

Bilgisayar Grafik Yazılımı Görüntünün Matematiği

Bilgisayar grafiği donanımında son yıllarda büyük gelişmeler gözlemlendi, karmaşık işlemleri çok kısa sürelerde yapabilecek değişik aygıtlar üretildi. Ancak bir bilgisayar sistemi, başarısını donanımdan çok, yazılıma borçludur. Fotoğraf gerçekliğinde yapay görüntüler elde etmeyi amaçlamış olan bilgisayar grafiği bilim dalı, yazılım konusunda da başarılı örnekler vererek, tiptan reklamcılığa, makine mühendisliğinden filmciliğe kadar sayısız uygulama alanında çok etkileyici sonuçlar elde etmiştir.

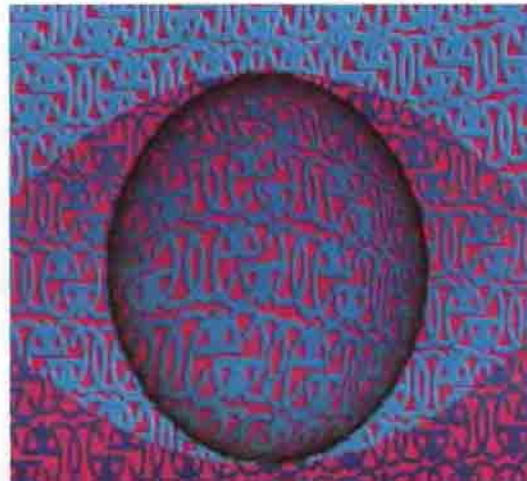
Bülent Özgüç
Bilkent Üniversitesi
Mühendislik Fakültesi

GRAFİK yazılımı, kullanım amacına göre ikiye ayrılır. Bunlar genel grafik paketleri ve özel amaçlı uygulamalar için kullanılan paketlerdir. Genel grafik program paketleri geniş kapsamlı grafik uygulamaları

için kullanılır. Bunlar C veya benzeri üst düzeyli programlama dilleri için kullanılırlar. Genel grafik program paketleri için bir örnek Silicon Graphics® iş istasyonlarında bulunan GL (Graphic Library) sistemidir. Genel amaçlı paketlerin sağladığı işlemler temelde şunlardır: Düz çizgiler, çokgenler,

daireler, diğer geometrik şekil örneklerini oluşturmak, renk ve yoğunluk değerleri, görüntülerin seçimi ve dönüştürme uygulamaları. Boyama - tonlama - aydınlatma yöntemlerinden bir kısmı da artık genel amaçlı yazılım sistemlerinde bulunmaktadır. Bu tür yazılımları kullana-

cak kişinin bilgisayar grafiği ile programlama bilgisi olması beklenir. Özel amaçlı grafik yazılımları ise kullanıcılardan grafik işlemlerinin nasıl yapılacağı ve nasıl program yazılacağı bilgisini beklemeden rahatça kullanılmak üzere hazırlanmıştır. Bu tür paketlerde ara birimler, kullanıcının kendi is-



tekleri doğrultusunda programlarla bağlantı kurmasını sağlar. Bu uygulamalara boyama - aydınlatma programları, çeşitli işletmecilik, tıbbi ve bilgisayar destekli tasarım (CAD) yazılımları örnek verilebilir.

Koordinat Tanımları

Genel grafik paketleri, çok azı dışında, kartezyen koordinatları ile kullanılmak üzere tasarlanmışlardır. Eğer koordinat değerleri başka şekillerde (küresel, parabolik vb.) alındıysa, bu değerler grafik paketlerine girilmeden önce kartezyen koordinatlarına çevrilmelidir. Özel amaçlı paketler, uygulamalara elverişli hale getirilmiş diğer koordinat türleri kullanma olanağı verir. Genelde ekranı oluşturmak ve görüntülemek için değişik hiyerarşik koordinat düzenlemeleri kullanılır. Ekranda gösterilecek ağaç, mobilya gibi bir çok nesne, önce model veya yerel koordinat ile şekillendirilebilir. Nesnenin şekli belirlendikten sonra "dünya" veya gerçek koordinat olarak adlandırılan sistem içinde alması gereken uygun konuma getirilir. Son olarak gerçek koordinat sistemi ekranda görüntülenmek için ekran koordinat sistemine dönüştürülür. Ekran koordinat sistemi, ekranı oluşturan görüntü elemanları birimindedir. Model veya gerçek koordinatlar ise, uzunluk birimleri, zaman birimleri gibi gösterilecek nesnelerin hakiki boyutları ile ilintilidir.

İki boyutlu sistemler için model koordinatlarından aygıt koordinatlarına yapılan bir dizi koordinat dönüşümlerini tanımlayalım. Başlangıç model koordinat konumu (Xmc , Ymc) aygıt koordinat konumlarına (Xdc , Ydc) şu sırayla dönüştürülebilir:

$(X_{mc} , Y_{mc}) \rightarrow (X_{wc} , Y_{wc}) \rightarrow (X_{nc} , Y_{nc}) \rightarrow (X_{dc} , Y_{dc})$

Model (Xmc , Ymc) ve gerçek (Xwc , Ywc) koordinat konumları herhangi bir kayan noktalı değer olabilir. (Xnc , Ync) ile belirtilmiş normalize koordinatlar ise şu eşitsizliği sağlarlar: $0 \leq X_{nc} \leq 1$, $0 \leq Y_{nc} \leq 1$

Her çıktı aygıtı için koordinatlar Xdc ve Ydc, (0, 0) dan (Xmax, Ymax)'a kadar olan alanda birer tamsayıdır.

Grafik Fonksiyonları

Genel amaçlı grafik yazılımlar çeşitli işlemler kullanılarak resimleri yaratmayı ve uyarlama yapmayı sağlar. Bu işlemler girdi, çıktı, özellikler, dönüşümler, bakış, gösterim veya genel denetimleri içerir.

En temel yapı blokları çıktı işlemleri olarak adlandırılırlar. Bunlar bazı karakter biçimleri, noktalar, düz çizgiler, eğri çizgiler ile bazı alanlar (çokgenler, daireler) gibi geometrik şekilleri ve renkli noktalarla doldurulabilen alanları içerirler. Çıktı işlemleri resimlerin ortaya çıkarılması için en temel yazılım parçalarıdır.

Özellikler, çıktı işlemleri öğelerinin nasıl gösterileceğini açıklar. Bunlar yoğunluk ve renk bilgileri, çizgi türleri, yazı stilleri ve alan doldurma desenleri içerir. Bu kategorideki işlemler her bir çıktı ögesi için gerekli vasıf ve özellikleri oluşturmayı sağlar.

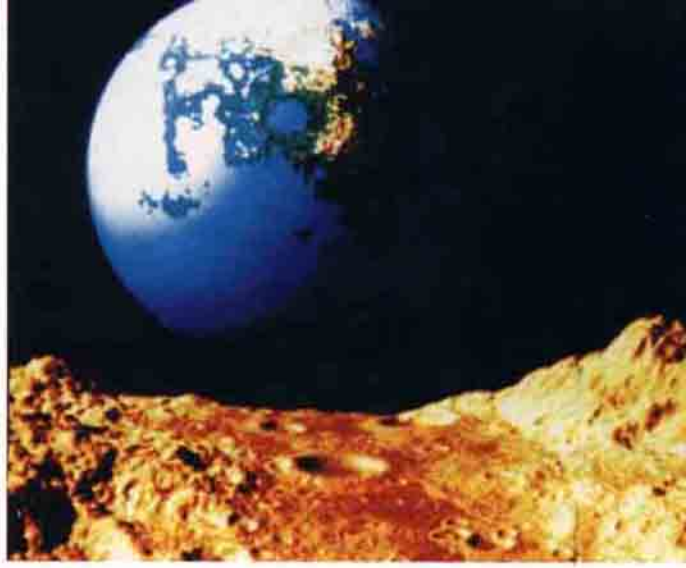
Geometrik dönüşümler kullanarak ekran içinde nesnenin boyutu, konum veya yönlerini değiştirebiliriz. Benzer olarak, model dönüşümleri, model koordinatlarındaki nesne tanımları kullanılarak ekranın oluşturulması için kullanılır.

