

產學合作案結案報告書

華資工 105 產學字第 005 號

三維實體建構之研究

甲方：上奇資訊股份有限公司

乙方：中華學校財團法人中華科技大學

資訊工程系

計劃主持人：陳建成

目次

圖次	3
表次	4
摘要	5
第一章 緒論	7
第一節 研究緣起	7
第二節 研究背景	8
第二章 執行策略	10
第一節 八元樹	10
第二節 線性結構	11
第三章 研究方法	12
第四章 研究內容與結果	17

第五章 結論 20

參考書目 21

圖次

圖 3-1 ALGM() 演算法 15

圖 4-1 範例:以線性四元樹 (100, 102, 12x, 2xx, 312)

編碼 18

表次

表 3-1 節點型態的合成關係	16
---------------------------	----

摘 要

關鍵詞：八元樹、平移掃瞄、遞迴

一、研究緣起

誠如我們所知，三維實體的塑模方式在很多工程和科學的領域都是一項重大的課題，因為實體的表示法將影響資料處理與儲存空間的效率。而常見的方式有曲面表示法、分裂幾何法、掃瞄表示法、實體幾何構建法與八元樹表示法等。而實體塑模的核心內容是三維物體的數學模型和電腦表示方法，以及電腦表示法的表現方式，使用這些塑模方式的系統都在電腦內提供了對物體完整的幾何定義，因此，可以隨時獲得所需要的資訊，支援電腦輔助幾何設計與製造的任何一個層面，如電腦繪圖、數控加工等應用，因此，是值得我們研究的課題。

二、研究方法及過程

由於線性的資料結構和掃瞄建構法的簡潔性與可用性，使得線性八元樹與掃瞄運算已成為實體塑模領域中廣泛被使用的技術。因此，我們可藉由這些技術來發展一個有效率的平移掃瞄演算法，將以線性四元樹編碼的影像表示二維資料，並建構為以線性八元樹編碼的三維實體。演算法的設計方向是藉由不同的掃瞄區間類型之由上往下的策略來控制整個掃瞄的步距，這是一種植基於遞迴的演算法，其以類似樹狀結構走訪的方式來進行，將會產生一串不需要後續額外運算處理排好序且緊密的線性八元樹碼當成輸出，我們可藉由這些完整的設計

過程，認知到三維實體塑模的方法。

三、研究結果

在本研究中，我們設計一個有效率的平移掃瞄演算法，它將以線性四元樹編碼的影像建構為以線性八元樹編碼的實體，它將產生一串排好序且緊密的線性八元樹碼當成輸出，而不需要後續額外的排序或合併等運算處理。藉此三維實體建構的方法，可將其應用於電腦輔助幾何設計上。

第一章 緒論

第一節 研究緣起

誠如我們所知，三維實體的塑模方式在很多工程和科學的領域都是一項重大的課題，因為實體的表示法將影響資料處理與儲存空間的效率。

而常見的方式有曲面表示法、分裂幾何法、掃描表示法、實體幾何構建法與八元樹表示法等。而實體塑模的核心內容是三維物體的數學模型和電腦表示方法，以及電腦表示法的表現方式，使用這些塑模方式的系統都在電腦內提供了對物體完整的幾何定義，因此，可以隨時獲得所需要的資訊，支援電腦輔助幾何設計與製造的任何一個層面，如電腦繪圖、數控加工等應用，因此，是值得我們研究的課題。

第二節 研究背景

三維物體的表示法在很多工程與科學的領域中都是一個重要的議題，因為他決定了物體被儲存與處理的效率，換句話說，對三維的物體而言，資料結構的選擇將影響整個系統的效能。

而八元樹已經廣泛地被使用，他是一種簡潔的階層式表示法，由於它具有像樹般的結構，因此，很多有關於物體的問題都能夠以樹的走訪方式來實現，此外，由於它的空間可定址性、已排好序等特性，很多的運算：像布林運算、隱藏面去除等等，也相對地被簡化了。而在很多的實際應用中：如實體模型化、計算機圖學和影像處理等方面，也都在在地顯現出它的優點與實用性。

