

# Sanal Ortamda Nesnelerin Haptic Kol ile Manipülasyonu

Alpaslan Duysak<sup>1</sup>, Sevcan Aytekin<sup>1</sup>

<sup>1</sup> Dumlupınar Üniversitesi, Bilgisayar Mühendisliği Bölümü, Kütahya

aduyusak@dpu.edu.tr, saytekin@dpu.edu.tr

**Özet:** Bilişim teknolojileri gündelik hayatımıza her geçen gün daha fazla nüfuz etmekte ve kullanım alanları genişlemektedir. Eğitim, sağlık, ulaşım, ekonomi ve eğlence gibi birçok alandaki ihtiyaçlar yazılım teknolojileri ile karşılanabilir hale gelmiştir. Fiziki çevre ve şartlarının bilgisayar ortamında Üç Boyutlu (3B) olarak simüle edilmesi hızla artarak yaygınlaşmıştır. Gerçek ortamların ve uygulamaların eğitim veya test için sanal ortamda 3B olarak simüle edilmesi, gerçek hayattaki uygulamaları kolaylaştırdığı birçok araştırmada görülmektedir. Bu çalışmada nesnelerin 3B olarak nasıl modellenecekleri, elde edilen modellerin sanal ortamda nasıl manipüle edilecekleri ve interaktif olarak fare ve haptic kolun kullanımı incelenmiştir. Modeller 3Dmax programı ile oluşturulmuş ve OpenGL ortamına taşınmıştır. Sanal dünya ve interaktif etkileşim OpenGL ve C++ programlama dili ile gerçekleştirilmiştir. Geliştirilen sanal ortam programı modellerin ağ yapısının Kütle-Yay sistemleri (Mass-Spring Systems) ile modellenmesine olanak vermektedir. Böylece cisimler haptic kol ile hem manipüle (döndürme ve öteleme) hem de deformasyona tabi tutulabilmektedirler. Bu çalışma ile elde edilen sanal ortam programı deformasyon simülasyonu için ve eğitim programları (nesne öğrenimi gibi) için kullanılabilirlerdir.

**Anahtar Sözcükler:** Sanal Ortam, 3B Modelleme, Kütle Yay Sistemi, Haptic Kol

## Manipulation of Objects With Haptic Device in Virtual Environment

**Abstract:** Information technologies are taking much more place in our daily life day by day and usage areas of them are getting wider. Needs in many subjects such as education, health, transportation, economy and entertainment can be done by software technology. The Three Dimensional (3D) simulation of physical environment and conditions in computers becomes widespread. The applications in real life become very easy by assistance of the simulation of real environments and applications for education or test, and this can be seen in many researches. In this research how 3D objects can be modeled, how obtained models can be manipulated in virtual environment and usage of mouse and haptic device are examined. The models are formed by 3Dmax software and move to OpenGL platform. Virtual world and interactive interaction is materialized by OpenGL and C++. The materialized virtual environment software gives an opportunity to shape the models with web structure of Mass-Spring Systems. By doing this, objects can be both manipulated and subjected to deformation by haptic arm. Virtual environment software which is gotten by this research can be used for deformation simulation and education programs (i.e. learning objects).

**Keywords:** Virtual Environment, 3D Modelling, Mass-Spring System, Haptic Device

## 1. Giriş

Sanal ortamda yapılan çalışmalar uzun yıllardan beri devam etmektedir. 1980 yılı ve sonrasında kişisel bilgisayar kullanımının artması ve internetin gündelik hayata hızla yerleşmesi sanal ortamda yapılan çalışmalara yön vermiştir. Bu çalışmalar, yazılımsal ve donanımsal araçlar yoluyla fiziksel dünyayı sanal dünyaya taşımayı hedefler. Yani gerçek ortamlar benzetim yoluyla sanal ortamda 3B olarak modellenir ve simüle edilir. Gerçek ortamlarda pratikte öğrenmenin ve uygulamanın zor veya pahalı olduğu durumlarda veriler sanal ortamda işlenerek öğrenme gerçekleştirilir ve gerçek uygulamalar için deneyim kazanılır. Örneğin bir tıp öğrencisinin cerrahi eğitiminde simülasyon önemli bir öğrenme aracı olabilmektedir. Hastanın operasyon öncesi ve sonrası durumlarını analiz

