

Sanal Cerrahi Uygulamalarında Görselleştirme Altyapısı

Dionysis Goularas

Yeditepe Üniversitesi,
Bilgisayar Mühendisliği Bölümü
goularas@cse.yeditepe.edu.tr

Başar Atalay

Yeditepe Üniversitesi,
Tıp Fakültesi
Nöroşirürji Anabilim Dalı
batalay@yeditepe.edu.tr

Esin Onbaşıoğlu

Yeditepe Üniversitesi,
Bilgisayar Mühendisliği Bölümü
esin@cse.yeditepe.edu.tr

Koray K. Şafak

Yeditepe Üniversitesi,
Makine Mühendisliği Bölümü
ksafak@yeditepe.edu.tr

Fethi Okyar

Yeditepe Üniversitesi,
Makine Mühendisliği Bölümü
okyar@yeditepe.edu.tr

Ahu Soydan

Yeditepe Üniversitesi,
Bilgisayar Mühendisliği Bölümü
asoydan@cse.yeditepe.edu.tr

Özetçe

Sanal gerçeklik kullanılan cerrahi eğitimi geleneksel yöntemlere göre büyük potansiyel taşımaktadır. Nöroşirürji gibi hassas mikrocerrahi yöntemler kullanılan cerrahi branşlarda cerrahların yetiştirilmesi için kullanılan araçlar kısıtlıdır. Cerrahi tecrübe ancak çok sayıda cerrahi işlem sonrasında kazanılabilmektedir. Tecrübesiz cerrahların yaptığı hatalı ya da eksik cerrahi müdahaleler hastalarda ciddi doku hasarı ve fonksiyonel kısıtlanmaya neden olmakta ve hatta felç ile sonuçlanabilmektedir. Günümüzde cerrahi eğitim öncelikle gözlem, daha sonra da deney hayvanları, kadavralar ve en sonunda hastalar üzerinde yapılmaktadır. Bu çalışmanın amacı nöroşirürjide öncelikle spinal ve kranial cerrahide eğitim ve cerrahi planlamada kullanılmak üzere yeni bir sanal gerçeklik eğitim sistemi geliştirmektir. VISPLAT bir sanal gerçeklik simülasyon platformudur. Hastaya-özel bir sistem olarak tasarlanmıştır. BT görüntülerinden 3B nesnelerin modellenmesi, cerrahi işlemlerin görüntülenmesi, dokunsal etkileşim ve cerrahi işlemlerde mekanistik materyal çıkarma modelleri için bir çerçeve oluşturur.

Anahtar Sözcükler : sanal cerrahi, görselleştirme, dokunsal, çarpışma sezimi

Visualisation Platform for Virtual Surgery

Abstract

Virtual reality based surgical training have a great potential as an alternative to traditional training methods. In neurosurgery, state-of-the-art training devices are limited. The surgical experience accumulates only after so many surgical procedures. Incorrect surgical movements can be destructive. Traditional techniques for training in surgery is based on observations, using animals and cadavers, and finally real patients. The aim of this research is the development of a virtual reality training system for spinal and cranial surgery in neurosurgery. VISPLAT is a platform for virtual surgery simulation. It is designed as a patient-specific system. It acts as a framework for modeling 3D objects from CT images, visualization of the surgical operations, haptic interaction and mechanistic material-removal models for surgical operations.

Keywords : virtual surgery, visualization, haptic, collision detection

1 Giriş

Sanal gerçeklik teknikleri, doktorların, hastalar için risk teşkil etmeyecek sanal bir ortamda tecrübe edinmelerini sağlamaktadır. Aynı zamanda bu teknoloji görselleştirme konusundaki zorlu problemlere yeni çözümler de sunmaktadır.

VISPLAT (Virtual Surgical Planning and Training Platform), insan kafatası ve omurga üzerinde cerrahi eğitimi ve planlaması için geliştirilmiş bir simülatördür. Hastaya-özel bir sistem olarak tasarlanmıştır. Hastaların bilgisayarlı tomografi (BT) görüntülerinin, bunların 3-boyutlu (3B) modellerinin ve diğer hasta bilgilerinin saklandığı bir veritabanı vardır. Simülator, cerrahi işlemdeki mekanistik materyal çıkarma modellerini, görselleştirme sistemi ve dokunsal arayüzle birleştirir. Dokunsal arayüz, cerrahi işlem sırasında kullanıcıya dokunma hissi verir. Görselleştirme sistemi gerçek zamanlı etkileşim sağlayabilmelidir.