Model ve gerçek koordinatlarda görüntülenecek sahne hazırlanırken, grafik yazılımlar bakış yönü ve bakış uzaklığına göre gösterim dönüşümlerini yaparlar. Bu dönüşümler ortaya çıkacak görüntüleri, kullanılacak çıktı aygıtına göre izdüşüm olarak ekran veya ekran parçasına yerleştirirler.

Resimler gerekirse belirli parçalara bölünebilirler. Bu parçalara yapı, bölüm veya nesne adı verilir. Her bir yapı, resmin bir mantıksal birimini tanımlar. Bu birimler bölüm veya nesne diye tanımlanan alt birimlerden oluşur.

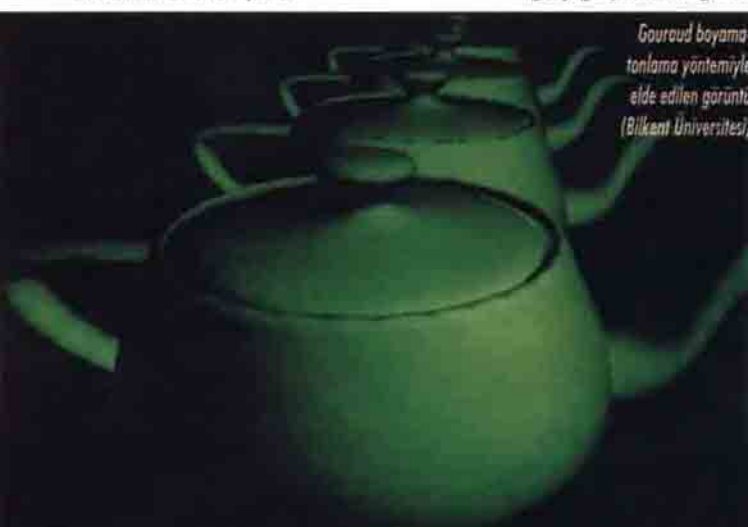
Boyama, Aydınlatma, Tonlama

Geometrik tanımları hazırlanmış ve gösterime gönderilecek üç-boyutlu nesnelerin, gerçeğe yakın bir görüntü verebilmeleri için

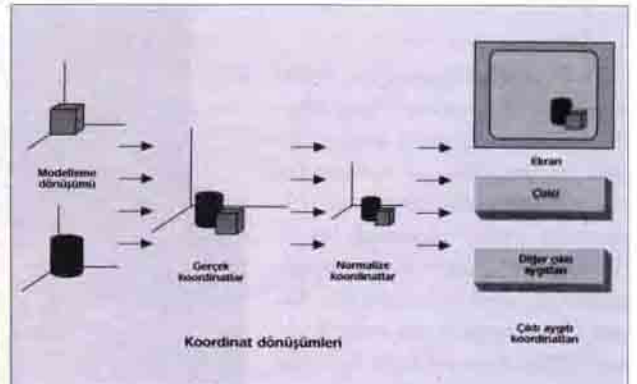


boyanıp tonlanmaları gerekir. Aksi halde, sadece sınırları belirleyen birtakım karmaşık çizgiler görülecektir. Ayrıca, saydam olmayan nesnelere, bakış yönüne göre, bazı yüzeyler tamamen, bazı yüzeyler kısmen görünecek, bazı yüzeyler ise hiç görünmeyecektir. Yüzeylerin görünürlüğünü belirlemek ve boyama - aydınlatma - tonlama yapmak (renk ve ışık yoğunluğu belirlemek) çoğunlukla aynı zamanda yapılır. Görünmeyen yüzeyleri belirlemek için genelde şu yöntem kullanılır: Yüzey, sonlu bir noktalar kümesine dönüştürülür. Herhangi bir noktanın ekrana gönderilmesine sıra geldiği zaman, z-belleği diye adlandırılan ikinci bir görüntü belleğinde, söz konusu noktanın yerleşmesi gereken konuma bakılır. Daha önce bu konuma bir nokta yerleştirilmişse, iki noktanın kullanıcıya göre olan uzaklıklarına bakılıp, hep kullanıcıya en yakın nokta z-belleğinde saklanır. Eğer noktalar saydam veya yarı saydam yüzeylere aitse, noktaların birleşik etkileri z-belleğine yerleştirilir. Bütün noktalar için bu işlem tamamlandıktan sonra, z-belleği ekrana aktarılır. Görünmeyen yüzeylerin bulunması özellikle çok yüzeyli olan nesnelere vakit alıcı bir işlem olup, hızlandırılması hâlâ ilginç araştırmalara konu olmaktadır.

Boyama - aydınlatma veya renklendirme, gene sonlu noktalara dönüştürülmüş yüzeylerin istenilen renge, ışık kaynaklarının uzaklığı, konumu ve yoğunluğu göz önüne alınarak boyanmasıdır. Bu da fiziksel olarak tanımlı bir işlem olmasına rağmen, kullanılan



Gouraud boyama-tonlama yöntemiyle elde edilen görüntü (Bilkent Üniversitesi)





yöntemler zaman alıcıdır. Bilgisayar grafiğinin hâlâ cazip bir araştırma konusudur. Gerçeğe uygunluktan biraz özveri yapılırsa, basitleştirilmiş yöntemler kullanılabilir. En basit boyama - aydınlatma, her yüzeyin orta noktasındaki ışık yoğunluğunu, yüzey normal vektörünün ışık kaynağı ile olan açısını ve orta noktanın ışık kaynağından uzaklığını göz önüne alarak hesaplamak ve bütün yüzeyi, bulunan bu tek değer ile aydınlatarak boyamaktır. Bu yöntem 'sabit aydınlatma' denir. Bu, özellikle büyük yüzeylerde yarıltıcı sonuçlar verebilir. Bu nedenle, aydınlatma değerleri, yüzeylerin birkaç uzak noktası için bulunup, ara değerler belirli ortalamalar alınarak hesaplanabilir. Phong ve Gouraud isimli iki bilim adamının bu konudaki çalışmaların oldukça gerçekçi görüntüleri kısa sürede elde etmeyi sağlamıştır. Yukarıda anlatmış olduğumuz boyama - aydınlatma yöntemleri, sadece bireysel yüzeyler düzeyinde ele alınmakta ve birden fazla yüzeyin görsel etkileşimi hesaplanmamaktadır. Ancak, gerçekçi görüntülerde, nesnelerin birbirleri üzerine gölgelerinin düşmesi, nesnelerin parlak yüzeyler üzerinde yansımaları, ışığın geçtiği ortam değişince kırılma meydana gelmesi ve yakın nesnelere renk akması çok sık oluşan fiziksel kurallardır. Bunları da içeren aydınlatma - boyama yöntemleri daha karmaşık olup, en yaygın olanları *ışın* (radiosity) ve *ışın izleme* (ray tracing) uygulamalarıdır.

