

行政院國家科學委員會補助專題研究計畫成果報告

多媒體系統晶片設計技術之研究—總計畫 (3/3) 子計劃三：單晶片多媒體系統整合技術之研究

計畫類別： 個別型計畫 整合型計畫

計畫編號：NSC89 - 2218 - E - 009 - 080 -

執行期間： 89年 8月 1日至 90年 7月 31日

計畫主持人：李鎮宜教授

本成果報告包括以下應繳交之附件：

赴國外出差或研習心得報告一份

赴大陸地區出差或研習心得報告一份

出席國際學術會議心得報告及發表之論文各一份

國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學電子工程學系

中 華 民 國 九 十 年 十 月 二 十 八 日

行政院國家科學委員會專題研究計畫成果報告

多媒體系統晶片設計技術之研究—總計畫 (3/3)

子計劃三：單晶片多媒體系統整合技術之研究

The Study of System-on-Chip (SOC) Design Technology for Multimedia System Integration

計畫編號：NSC89-2218-E-009-080

執行期限：89年8月1日至90年7月31日

主持人：李鎮宜 國立交通大學電子工程學系

計畫參與人員：謝百舉、陳正賢、游政謀、王經楷、蔡侑庭 國立交通大學電子工程學系

一、 中文摘要：

本計畫主要是針對在於單晶片設計中多媒體系統整合技術之研究，在此我們的多媒體系統為一三度空間電腦繪圖系統，主要的研究內容為如何降低記憶體頻寬瓶頸以提升整體繪圖系統之效能。為解決記憶體頻寬的問題，我們分別從三方面著手，一為降低從處理器或主記憶體進入繪圖系統之頻寬，主要是利用貝式(Bezier)曲線來表示物體的幾何外觀，如此可大幅降低幾何資料的傳輸，並設計一適合於硬體實現的簡化曲線切割(tessellation)方法；第二個方法是降低繪圖晶片內部不必要的運算和 Z-Buffer 的頻寬，在此我們提出一 two-level hierarchical Z-Buffer 架構，藉由此架構我們可在前端就可以除去一些隱藏的三角形，並且在 rasterization 單元中可以快速地除去隱藏的點以及減少 Z-Buffer 存取的次數，使整體隱藏面消除的效率提升；第三個方法是對貼圖的材質作資料壓縮，以達到降低貼圖所佔記憶體容量及貼圖單元所需之頻寬；綜合以上三個方法，我們可以大幅降低整體系統所需之記憶體頻寬，進而提升整體效能。

關鍵字：單晶片、多媒體系統、三度空間電腦繪圖、貝式曲線切割、階層式深度測試、材質壓縮。

英文摘要：

In this project, we focus on the integration techniques of multimedia system-on-chip. Our multimedia system is a three-dimensional computer graphics rendering system.

The research is focus on how to reduce the memory bandwidth bottleneck and improve the overall system performance. We develop three techniques to solve the bandwidth problem. First we reduce the bandwidth requirement from CPU and main memory to graphics system. This is done by using Bezier curves to represent the geometry of objects. And we also develop a simple Bezier curves tessellation algorithm, which is suitable for hardware implementation. Second we reduce the Z-buffer bandwidth and unnecessary operations in graphics chip. We propose a two-level hierarchical Z-Buffer architecture. By using this technique, we can discard hidden triangles at front-end pipeline and discard hidden pixels in rasterization stage. This can reduce the Z-Buffer access and improve the efficiency of hidden surface removal. The third technique is texture compression. It can reduce the texture memory size and the bandwidth requirement of texture mapping hardware. By combining above three techniques, we can dramatic reduce the bandwidth requirement of the system and boost the overall performance.

Keyword : SOC, System On Chip, Multimedia System, 3D Graphics, Bezier curve tessellation, Hierarchical Z-Buffer Test, Texture Compression.