2 İlgili Çalışmalar

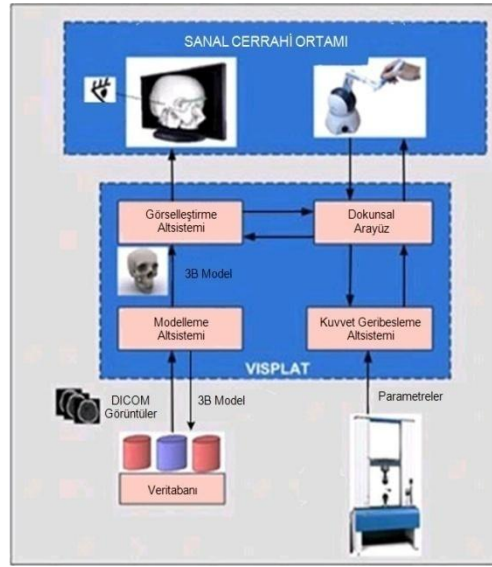
Literatürde değişik amaçlarla tasarlanmış simülatorler mevcuttur. Eğitim ve planlama araçları olarak tasarlanmış ticari simülatorler arasında ortakluk cerrahisi [1], yaraya dikiş atma ve düğüm bağlama [2], abdominal cerrahi [3] için geliştirilmiş çalışmalar gösterilebilir. Gerçek zamanlı yumuşak doku simülasyonu [4], kafatası cerrahisi [5, 6, 7], sanal temporal kemik diseksiyonu [8, 9], ve dokunsal kuvvet hesaplanması kullanarak delici-kemik etkileşimi [10] ise açık kaynak ve araştırma projeleri arasında sayılabilir.

Mevcut cerrahi simülatorlerinin çoğunda 3B veriler voxel olarak gösterilmektedir [11]. Yüzey verisi üzerinde dokunsal gerçekleştirilen bir çalışma vardır [12, 13]. Bu çalışmada implantları yerleştirmek için delme işlemi yapılmıştır.

Gerçekçi dokunsal gerçekleştirme zorlu bir konudur. Ortamın mekanik modelinin ve dokunsal cihaz üzerinde sabit kuvvet kontrol döngüsünün uygulanması gerekir. Dokunsal arayüzlerin kontrolü konusunda araştırmalar devam etmektedir [14, 15, 16, 17]. Sanal cerrahideki kuvvetlerin gerçeğe uygun olarak yansıtılması da aktif bir araştırma alanıdır [18, 19, 20, 21].

3 Visplat Ortamına Genel Bakış

VISPLAT [26], hastaların (BT) görüntüleri kullanılarak 3B modellerinin oluşturulması, cerrahi işlemlerin görselleştirilmesi, dokunsal etkileşim ve cerrahi işlemlerdeki mekanistik materyal çıkarmanın modellenmesi için geliştirilmiş bir çerçevedir. Dört alt sistemden oluşur (Şekil 1). Modelleme alt sisteminde (Modeling Subsystem), veritabanındaki BT görüntüleri kullanılarak hastanın 3B anatomik modeli oluşturulur. Bu model doğrudan simülasyonda kullanılabileceği gibi, sonradan kullanılmak üzere veritabanında da saklanabilir. Bu 3B model ve eksensel, oksal ve taç (axial, sagittal, coronal) bakış görüntüleri Görselleştirme alt sistemi (Visualization Subsystem) tarafından ekranda gösterilir.



Şekil-1: Genel Bakış

Görselleştirme alt sistemi ayrıca simülasyon sonuçlarını etkileşimli olarak gösterir. Delici kemiğe dokunduğunda görselleştirme alt sistemi yüzeyde bir çukur oluşturur, yüzey örgüsünü değiştirir ve ekranda gösterir. Bu amaçla çarpışma sezimi (collision detection) ve örgü ayrıştırma (mesh refinement) algoritmaları uygular. Görselleştirme alt sistemi dokunsal arayüzden dokunsal kalemin konumunu alır ve çukur

oluşturduktan sonra örgüyü değiştirerek yeni yüzey örgüsünü oluşturur.

Dokunsal arayüzün (Haptic Interface) fonksiyonu, dokunsal cihaz, görselleştirme alt sistemi ve kuvvet geri besleme sistemi arasındaki veri iletimini sağlamaktır.

Kuvvet geri besleme sistemi (Force feedback subsystem), yazılım ortamıyla dokunsal cihazı birleştirir. Dokunsal cihaz 6 serbestlik derecesinde (degrees-of-freedom) robotik bir mekanizmadır. Sanal delme işleminde, cerrahın el hareketleri ile ilgili bilgi kuvvet hesaplama alt sistemine iletilir. Kuvvet geri besleme bilgisi tekrar dokunsal cihaza gönderilerek operasyon sırasındaki dokunma hissini hissedilmesi sağlanır.

