比較表如表一所示。



圖四、作者提出 Pipeline 架構圖 [2]

表一、文獻比較表 [2]

Reference	[3]	[6]	This work 0.13 μ m	This work 65nm
Modulation	64-QAM	64-QAM	64-QAM	64-QAM
Method	K-Best	K-Best	K-Best	K-Best
<i>K</i> -value	64	64	10	10
Process	0.13 μ m	65 nm	0.13 μ m	65 nm
Core Area	5270 KG	174 KG	174 KG	298 KG
Max Freq.	270 MHz	200 MHz	270 MHz	833 MHz
Throughput	100 Mbps	115 Mbps	655 Mbps	2000 Mbps
Latency	N/A	N/A	0.7 μ s	0.23 μ s
Power	847 mW	11 mW	195 mW	280 mW
Energy/bit	8470pJ/b	96pJ/b	298pJ/b	140pJ/b
Soft/Hard	Soft	Soft	Soft	Soft
SNR Dep.	Yes	Yes	No	No
Domain	Complex	Complex	Real	Real

三、考察參觀活動(無是項活動者略)
無。

四、建議

ISCAS 是國內這個領域重量級學者與年輕學者每年均會前往參與之會議，在此會議上，可以與國外大師面談並可以參與一些重要的 Technical Committee，但參與 Technical Committee 需要有人引見與推薦，否則會不得其門而入，推薦後也必須經過投票通過，才能成為該 TC 核心成員，此次個人就是經由 Prof. Sobelman (at University of Minnesota) 推薦才有此機會。另外，低能量/低耗能/整合多功能之多媒體與通訊設計絕對是未來電子產品呈現與溝通的設計重點，另一方面也要符合標準，後者將是產品化的門檻。歐洲與中國大陸在多媒體與通訊方面的進展速度相當快且經費相當充裕且標準通常以他們為主，所以我們更應該將經費投注在重點研究上。總結來說，該次大會所接受之論文絕大部分為高水準之多媒體與通訊與 SOC 之論文研究。

五、攜回資料名稱及內容

本次研討會本人攜回論文數位資料 CD 乙份。在此特別感謝國科會計畫之支持，讓個人對目前與次世代的多媒體與通訊與 SOC 技術與研究有相當豐富的收獲，再次感謝！

六、其他

無。

七、參考文獻

[1] Wu-Sheng Lu and Takao Hinamoto, "Digital Filters with Sparse Coefficients," in Proc. IEEE ISCAS, pp. 169-172, 2010.

[2] Dimpesh Patel, Vadim Smolyakov, Mahdi Shabany, P. Glenn Gulak, "VLSI Implementation of a WiMAX/LTE Compliant Low-Complexity High-Throughput Soft-Output K-Best MIMO Detector," in Proc. IEEE ISCAS, pp. 593-596, 2010.

國科會補助計畫衍生研發成果推廣資料表

日期:2011/01/28

國科會補助計畫	計畫名稱: 支援3-D立體視訊的數位電視多媒體平台設計(3/3)
	計畫主持人: 蔡淳仁
	計畫編號: 98-2220-E-009-012- 學門領域: 晶片科技計畫--目標導向型研究計畫

無研發成果推廣資料

98 年度專題研究計畫研究成果彙整表

計畫主持人：蔡淳仁		計畫編號：98-2220-E-009-012-					
計畫名稱：支援 3-D 立體視訊的數位電視多媒體平台設計(3/3)							
成果項目		量化			單位	備註（質化說明：如數個計畫共同成果、成果列為該期刊之封面故事...等）	
		實際已達成數（被接受或已發表）	預期總達成數(含實際已達成數)	本計畫實際貢獻百分比			
國內	論文著作	期刊論文	0	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	5	0	100%		
		專書	0	0	100%		
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力 (本國籍)	碩士生	0	0	100%	人次	
		博士生	0	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		
國外	論文著作	期刊論文	5	0	100%	篇	
		研究報告/技術報告	0	0	100%		
		研討會論文	23	0	100%		
		專書	0	0	100%		章/本
	專利	申請中件數	0	0	100%	件	
		已獲得件數	0	0	100%		
	技術移轉	件數	0	0	100%	件	
		權利金	0	0	100%	千元	
	參與計畫人力 (外國籍)	碩士生	19	0	100%	人次	
		博士生	5	0	100%		
		博士後研究員	0	0	100%		
		專任助理	0	0	100%		

<p>其他成果 (無法以量化表達之成果如辦理學術活動、獲得獎項、重要國際合作、研究成果國際影響力及其他協助產業技術發展之具體效益事項等，請以文字敘述填列。)</p>	<p>目前本計劃相關人員積極參與國際標準組織 ITU-T VCEG 及 ISO/IEC MPEG，未來有機會將本子項之技術推展到標準組織當中。另外，本計畫參與學生也以本計畫開發之部份成果參加民國 99 年教育部主辦之嵌入式系統設計競賽獲得佳作。</p>
--	--

	成果項目	量化	名稱或內容性質簡述
科 教 處 計 畫 加 填 項 目	測驗工具(含質性與量性)	0	
	課程/模組	0	
	電腦及網路系統或工具	0	
	教材	0	
	舉辦之活動/競賽	0	
	研討會/工作坊	0	
	電子報、網站	0	
	計畫成果推廣之參與(閱聽)人數	0	

國科會補助專題研究計畫成果報告自評表

請就研究內容與原計畫相符程度、達成預期目標情況、研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）、是否適合在學術期刊發表或申請專利、主要發現或其他有關價值等，作一綜合評估。

1. 請就研究內容與原計畫相符程度、達成預期目標情況作一綜合評估

達成目標

未達成目標（請說明，以 100 字為限）

實驗失敗

因故實驗中斷

其他原因

說明：

2. 研究成果在學術期刊發表或申請專利等情形：

論文： 已發表 未發表之文稿 撰寫中 無

專利： 已獲得 申請中 無

技轉： 已技轉 洽談中 無

其他：（以 100 字為限）

3. 請依學術成就、技術創新、社會影響等方面，評估研究成果之學術或應用價值（簡要敘述成果所代表之意義、價值、影響或進一步發展之可能性）（以 500 字為限）

本計畫所開發的系統，包含四大關鍵技術：視差估測演算法、可重組式 3D 繪圖加速器、3-D 視訊編碼、及異質雙核心的 Java 處理器，針對每一項關鍵技術，我們都提出了學術上及技術上的創新。我們在此計畫下發表了四篇 IEEE transaction 期刊論文、一篇 ACM transaction 期刊論文、二十八篇國內外研討會論文、另外有兩篇 IEEE transaction 期刊論文在審查中、以及一篇 ACM transaction 期刊論文準備投稿中。

而且我們也成功地把計畫所全新開發的三個複雜電路 IPs（一顆 Java 處理器核心、一個 3-D 繪圖加速核心、以及一個視差估測電路）成功的整合在同一個 SoC 中（以 FPGA 驗證）。並寫出完整的系統軟體來進行即時展示 3-D 立體的視訊，圖形，以人機介面輸出。本計畫所設計的系統，儘可能符合現有國際標準，因此所開發的技術可以讓業界參考，大幅縮減開發下一代 3-D 數位電視應用及多媒體系統的時間。目前台灣在大型、複雜的嵌入式系統軟硬體整合設計方面較缺乏人才。本計畫訓練出來的學生，將可獲得相當紮實的相關技術訓練，對未來台灣產業昇級會有相當的助益。