İleri Grafik Uygulamaları

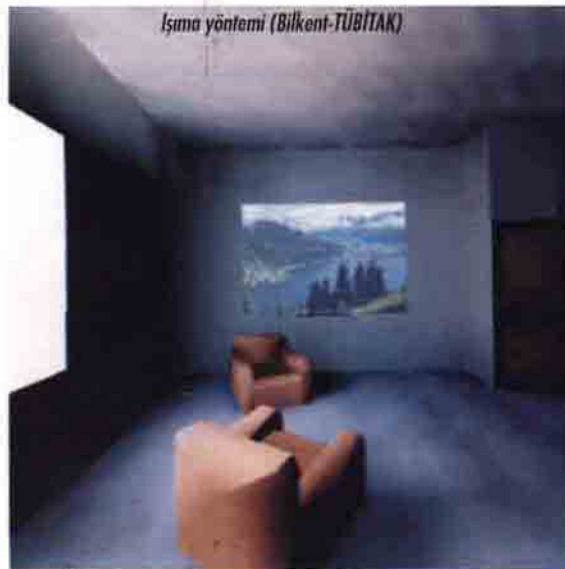
İşın

Bilgisayarda gerçeğe uygun sentetik görüntülerin üretilmesi uzun yıllardan beri bilgisayar grafiği alanında çalışan araştırmacıların ilgi odaklarından biri olmuştur. Amaç, gerçekte var olmayan bir ortamın bilgisayarda oluşturularak gerçeğe uygun bir şekilde bilgisayar ekranında gösterilmesidir. Bilgisayarda gerçeğe uygun sentetik görüntü üretilmesinin birçok kullanım

alanı vardır. Örneğin, bir iç mimarî tasarım ve dekorasyon alanında, bilgisayarda oluşturulacak bir bina modelinin gerçeğe uygun bir şekilde gösterilebilmesi ile bina daha tasarım halinde iken, sanki bitmiş gibi içerisinde gezilip gerektiğinde, tasarımda kolayca değişiklikler yapılabilmesi olasılığı vardır.

Bilgisayarda gerçeğe uygun sentetik görüntü üretebilmek için bir yüzeyden yansıyan ışığı doğru olarak hesaplamak ve simüle etmek gerekir. Çevremizdeki cisimleri yüzeylerinin üzerlerine düşen ışığın yansıması ile görürüz. Kapalı bir ortamda (örneğin oda, büro, ev gibi) bulunan bir cisimden veya yüzeyden yansıyan ışık miktarı ve rengi, yüzeyin rengine, doğrudan ışık kaynaklarından ve ortamdaki diğer yüzeylerden yansıyan ve o yüzeyin üzerine düşen ışık miktarına bağlıdır. Bir yüzeyin üzerine düşen ışık ise, yüzeyin ortamdaki diğer yüzeylere ve ışık kaynaklarına göre geometrik konumuna bağlıdır. Örneğin, ışık kaynaklarını doğrudan görmeyen yüzeyler, doğrudan gören yüzeylere göre daha az ışık yansıtacaktır. Böylece, ortamda gölgeler oluşacaktır.

Günümüzde bilgisayarlardaki hız ve kapasite bakımından sağlanan gelişmeler, gerçeğe uygun sentetik görüntü üretmek için daha karmaşık yansıma ve tonlama modellerinin geliştirilmesini ve kullanılmasını olanaklı kılmıştır. Son yıllarda geliştirilen yöntemlerden biri olan ışın yöntemi kapalı bir alanda mat yüzeylerden oluşan bir ortamdaki yansımaları gerçeğe uygun olarak hesaplamak ve simüle etmekte kullanılmaktadır. Bu yöntem 1984 yılında Cornell Üniversitesi'nde geliştirilmiştir.



Bu yöntemde, önce bilgisayarda oluşturulan ortamdaki yüzeyler küçük yüzey parçalarına bölünür. Ortamda bulunan ışık kaynakları da birer yüzey olarak ele alınır. Parçalara bölme işleminden sonra, ortamdaki her parça arasında durum katsayısı adı verilen değerler hesaplanır ve bu değerler bir matris şeklinde saklanır. Durum katsayısı değeri, bir parçadan yayılan ışık miktarının, diğer bir parça üzerine düşen kısmının oranını verir. Bu değer, parçaların ortamda birbirlerine göre geometrik konumlarına bağlıdır. Örneğin, birbirlerini görmeyen iki parçanın durum katsayısı değerleri sıfırdır. Her parça için bu değerler hesaplandıktan sonra durum katsayısı matrisi, parçaların renkleri ve ışık kaynaklarının şiddetleri kullanılarak bir denklem sistemi oluşturulur. Bu denklem sistemi çözüldükten sonra ortamdaki her parçadan yayılan ışın değeri hesaplanmış olur. Bu değerler renk tonlarına dönüştürülerek ortam, bilgisayar ekranında görüntüleme (rendering) algoritmaları kullanılarak gösterilebilir. Işık kaynaklarını görmeyen parçalar daha az ışın miktarına sahip olacakları için bu parçaların renkleri daha koyu olacaktır. Bu sayede elde edilen görüntülerde gerçeğe uygun şekilde gölgeler oluşacaktır. Şayet sadece ışık kaynaklarının rengi, ışık şiddeti ve/veya yüzeylerin rengi değiştirilirse görüntüyü elde etmek için denklem sisteminin çözümünü yeterli olur. Bu temel yöntem "toplama yöntemi" adı da verilmektedir.

Yukarıda kısaca bahsettiğimiz temel yöntem çok hesaplama gerektirir. Durum katsayısı matrisinin oluşturulması ve çözümü özellikle çok parçadan oluşan ortamlarda uzun zaman gerektirir. Bu nedenle ekranda ilk görüntülerin oluşması uzun sürmektedir. Çok parçadan oluşan karmaşık sahneler için oluşturulan matrisin saklanması için de geniş bilgisayar belleğine gereksinim vardır. Zaman ve bellek gereksinimini azaltmak için 1988 yılında yine Cornell Üniversitesi'nde "dereceli geliştirme yöntemi" (progressive refinement radiosity) adı verilen yeni bir yöntem geliştirilmiştir. Bu yöntem iteratif (yineleyen) bir yöntem olup, sonuca dereceli olarak yaklaşır. Bütün parçalar arasındaki durum katsayısı değerlerini hesaplamak yerine, iterasyonun (yinelemenin) o basamağında en çok ışık yayan parça seçilerek, o parça ile diğer parçalar arasındaki durum katsayısı değerleri hesaplanır. Tüm ışın değeri geçici olarak bu katsayılar göre bulunup, görüntü ekrana

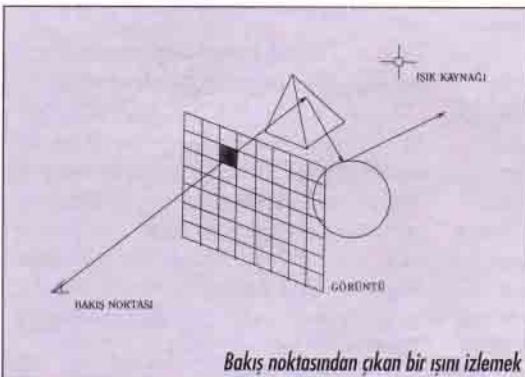
aktarılır. Sonra, seçilmiş olan bu parçanın ışın miktarı azaltılır ve yeniden aynı işlem tekrar edilir. Işık yayma değeri azalmış olan son parça artık en fazla ışık yayan parça olmayabileceği için, bu iterasyonda yepyeni bir parça seçilmiş olabilir. Böylece, tüm parçaların kullanılmış olması sağlanır. İzleyici ise ekranda, ka-ba bir görüntü ile başlayıp gittikçe duyarlılık ve netlik kazanan bir görüntü dizisi izler. İlk iterasyonlar sırasında sadece ışık kaynakları parlak iken ilerleyen adımlarda diğer yüzeyler de aydınlanmaya başlar. Kullanıcının daha önce belirlediği bir duyarlılık değerine (örneğin, iterasyonların sayısı) ulaşıncaya işlem sona erer.