4 Görselleştirme Alt sistemi

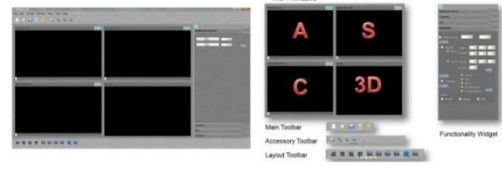
4.1 VISPLAT Kullanıcı Arayüzü

Kullanıcı arayüzü, cerrahların ve doktorların ihtiyaçlarını karşılayacak şekilde tasarlanmıştır. Monokromatik medikal görüntülerin kullanıcıyı rahatsız etmemesi için gri renk seçilmiştir. Dört pencere kullanılmıştır (Şekil 2). “A”, “S” and “C” pencereleri DICOM görüntülere eksensel, oksal ve taç bakış için kullanılır. “3D” penceresinde, oluşturulan 3B model görüntülenir. Değişik bakış seçeneklerine özen gösterilmiştir. “Layout toolbar” yardımıyla kullanıcı pencereyi uygulama arayüzünden büyütebilir veya tam ekran olarak gösterebilir. Ayrıca yaklaştırma seçeneği her pencerenin sağ üst köşesine yerleştirilmiştir.

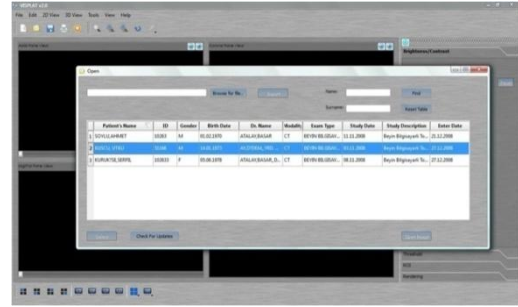
Bütün işlemler “functionality widget” üzerinde bir araya getirilmiştir. Kullanıcı, kontrast ve parlaklığı değiştirme, yüzey oluşturma parametrelerini değiştirme, hacim gerçekleştirme parametrelerini belirleme gibi çeşitli işlemleri yapabilir.

Görüntüler bir diyalog kutusu (dialog box) yardımıyla yüklenebilir. (Şekil 3). Diyalog kutusu hasta bilgilerini ve inceleme türünü sunar. DICOM dosyalarının her yüklenişinde hasta bilgileri de çıkartılarak incelemenin daha iyi anlaşılması sağlanır.

Bu projede, veritabanı kullanımının ihtiyaçlarla orantılı olmayacağı düşünülerek, veriler XML formatında düzenlenmiştir.

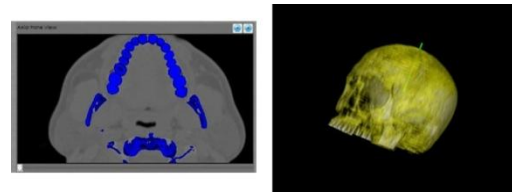


Şekil-2: Kullanıcı Arayüzü



Şekil-3: BT Görüntülerinin Yükleme

Kullanıcı, 3B modelin hazırlanması için, oluşturulacak olan bölgeyi bir eşik (threshold) seçerek belirler (Şekil 4). Bu bölge görselleştirilerek aynı zamanda kullanıcının seçtiği eşik etkisini izlemesi de sağlanır. Bu seçenek kullanıcı için önemlidir çünkü böylece seçtiği bölgeyi her görüntü üzerinde görebilir. Kullanıcı bütün görüntüler üzerinde gezinebilir. Pencerenin altındaki sürgü, kullanıcının bütün görüntü serisini incelemesini ve her görüntüde modeli oluşturulacak bölgeyi görmesini sağlar. Sonuç olarak, neyin modelinin oluşturulacağını görüntülenmesi gereksiz modellemeleri önler.

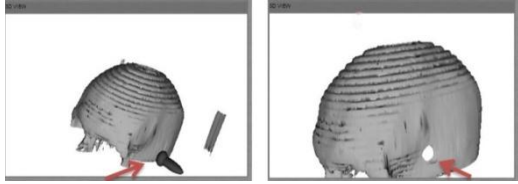


Şekil-4: Modeli oluşturulacak bölgenin görüntü üzerinde belirlenmesi.

Delme işlemi sanal bir delici kullanılarak yapılır. (Şekil 5). Kullanıcı bir delici seçerek işleme devam edebilir.

Modelleme alt sistemi ITK ve VTK araçları kullanılarak geliştirilmiştir. Görüntülerin

yüklenmesinde ITK kullanılmıştır. Modelleneyecek alanın seçimi ve 3B modelin oluşturulması için VTK kullanılmıştır. Kemik modeli BT görüntülerinden “marching cubes” algoritması kullanılarak oluşturulur. Görüntülerin eksensel, oksal ve taç bakış olarak ekranda gösterildiği kullanıcı arayüzü ise QT ortamında geliştirilmiştir.