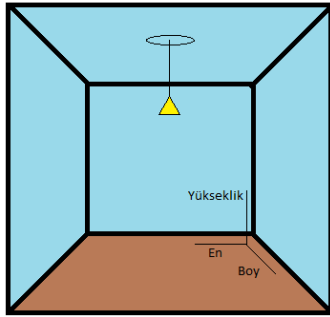
etme, ortaya çıkabilecek faktörleri önceden görebilme imkanı bu yolla sağlanabilmektedir. Tıp alanında olduğu gibi havacılıkta da, pilotların uçuş eğitimlerinde simülasyonlar ile öğrenme gerçekleştirilmektedir. Simülasyonların pek çok alanda kullanımının giderek artması bilgisayar destekli tasarıma ve bilgisayar grafikleri yazılımlarındaki çalışmalara hız kazandırmaktadır.

Simülasyonlar, gerçekte var olan uygulamalara ortam yaratılması, problemlere çözüm üretilmesi, verilerin test edilmesi, fiziki şartlarda denenmesi zor ve risk taşıyan uygulamaların hayata geçirilmesi gibi çalışmalara imkan vermektedir. Sanal ortam uygulamaları sadece gerçek hayattan örneklerle sınırlı kalmayıp, gerçekte var olmayan senaryoların da geliştirilip gerçek hayatta işlev kazandırılmasını sağlar nitelikte olabilir [1].

Bu çalışmada sanal ortamın ve 3B nesnelerin nasıl modellendiği, kullanıcı ile sanal ortam etkileşimini sağlayan çevre birimlerinin neler olduğu ve hangi hesaplama yöntemlerinin kullanıldığı gibi konular incelenmiştir.

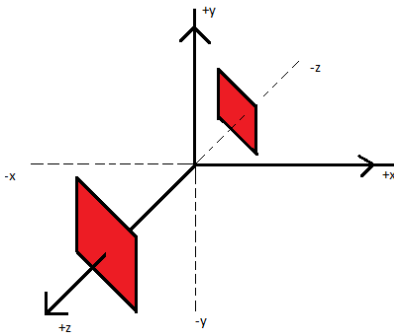
## 2. Üç Boyutlu (3B) Modelleme

3B modelleme, bir nesnenin x, y, z koordinatlarında matematiksel olarak ifade edilmesidir. Tasarlanan model geometrik bir yapıdır. 3B bir yapının simülasyonunda, yapıyı oluşturan x, y ve z koordinatlarının her bir yeni durum için hesaplanması gerekir.



Şekil 1. 3B oda

Şekil 1'de OpenGL ile modellenmiş bir oda yer almaktadır. x, y, z koordinat düzleminde Şekil 1'de en; düzlemin x eksenini, yükseklik; düzlemin y eksenini, boy; düzlemin z eksenini olarak tanımlanmaktadır (Şekil 2).



Şekil 2. x, y, z koordinat eksenleri

Bir koordinat düzleminde bir nesne +z eksenine doğru hareket ettirilirse nesne kullanıcıya yaklaşma, -z eksenini boyunca hareket ettirilirse uzaklaşma hissini vermektedir.

Sanal ortamın ve bu ortam içerisinde yer alan nesnelerin gerçeğine benzer şekilde modellenmesi son derece önemlidir. Bu nedenle hem sanal ortamın oluşturulmasında hem de nesnelerin modellenmesinde ve manipülasyonunda kullanılacak hesaplama yöntemleri ile yazılım ve

donanım kaynaklarının kombinasyonu iyi sağlanmalıdır.

## 2.1 Sanal Ortamın Oluşturulması

Bir simülasyon uygulaması, kullanıcının o uygulama ile ilgili taleplerini karşılayabilir nitelikte olmalıdır. Kullanıcının sanal ortamlarla arasında köprü vazifesi gören çevre birimlerinin kontrolü iyi sağlanmalıdır. Simülasyon ortamını oluşturmadan önce yazılımsal ve donanımsal ihtiyaçların belirlenmesi gerekir. Simülasyonda kullanılacak aygıtların kullanıcı ile etkileşiminin sağlanabilmesi için ilgili bağlantıların ve iletişim yollarının hazırlanması gerekir. Ve bu bağlantıların bir takım yordamlarla yazılımın programa dahil edilmesi sağlanır.

