

HACİMSEL GÖRÜNTÜ GRAFİĞİNİN BLİST GÖSTERİMİ

Sema KOÇ, Ulus ÇEVİK
 Elektrik ve Elektronik Mühendisliği Bölümü
 Gaziantep Üniversitesi
 27310 Gaziantep
 E-mail : skoc@gantep.edu.tr
 ulus@gantep.edu.tr

ÖZET

Hacim grafiği, hacimsel veri tarafından tanımlanan görüntü ve bu görüntünün işlenmesi ile ilgilidir. Bu makalede Boolean liste gösterimi kullanılarak hacimsel görüntüyü tanımlamak ve hacimsel verileri işlemek için yeni bir model tanımlanacaktır. Bu gösterimin en önemli özelliği, geleneksel modellemelerden farklı olarak özyineli işlemlere ve yığın yapısına ihtiyaç duyulmamasıdır. Boolean liste hesaplaması ile her bir voksel için gerekli olan bellek gereksinimi $\log(H+1)$ bite indirgenmiştir. Hacimsel görüntü ekranının oluşturulması için 2D CSG (yapısal katı geometri) görüntü işlemcisi kullanılmıştır. Anahtar kelimeler: hacim grafiği, CSG, Boolean operatör.

1. GİRİŞ

Hacimsel görüntü çeşitli geometrik ve hacimsel cisimler kullanılarak oluşturulabilir; örneğin, eğriler, yüzeyler, katı modeller ve CT (bilgisayar tomografi) verileri kullanılarak. Bu cisimleri birleştirerek görüntü oluşturmak için kullanılan en popüler tekniklerden birisi “görüntü grafiği” dir. Görüntü grafiği, cisimleri ve görüntü parametrelerini ağaç yapısı şeklinde düzenler.

Hacimsel görüntünün oluşturulması için vokelizasyon algoritması kullanılmaktadır. Hacim üzerindeki her bir nokta voksel olarak tanımlanır. Bu konuda yapılan pek çok çalışma bulunmaktadır [1-6]. Fakat bu çalışmaların çoğu cisme özel çalışmalardır. Yani farklı cisimlerin vokelizasyonu için farklı algoritmalar gerekmektedir.

[7]’ de donanım kullanılarak hızlandırılmış bir vokelizasyon algoritması sunulmuştur. Bu algoritma küçük çaptaki etkileşimli uygulamalar için hızlı bir şekilde çalışsa da Boolean operasyonunun uygulanacağı her bir düğüm için ara hacimsel cisimler oluşturmak gerektiğinden performansı kısıtlıdır.

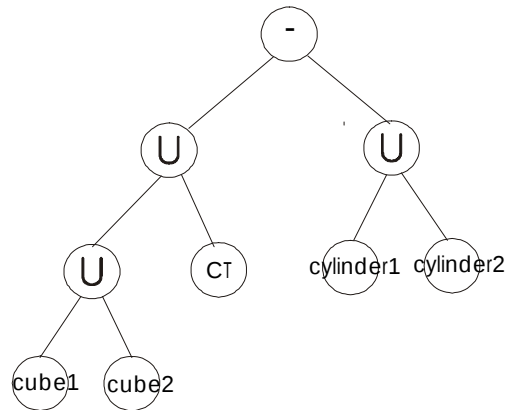
Başka hacimsel bir model [8]’de sunulmuştur. Bu çalışmada ağacın her bir düğümü için hacimsel seviyede vokelizasyon işlemi yapmak yerine her bir hacimsel kesit tüm ağaca uygulanmıştır. Bu çalışmada temel amaç, her bir cisim için öncelikle kesitler üretmek ve ağaç yapısının postfix sıralamasına göre

her bir kesit üzerinde Boolean operatörlerini uygulamaktır. Ancak bu algoritma ara işlemler ve sonuçları tutmak için yığın veri yapısı gerektirmektedir.

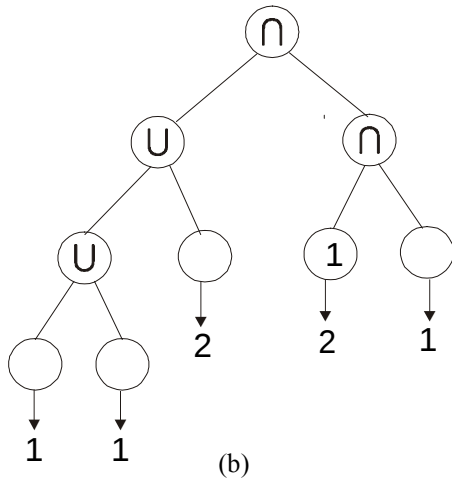
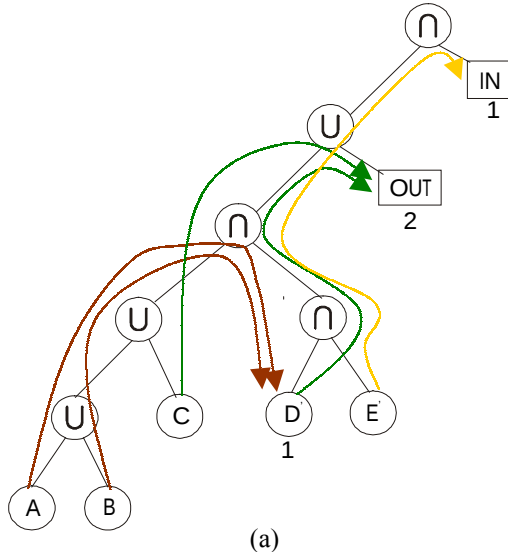
Bu çalışmada hacimsel ağaç grafiğini temsil etmek için Blist (Boolean listeleme) [9] kullanılmıştır. Blist gösteriminde Boolean açılımı ağaç yapısı yerine görüntüyü oluşturan cisimlerin listesi şeklinde hesaplanmaktadır. Blist metodunda hacimsel ağaç grafiğinin hesaplanması için herhangi bir yığın yapısına ihtiyaç duyulmamaktadır. Ve her bir voksel için gerekli olan bellek gereksinimini $\log(H+1)$ ’e indirmiştir (H ağacın yüksekliği).