然而，八元樹需要較多的空間，因此，八元樹另一種線性的表示方式，被稱為線性八元樹，這種線性的結構僅儲存黑色節點的位址碼，並將之安排成一排好序的陣列。此外，他已經被視為是很有效率的，特別是在節省記憶體方面和一些運算的處理時間上。

正如我們所知的，很多關於八元樹建構與處理的演算法已經被發展出來，而這也解決了相當多的問題，我們對這些問題亦是相當地有興趣，因此，在本研究中，我們將針對線

性八元樹三維實體的建構問題作探討，以期能發展出更有效率的實體建構演算法。

第二章 執行策略

第一節 八元樹

八元樹空間經常被模型化成一 $2^n * 2^n * 2^n$ 的立方體區域，這裡 n 為解析度參數，而 2^n 是八元樹空間的長度。

如何去建構一個八元樹呢？一開始，我們有一個大的足以包含整個物體空間的立方塊，然後，這物體的八元樹能夠是被產生的，乃藉著分割這大立方塊成八個相等大小的小立方體，我們稱為八分體，這每一個八分體再重複地被分解成子八分體，直到每一個子八分體有相同的屬性或是設定的解析度已經到達。

這最後被建成的八元樹，其根節點代表這整個物體空間，且每個非終端的八元樹節點其分支度為八，一個終端節點可以是黑色的或白色的乃端視於他相對應的八分體是完全落於物體內或物體外。一個非終端節點乃表示具有不均勻的屬性，因此需要更進一步的分解。所以，我們可以知道：基本上，八元樹是用一組二的指數次方大小的立方塊來約略三維的物體。

第二節 線性結構

誠如前一節所述，不可避免地，八元樹需要去維護指標，因而佔用較多的空間，為了記憶體的效率，線性的觀念於是出現，我們稱之為線性八元樹，他可被視為是八元樹的緊縮形式，因為這種線性的結構僅儲存黑色節點的位址碼。此外，他在節省記憶體與運算時間上的效率也散見於相關文獻上。

這裡，我們將線性八元樹的主要特性歸納如下：

- 僅儲存黑色的節點。
- 指標是被去除。
- 節點的編碼方式隱含了從根節點到那個節點的路徑。
- 這排好序的編碼節點維持著八元樹後序走訪的順序。
- 線性八元樹能與八元樹互相轉換。
- 線性八元樹的八元碼是一串嚴格遞增的數碼。

我們將利用這一些特點與優點，來設計我們的演算法。

第三章 研究方法

目前，使用八元樹去塑造三維的物體，已經受到廣泛的注意與研究。各種建構八元樹的方法也已發表於各學術文獻中，而在這些方法中，我們比較有興趣的建構型態是平移掃描法的建構方式。

基本上，它的意義是沿著某一個軸向，將一個平面區域往前掃出，而得到三維的物體。因為這種方式提供了一個相當簡易的方法來定義某一類的物體，因此，觸發了我們嘗試經由掃描運算來塑造物體的動機，所以，我們設計了一個掃描演算法來建構線性八元樹編碼的物體。

如圖 3-1 所示為我們所設計的 ALGM() 演算法，可簡述如下：有三種不同的情形考量；即 G、W、Q 等三種不同的節點型態。

當遇到 G 時：我們拷貝相對應的四元樹節點，當成八元樹節點。若此時有黑色的節點，則須分裂成四個一樣大的小節點。

當遇到 Q 時：我們亦是拷貝相對應的四元樹節點，當成八元樹節點。若此時是黑色或白色的節點，則可停止遞迴呼叫，否則，須要更進一步的處理。

當遇到 W 時:我們只須拷貝四個白色的節點，當成八元樹的節點即可。

在此演算法中，我們使用了假區間樹的概念，他給我們一個鳥瞰的視野，去了解整個掃描空間的分布情形，並給我們完整的訊息，以產生儘可能大的八元樹節點。我們即是根據掃描的起始與最終之掃描索引值來決定目前為何種節點型態，其節點型態的合成關係如表 3-1 所示。