Sanal ortamda kullanılan başlıca çevre birimleri ise şunlardır [1] :

- Klavye, Fare
- Kumanda Paneli (JoyStick)
- Haptic Kol
- Sanal Gerçeklik (Virtual Reality)

Aygıtları

- Kask (VR Helmet)
- Gözlük (VR Glasses)
- Eldiven (VR Gloves)

Klavye ve fare, sanal ortamla iletişimin en kolay yoldan sağlandığı ve kullanıcıların kullanmakta en alışık olduğu birimlerdir. Klavyedeki tuşlar ve farenin sağ, sol (click) tuşları ile nesnelere istenilen şekilde yönlendirilebilir. Sanal ortamda bu aygıtlar ile nesne bir yerden bir yere taşınabilir, istenilen yöne döndürülebilir, şekli değiştirilebilir ya da klavyeden girilen bilgi ile manipülasyonu sağlanabilir.

Haptic kol, son yıllarda gerçek zamanlı çalışmaların yapıldığı uygulamalarda önemli bir araç haline gelmiştir. Kullanıcıya dokunma hissi verebilme özelliği ile diğer çevre birimlerinden ayrılır. Sanal ortamda Haptic kol ile dokunma duyusunun algılanması sağlanabilmektedir.



Resim 1. Sensable Haptic Kol

Bir simülasyon programı için gerekli aygıtların belirlenmesi ve iletişiminin sağlanmasından sonra uygulamanın amacına uygun yazılımın yazılması gerekmektedir.

Kullanılan platforma bağlı olarak başlıca yazılımsal ihtiyaçlar ise şunlar olmuştur:

- C Derleyicisi
- Kütüphaneler
- Yardımcı grafik programları (3Dmax, vb.)

Yazılımda programlama dili olarak C kullanılmıştır. C, kullanımı kolay ve basit bir dildir. Bir C derleyicisi ile program verimliliği test edilebilir. Kullanılan programlama diline bağlı olarak seçilen bir derleyici, programa hızlı cevap verebilme ve etkin kullanılabilme özelliğine sahip olmalıdır. C ile yazılmış uygulamalar diğer platformlarda rahatlıkla kullanılabilir.

Programın yazma aşamasında ihtiyaç duyulan kütüphane dosyaları sisteme aktarılabilir ve kullanılabilir hale getirilmelidir. Nesnelerin tasarım aşamasında ise 3B grafik programlarından yararlanılabilir [1].

### 2.1.1 Görüntü Oluşumu

Bir görüntünün oluşumunda kamera ve ışık önemli faktörlerdir. Kameranın konumu, geliş açısı ve kullanıcının bakış açısı birlikte hesaplanarak ekrana yansıtılır. OpenGL'de görüntü oluşumu esnasında içinde bulunulan ortamın koordinat ve piksel bilgileri de göz önüne alınarak kullanıcının görmek istediği alana bağlı olarak kameranın bulunduğu yer, kameranın yönü ve bakış açısı bilgileri `glPerspektif()`, `glOrtho()`, `glLookAt()` gibi fonksiyonlarla hesaplanır.

Işık görmemizi sağlayan en önemli faktördür. Bir ortamın aydınlatılmasında ışığın geliş yönü, rengi, yansıttığı nesnelerin cinsi ve yüzey bilgileri gibi etkenlerin bilinmesi gerekir. Işık doğrudan bir ışık kaynağından gelebildiği gibi bir yüzeyden yansıyarak da gözümüze ulaşabilir. Işığın nesnelere yansıttığı bilgiler nesnenin malzemesine ve yüzeyine bağlı olarak değişir. OpenGL'de ışık kaynağının eklenmesi, ışık yönünün belirlenmesi, açısı, ışık çeşitleri gibi bilgiler OpenGL'in kendi çağruları ile programa kolaylıkla eklenebilir [2].

### 2.1.2 OpenGL

OpenGL, gelişmiş donanım desteğini kullanarak 2B veya 3B grafik çizdirmek için kullanılan bir uygulama geliştirme aracıdır [1].

