

## Üç Boyutlu Bilgisayar Grafikleri

### 1. Üç Boyutlu Nesne Tanımlama Yöntemleri

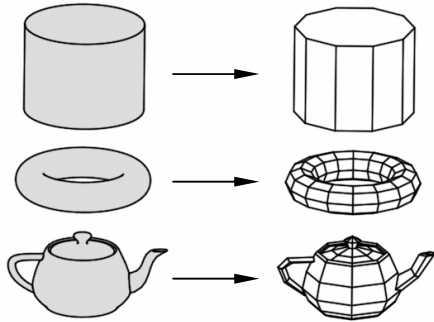
Bilgisayar grafiklerinde üç boyutlu nesnelere tanımlamak için birçok yöntem geliştirilmiştir. Hangi tanımlama yönteminin avantajlı olduğu üç boyutlu uygulamanın amaç ve gereksinimleri, modellenme yöntemleri ve kullanılan modelleme programına göre farklılık gösterir.

Nesne tanımlama yöntemlerini iki sınıfta incelemek mümkündür: Bu yöntemlerin ilkinde nesnenin iç kısmını dış ortamdaki ayıran sınır (nesnenin yüzeyi) tanımlanır. Bu yöntemin belirgin örnekleri arasında poligon nesnelere ve parametrik yüzeyler bulunur. İkinci yöntemde ise nesnenin uzayda kapladığı hacim göz önünde bulundurulur ve genellikle bu yöntem nesnenin sınır bilgisinin uygulama için yetersiz kaldığı durumlarda kullanılır.

#### 1.1 Poligon Nesnelere

*Poligon nesnelere* en yaygın kullanılan nesne tanımlama yöntemidir. Poligon tanımlama yöntemlerinde, birbirlerine komşu birçok üç boyutlu poligon nesnenin yüzeyini oluşturur.

Poligon nesnelere diğer nesne tanımlama yöntemlerine göre çok daha yaygın olarak kullanılmasının başlıca nedenleri, poligon tanımlama yöntemlerinin hızlı, kolay ve esnek olmasıdır. Ancak ayrıntılı bir nesneyi tanımlamak için yüksek sayıda poligon kullanılması gerekliliği, poligon nesnelere önemli bir dezavantajdır. Örneğin, bir Boeing-777 uçağının üç boyutlu modeli yaklaşık olarak 500.000.000 adet poligon içerebilir [1]. Bu kadar fazla sayıda poligon ile çalışmak fazla bellek, yüksek işlemci gücü ve hızlı veri yolları gerektirecektir [2].



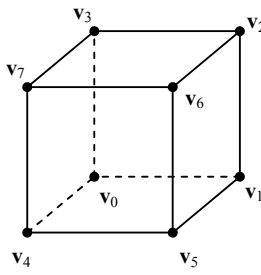
Şekil 1. Poligon nesnelere

Şekil 1'de bazı nesnelere ve poligonlarla tanımlanmış halleri gösterilmektedir. Kullanılan poligon sayısı arttırıldıkça, tanımlamadaki hatalar (gerçek nesne ile poligonlarla tanımlanmış hali arasındaki fark) o kadar azalacaktır. Bu tür hataları görsel olarak azaltmak için Gouraud [3] ve Phong [4] tonlandırma yöntemleri geliştirilmiştir.

Poligon nesnelere birbirlerine komşu birçok poligon oluşturulduğu için, herhangi bir köşe koordinatı bilgisi birçok poligon tarafından paylaşılır. Eğer poligonlar birbirinden bağımsız olarak kabul edilirse, paylaşılan köşelerin koordinatları veri yapısında gereği olmadığı halde defalarca saklanır ve gereksiz bellek kullanımı sorunuyla karşılaşılır. Bu tür bir soruna neden olmamak için kullanılacak basit yöntemlerden birisi, poligon nesneyi saklamak için kullanılacak veri yapısını iki ayrı listeden oluşturmaktır:

- **Köşe listesi:** Köşe listesinde, nesneyi oluşturan poligon köşelerinin üç boyutlu koordinatları saklanır.
- **Poligon listesi:** Poligon listesinde, poligonun her köşesi için nokta listesine bir indis (ya da işaretçi) saklanır.

Bir küp nesnesini saklamak için oluşturulan veri yapısı Şekil 2'de gösterilmiştir.