Toplama ve dereceli geliştirme yöntemlerinin farklı kullanım alanları vardır. Dereceli geliştirme yöntemi animasyon gibi uygulamalarda görüntüler daha çabuk elde edilebildiği için daha uygundur. Diğer taraftan, toplama yöntemi bir ortamda ışık etkilerinin incelenmesi bakımından daha uygun bir yöntemdir.

Işın izleme

Bilgisayar grafiğinde oldukça popüler olan ışın izleme yöntemi, matematiksel olarak tanımlanmış nesnelere oluşan bir sahneyi gerçeğe uygun olarak görüntülemeye yarar. Bilgisayarda elde edilen bu tür görüntülere gerçeğe uygun en önemli yolu, sahnedeki ışık kaynakları ve nesnelere arasındaki etkileşimden kaynaklanan optik etkileri bulmaktır. Bu etkileşimden kaynaklanan optik etkilerden en önemlileri şunlardır:

- Işık kaynaklarının nesnelere tarafından engellenmesi ile ortaya çıkan gölgeler,
- Işık kaynaklarının nesne yüzeylerinin rengine etkisi,
- Cam gibi parlak yüzeylerde oluşan parlaklıklar,
- Sahnedeki bir nesnenin parlak bir nesne yüzeyindeki yansıması,
- Saydam bir nesnenin arkasındaki nesnelere görülmesi.



Işın izleme yöntemiyle elde edilen bir görüntüdeki bu optik etkiler, fotoğraf makinası ile çekilmiş gerçek bir görüntüdeki etkilerin benzerini vermeye çalışır. Işın izleme, optik etkilere ek olarak görülmeyen yüzeylerin atılması ve perspektif izdüşümü de yapmaktadır. Aslında, listelenen optik etkileri bulmak ve diğer işlemler için ayrı ayrı bilgisayar grafiği yöntemleri bulunmaktadır. Işın izleme bunların tümünü bir arada toplayan, programlaması oldukça basit bir yöntemdir. Bir ışın izleme programına verilen girdiler şunlardır:

- Matematiksel olarak tanımlanmış nesnelere. Örneğin, küre, merkezinin koordinatları ve yarıçapı; üçgen, üç köşe noktasının koordinatları; dörtgen, dört köşesinin koordinatları; kutu, altı dörtgen veya sekiz köşe noktasının koordinatları ile matematiksel tanımları yapılabilir.
- Bu nesnelere renkleri, yapıldıkları maddelerin ve yüzeylerin özellikleri (saydam, yarı saydam, parlak yüzey, mat yüzey, v. s.)
- Işık kaynaklarının koordinatları ve ışığı yayma şiddetleri.
- Bakış noktasının koordinatları ve bakış ile ilgili diğer parametreler.

Işın izleme bu girdileri kullanarak matematiksel tanımları verilmiş nesnelere ışık kaynakları arasındaki etkileşimi içeren bir görüntü üretir. Işın izleme yöntemi üretilen görüntüyü iki boyutlu bir matris olarak ele alır ve matrisin her bir elemanının renk değerini bulmaya çalışır. Bu matrisi yatay ve dikey çizgilerden oluşmuş bir ızgara olarak da düşünmek olasıdır. Bu durumda, ızgara delikleri matrisin elemanlarına karşılık gelir. Yöntem, verilen bakış noktasından başlayan ve ızgara deliklerinden (matris elemanlarından) geçen ışınların (vektör) sahneye gönderilmesine dayanır. Bu ışınların herbiri, görüntünün ilgili noktasının renk değerini bulmak için sahnedeki nesnelere ile kesilir. En yakın yüzey kesim noktasında hesaplanan renk değeri, görüntünün ilgili noktasına atanır. Yüzey

noktasının rengini bulmak için ise kullanılan tonlama modelinin öngördüğü hesaplamalar yapılır. Buna göre bakış noktasından gönderilen ilk ışın, kesişme noktasından yansır. Yansıyan ışının kestiği başka bir yüzey noktası varsa bu yeni noktanın renk değeri ilk kesişme noktasının renk değerinin hesaplanmasında kullanılır. Bu sayede bir nesnenin başka bir nesne üzerindeki yansıması bulunmuş olur. Eğer yansıyan ışın herhangi bir ışık kaynağında son bulursa ve nesne yeteri kadar yansıtıcı bir maddeden yapılmışsa, kesişme noktasından parıltıların görüldüğü varsayılır. Nesnenin saydam bir maddeden yapılmış olması halinde ise nesnenin içinden geçecek şekilde bir ışın daha üretilir ve diğer ışınlar gibi izlenir. Bu sayede saydam ve yarı-saydam nesnelere arkasındaki nesnelere de gerçeğe uygun üretilir.



Işın izleme yöntemi bu denli yararlı olmasının karşın, en önemli kusuru aşırı hesaplama gerektirmesidir. Örneğin bir görüntüyü üretme süresi, girdiye bağlı olarak saatlerce zaman alabilmektedir. Bu nedenle pratik ve kullanılabilir olması için yöntemi hızlandırmak gerekmektedir. Yapılan incelemelerde, bu zamanın büyük çoğunluğu ışınlar ve nesnelere arasındaki kesişme hesaplarında tüketilmektedir. Çünkü ortaya çıkan her ışının tüm nesnelere ile kesişme zorunluluğu vardır. Bu nedenle ışın izleme yönteminin hızlandırılması ilk kesişme noktasının hızlı bir şekilde bulunmasına bağlıdır. Geliştirilen hızlandırma teknikleri sayesinde görüntü üretme süresi oldukça azalmıştır.

Karmaşık Nesnelerin Modellenmesi

Belirli nesnelerin basit geometrik kural- lar ile elde edilmesi çok zordur. Doğada gör- düğümüz dağlar, ağaçlar ile eğilmiş, bükül- müş, ezilmiş nesnelere modellemek sorunlar yaratabilmektedir. Bu gibi nesnelere göstere- bilmek için, Bilim ve Teknik dergisinin 325.

sayısında yer alan Ka- os kuramından yola çıkarak, Fraksiyonel Geometri (Fractals) ile, defor- masyon ve değişim yöntemleri kulla- nılabilir.

Deformasyon Teknikleri Kullanılarak Düzensiz Nesnelerin Modellenmesi

Düzensiz nesnelerin modellenmesinde deformasyon yöntemleri uygulamak, üzerin-



de oldukça çok ça- lışılan konulardan bi- risidir. Böyle nesnelere, yü- zeylerindeki tüm noktalar verile- rek tanımlanabilir. Bu yol kullanı- lduğunda bu nesnelere yapılabilecek işlemler açık olmamakta ve bazı zor- luklar ortaya çıkmaktadır. Bu nesnele- rin modellenmesinde etkili bir yöntem de düzenli şekilleri oluşturduktan sonra on-

Çizgi Film Üretimine Geleceği

Bilgisayar grafiği bilim dalının, özellikle geliştirdiği Jack animasyon sistemiyle, en önde gelen bilim adamlarından birisi olan Prof. Dr. Norman Badler ile elektronik posta aracılığıyla bir söyleşi yaptık. Pennsylvania Üniversitesi Mühendislik Fakültesi Bilgisayar Grafiği Laboratuvarı'nın kurucu ve yöneticisi olan Dr. Badler'i belki bazı okurlarımız geçtiğimiz günlerde CNN'de yayınlanan bir programda izlemiş olabilirler. Söyleşimizin konusu, bilgisayar grafiği ve bilgisayar destekli animasyon - çizgi film üretiminin geleceği.

Bilim ve Teknik: Bilgisayarları neden animasyon yapmada kullanıyorsunuz?