Grafik donanımına bir yazılım arayüzü sağlayan OpenGL, içinde yüzlerce yordam ve fonksiyonu barındırır. Özellikle 3B grafiksel modellemede

yüksek kalitede görüntüleme ortamı sağlar. OpenGL çağruları ile program içinde nokta, doğru, poligon (polygon), resim gibi primitifleri kullanma imkanı sağlar [2].

OpenGL, API (Application Programming Interface) uyumlu donanımlar üzerinden ekrana şeklin çizilmesini sağlar. OpenGL, API içerisinde bulunan çağruları kullanmak için, içinde 750'den fazla fonksiyonu ve komutu barındırır. Bu komutlar temel grafik öğelerini (nokta, poligon, doğru, vb.) çizdirmeye yarar. Fonksiyonlar yardımıyla tasarlanan modellere bazı gelişmiş özellikler eklenebilir. Işıklandırma ve gölgeleme, karıştırma (Blending), sis (Fog), doku kaplama, transparanlık, hareket, ses gibi efektler kazandırılır.

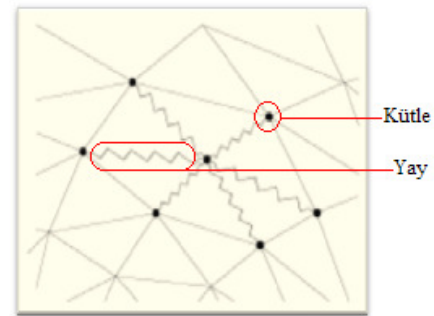
Birçok platformda OpenGL kendi islev kutuphanesi olan GLU ile birlikte gelmektedir. Bu kütüphanenin içerisinde yüzey eğri ve bazı özel matris işlevlerini yapmaya yarayan yordamlar tanımlanmıştır. Glut kütüphanesini kullanarak, çok daha az bir kod satırıyla daha yüksek bir performansa sahip uygulamalar geliştirmek mümkündür.

## 2.2 Hesaplama Yöntemleri

### 2.2.1 Kütle- Yay Sistemi

Simülasyon uygulamalarında nesnelere hareket, biçim, yön gibi kavramların verilmesi; nesnelerin çarpma, kırılma, eğilme gibi bir etkiyle karşılaşması durumları olabilir. Cisimler bu durumlar karşısında geçici şekil bozulmalarına veya kırılmalara maruz kalabilir. Bu durumların hesaplanmasında fiziksel şartlar göz önüne alınarak uygun hesaplama yöntemleri geliştirilmektedir.

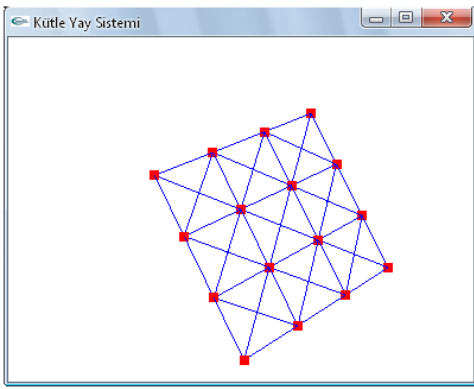
Cisimler çeşitli geometrik yapılara (üçgen, dörtgen, vb.) göre modellenebilir. Bu modelleme ağ yapısı (meshing) şeklindedir. Şekil 3'de ağda bulunan her bir düğüm (vertex) sanal olarak belirlenen kütle noktalarını temsil eder. Noktaları birbirine bağlayan doğrusal bağlar ise yayları (spring) temsil eder.



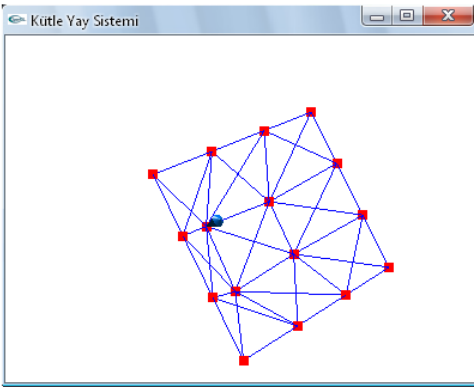
Şekil 3. Kütle Yay modeli [4]