Şekil-5: Delme İşleminin Önce ve Sonra

4.2 Delme İşlemi

Literatürdeki simülasyonların çoğunda voxel modelleri kullanılmaktadır. Bu projede yüzey modeli kullanılmıştır. Kemik ve delici üçgenel örgü yapısıyla modellenmiştir. Kemik modeli BT görüntülerinden oluşturulur. Delici bir kesel (geodesic) küredir.

Yüzey modeli üzerinde delme işlemi aşağıdaki basamaklardan oluşur:

- Üçgenler arası çarpışma sezimi
- Çukur oluşturma
- Yüzey örgüsü ayrıştırması

Üçgenler arası çarpışma seziminde, delici kemiğe dokunduğunda çarpışan üçgenler belirlenir. Daha sonra delici kürenin kemik yüzeyinin altında kalan kısmı kullanılarak çukur oluşturulur. Yüzey örgüsü ayrıştırmasında, bu çukur yüzey örgüsüne bağlanır.

4.3 Üçgenler arası Çarpışma Sezimi

Üçgenler arası çarpışma sezimi algoritmaları arasında [22], [23], [24], ve [25] yer almaktadır. Bu projede kullanılan algoritma [22] ve [23]'te kullanılan algoritmalara dayanmaktadır. [22]'de, üçgenlerin yüzey denklemleri kullanılmıştır. Kesişme noktaları, üçgenlerin köşelerinin kesişme çizgisi üzerindeki skaler izdüşümleri kullanılarak bulunur. [23]'te ise, bir üçgenin yüzeyi ve diğerinin kenar vektörleri kullanılmıştır. Bütün kenar vektörleri için $Ax=b$ türü denklem setlerinin çözümü ile sabit sayılar hesaplanır. Eğer, bir kenar

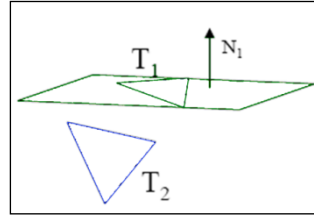
vektörü ile yüzeyin kesişme noktası üçgenin içindeyse geçerli bir sabit mevcuttur.

Çarpışma sezimi algoritmaları önemli miktarda hesaplama gerektirir. Bu projede delicinin merkezi ve büyüklüğü bilinmektedir. Bu nedenle deliciye uzak olan yüzey üçgenleri çarpışma testlerinden önce elenebilir ve çarpışma algoritmaları sadece deliciye yakın olan üçgenlere uygulanabilir.

Üçgenler arası çarpışma sezimi algoritmamız aşağıdaki basamakları her üçgen çifti üzerinde (biri yüzeyde (T_1) ve diğeri delicide (T_2)) uygular:

1. Bir üçgenin bütün köşeleri diğer üçgenin düzleminin aynı tarafındaysa, kesişme yoktur.
2. T_1 ve T_2 düzlemlerinin kesişme çizgisi (L) bulunur.
3. Kesişme çizgisi üzerinde bir nokta p_0 ve bir vektör \vec{d}_l bulunur.
4. L ile üçgenin her kenarının kesişme noktaları hesaplanır.
5. İki üçgenin kesin kesişme noktaları hesaplanır.
6. Kesişme çizgisi üzerindeki ortak doğru parçası bulunur.

Birinci basamak, aralarında kesişme olmayan üçgenleri daha fazla hesaplama yapmadan elemeyi amaçlar. Üçgenlerin düzlem denklemleri hesaplanır. T_1 üçgeninin bütün köşeleri T_2 düzleminin aynı tarafındaysa kesişme yoktur. Aynı şekilde, T_2 üçgeninin bütün köşeleri T_1 düzleminin aynı tarafındaysa kesişme yoktur (Şekil-6).



Şekil-6: Kesişmeyen Üçgenler

İkinci basamakta, iki üçgenin düzlem denklemleri birbirine eşitlenerek

$$A_1x + B_1y + C_1z + D_1 = A_2x + B_2y + C_2z + D_2 \quad (1)$$

iki düzlem arasındaki kesişme çizgisinin (L) denklemi bulunur.

$$A_Lx + B_Ly + C_Lz + D_L = 0 \quad (2)$$

Buradaki A_L, B_L, C_L, D_L aşağıdaki gibidir

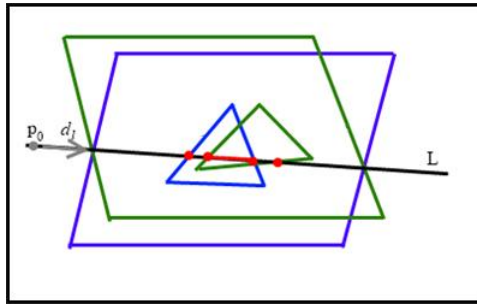
$$A_L = A_1 - A_2$$

$$B_L = B_1 - B_2$$

$$C_L = C_1 - C_2$$

$$D_L = D_1 - D_2$$

Şekil 7'de üçgenler ve düzlemleri arasındaki kesişme çizgisi (L) gösterilmiştir.