Prof. Dr. Norman Badler: Bilgisayarlar, animatörlerin elle çok zor veya hiç yapamayacakları çizimler için kullanılırlar. Animatörler iki boyutlu şekillerin çiziminde çok iyi de olsalar, üç boyutlu şekiller ve bunların hareketleri, en deneyimli göz ve ellere bile meydan okuyabilir. Eğer üç boyutlu bir uzam resmedilecekse, bu sanal ortamda hareket eden bir kameranın sürekli ürettiği perspektif görüntüleri elle yapabilmek olanaksızdır. Tanımlanmış üç boyutlu nesnelere etrafında kamera hareket ederken veya nesnelere kendi kendilerine hareket ederlerken, deformasyona uğrarlar ve görünümünü değiştirirken, bilgisayar tüm karmaşık ve yoğun matematiksel işlemleri yapabilir. Buna benzer olarak, duyarlı ışık efektleri elle tek bir görüntü karesi içinde ustalıkla çizilebilir; ancak görüntü video veya sinemada olduğu gibi bir kare- den diğer kareye geçerken bu efektler gerçek dışı olarak değişmemelidir. Burada da bilgisayarlar uygun ve tutarlı aydınlatma efektleri için gereken boyama, tonlama ve aydınlatma hesaplarını rahatlıkla yapabilirler.

Bilim ve Teknik: Neden insan hareketi animasyonlarında bilgisayar kullanılıyor?

Badler: Üç boyutlu hareketli nesnelerin üretimi bilgisayarla daha iyi yapılabildiğinden ve insanlar ilginç ve önemli üç boyutlu varlıklar olduğundan, biz bilgisayarların hareket halindeki yapay insanları üretmelerini, çizmelerini ve benzetimlerini yapmalarını istiyoruz. Fakat ne yazık ki, insan animasyonu özellikle zor kendimiz de insan olduğumuz için, hareketlerimizi stil, kişilik, iletişim, tavır, sosyo-ekonomik yapı ve benzeri etkenlerle yorumluyoruz. Bu nedenle bilgisayarla insan hareketi benzetimi zor, çünkü sadece şekil ve konum hesabı değil, kabul edilir bir hareket kalitesi ve uygunluğu söz konusu. Örneğin, el hareketleri ge-

lişgüzel değil. Bunların şekli, zamanlama ve vurgulaması, kişinin duyguları ve konuşması ile kendine özgü biçimlerde senkronize. Gene buna benzer olarak, yüz hareketleri bilgi aktarabildiği gibi, duygu, iletişim sinyalleri, veya göz kırpmak gibi doğrudan biyolojik bir gereksinimin göstergesi olabilir.

İnsan hareketi animasyonunda birkaç değişik yaklaşım vardır. Bunların hepsinde, bilgisayar hesaplaması ile, animatörün deneyimi, yaratıcılığı ve katkısı arasında seçimler yapılmıştır. Örneğin, en iyi insan hareketi benzetimleri tamamen sentetik olanları değil, canlı bir insandan kaydedilmiş ve örneklenmiş olanlardır. Buna 'performansa dayalı' animasyon denir. Yapılması oldukça kolay olmasına rağmen, bir kişinin tüm hareketleri yapmasını zorunlu kılar. Hareketler bilgisayara kaydedilince, üç boyutlu bir insan modeline uygulanıp, animatör tarafından yepyeni bir ortamda değiştirilerek kullanılabilir. Diğer uçtaki yaklaşım ise, hareketlerin tümünü yapay olarak üretmektir. Bir insanın tehlike nedeni ile yapamayacağı hareketler, yer çekimi olmayan yerlerdeki hareketler, istenilen boyda bir kişinin bulunamaması gibi zor durumlarda, hareket benzetiminin tümü bilgisayar tarafından gerçekleştirilebilir. Buna verebileceğimiz en iyi örnek, henüz tasarım aşamasında olan iş ortamlarında insan davranışı benzetimleridir. Bir insanın gereken işlemleri (nesnelere görebilme, onlara erişebilme, uzama sığma, bir ayağı kaldırıp kullanabilme, vs.) yapılabilmek için yazılacakları bu şekilde, iş ortamı oluşturulmadan, gözlemleyebiliriz.

Bu iki uç nokta arasında, animatör ve bilgisayarın etkileşimli olarak kullanıldığı ve birbirlerini destekleyen roller üstlendiği insan hareketi benzetimi sistemleri vardır. Animatörün rahatça kullanıp yönlendirebildiği sistemlerde bilgisayarlar, duygu, kişisel tepki, iletişim tarzı gibi zor hareketlerden anndırılabilir.

Bilim ve Teknik: Bilgisayar animasyonu, yapılan filmlerin kalitesini arttıracak mı?

Badler: Bilgisayarların filmlerdeki özel efektlerin kalitesini arttırdığına dair hiçbir kuşku yok. Ancak, bilim dalımız henüz 'yapay aktör' yaratarak, gerçek aktörlerin işlerine son verecek düzeye gelmedi. Gerçek aktör bulunamadığı durumlarda, örneğin Jurassic Park'taki dinozorlarda olduğu gibi, bilgisayarın değeri belli olmakta. Yinede, gerçeğe benzer hareketler, inandırıcı mimikler, ko-

nuşma, ifade, elbise, saç ve etkileşimli dinamizm gibi konularda daha çok yol katetmemiz gerekiyor.

Bilim ve Teknik: Yakın bir gelecekte, seyrettiğimiz filmle etkileşimimiz olası mı?

Badler: Bu tür etkileşim zaten video oyunlarında var. Bu nedenle, sorunun daha derin boyutlarına bakalım: Bir filmin, etkileşimli olarak, önceden belirlenmemiş bir şekilde akışını değiştirebilir miyiz? Bunun yanıtı, sanırım, evet. Ancak, bu iş için, gerçek zamanda çevreye ve kullanıcıya tepki gösterebilecek karakterler modellememiz gerekir. Böyle bir sentez yapmak, hele gerçeğe yakın görüntü isteniyorsa, çok zor. Video oyunlarında sadece belirgin bir hikaye olduğu için, genellikle, görüntü kalitesi ile hız lehine feda edilebilir. Ancak, genellikle isteniyorsa, o zaman gerçek yaşamın tüm değişkenlerini modellememiz gerekecek.

Bilim ve Teknik: Bilgisayar animasyonlarının maliyeti nedir?

Badler: Aslında maliyetin iki bileşeni var. Birincisi bilgisayarların maliyeti (donanım ve yazılım), diğeri ise bu sistemi kullanan kişilerin maliyeti. Zaman içinde, kullanıcı maliyeti çok daha fazla olmaktadır. Donanımda düşük bir yatırım ile alınabilecek kişisel bilgisayarlar ve gene fazla maliyet yaratmayacak yazılımlar ile çok kaliteli animasyonlar yapılabilir. Fakat gösterilecek görüntüler karmaşıklıkla, karakterlerin hareketleri çeşitlendikçe, gerçek zamana yaklaşan hız istenildikçe, Silicon Graphics iş istasyonları gibi güçlü hesaplama ortamlarına gereksinim doğar. Çok işlemcili bilgisayarlara kadar büyüyecek yüksek performanslı sistemlerin maliyetleri de büyüktür. Yazılım maliyeti de kullanılan donanıma bağlı olarak artar ve bazı animasyon gereksinimleri özel olarak yazılım geliştirilmesini gerektirebilir. Bilgisayar grafiğindeki kural, kullanıcı veya animatörün erişebildiği sistemi en iyi ve en yoğun şekilde kullanabilmesidir!

Bilim ve Teknik: Kişisel bilgisayarımızda çok ileri düzeyde grafik yapabilir miyiz?