Şekil 4a'da 16 adet düğüm ve bu düğümlerin her biri birbirine komşusu olan düğümlere üçgensel bir ağ yapısıyla bağlanmıştır. Model üzerindeki

herhangi bir noktaya dışarıdan bir kuvvet uygulandığında noktaya bağlı yaylar kuvvetin etkisiyle sıkışma veya esneme hareketi yapmak istemektedir. Kuvvetin etkisiyle birbirine bağlı noktalar birbirleriyle iletişime geçer. Kuvvetin uygulandığı yönde etkilenen düğüm belirli bir oranda ilerleyerek komşu düğümlerle bağlı olduğu yayları sıkıştırmakta veya esnekletmektedir (Şekil 4b). Yaylar bağlı bulunduğu noktayı kuvvetin uygulanma yönüne göre itme veya çekme görevini yerine getirmektedir. Böylece ilk iterasyon ile en yakın noktalar, kuvvetin uygulanma yönünde ve kuvvetin şiddetiyle yer değiştirmektedir. Bu şekilde işlemler dizisi uygulanan kuvvetin büyüklüğü derecesinde ilerleme kaydeder. Uygulanan kuvvetin etkisi geçince kütleler ve yaylar denge konumlarına geri dönmektedir [4].



Şekil 4a. Üçgenel ağ yapısı ile oluşturulan bir model



Şekil 4b. Modelin bir noktasına etki eden kuvvet sonucu deformasyonu

Dışarıdan etki eden bir kuvvete karşı noktaların hareket etmesi nesne üzerinde deformasyona veya şekil değişikliklerine yol açmaktadır. “Kütle-yay sistemi bu noktalara bağlı elastik bağların ve nokta kümesi elemanlarının birbirleriyle olan dinamiksel davranışlarını inceleyen bir sistem modeli olarak tanımlanır” [1].

Kütle yay sistemlerinde, kütle noktaları üzerinde oluşan toplam kuvvet, her bir kütle için yaylar üzerine yaptığı kuvvet ve dışarıdan gelen diğer kuvvetlerin birleşiminde oluşmaktadır.

$$f_{top} = f_k + f_d(k) + f_{çevre}$$

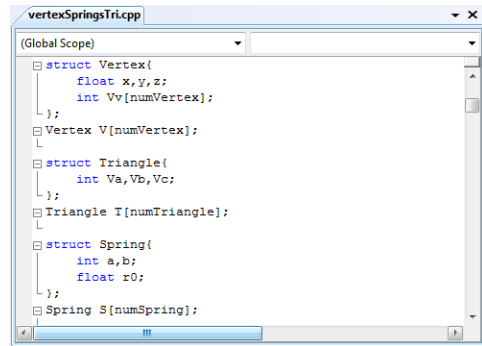
Toplam kuvvetin hesaplanmasında, dışarıdan gelebilecek olan kuvvetlerin ( $f_{çevre}$ ), yay sertliği ve sönümü ile ilgili yay kuvvetlerinin ( $f_k$  ve  $f_d$ ) bilinmesi gerekir. Kullanıcı tarafından belirlenecek kuvvetlerin hesaplanmasında kütlelerin hız, uzaklık ve kuvvetin etkisine bağlı olarak yer değiştirme ilişkisi ise şöyle ifade edilmektedir:

$$f_{top} = f_k + f_d = \sum [k(r - r_0) - d(\|v_a - v_b\|)] \frac{x_b - x_a}{\|x_b - x_a\|}$$

$x_a$  ve  $x_b$ , kütlelerin pozisyon vektörleri,  $v_a$  ve  $v_b$  kütlelerin hız vektörleri,  $r_0$  kütlelerin arasında kalan yay uzunluğunu ifade eder.  $k$  ve  $d$  sertlik ve sönüm sabitleri olarak kullanılmaktadır [5].

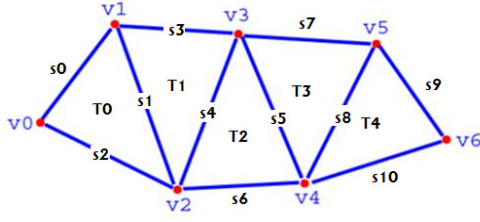
### 2.2.2 Ağ Oluşumu (Meshing)

Simülasyon uygulamalarında 3B modellerin tasarımında program içerisinde vertex (nokta) ve spring (yay) bilgilerinin tutulduğu bir structer (yapı) tanımlanmıştır (Şekil 5).