基本上，這個假區間樹不是一棵真正的樹，而僅是具有樹般的控制結構，他是經由遞迴式的呼叫，從上往下開展。因此，在他的控制下，可長出緊密的八元樹。此外，由於它總是讓最大的節點先長出來，因此，它也具有貪心策略法的精神。

此演算法其輸入與輸出皆是線性的結構，且不需要更進一步的緊密運算，也不需要維護繁複的資料結構，他的基本觀念是每個在八元樹上的黑色節點，恰存在一條從根節點到那個節點的路徑，而此路徑的訊息正恰好是線性八元樹的節點碼，藉由這個特性，配合前述假區間樹的控制，我們即可達成目的。

明顯地，考量節省時間與空間起見，線性八元樹當成掃描結果的輸出是合理且必須的。因掃描法所產生的八元樹將會有很多的節點，因此，需要很大的空間來儲存輸出的結

果，亦將有大量的時間花在維護與處理指標上，並隨時需要去配置動態的堆積儲存體。因此，以原本排好序的線性四元碼樹為輸入，並模擬樹狀結構產生八元樹的方式，將可產生簡潔且排好序的線性八元樹碼為輸出。

```

ALGM(L:level; Front, Back:array-index; Buffer:octcode; Expand:Boolean)
/* ALGM() is a recursive procedure of linear quadtree to linear octree;
  L : recursive level;
  Front, Back : the first and last nodes of quadcode array to be examined;
  Buffer : a temporary storage to record octcode;
  Expand : a Boolean flag to record splitting condition */
Begin /* ALGM() */
  To specify sweeping-range type;
  Case sweeping-range type of
    'W': do nothing; /* linear octree only cares black nodes */
    'Q': if (not Expand) then
      if (L < 1) then
        Repeat
          finding the subquadrant pointed by Head and Tail;
          copy the corresponding quadcode-array[Head] to Buffer;
          if (Color(Head) = Black) then
            output Buffer;
          else
            ALGM(L-1, Head, Tail, Buffer, Expand);
          advance Head to the next index of Tail;
        Until (Head > Back)
      else
        for i = Front to Back do
          copy quadcode-array[i] to Buffer and output the Buffer;
      else
        for son-index = 0 to 3 do
          save son-index to Buffer and output the Buffer;
    'G': if (not Expand) then
      Repeat
        Expand = False;
        finding the subquadrant pointed by Head and Tail;
        copy the corresponding quadcode-array[Head] to Buffer;
        if (Color(Head) = Black) then
          Expand = True;
        ALGM(L-1, Head, Tail, Buffer, Expand);
        advance Head to the next index of Tail;
      Until (Head > Back)
    else
      for son-index = 0 to 3 do
        Begin
          save son-index to Buffer;
          ALGM(L-1, Head, Tail, Buffer, Expand);
        End
      End /* Case */
  End /* ALGM() */

```

圖 3-1 ALGM() 演算法

*	W	Q	G
W	W	G	G
Q	G	Q	G
G	G	G	G

表 3-1 節點型態的合成關係

第四章 研究內容與結果

我們設計一個植基於遞迴方式的實體建構演算法 ALGM()，此演算法的呼叫方式為 ALGM(n, initial-index, final-index, BUFFER, EXPAND)，其中：n 為解析度參數，initial-index，final-index 各代表線性四元樹碼陣列的起始與最終之掃描索引值，BUFFER 是一個緩衝記憶區用以儲存線性八元樹碼，EXPAND 是一個布林變數用以處理分裂的情況，此演算法的輸出是一系列排好序且緊密的線性八元樹碼。