Badler: Elbette, fevkalade yetenekli yazılımlar var, ancak animasyon için bunlar iki boyutlu sınırlı. Perspektif, görünen yüzeyler, boyama, tonlama ve aydınlatma hesapları, kişisel bilgisayarların gücünü çok zorlamakta. Bunlar isteniyorsa, iş istasyonları, özel amaçlı yazılım ve donanımlar kaçınılmaz oluyor. Ama elimizde kişisel bilgisayar varsa neden grafik dünyasının sınırsız olanaklarından yararlanmayalım...

lar üzerinde deformasyon tekniklerini uygulamaktır. Bu yolla gerekli nesnelere elde edilirken ortaya çıkan zorluklar da önlenmekte ve işlemlerde açıklık sağlanmaktadır. Yazımızın bu kısmı, bazı modelleme yöntemlerine deformasyon tekniklerinin bugüne kadar kullanılan değişik yaklaşımlarının uygulanması ile ilgilidir. "Superquadric" ve "Bezier" yüzeyler, nesnelere modellemekte kullanılan değişik yöntemlerdir. Kıvrıma, bükme ve inceletme gibi düzenli deformasyonlar (regular deformations), serbest deformasyon (free-form deformations) tekniği ile birleştirilerek yeni bir deformasyon yöntemi elde edilmiştir. Bu yolla her iki deformasyon tekniği hiyerarşik ve dönüşümlü olarak uygulanabilmektedir. Böylece bu yöntemlere özgü bazı dezavantajlar yok edilmiş ve her iki yöntemin avantajlarından etkin bir şekilde yararlanılmıştır.

Fraksiyonel Geometri (Fraktaller) ile Düzensiz Nesnelere Modelleme

Bilim adamlarınca doğadaki galaksiler, bulutlar, dağlar vb. gibi pek çok karmaşık yapıya uzun süre bir model getirilememiştir. Öklid Geometrisi yöntemleri bu düzensiz yapıları tanımlamakta yetersiz kalmıştır.

Fransız matematikçi Mandelbrot, 1975'de Öklid Geometrisi'ne seçenek olarak Fraktal geometri'yi tanıtarak bu karmaşık yapıları hem bir tanım, hem de bir model kazandırmıştır. Fraktal sözcüğü, Latince fractus'dan gelen frangere (kırmak) fiilinin sıfır halinden alınmıştır. Mandelbrot'un 1977'de yayımladığı "Fractals" ve özellikle 1983'de yayımladığı "The Fractal Geometry of Nature" adlı kitaplarla fraktal kavramı bilim dünyasında iyice yerini almıştır.

Sezgisel olarak kavramaya yatkın olduğumuz Öklid Geometrisi, cisimleri yalnızca bir, iki ya da üç boyutta tanımlamıştır. Fraktal geometri bizlere sıfır boyutlu noktalar, bir boyutlu çizgiler, iki boyutlu düzlemler ve üç boyutlu hacimler arasında yer kaplayan ve boyutları kesirli sayılarla ifade edilen yapıları, fraktalleri tanıtmıştır.

Öklid Geometrisi'nde düzlemsel bir bölgeyi doldurmak için sonsuz sayıda doğru gerekir. Oysa sonsuz kıvrımlı bir doğru, sonunda bir kağıt parçasını doldurabilir. İşte bu sonsuz kıvrımlı doğru, artık bir fraktal eğridir. Görüldüğü gibi, fraktal eğri bir boyutlu değil, bir ile iki arasında, bir tamsayı ile ifade edilemeyen bir boyuta sahiptir. Gerçekte fraktal cisimlerin hepsi böyle kesirli bir boyuttadırlar; buna da *fraktal boyut* adı verilir. Benzer şekilde, bir kağıt parçasını alır, elinizde top halinde buruşturursanız, boyutu ikiye, üçe yaklaşmış olur. Buna rağmen kağıt parçası topolojik olarak hâlâ iki boyutlu bir yüzeydir. Fraktal eğriler de topolojik olarak hâlâ iki boyutlu yüzeydiler; ancak bir eğri oldukları için de bir boyutludurlar. Fraktaller Mandelbrot tarafından kısaca, "fraktal boyutu, topolojik boyuttan büyük olan kümeler" olarak tanımlanmışlardır.

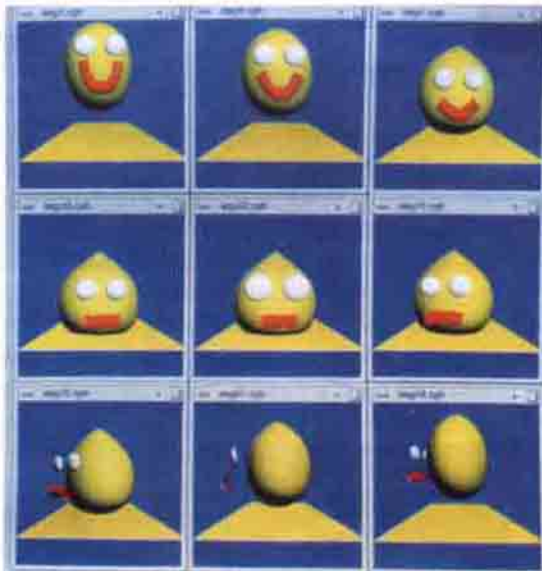
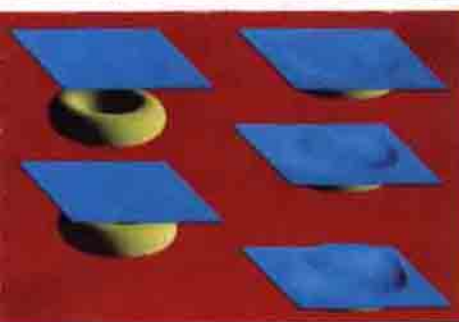
Türkiye'nin Ege kıyı şeridinin uzunluğu nedir? Bu uzunluğun kıyı şeridinin iki uç noktası arasında uzanan doğru parçasının uzunluğundan fazla olduğu hemen söylenebilir. Daha ayrıntılı bir ölçüm yapmak istendiğinde ölçü birimimizi (doğru parçasının uzunluğunu) küçültmemiz gerekecektir. Fakat birimi ne kadar küçültürsek bulduğumuz uzunluk da o kadar artacaktır. Bütün ölçüm teknikleri göstermiştir ki, tipik bir kıyı şer-



idin uzunluğu, limiti sonsuza ıraksayacak kadar büyüktür. Bu örnek fraktal eğrilerle ilgili önemli bir özelliği göstermektedir. Fraktal eğriler her noktada sonsuz ayrıntıya sahiptir ve bu yüzden gerçek uzunluklarının ne olduğunu belirlemek olanaksızdır.

Fraktallerin diğer bir önemli özelliği de kendine benzer yapıda olmalarıdır. Ne denli küçük ölçekte olursa olsun cismin her parçası ana şeklin tamamını oluşturacak şekilde büyütülebilir. Kendine benzerlik, doğadaki cisimleri modellemede de önemli bir özelliktir. Örneğin, yine bir kıyı şeridini ve bundan çok yüksekte uçan bir uçağı düşünelim. Kıyı şeridi bu yükseklikten ne kadar girintili çıkıntılı görünüyorsa, yaklaştıkça yine aynı derecede girintili çıkıntılı görünecektir. Fakat bu tam bir kendini tekrar değildir. Gerçekte doğal yapılar istatistiksel olarak kendine benzerdir fakat ayrıntıda farklıdır. Buna *istatistiksel kendine benzerlik* adı verilir. Bu özellikteki fraktallara *gelişigüzel fraktaller*, kesin ve tekrar edilebilir yapıdaki diğer gruba da *gerekirli fraktaller* adı verilir.