Şekil 5. Nokta (Vertex) ve Yay (Spring) bilgilerinin tanımlanması

Buradaki amaç, bu bilgileri kullanarak nesneyi hacimsel olarak bir nokta kümesi olarak ifade etmektir. Birbirinden bağımsız olarak düşünülen her bir nokta birbirine doğrusal bağlarla bağlanarak bir nesne modeli oluşturur. Böylece bir ağ yapısı ortaya çıkarak nesnelerin modellenmesi sağlanmış olur (Şekil 6a).



**Şekil 6a.** Modelin ağ yapısı

Modelin 3B olarak tasarımında modelin kütle, yay ve üçgenin hangi noktalardan oluştuğunu tutan veriler bir veri dosyasında tutulmaktadır. x,y,z koordinatında modellenecek nesnenin noktalar veri kümesinde her bir noktanın x,y,z bilgisi, iki nokta arasındaki yay bilgisi ve oluşacak her bir üçgenin nokta bilgileri veri dosyasında olması gereken bilgilerdir. Şekil 6b'deki modelde nokta, yay ve üçgen koordinat bilgileri çizelgelere listelenmiştir.

Vertex	x,y,z	Vertex Meshes
V0	0.8, 1.0, -1.0	V1, V2
V1	1.8, 1.5, -1.0	V0, V2, V3
V2	2.5, 0.0, -1.0	V0, V1, V3, V4
V3	3.2, 1.4, -1.0	V1, V2, V4, V5
V4	3.8, 0.2, -1.0	V2, V3, V5, V6
V5	4.3, 1.2, -1.0	V3, V4, V6
V6	5.0, 0.5, -1.0	V4, V5

**Tablo 1.** Noktaların (düğüm) koordinat bilgisi

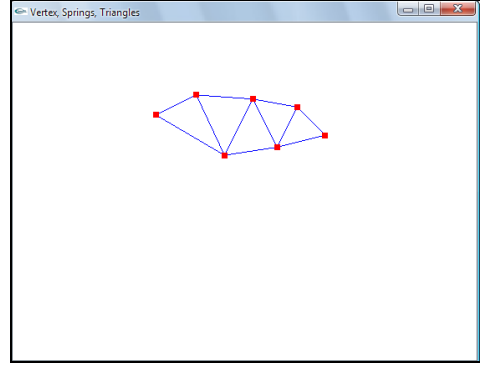
Springs	Vertex List
s0	V0, V1
s1	V1, V2
s2	V0, V2
s3	V1, V3
s4	V2, V3
s5	V3, V4
s6	V2, V4
s7	V3, V5
s8	V4, V5
s9	V5, V6
s10	V4, V6

**Tablo 2.** Nokta yay ilişkisi

Triangles	Vertex List	Springs List
T0	V0, V1, V2	s0,s1, s2
T1	V1, V2, V3	s1, s3,s4
T2	V2, V3, V4	s4, s5, s6
T3	V3, V4, V5	s5, s7, s8
T4	V4, V5, V6	s8, s9, s10

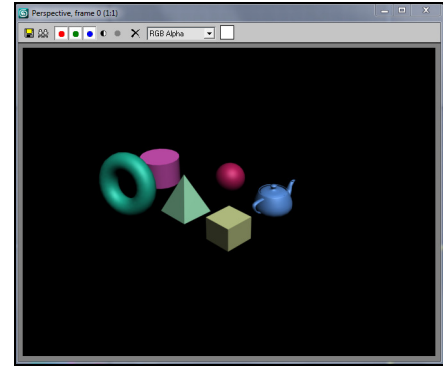
**Tablo 3.** Her bir üçgenin nokta yay bilgisi

Tablolarda yer alan bilgiler vertex.txt, springs.txt, triangles.txt dosyalarında saklanmaktadır. vertex.txt dosyası her bir noktanın üç boyutlu düzlemde x,y,z koordinat bilgilerini, springs.txt dosyası hangi noktalar arasında bağlantı kurulacağı bilgisini, triangles.txt dosyası ise oluşturulacak üçgenlerin hangi noktalardan oluşacağı bilgisini vermektedir.