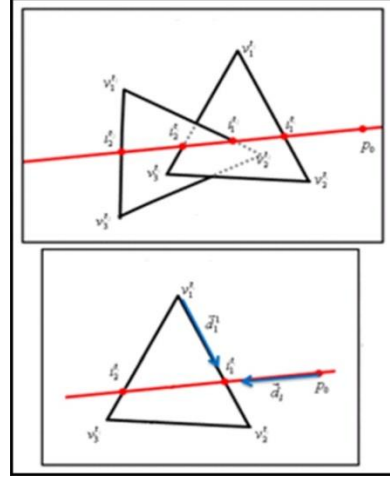
Şekil-7: Üçgenlerin Düzlemleri Arasındaki Kesişme Çizgisi

Üçüncü basamakta, kesişme çizgisi üzerinde bir $p_0 = (x_0, y_0, z_0)$ noktası ve bir \vec{d}_l vektörü bulunur. Kesişme çizgisi x eksenine paralel değilse p_0 , L çizgisinin yz-düzlemini kestiği nokta olabilir. L doğrusu üzerindeki \vec{d}_l vektörü ise düzlemlerin normalleri kullanılarak bulunabilir. N_1 ve N_2 iki düzlemin normal vektörleri ise, $N_1 \times N_2$ kesişme çizgisi üzerindeki $\vec{d}_l = (x_l, y_l, z_l)$ vektörünü verir. p_0 ve \vec{d}_l Şekil 7'de gösterilmiştir.

Dördüncü basamakta, $\vec{d}_l = (x_l, y_l, z_l)$ vektörü ve üçgenin kenar vektörlerinin kesişme noktaları bulunur (Şekil 8). Eğer bir üçgenin köşeleri $v_j = (x_j, y_j, z_j)$, $j=1,2,3$, ve kenar vektörleri \vec{d}_j , $j=1,2,3$, ise, (\vec{d}_j) vektörü, v_j köşesinden başlayacak şekilde, \vec{d}_l vektörünün j kenarı ile kesişmesi aşağıdaki eşitliğin çözümü ile bulunur

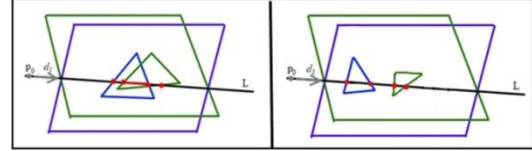
$$v_j + (\beta_j \vec{d}_j) = p_0 + (\alpha_j \vec{d}_l) \quad (3)$$

Burada β_j ve α_j sabit sayılardır, ($j=1,2,3$). Eğer \vec{d}_l ve j kenarı arasında bir kesişme noktası varsa, β_j 0 ile 1 arasında bulunur, ($0 \leq \beta_j \leq 1$).



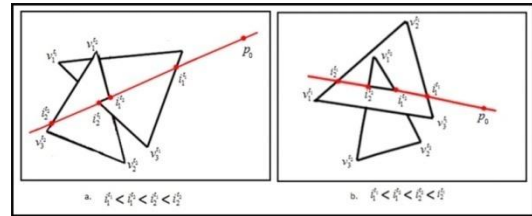
Şekil-8: Üçgenin Kenarları ile L Doğrusunun Kesişimi

Dört kesişme noktası bulunduktan sonra bu noktalar arasındaki doğru parçaları üst üste geliyorsa üçgenler çarpışmaktadır (Şekil 9(a)). Doğru parçaları üst üste gelmiyorsa üçgenler çarpışmaz (Şekil 9(b)).



(a) Çarpışan Üçgenler (a) ve Çarpışmayan Üçgenler (b)

Çarpışma durumunda, üçgenler ya kesişiyordur (Şekil 10(a)) yada birbirinin içinden geçiyordur (Şekil 10(b)). Kesin kesişme noktaları buna göre belirlenir.

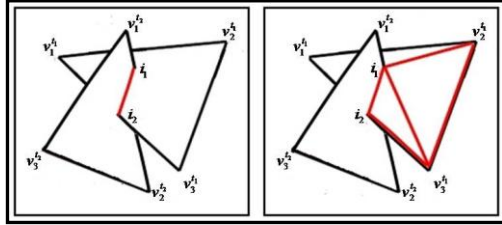


(a) Kesişen Üçgenler (a) ve İçinden Geçen Üçgenler (b)

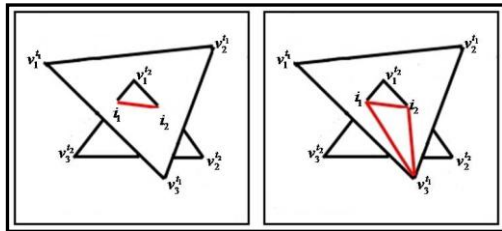
Kullandığımız algoritmada, ilk aşamada, [22]'deki gibi düzlem denklemleri kullanılarak kesişme olmayan üçgenler elenmektedir. Daha sonra [23]'teki gibi vektör işlemleri yapılmaktadır. Fakat [23]'te bir üçgenin kenarlarının diğerinin düzlem denklemi ile kesişme noktaları hesaplanmakta, ancak bizim algoritmamızda bir üçgenin kenarlarının yüzey kesişme çizgisiyle (L) olan kesişme noktaları bulunmaktadır. Bu nedenle yapılan hesaplamalar her iki algoritmadan da farklıdır.