二、 計畫緣由與目的：

在三度空間電腦繪圖之多媒體系統中，記憶體頻寬乃是目前限制整個系統效能的主要瓶頸，雖然處理器的速度大幅提昇，然而記憶體存取的速度卻始終跟不上運算單元運作的速度，因此縱使再快速的運算單元仍無法大幅提昇整體的效能，在此我們選擇研究降低頻寬需求的技術以求提升整個三度空間多媒體系統晶片之整體效能。而從整個 3D 電腦繪圖的架構上來看，主要的頻寬瓶頸在三個部分，一是在資料由主記憶體或 CPU 傳送到加速卡上這部分的頻寬，二是繪圖晶片上和外部記憶體的存取上，這邊主要包含兩部分，一部份是 Z-Buffer 存取，另一部份是貼圖存取，此三個部分是目前耗費記憶體頻寬最大的部分，因此我們分別從這三方面著手解決頻寬不足的問題。

三、 研究方法與成果：

(1) 貝式曲面切割演算法

在 3D 繪圖中一般都是使用三角形來表示物體的資料，但是三角形的資訊量很大，會造成頻寬上的瓶頸，而一般 CAD 軟體大都會採用曲線來建製物體的外觀，表示曲線所需的資料量相對地少很多，但是目前的硬體架構並不支援曲線的表現方式，因此我們嘗試研究一適合於硬體架構的曲線切割演算法，如此一來物件可直接用曲線方程式來描述，大大降低前級所需要的資料量，在此我們採用的曲線是一般常用的貝式曲線。

簡化式的貝式曲面切割[4]演算法是由一致性的貝式曲面切割演算法[1][2][3]所改良而成，不但能夠如適合性貝式曲面演算法一樣能夠有效的防止產生不必要的三角形，並且在複雜度方面比適合性貝式曲面演算法更低，在硬體設計上更容易實現以及對其作最佳化。

以一個貝式曲面為例來說明此演算法的流程和內容如下：

- 1.使用一致性貝式曲面切割法來切割貝式曲面，切割之後的三角形網路如圖一所示。
- 2.將步驟 1 所切割出來的三角形分成一個個

的區塊，每個區塊以 $N \times N$ 個三角形頂點所組成，如圖一的 B 所示。

3.將步驟 2 中分隔出來的區塊，分別一一作三角形頂點簡化的處理，為了讓各區塊之間能夠個別獨立的作處理，保留其區塊邊界部分(如圖一所標示的各點)不作處理，只針對內部的三角形頂點作簡化的運算。

4.對區塊內部的三角形計算其所謂的比重值，首先算出此一頂點與相鄰頂點之平均中心點 v 與平均法向量 n ， v 與 n 及構成了所謂的平均平面 P ，接著再以此頂點與平均平面的距離來當作此頂點的比重值

5.將步驟 4 所計算出來的比重值與事先所設定好的容忍值作比較，若比重值小於容忍值，則判定此三角形頂點是可去除的。

6.將步驟 5 中判定可去除的三角形頂點消除，並針對區塊中剩下的三角形頂點重新規劃出新的三角形網路。

7.各區塊可以獨立作簡化的運算，因此在步驟 4 一完成之後，各區塊所切割出來的三角形網路便可立即輸出到整個 3D 繪圖硬體管線的下一級去作著色處理，最後整個曲面切割之後的三角形網路如圖二所示。

由圖一、二當中可以清楚看出，區塊 B 中較平滑的右半邊，三角形個數有明顯的減少，如此一來對於下一級的著色處理所需的運算量就大大地減少。

(2) Hierarchical Z-Buffer

圖三所示為一加入 Hierarchical Z-Buffer test 的硬體方塊圖，灰色部分是我們所加入的單元，主要的功能是希望能快速地消除掉隱藏的三角形和點，以減少不必要的運算和降低之後的記憶體存取。

傳統 3D 繪圖中所採用的隱藏面消除方式為 Z-Buffer[5]，此乃最容易實現之方法，但其效率卻很低，因此當三角形數目增加時，其頻寬需求也會相對應的提高，因此許多進階的方法被提出來解決隱藏面消除的問題，比較常見的是 BSP[7][8]，但是這些方法皆需要大量的前置處理動作，沒有辦法用硬體來實

現，因此一個硬體支援的快速隱藏面消除的演算法是必要的，而 Hierarchical Z-Buffer[6][9][10]正是一個適合於硬體實現的方法。

Hierarchical Z-Buffer 乃是將原本 Z-Buffer 做 down sampling 的動作所得的值，而 N. Greene[6]之前所提出之 hierarchical Z-Buffer 方式並不適合硬體設計，因此我們提出一適合硬體設計之 Two-level Hierarchical Z-Buffer 架構(圖三)，圖三中第一個 HZ test 是在三角形層級作隱藏面測試，若是測試不通過的話，此三角形將被剔除，如此一來可省下後面 lighting rasterize 和貼圖等等的運算及記憶體存取；第二個 HZ test 是在 rasterization stage 做，主要是希望能快速地消除隱藏的點，rasterizer 可以經由良好的設計和 HZ Test 結合進而一次消除數個點或一整條線，如此一來可提升隱藏點消除的速度。經由這兩次的 HZ test，我們可降低許多不必要的運算和記憶體存取。