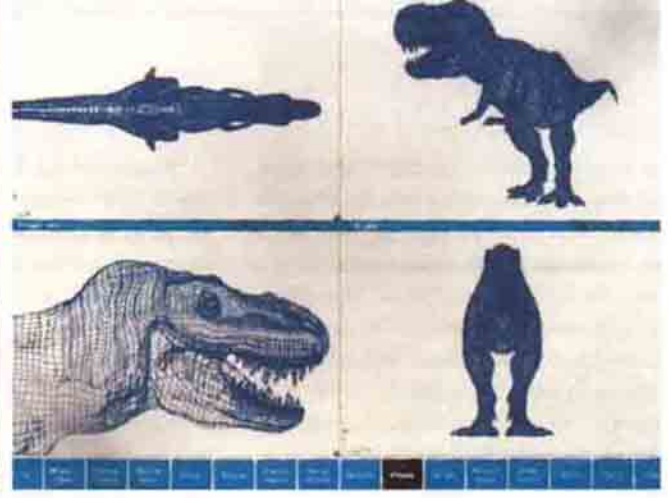
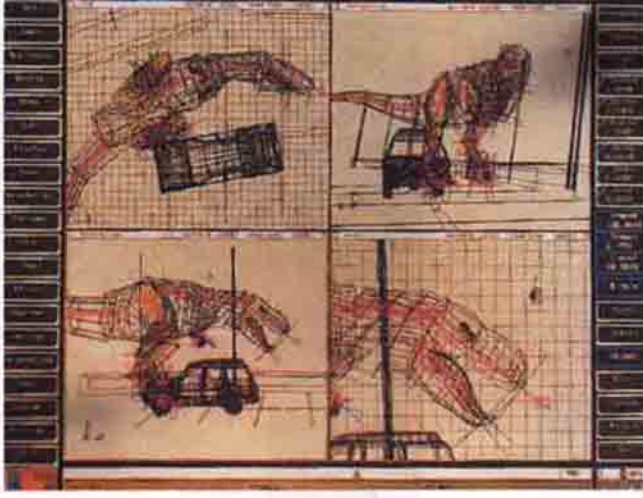
Fraktal cismin yinelenen dokusu, onun fraktal, başka bir deyişle benzerlik boyutunu



↑ İki boyutlu çizgi film çalışması ↑ Nesneye yönelik animasyon sistemi

← Deformasyon yöntemi (Bilkent) →





Jurassic Park filminin yapım ayrıntıları
(© Industrial Light & Magic, Universal Studios)

tanımlar. Kendine benzerlik sezgisel boyut kavramımızla çok yakından ilgilidir.

Fraktal geometrinin temel öğeleri Öklid Geometrisi'nin çizgileri, daireleri gibi kolayca görülemezler, onları tanımlayan basit cebirsel formüller yoktur. Genellikle - bilgisayarca uygulanması çok kolay olan - yinelemeli işlemler veya algoritmalarla tanımlanırlar. Sistem aynı işlemi tekrar tekrar yaparken, bir işlemin çıktısını bir sonraki girdi olarak sunan basit bir geri besleme halindedir.

Matematikçiler bugün fraktal olarak bildiğimiz yapıları 19. yüzyıldan beri biliyorlardı. Ancak bu yapıların olağanüstü karmaşıklıkları matematikçilerin cesaretini kırmıştı. Modern bilgisayar grafiği olmadan bu ince fikirlerle ulaşmak olanaksızdı. Bilgisayar grafiği, matematikçi ve fizikçiler gibi şekillerle düşünenlerin yaratıcı potansiyellerini açığa çıkarmış, sürekli gelişerek, algılama yeteneğimizi, bilim dünyasında şimdiye kadar hiçbir aracın başaramadığı kadar zenginleştirmiştir. Önceki kuşakların bilim adamları denklemlerini ya iyice basitleştirmek ya da

tümünü bırakmak zorunda kalırken, bizler bunların tüm içeriğini bilgisayar ekranında görebilmekteyiz. Bilgisayar grafiği, fraktalları tanımlayan matematiksel işlem kümelerini geometrik şekillere çevirerek, gelişimlerinde ve hızla artan popüleritelerinde büyük rol almıştır. Aynı zamanda fraktal geometri de bugün, doğal yapıların ve fantastik şekillerin modellenerek görüntülenmesi ve hareketlendirilmesinde önemli bir yer tutmaktadır.

Canlandırma ve Animasyon

Bilgisayar grafiğinin en çok kullanıldığı alanlardan biri, hareket tanımlarını da içeren canlandırma ve animasyon (çizgi film) uygulamalarıdır. Animasyon bir nesneye hareket vererek gerçeklik ya da karakter kazandırma. Bilgisayar kullanılarak yapılan animasyonlarda temel olan, modellenmiş cisimlere hareket dizileri uygulayarak gerçeğe yakın görüntüler elde etmektir. Yukarıda anlatılan modelleme ve aydınlatma - boyama

yöntemlerinin 4. boyuta aktarılması ile hareket elde edilir. Hareket temelde iki değişik şekilde olur: Birincisi, nesnelerin zaman içinde konum değiştirmesi ki, bu eklemli nesnelere karmaşık bir işlemdir; ikincisi ise nesnelerin eğilip, bükülüp ve hatta kırılıp şekil değiştirmesi şeklindedir.

Eklemli Nesnelerin Animasyonu

Animasyon, oluşturulan bir senaryo üzerine modeller yaparak gerçekleştirilir. Çocukların çok severek izledikleri çizgi filmler en çok rastladığımız animasyon örnekleridir. Animasyonun senaryosu izlenerek görüntüler yaratılır. İlk yapılan çizgi filmlerde, senaryoya göre belli anahtar sahneler oluşturulmuş, ara görüntüler daha sonra hazırlanıp, hepsi arka arkaya gösterilerek iki boyutlu animasyonlar hazırlanmıştır. Daha sonra bilgisayar kullanılarak ara görüntülerin üretimi hızlandırılmıştır. Günümüzde animasyonların çoğu tamamen bilgisayarla üretilebilmektedir. Bilgisayarlara, çeşitli şekillerde elde edilen ve nesnelerin yapısını tanımlayan sayısal veriler ile hareket basamakları parametrik şekilde yüklenerek senaryo gerçekleştirilmekte ve gerçeğe yakın üç boyutlu görüntüler ortaya çıkarılmaktadır.

Bilgisayar animasyonunun en önemli özelliklerinden biri, gerçek hayatta görüntüsünü alamayacağımız nesnelere ya da gerçekleştirildiğinde çok tehlikeli sonuçlar doğurabileceği düşünülen hareket dizileri yaratmaktır. Örneğin, günümüzde dinazorları görüntülemek olanaksızdır. Ayrıca bir filmdeki uçak kazası sahnesinde gerçek bir uçak kullanmak pek akıllıca bir iş değildir. Pek çok otomobil firması, üretim aşamasındaki otomobillerin rahatlığını ve bazı koşullardaki dayanıklılığını ölçmek için, bilgisayarda modellerini oluşturup belirlenen hareketleri ve kaza sırasında olabilecek durumları modeller üzerinde uygulayarak ürün hakkında bir fikir edinmektedirler.

Animasyon, son yıllarda birçok filmde etkili bir şekilde kullanılmıştır. Başta Jurassic Park olmak üzere Total Recall, Robocop, Oyuncaklar, Özgür Willy ve Terminator 2 örnek olarak verilebilir.