4.4 Çukur Oluşturma ve Örgü Ayrıştırma

Yüzey örgüsünün ayrıştırmasında, çarpışan üçgenler kesişme çizgisi üzerinden ikiye bölünür. Üçgenin delici içinde kalan noktası ortadan kaldırılır. Bir köşe delicinin içindeyse kesişme noktaları üçgenin diğer köşelerine Şekil 11'deki gibi iki yeni üçgen yaratarak; iki köşe delicinin içindeyse kesişme noktaları üçgenin diğer köşesine Şekil 12'deki gibi bir yeni üçgen yaratarak bağlanır. Böylece üçgenlerin kesişme noktaları yüzey örgüsüne bağlanmış olur.



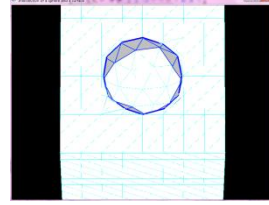
Şekil-11: Bir Köşe Delicinin İçinde



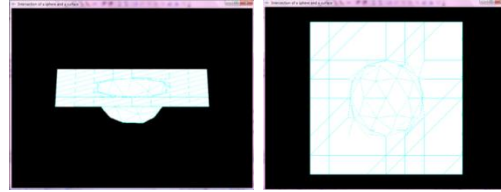
Şekil-12: İki Köşe Delicinin İçinde

Daha sonra, delici kürenin, yüzey örgüsünün altında kalan üçgenleri yüzey modeline dahil edilerek çukur oluşturulur. Çukur kenarları, yine yukarıda anlatılan

yöntemle yüzey örgüsüne bağlanır (Şekil 13). Sonuç olarak Şekil 14'teki örgü elde edilir.



Şekil-13: Küre Örgüsü Ayrıştırma



Şekil-14: Sonuç Örgüsü, Yandan Görünüm (a), Üstten Görünüm (b)

4.5 Dokunsal Arayüz

Dokunsal cihaz olarak SensAble Phantom Omni kullanılmıştır. Delme işlemi ve dokunsal arayüz H3D ortamında geliştirilmiştir. H3D, dokunsal ve grafik gerçekleştirme (rendering) işlemlerini OpenHaptics ve OpenGL kütüphaneleri kullanarak sağlamaktadır. Ayrıca dokunsal gerçekleştirme ve grafik gerçekleştirme işlemlerinin farklı izlekler (thread) tarafından yapılmasına da imkan vermektedir.

5 Performans Analizi

Üçgenler arası çarpışma sezimi algoritmamızın performansı [22] ve [23] ile karşılaştırılmıştır. Çizelge 1, üç algoritmadaki işlem sayılarını karşılaştırmaktadır. Çizelgede soldaki sayılar minimum ve sağdaki sayılar maksimum değerlerdir. Çizelge 2 algoritmaların bir çarpışma için çalışma zamanlarını karşılaştırmaktadır. Bu çizelgedeki değerler [27]'deki yöntem kullanılarak, 3.0 GHz Intel Pentium 4 işlemci üzerinde hesaplanmıştır. Bu yöntemdeki hata oranı %10-15'tir. Sonuçlar, kullandığımız algoritmanın [22]'den daha iyi sonuç verdiğini ve [23]'e yakın sonuç verdiğini göstermektedir.

Çizelge-1: Çarpışma Sezimi Algoritmalarındaki İşlem Sayıları

	Algoritma		
	Möller [22]	Tropp et al. [23]	Kullandığımız
+ / -	54	26/27	44/50
×	57	56/57	63/71
÷	0	0	6/8
karşılaştırma	12/28	13	10/12
abs	3/9	0	0
=	69/75	31/35	17/19

Çizelge-2: Çarpışma Sezimi Algoritmalarının Çalışma Zamanları (ns)

	Algoritma		
	Möller [22]	Tropp et al. [23]	Kullandığımız
+ / -	101	49/51	82/93
×	107	105/107	117/132
÷	0	0	16/21
karşılaştırma	39/92	43	32/39
abs	14/42	0	0
=	186/203	84/95	45/51
Toplam	447/545	281/296	292/336

Sistemin performansı değişik delici ve yüzey modelleri kullanılarak ölçülmüştür. Modellerdeki üçgen ve köşe sayıları Çizelge 3 ve Çizelge 4'de verilmiştir.