在 Hierarchical Z-Buffer 建製的過程中會遇到更新 HZ buffer 值的問題，若是按照原始 hierarchical Z 的定義，則當 Z Buffer 每次更新時，我們也必須更新相對應的 HZ Buffer 的值，而要更新此值則必須比對在此 HZ Buffer 所覆蓋的區塊中所有深度的值，找到最遠的然後儲存於 HZ Buffer 中，此一動作會增加記憶體存取次數和運算的複雜度，進而讓 Hierarchical Z-Buffer 的優勢完全喪失，因此我們在 HZ Buffer Management 單元內加入一 bit mask cache 來解決 HZ Buffer 更新的問題，此 bit mask 的內容如圖四所示，其中必須儲存被描繪過點的位置及一最遠的深度值 tempZ，block 只有當在此 bit mask cache 中的 mask 都被設定為 '1' 時才將此時的 tempZ 深度值寫入 HZ Buffer 中，如此一來可確保 HZ Buffer 中所存的值是正確的。圖五中所示是 HZ Buffer management 單元更新 HZ Buffer 的流程圖，此 bit mask cache 的取代機制為 FIFO(First In First Out)並不是一般 cache 中所用的 LRU，而經由我們模擬的結果，只要 16 個 block 大小的 bit mask cache 即可達到不錯的效果。

表一列出在不同區塊大小下之 Two-level HZ buffer test 之模擬結果，在此表中我們選出四個複雜度在數十萬三角形以上的場景來模擬，看其在四個不同

HZ Buffer 配置下記憶體存取降低的程度以及三角形和圖素的消除比例，其結果顯示 two-level HZ Buffer test 可降低 10~35%的頻寬，而不同的區塊大小也會影響到三角形和圖素的消除率，對於大的區塊其三角形消除率比較高但圖素消除率比較低，反之小的區塊對三角形的消除率會降低，但圖素消除率會提高，因此在選擇 hierarchical 區塊大小時必須考慮到消除率和 HZ Buffer 的大小，最後我們所採用的是 16x16-8x8 的區塊大小，加上 16 個區塊大小的 bit mask cache 及 8bit 精確度的 HZ Buffer。

(3) 貼圖壓縮

在繪圖的運算單元中耗費最大記憶體容量和頻寬的部分為材質貼圖的部分，且會隨著所使用材質的多寡和濾波器的形式而有大幅度的變化，因此降低此部分的頻寬是一大課題，在此我們提出的方法為壓縮貼圖所需之材質，貼圖之材質因繪圖時使用的特性和在傳統二維影像上的特性有所不同，在貼圖時，必須要能快速地存取材質中任意一個區塊的資料，且壓縮後材質的失真度也不能太差，這兩項原則，使得傳統應用於二維影像壓縮的方法無法輕易套用於材質壓縮，尤其在快速存取方面，因傳統影像壓縮方法大都會造成資料長度的不同，此點造成隨機存取的困難。

目前已有少數人提出材質壓縮的方法 [11][12][13]，但這些方法一者無法達到高壓縮比 [11]，或無法達成輕易隨機存取的需求 [12][13]，因此我們研究出一可達成目前產品上兩倍壓縮比以上且影像品質仍不失真太多的壓縮方式。

我們研究的結果是修改 JPEG 的壓縮方式，使其適用於貼圖壓縮，改變的地方乃是在 Quantization 的地方採用 adaptive quantization 的方法，在此我們事先定義好幾種不同的 quantization coefficient，然後依據影像的複雜度選取適當的參數，使其壓縮出來的資料長度能趨近於我們所要求的長度，其架構如圖六所示，而表二列出在各種不同固定長度的壓縮比下的影像品質，整體影像失真仍在可接受之範圍，但壓縮比比目前方法高兩至三倍，且能達到固定長度，方便於貼圖單元擷取所需要之材質區塊。