如圖4-1所示為一簡單的範例說明，此範例展示出我們所設計的演算法執行的過程與步驟，其中：輸入為以線性四元樹 (100, 102, 12x, 2xx, 312) 編碼的影像，而其相對應的線性八元樹編碼實體輸出為 (144, 146, 164, 165, 166, 167, 244, 245, 246, 247, 254, 255, 256, 257, 264, 265, 266, 267, 274, 275, 276, 277, 356, 500, 502, 504, 506, 712, 716)，這些產出的實體編碼序列是緊密且排好序的。

LEVEL=3	LEVEL=2	LEVEL=1
Front=1 Back=5		
Head=1 Tail=3 1	Front=1 Back=3	
	Head=1 Tail=2 14	Front=1 Back=2 144 146
	Head=3 Tail=3 16 Expand=TRUE	164 5 6 7
Head=4 Tail=4 2 Expand=TRUE	Front=4 Back=4 24 5 6 7 Expand=TRUE	244 264 5 5 6 6 7 7 254 274 5 5 6 6 7 7
Head=5 Tail=5 3	Front=5 Back=5	
	Head=5 Tail=5 35	Front=5 Back=5 356

圖 4-1 範例：以線性四元樹 (100, 102, 12x, 2xx, 312) 編碼

LEVEL=3	LEVEL=2	LEVEL=1	
Front=1 Back=5			
Head=1 Tail=3 5	Front=1 Back=3		
	Head=1 Tail=2 50	Front=1 Back=2 500 502	Front=1 Back=2 504 506
	Head=3 Tail=3 52X		
Head=4 Tail=4 6 Expand=TRUE	60X 1 2 3		
Head=5 Tail=5 7	Front=5 Back=5		
	Head=5 Tail=5 71	Front=5 Back=5 712	Front=5 Back=5 716

圖 4-1 (續)

第五章 結論

一般而言，線性的資料結構是簡潔的，而且，掃描運算是簡單的。因此，線性八元樹和掃描表示法在實體塑造的領域中，已是兩種廣泛被使用的技巧。

在本研究中，我們設計一個有效率的掃描演算法，去轉換平面的四元樹編碼成相對應的線性八元樹編碼實體，此演算法產生緊密且排好序的線性八元樹碼輸出，而不需要後續額外的排序或合併等運算處理。

此外，從平移掃描法延伸到旋轉掃描法也可以利用類似的觀念去設計有效率的演算法，而這些有趣的課題都是值得我們更進一步去研究的。

參考書目

中文部分

唐榮錫等。計算機圖學。

網奕資訊，2005 年。

葉怡蘭等。3D 電腦動畫及數位特效。

全華，2006 年。

胡昭民。遊戲設計概論。

博碩文化，2007 年。

吳亞峰，蘇亞光。深入淺出 Android 遊戲程式開發範例大全。

博碩文化，2011 年。

英文部分

Angel, E. and Baker, M. P.. Interactive Computer Graphics : A Top-Down Approach wth OpenGL. Addison-Wesley, 2008.

Terzopoulos, D., Platt, J., Barr, A., Fleischer, K., Elastically Deformable Models, ACM Computer Graphics, 21(4): 205-214, 1987.

Hearn, D. and Baker, M. P.. Computer Graphics. Addison-Wesley, 1997.

McGregor, J. and Watt, A.. The Art of Microcomputer Graphics, Addison-Wesley, 1984.

Mandelbrot, B. B.. The Fractal Geometry of Nature, San Francisco, Freeman, 1982.

Prusinkiewicz, P.. Developmental Models of Herbaceous Plants for Computer Imagery Purposes. Computer Graphic, 22(4): 141-150, 1988.

Watt, A. and Watt, M.. Advanced Animation and Rendering Techniques. Addison-Wesley, 1992.

J. Camata and A. Coutinho, Parallel linear octree meshing with immersed surfaces, Proceedings of the 22nd

International Symposium on Computer Architecture and High Performance Computing, 151-158, 2010.

R. S. Sampath and G. Biros, A parallel geometric multigrid method for finite elements on octree meshes, *SIAM Journal on Scientific Computing*, 32, 1361-1392, 2010.