Çizelge-3: Yüzey Modellerindeki Köşe ve Üçgen Sayıları

Model	Köşe sayıları	Üçgen sayıları
Yüzey 1	386	768
Yüzey 2	4467	9216
Yüzey 3	8577	17335

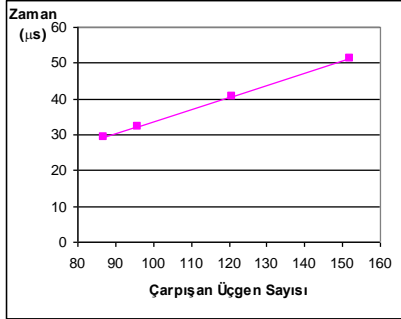
Çizelge-4: Delici Modellerindeki Köşe ve Üçgen Sayıları

Model	Köşe sayıları	Üçgen sayıları
Delici 1	42	80
Delici 2	92	180
Delici 3	162	320
Delici 4	252	500

Çarpışma sezimi algoritmaları iki yüzeydeki üçgenlerin birbiriyle karşılaştırılmasını gerektirir. Her delici-yüzey çifti için, bütün üçgenlerin birbiriyle karşılaştırıldığı durumda, kullandığımız çarpışma sezimi algoritmasının çalışma zamanı Çizelge 5'te verilmiştir. Küçük modeller için (Delici 1 – Yüzey 1) çarpışma sezimi algoritmasının kabul edilebilir bir performansı vardır. Fakat üçgen sayısı arttıkça (Delici 4 – Yüzey 3), daha gerçekçi bir görüntü elde edilebilmesine karşı çalışma süresi de artmakta ve sonuçların gerçek zamanlı elde edilmesi zorlaşmaktadır. Ancak kullandığımız yöntemde, çarpışma sezimi algoritması, kemik yüzeyindeki üçgenlerden sadece deliciye yakın olan üçgenler üzerinde uygulanmaktadır. Çarpışan üçgen sayısına göre performans Şekil 15'te verilmiştir. Yatay eksen, Yüzey 3 ve farklı dört delici (Delici 1 – Delici 4) arasındaki çarpışma sayılarını, ve dikey eksen algoritmanın çalışma zamanını göstermektedir. Algoritmanın performansı çarpışma sayısına göre lineer artmaktadır.

Çizelge-5: Çarpışma Sezimi Algoritmasının Çalışma Zamanı (ms)

	Delici 1	Delici 2	Delici 3	Delici 4
Yüzey 1	18	40	72	112
Yüzey 2	215	484	861	1346
Yüzey 3	405	911	1620	2531



Şekil-15: Çarpışan Üçgen Sayısına Göre Çarpışma Sezimi Algoritmasının Çalışma Zamanı (µs)

9 Sonuç

Bu makalede, kafatası ve omurga cerrahisi simülasyonu için geliştirilmiş olan VISPLAT yazılımı tanıtılmış ve görselleştirme altsistemi ayrıntılı olarak sunulmuştur.

Geliştirilmiş olan kullanıcı arayüzü tanıtılmış, delme işleminde kullanılan üçgenler arası çarpışma sezimi ve örgü ayrıştırma algoritmaları anlatılmış, ve dokunsal arayüz hakkında bilgi verilmiştir. Kullanılan algoritmalarla üçgen sayısı sınırlı modellerde gerçek zamanlı görüntü elde edilebilmektedir. Üçgen sayısı arttıkça gerçek zamanlı görüntü elde etmek zorlaşmaktadır.

Bir cerrahi eğitim aracı olarak VISPLAT platformunun cerrahların öğrenme sürecini hızlandırması beklenmektedir. Gelecekte VISPLAT ortamının cerrahi planlama aracı olarak da geliştirilmesi hedeflenmektedir.