Total Recall filmindeki yapay iskeletler (© Metrolight Studios)



Terminator 2

Jurassic Park, maket ve bilgisayarda modellenmiş dinazorları başarılı bir şekilde birlikte kullanması açısından, filmlerde bilgisayar kullanımını daha farklı bir düzeye getirmiştir. Animasyon ekibi filmdeki dinazor görüntülerinin bilgisayarda gerçekçi şekilde oluşturabilmek için önce bir dinazor iskeletini üç boyutlu olarak görüntülemiştir. Bu iskeletin üzerine kas ve en üstte de deri tanımlanmıştır. Dinazorların hareketlerini gerçekçi bir biçimde yansıtabilmek için - animasyonun en önemli amaçlarından birisi - kasların ve derinin davranışları incelenerek en iyiye ulaşılmaya çalışılmıştır. Filmdeki bazı sahneleri örnek olarak gösterecek olursak, yaratıkların bina içinde insan kovalamaları, mutfak sahnesinde koşma, zıplama ve ayaklarının kayması gibi zor hareketler oldukça başarılı bir şekilde gerçekleştirilmiştir.

Total Recall filminde ise insan iskeletleri modellendikten sonra gerçek aktörlerin hareketleri incelenerek, yürüme, koşma dövüşme gibi hareket dizileri iskelet modellerine uygulanmıştır. Arnold Schwarzeneg-



Abyss filminden bir görüntü

er'in başrolünü oynadığı bu filmdeki X-ışını panosunun arkasındaki iskeletlerin hareketi oldukça gerçekçi bir görüntüyle verilmiştir.

Robocop 2 filminde ise bilgisayarla oluşturulmuş bir yüz kullanılmıştır. Gerçek bir insanın yüzü kullanılarak alınan veriler üzerinde parametrik fonksiyonlar kullanılarak, yüzdeki mimikler gerçekleştirilmeye çalışılmıştır.

Özgür Willy filminin bir bölümünde, balinanın açık denize ulaşması için yüksek ve kayalık bir dalgakıran üzerinden denize atlaması gerekmektedir. Ancak, bir balina için o yükseklikteki bir kayanın üzerinden atlamak oldukça zordur. Ayrıca açık denize ulaşan eveil bir balina geriye döndürülemeyebilir. Bu yüzden bilgisayarla bir balina modeli yaratılıp bu model kullanılarak bir animasyon hazırlanmıştır. Balinanın sudan çıkma

sahnesinin görüntülenmesi için önce bir maket sudan çıkarılmış, bunun arkasına animasyon eklenmiştir. Maket balina yeterince esnek ve oluşturulan görüntü çok gerçekçi olmayacağından sahnenin tümünde maket bir balina kullanılması uygun bulunmamıştır. Böylece gerçekleştirilmesinde böylece zorluklar olan bu eylemler, bilgisayar aracılığıyla gerçeğe oldukça yakın bir biçimde izleyiciye sunulmuştur.

Fizîğe Dayalı Animasyon

Bilgisayar grafiğinde, üç boyutlu tasarım ve modelleme için kullanılan sayısal yöntemler tasarımcıların nesnelerin şekillerini tanımlamaları için etkileşimli bir ortam sağlar. Nesnelerin cebirsel ve geometrik olarak tanımlanan basit şekillerinin birleştirilmesi ile modellenmesi, bilgisayar destekli tasarım ve üretim gibi uygulamalar için en önemli gereksinimdir.

Günümüzde kullanılan modelleme yöntemlerinin çoğu yukarıda sözü geçen gereksinimlere paralel olarak geliştirilen kinematik yöntemlerdir. Bu yöntemler kullanılarak gerçeğe uygun hareket ve animasyon elde etmek oldukça zor olduğundan, ancak bu konuda uzmanlaşmış kişiler istenen hedeflere ulaşabilir. Bunun nedeni, kinematik modelleme yöntemlerinin edilgen olması ve bu yöntemlerle elde edilen modellerin birbirleriyle ve etrafındaki diğer etkilerle etkileşim içinde olmamasıdır. Bir animasyonun gerçeğe uygun olması için nesnelerin önceden tanımlanmış yolları aynen izleyebilmesi, aynı zamanda gerçek fiziksel nesnelere gibi ilginç hareketler yapabilmesi ve birbirleri ile etkileşim içinde olması gereklidir. Bu şekilde etken modellerin üretilmesi ve animasyonu için fizîğe dayalı modelleme yöntemleri kullanılmalıdır. Bu yöntemler, edilgen modellere fiziksel özellikler ekler. Bu özellikler kuvvet, döndürme kuvveti, hız, ivme, kinetik ve potansiyel enerji, sıcaklık ve buna benzer diğer özelliklerdir. Fiziksel benzetim, bu özellikler kullanılarak animasyon üretmek için kullanılır.



RoboCop 2 filmindeki yapay insan yüzü (© deGraf/Wahrman)



Mask filminden iki kare



Gerçeğe uygun animasyon elde etmede diğer önemli bir konu da katı olmayan (deforme olabilen) nesnelerin hareketlerinin modellenmesidir. Deforme olabilen nesnelere sonlu sayıda noktalar kullanarak yaklaşık bir şekilde modelleyebiliriz. Nesnelerin hareketleri de bu noktaların birbirlerine göre hareketleriyle tanımlanabilir. Noktaların birbirlerine göre hareketlerinin değişik şekillerde tanımlanmasıyla nesnelerin elastik ve plastik özelliklerini modelleyebiliriz. Örneğin elastik bir yüzey elde etmek için yüzey üzerindeki noktaları birbirlerine yaylarla bağlayabiliriz. Deforme edilebilen nesnelere modellemeye ve hareket ettirmeye yarayan yöntemler temelinde elastik ve plastiklerin kuramına dayanmaktadır. Belirtilen yöntemlerle animasyon elde etmek için modellerin diferansiyel denklem sistemlerini oluşturmak, bu denklemleri çözmek ve elde edilen verilerden animasyon karelerini üretmek gereklidir.

Teşekkür

Bu yazının hazırlanmasında emeği geçen Bilkent Üniversitesi öğretim elemanları ve lisans üstü öğrencileri Dr. Veyis İşler, Dr. Uğur Güdükbay, Tahsin Kurç, Gamze Tunali ve Sebnem Levlak'a teşekkür ederim. Ayrıca, öğretimden görüşmelerden yararlandığım Ersin Ünal, Bülge Erkan, Uğur Akdeniz ve Tuğba Çapın'a da teşekkür ederim. Üniversitemizde geliştirilmiş olan yazılımların sonuçlarını da örneklemiş olduğumuz bu yazıda sunulan yazılım alanlarında TÜBİTAK'ın bize çok büyük destekler sağladığına da önemle belirtmek isterim.

Kaynaklar

- Hearn, D. ve Baker, M. P., Computer Graphics, ikinci baskı, Prentice Hall, New Jersey, 1994.
- Foley, J.D. ve van Dam, A., Computer Graphics: Principles and Practice, Addison-Wesley, 1990.
- Watt, A., Fundamentals of Three-Dimensional Computer Graphics, Addison-Wesley, 1989.
- Burger, P. ve Gillies, D., Interactive Computer Graphics, Addison-Wesley, 1989.
- Rogers, D.F., Procedural Elements of Computer Graphics, McGraw-Hill, 1985.
- Cohen M. F., Chien S. E., Wallace J. R., & Greenberg D. P., A Progressive Refinement Approach to Fast Radiosity Image Generation, ACM Computer Graphics, Vol. 22, No. 4, pp. 75-84, (1988).
- Cohen M. F., & Greenberg D. P., The Hemi-Cube: A Radiosity Solution for Complex Environments, SIGGRAPH 88, San Francisco, Vol. 19, No. 3, pp. 254-263, (1985).
- Goral C. M., Torrance K. E., Greenberg D. P. & Battaglia B., Modeling the Interaction of Light between Diffuse Surfaces, Computer Graphics, 18 (3), pp. 213-222, (1984).
- Watt A. & Watt M., Advanced Animation and Rendering Techniques: Theory and Practice, Addison-Wesley, (1992).