四、 結論與討論

在此研究計畫中,我們針對整個電腦繪圖多媒體系統的頻寬瓶頸問題做探討,同時提出了三個方法分別針對系統到處理器及主記憶體的頻寬和晶片對外 Z-Buffer 和貼圖所需之頻寬做一探討,提出一簡化的貝式曲線切割方式來降低幾何物體傳輸所需之頻寬; Two-level Hierarchical Z-Buffer 用來快速消除隱藏面和點,以減少不必要之運算和 Z-Buffer 之存取;最後再加上材質壓縮的方式來減少貼圖時所需的記憶體頻寬以及儲存材質的記憶體容量,藉由這三方面的技術達到降低整體頻寬的需求,提升整個系統的效率。

多媒體單晶片的複雜度日益提升,如何快速的實現出一系統是一大挑戰,除了面臨系統內部各個單元的設計外,還必須考量到整體系統對內、對外之整合;在此我們希望藉由對 3D 繪圖系統的探討,日後能建立一系統層級的整合環境,將軟硬體設計、模擬及驗證整合在一起,如此才能縮短整個系統晶片之設計流程,並降低設計時的錯誤。

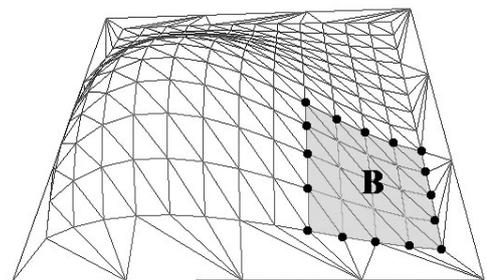
五、 參考文獻：

- [1] E. Catmull, "A Subdivision Algorithm for Computer Display of Curved Surfaces", UTEC-74-133, University of Utah, 1974
- [2] Salim S. Abi-Ezzi, Leon A. Shirman, "Tessellation of Curved Surfaces under Highly Varying Transformation", EUROGRAPHICS '91, 1991, pp. 385-397
- [3] Michael Garland, Paul S. Heckbert, "Surface Simplification Using Quadric Error Metrics", Computer Graphics(Proc. SIGGRAPH '96), Vol. 31, ACM Press, New York, 1997, pp. 204-241
- [4] David P. Luebke, "A Developer's Survey of Polygonal Simplification Algorithms", IEEE Computer Graphics and Applications, Vol. 21 Issue 3, May/June 2001, pp. 24-35
- [5] Catmull, Edwin, "Computer Display of Curved Surfaces", *Proceedings of the IEEE*

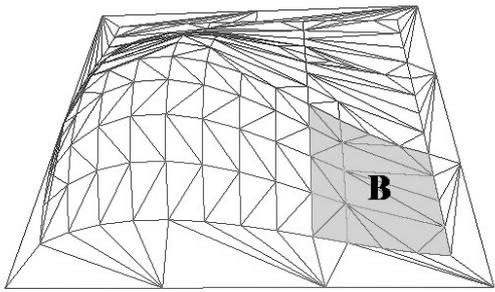
Conference on Computer Graphics, Pattern Recognition and Data Structures, pp. 11-17, May 1975

- [6] N. Greene, M. Kass, and G. Miller, "Hierarchical Z-Buffer Visibility," *Proceedings of SIGGRAPH '93*, pp. 231-238, July 1993.
- [7] Fuchs, H., Z.M. Kedem, and B.F. Naylor, "On Visible Surface Generation by a priori tree structures", *Computer Graphics*, vol. 14, no.3, pp.124-133, July 1980
- [8] Gordon, Dan and Shuhong Chen, "Front-to-back display of BSP trees", *IEEE Computer Graphics and Applications*, vol. 11, no. 5, pp.79-85, September 1991
- [9] Steve Morein, "ATI Radeon HyperZ Technology", *In: Eurographics Hardware Workshop 2000, Hot3D Panel*.
- [10] nVidia, "Technical Brief: Geforce3: Lightspeed memory architecture", <http://www.nvidia.com>
- [11] 3dfx, "FXT1 texture compression whitepaper", 3dfx technology. 1999
- [12] Microsoft, "Escalante Hardware Overview, Talisman Graphics & Multimedia System", 8 Aug 1997
- [13] Microsoft, "Method and System for Accessing Texture Data in Environments With High Latency in a Graphics Rendering System", United States Patent, Patent Number: 5880737, Mar. 9, 1999