10 Kaynakça

- [1] Voxel-Man, On the WWW, URL <http://www.voxel-man.de/news/>
- [2] Desai, J., 2008. Robotic Haptic Feedback System for Bx/RFA of Breast Tumor Under Continuous MRI. On the WWW, URL <http://www.researchgrantdatabase.com/g/5R01EB008713-02/Robotic-Haptic-Feedback-System-for-Bx-RFA-of-Breast-Tumor-under-Continuous-MRI/>
- [3] LapSim, On the WWW, URL <http://www.virtualsurgery.vision.ee.ethz.ch/SurgicalScience/LapSim>
- [4] Montgomery, K., "The Spring Simulation Platform", On the WWW, URL http://simworkshops.stanford.edu/06_0627/SPRING-Kevin%20Mont-ppt.pdf
- [5] Eriksson, M., "A Haptic and Virtual Reality Temporal Bone Surgery Simulator", On the WWW, URL <http://www.md.kth.se/research/projects/mda/p13.shtml?eng>
- [6] Eriksson, M., Flemmer, H., and Wikander, J., "A Haptic and Virtual Reality Skull Bone Surgery Simulator", 2005. *World Haptics Conference*, March.
- [7] Eriksson, M., Flemmer, H., and Wikander, J., "A Haptic and Virtual Reality Temporal Bone Surgery Simulator", *Submitted for publication in Advanced Robotics Magazine*, March.
- [8] Bryan, J., Stredney, D., Wiet, G., and Sessanna, D., "Virtual temporal bone dissection: a case study", On the WWW, URL <http://portal.acm.org/citation.cfm?id=601762>
- [9] Ohio Supercomputer Center, 2008 Annual Research Report, On the WWW, URL http://www.osc.edu/press/media/docs/2008_Research_Report.pdf
- [10] CRS4 Visual Computing, On the WWW, URL <http://www.crs4.it/vic/projects/>
- [11] Hearn, D., Baker, M.P., Computer Graphics, Prentice-Hall, 2004
- [12] Cho, J., Jung, H., Lee, J., Lee, D., and Ahn, H., 2007. "Haptic Rendering of Drilling into Femur Bone with Graded Stiffness", *International Conference Frontiers in the Convergence of Bioscience and Information Technologies*, pp. 525-530.
- [13] Cho, J., Jung, H., Yu, I., Lee, K., Lee, D., Ahn, H., Park, I., Yeo, S., and Han, S., 2007. "Surface-Data-Based Haptic Rendering for Simulation of Surgery of Closed Reduction and Internal Fixation", *Proceedings of the 29th Annual Int. Conference of the IEEE EMBS Cité Internationale*, Lyon, France, August.
- [14] Hoogen, J., Riener, R., and Schmidt, G., 2002. "Control aspects of a robotic haptic interface for kinesthetic knee joint simulation", *Control Engineering Practice*, November, 10(11), pp.1301-1308.
- [15] Park, H., and Lee, J., 2004. "Adaptive impedance control of a haptic interface", *Mechatronics*, April, 14(3), pp. 237-253.
- [16] Lee, C. D., Lawrence, D. A., Lucy, L.Y., and Pao, Y., 2004. "Isotropic force control for haptic interfaces, Control Engineering

- Practice”, *Mechatronic Systems*, November, 12(11), pp. 1423-1436.
- [17] Lee, S. S., and Lee, J. M., 2003. “Design of a general purpose 6-DOF haptic interface”, *Mechatronics*, September, 13(7), pp. 697-722.
- [18] Agus, M., Giachetti, A., Gobbetti, E., Zanetti, G., and Zorcolo, A., 2003. "Adaptive techniques for real-time haptic and visual simulation of bone dissection", *IEEE Virtual Reality Conference 2003 (VR 2003)*, pp. 102.
- [19] Balijepalli, A., and Kesavadas, T., 2003. "A Haptic Based Virtual Grinding Tool", *11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS'03)*, pp. 390.
- [20] Agus, M., Giachetti, A., Gobbetti, E., Zanetti, G., and Zorcolo, A., 2002. "Real-Time Haptic and Visual Simulation of Bone Dissection," *IEEE Virtual Reality Conference 2002 (VR 2002)*, pp. 209.
- [21] Chang, Y. H., Chen, Y. T., Chang, C.W., and Lin, C.L., 2008. "Development Scheme of Haptic-Based System for Interactive Deformable Simulation", *Computer-Aided Design, in press*. October.
- [22] Möller, T., 1997. “A Fast Triangle-Triangle Intersection Test”, *JGTOOLS: Journal of Graphics Tools*, 2(2), pp.25-30.
- [23] Tropp, O., Tal, A., and Shimshoni, I., 2006. “A Fast Triangle to Triangle Intersection Test for Collision Detection”, *Computer Animation and Computer Worlds*.
- [24] Held, M., 1997. “A Collection of Efficient and Reliable Intersection Tests”, *Journal of Graphics Tools*, 2(4), pp.25-44.
- [25] Devillers, O., Guigue, P., 2002. “Faster Triangle-Triangle Intersection Tests”, *Technical Report 4488, INRIA*.
- [26] Onbaşıoğlu, E., Atalay B., Goularas D., Soydan A., Okyar F. Şafak K., 2010. “Visualization of Burring Operation in Virtual Surgery Simulation”, *ASME 10th Biennial Conference on Engineering Systems Design and Analysis ESDA 2010*.
- [27] Onbaşıoğlu, E., Paker, Y., 1997. “A Comparative Workload-Based Methodology for Comparison of Parallel Computers”, *Future Generation Computer Systems*, 12, pp.521-545.