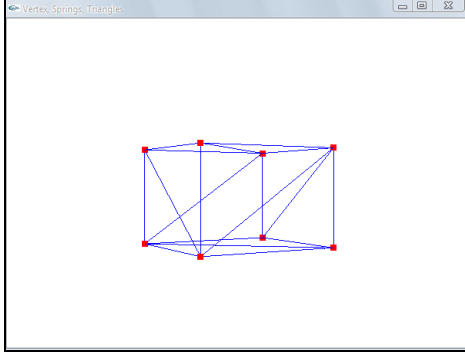
**Şekil 6b.** Modelin sanal ortamda görüntüsü

Şekil 6b'de bir modelin düğümlerini ve koordinat bilgilerini bilerek o modeli üçgensel bir ağ yapısına dönüştürebilmekteyiz. Peki bu düğüm ve koordinat bilgilerini nasıl elde edeceğiz? 3Dmax programı bize istediğimiz modeli yaratma imkanı sunabilmektedir. Şekil 7'de 3Dmax ile kolay bir biçimde oluşturduğumuz matematiksel nesnelere görülmektedir. Bu sayede içinde her bir nesnenin nokta ve bu noktaların koordinat bilgilerinin bulunduğu C dosyasını bir fonksiyon olarak OpenGL'e dahil edebilmekteyiz.



**Şekil 7.** 3DsMax'de çizilmiş 3B matematiksel modeller

Şekil 8'de 3Dmax ile çizilmiş bir dikdörtgenler prizması yer almaktadır. Bu modeldeki düğümler ve bu düğümlerin koordinat bilgileri yazılıma eklenmiştir. Her bir düğümün ve bu düğümlerin komşu düğümleri ile bağlantısı (yay) hesaplanarak OpenGL ile üçgensel ağ modeli oluşturulmuştur.



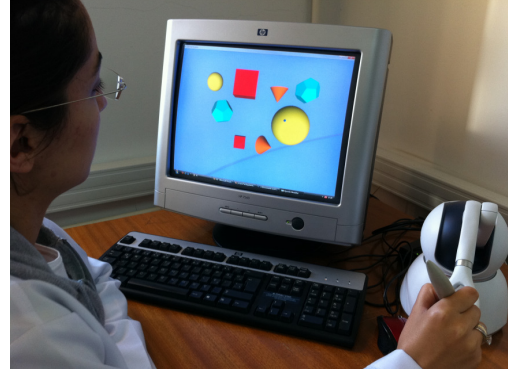
Şekil 8. 3B dikdörtgenler prizması

### 3. Haptic Teknolojisi

Haptic aygıtları, insan ve bilgisayar arasında 3B veri alışverişi sağlayan bir donanım birimidir. Kullanılan diğer çevre birimlerinden farkı, sanal cisme dokunuyormuş hissi vermesidir. Sanal gerçeklik ortamlarında haptic aygıtların kullanımı uygulamalara farklı bir boyut kazandırmaktadır. Özellikle tıp alanında kullanımı giderek artmaktadır [6].

Haptic cihazlar simülasyon ve mekanik kısmı olmak üzere kullanıcı ile iletişimi sağlanmaktadır. Kullanıcının cihaz ile verdiği mekanik sinyaller işlenerek kullanıcıya geri dönmektedir. Kullanıcı, gelen bu sinyal ile sanal nesneye gerçekten dokunduğunu geribildirim ile hissetmektedir. Haptic cihazları kullanım alanına göre çeşitli şekle karşımıza çıkmaktadır. Bunlardan bazıları mekanik kol haptic cihazlar, tel tahrikli haptic cihazlar, manyetik haptic cihazlar şeklindedir. Literatürlerde ilk mekanik haptic kol Sensable Technology Inc. tarafından üretilmiştir [7].

Resim 2’de Sensable Haptic Robot kol, kullanıcının elinde tuttuğu kalem ile sanal ortamdaki nesnelere dokunma eylemini gerçekleştirmektedir. Kalem kol olarak ifade edilir. Kullanıcının kolu hareket ettirmesiyle sanal ortamda istenilen yöne gidilebilir. Haptic kol, fare ve klavyede olduğu gibi benzer şekilde etkileşimi sağlanarak, nesnelere hareket, itme-çekme, yer değiştirme gibi işlevsellik kazandırılmaktadır [7]. Ayrıca bazı simülasyonlarda kalem fırça görevi görerek sanal nesnelere istenilen renklere boyanmaktadır.