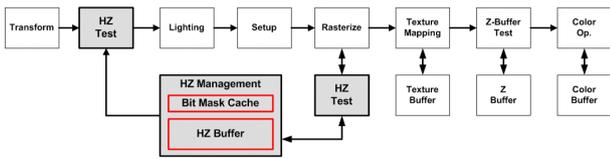
六、 圖表：



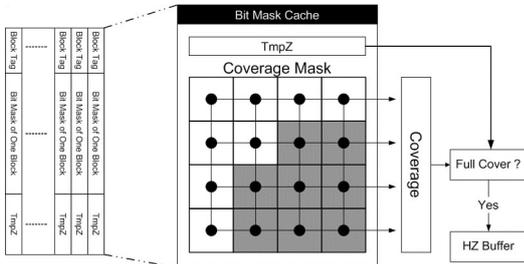
圖一 使用一致性貝式曲面切割法切割之貝式曲面
(450 個三角形)



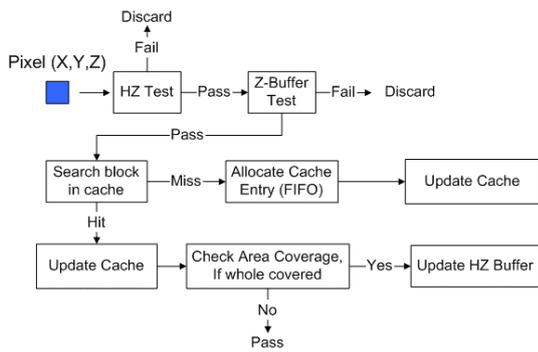
圖二 經過簡化運算過後的三角形網路
(247 個三角形)



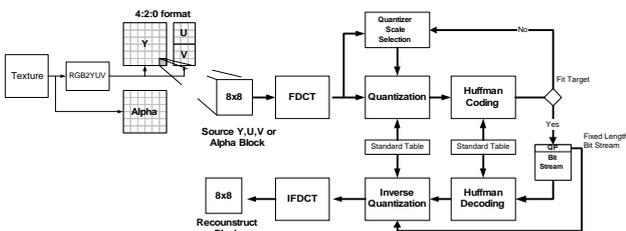
圖三 Two-level hierarchical Z-Buffer 架構



圖四 Bit-Mask Cache



圖五 HZ buffer 更新之 flow



圖六 Adaptive Quantization Texture Compression

表一 Triangles, pixels and bandwidth reduction percentage under various HZ configurations (with 64 blocks bit-mask cache and 1280x1024 resolution).

	House	Cars	Teapots	Coffee Shop	
Total Triangles	316671	454746	102400	288374	
Back Face Culling	158154	233484	52298	134545	
Memory Access	436234	387337	147354	736170	
	%	%	%	%	
8x8-4x4	Triangle	20.7	34.9	5.5	19.3
	Pixel	17.2	44.8	39.6	21.2
	Bandwidth	12.8	35	28	13.1
16x16-8x8	Triangle	23.6	34.6	12.3	18.9
	Pixel	10.2	30	28.7	20.2
	Bandwidth	9.4	27	24.8	12.9
32x32-8x8	Triangle	26.7	35.1	10.5	19.8
	Pixel	10.8	30.7	29	20.3
	Bandwidth	10.5	26.7	24.3	13
32x32-16x16	Triangle	14.9	32.5	14.4	16.4
	Pixel	4.67	14.5	14.8	17.4
	Bandwidth	5.2	18.3	19.8	11.7

表二 貼圖壓縮之影像品質

	FXT1 CR=6	S3TC CR=6	JPEG* *	TREC* **	Proposed CR=8	Proposed CR=10.67	Proposed CR=12	Proposed CR=19.2
Baboon	28.30	27.82	24.94 (15.34)	26.20 (12.16)	25.92	25.03	24.64	23.08
Lenna	34.91	35.03	31.71 (30.27)	31.32 (36.03)	33.16	32.89	32.72	31.51
Backwall	33.14	32.6	29.44 (21.10)	28.87 (24.46)	29.71	29.56	29.45	28.47
Banner	30.26	30.01	25.88 (14.39)	26.89 (12.08)	26.80	26.10	25.81	24.31
Texture	37.97	36.51	44.78	34.93	43.65	42.49	41.44	31.85

			(21.94)	(24.5)				
Venus	33.44	33.05	29.90	30.36	32.10	31.35	30.90	28.83
			(19.17)	(17.49)				

*Textures from 3D Winbench™ 2000, 3D Winbench™ is the trademark of ZD Inc.

**JPEG : UV down sample, and use default Quantizer Scale and default Huffman table

***TREC [12]: UV down sample, and use uniform quantization mode