2. HACİMSEL AĞAÇ YAPISININ BLİST GÖSTERİMİ

Blist gösterimi CSG ağacını karar grafiğine taşımaktadır. Bu gösterim CSG açılımını bir tablo halinde sunmaktadır. CSG açılımını blist gösterimine çevirmek için öncelikle ağaç, De Morgan kanunları kullanılarak pozitif forma dönüştürülür; örneğin $A-B = A \cap B^c$. Daha sonra ağaç sol ağırlıklı olacak şekilde döndürülür ve her bir ağacın yaprağı için BL tablosunda karşı gelen alan doldurulur. Şekil 1’de hacimsel görüntü ağacı ve Şekil 2’de bu ağacın Blist gösterimine dönüştürülmesi gösterilmiştir.



Şekil 1. Hacimsel görüntü ağacı



Şekil 2. Hacimsel görüntü ağacının Blist gösterimi

Şekil 2’de gösterilen hacimsel görüntü ağacının BL tablosu aşağıdaki gibidir.

P	Bl[p]. isim	BL[p]. işaret	BL[p]. cisim Referansı	Bl[p]. Türü
1	0	+	A	1
2	0	+	B	1
3	0	-	C	2
4	1	+	D	2
5	0	-	E	1

Blist gösteriminde listedeki her bir cisim, bir geometrik modeli veya hacimsel bir veri setini gösterir. Burada önemli olan listedeki her bir cisim için bir kesit oluşturmak ve daha sonra her bir vokselle için BL(p) tablosunu kullanarak o vokselin söz konusu hacim içerisinde yer alıp almadığına karar vermektir. Bu işlemin ayrıntıları [9]’da yer almaktadır.

Blist gösteriminin avantajı; daha önceki kullanılan metodlarda Boolean açılımının hesaplanması için gerekli olan özyineli işlemlere ve yığın yapısına ihtiyaç duyulmamasıdır.

2.1 Voxelizasyon Algoritması

Voxelizasyon algoritması 3D bir ekran üzerinde tanımlanan bir bölge için hacimsel veriyi oluşturur.

Biz bu çalışmada hacimsel görüntünün her bir kesitini oluşturmak için 2D CSG işlemcisi [10] kullandık. İki Z-düzlemi arasında kalan ince alan bir kesit olarak tanımlanır. Burada, algoritma Z-düzlemine paralel hareket ederek belirli aralıklarla önden arkaya doğru hacimsel kesitleri oluşturmaktadır. 2D görüntü işlemcisi kullanarak her bir kesitin görüntülenmesi gerçekte, oluşturulacak olan hacmin her bir kesitini temsil etmektedir.

2D görüntü işlemcisi cisimlerin ön ve arka yüzeylerinin derinlik değerini kullanarak konveks, konkav, poligon ve çizgileri görüntüleyebilmektedir. Görüntüleme işlemlerinin ayrıntıları [10]’da yer almaktadır.

3. SONUÇ

Bu çalışmada hacimsel görüntü ağacının Boolean liste gösterimi sunulmuştur. Blist gösterimi ile hacimsel bir görüntüyü oluştururken her bir vokselle için gerekli olan bellek gereksinimi $\log(H+1)$ bite indirgenmiştir. Ve herhangi bir yığın yapısına ihtiyaç duyulmamaktadır.

Hacimsel görüntünün her bir kesitini oluşturmak için 2D CSG işlemcisi kullanılmıştır. Bu görüntü algoritmasının avantajı algoritmanın cismin türüne bağlı olmayıp konveks, konkav, poligon ve çizgiler için kullanılabilir olmasıdır.

REFERANSLAR

1. Cohen D, Kaufman. A Scan-conversion algorithms for linear an quadratic objects. In: Kaufman A, editor. Volume Visualization, Los Alamitos, CA: IEEE Computer Society Press, p.280-301, 1991..
2. Cohen D, Kaufman. A. 3D linear voxelization and connectivity control. IEEE Computer Graphics and Applications 17(6) 80-7,1997.

3. Huang J, Yagel R, Fillippov V. Kurzion Y. An accurate method for voxelization polygon meshes. Proceedings of the IEEE/ACM symposium on Volume Visualization p.199-206, 1998.
4. Kaufman A. Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes. SIGGRAPH'87 p.171-9, 1987 .
5. Kaufman A. Efficient algorithms for 3D scan-conversion 3D polygons. Computer and Graphics; 12(2):213-9, 1988.
6. Nadeau D. Volume scene graphs. In Proc, IEEE/ACM Symposium on Volume Visualization. P. 49-56 Oct. 2000.
7. Fang S., Chen H. Hardware accelerated voxelization. Computer&Graphics 24(3):433-442. June 2000.
8. Fang S., Duoduo L. Fast CSG voxelization by Frame Buffer Pixel Mapping. In Proc. IEEE/ACM Symposium on Volume Visualization, p.43-48, Oct. 2000.
9. Rossignac J. Technical Report GIT-GVU-94-04. October 1998.
- 10.Çevik, U. Design of an FPGA Based Parallel Architecture Processor Displaying CSG Volumes and Surfaces. PhD. Thesis. University of Sussex, Brighton. June 1996.