Resim 2. Haptic kol ile sanal ortam etkileşimi

Simülasyonda haptic kullanımı için haptic kütüphanesinin programa dahil edilmesi ve gerekli yordamların çağırılması gerekmektedir. Haptic kütüphanesi (HL), OpenGL ‘de GL kütüphanesine benzer bir yaklaşım sergilemektedir. Sensable Phantom Omni Haptic aygıtı sürücülerini, haptic hütüphane API (HLAPI), kullanım klavuzu ve web sayfasındaki uygulamaları ile yazılımcı ve kullanıcılara geniş bilgiler sunmaktadır [8].

### 5. Sonuç ve Öneriler

Günümüz teknolojisi ile gündelik hayatımızda birçok uygulamayı sanal ortamlarda görebilmekteyiz. Özellikle risk taşıyan hayati uygulamalarda pratiklik ve tecrübe kazanılması açısından simülasyonlara büyük ihtiyaç duyulmaktadır. Bir sürücünün direksiyon eğitiminden bir pilotun uçuş eğitimine, bir cerrahın operasyonundan bir hastanın tedavisine kadar olan uygulamalarda geliştirilen metotların, yazılımların, üretilen donanımların ne kadar önemli olduğu ortaya çıkmaktadır. Bu çalışmada kullandığımız Haptic kol, gerçek ortamda algıladığımız dokunma duyusunu sanal ortamda modellenen nesnelere gerçekte dokunuyormuş gibi hissedilmesinde bir araç olmaktadır. Bu aygıt doku ve organ modellemede cerrahi simülasyonlar için önemlidir.

Bu çalışmada OpenGL ile grafik programlarını kullanarak sanal ortamda 2B ve 3B nesnelere oluşturulduktan sonra nesnelere, geliştirilen hesaplama yöntemleri ile geometrik ağ yapısına dönüştürülmektedir. Nesnelere bu şekilde modellemedeki amaç, dışarıdan gelebilecek bir etkiye karşı nesnenin gösterdiği tepkinin incelenmesidir. Özellikle deformasyon uygulamalarında, gerçek ortamda fizik kanunlarına göre ortaya çıkabilecek sonuçların sanal ortamda da benzer sonuçları ortaya çıkarmasına zemin hazırlamaktır. Sanal ortamdaki 3B nesnelere şekil, renk, biçim, hareket gibi kavramların öğretilmesinde interaktif eğitim programlarına ortam hazırlayacak niteliktedir.

## 6. Kaynaklar

- [1] Aytakin, S., “Sanal Ortamda Cisimlerin Haptic Kol ile Manipülasyonu ve Deformasyonu”, Y. Lisans Tezi, Temmuz 2011
- [2] The OpenGL Programming Guide: The Official Guide to Learning OpenGL Version 3.0 and 3.1
- [3] Şekercioğlu, A. S., “Moleküler Modelleme ile Yumuşak Doku Deformasyonunun Modellenmesi ve Simülasyonu”, Y.Lisans Tezi, Ocak 2010
- [4] Duysak, A., Jian J. Zhang, “Fast Simulation of Deformable Objects,” iv, pp.422-427, Eighth International Conference on Information Visualisation (IV’04), 2004
- [5] Alpaslan DUYSAK, A. Sait ŞEKERCİOĞLU, Molecular Modelling for Deformable Object Simulation, The 2009 International Conference on Computer Graphics and Virtual Reality, CGVR’09, pp. 28-31
- [6] Balkan, T., Konukseven, E. İ., Koku A., Başer, Ö., Bideci, S., 2008, “Haptic” Dokunma Hisli ve Kuvvet Geri Beslemeli Arayüz Sistem Tasarımı
- [7] Karal, H., Reisoğlu, İ., Haptic Teknolojisinin Simülasyon ve Geleneksel Yöntemlere Göre Öğrencilerin Akademik Başarılarına Etkisi
- [8] Sensable Technologie Inc., OpenHaptics Programmer’s Guide, 2005