	Köşe Listesi	Poligon Listesi
	$v_0(-1, -1, -1)$	[0, 3, 2, 1]
	$v_1(1, -1, -1)$	[2, 3, 7, 6]
	$v_2(1, 1, -1)$	[0, 4, 7, 3]
	$v_3(-1, 1, -1)$	[1, 2, 6, 5]
	$v_4(-1, -1, 1)$	[4, 5, 6, 7]
	$v_5(1, -1, 1)$	[0, 1, 5, 4]
	$v_6(1, 1, 1)$	
	$v_7(-1, 1, 1)$	

Şekil 2. Bir küp nesnesini poligonlar (dörtgenler) kullanarak tanımlamak için kullanılacak veri yapısı

#### 1.2 Eğriler ve Yüzeyler

Ayrıntılı bir nesneyi tanımlamak için yüksek sayıda poligon kullanılması gerekliliği, poligon nesnelere en önemli dezavantajdır. Örneğin, basit kahve kupasının poligon gösterimi binlerce poligon içerebilir ve kupa üzerinde ufak bir değişiklik yapmak

## Nesne Tanımlama Yöntemleri

için bile birçok koordinatın değiştirilmesi gereklidir. Dolayısıyla, ayrıntılı nesnelerin gösterimi için hem daha az bellek ihtiyacı olan hem de modelleme yapan kişiye kolaylıklar sağlayacak daha esnek yapılara ihtiyaç duyulur.

### 1.2.1 Kapalı ve Parametrik Fonksiyonlar

Eğri ve yüzeyleri tanımlamak için kapalı ya da parametrik fonksiyonlar kullanılabilir. Kapalı fonksiyonlara örnek vermek gerekirse, düzlemin kapalı formda gösterimi

$$f(x, y, z) = ax + by + cz + d = 0$$

olarak belirtilirken, küre yüzeyi ise

$$f(x, y, z) = x^2 + y^2 + z^2 - r^2 = 0$$

olarak belirtilir. Ancak kapalı fonksiyonlar kullanılarak tanımlanan yüzeylerin (ya da eğrilerin) üzerinde bulunan noktaların koordinatlarının hesaplanması zor ve uzun süren işlemler gerektirir. Bazı durumlarda da çözüm imkansız ya da kullanılmayacak derecede karışık olur. Bundan dolayı, eğri ve yüzeyleri tanımlamak için parametrik fonksiyonlar çok daha kullanışlıdır ve daha yaygın olarak kullanılır.

Üç boyutlu eğrilerin parametrik gösteriminde, eğri üzerinde bulunan noktaların  $x$ ,  $y$  ve  $z$  koordinatları, bir diğer  $u$  değişkenine bağlı fonksiyonlar olarak tanımlanır:

$$\mathbf{p}(u) = (x(u), y(u), z(u))$$

Bu gösterimde  $u$  değişkenine parametre,  $x(u)$ ,  $y(u)$  ve  $z(u)$  denklemlerine de *eğrinin parametrik denklemleri* adı verilir. Parametrik gösterim, eğri üzerindeki her noktanın  $u$  parametresi cinsinden hesaplanmasına olanak sağlar.

Yüzeylerin parametrik gösterimi ise iki parametreye ( $u$  ve  $v$ ) bağlı olarak belirtilir:

$$\mathbf{p}(u, v) = (x(u, v), y(u, v), z(u, v))$$

Parametrik eğri ve yüzeylerin tanımlanması sürecinde, parametrik fonksiyon olarak çoğunlukla polinomların kullanılması tercih edilir. Her ne kadar polinomlar gibi basit fonksiyonlar kullanılsa da, polinomların katsayılarını değiştirerek istenilen eğrinin ya da yüzeyin oluşturulması oldukça zordur. Bunun yerine, tasarımcının belirleyeceği kontrol noktalarının koordinatlarına göre parametrik fonksiyonun

katsayılarını hesaplayacak yöntemler geliştirilmiştir. Bu yöntemler arasında en yaygın kullanım alanı bulan Beziér eğrileridir.

### 1.2.2 Bézier Eğrileri

Bu yöntem, bir Fransız mühendis olan Pierre Bézier tarafından otomobil tasarımında kullanmak amacıyla geliştirilmiştir<sup>1</sup>. Bézier eğrilerinin sahip olduğu özellikler, onları eğri ve yüzey tasarımında bir hayli kullanışlı ve uygun hale getirmektedir. Ayrıca, Bézier eğrilerinin programlanması da oldukça kolaydır. Bu tür özelliklerinden dolayı, Bézier eğrileri bilgisayar grafiklerinde yaygın olarak kullanılır.

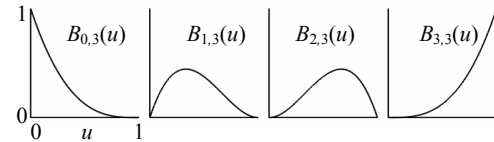
Pierre Bézier,  $n+1$  adet  $\mathbf{p}_i$  kontrol noktası tarafından kontrol edilen  $\mathbf{p}(u)$  eğrisini aşağıdaki gibi tanımlamıştır:

$$\mathbf{p}(u) = \sum_{i=0}^n \mathbf{p}_i B_{i,n}(u)$$

Denklemden bulunan  $B_{i,n}(u)$  fonksiyonu Bernstein polinomları olarak adlandırılır.

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}$$

Bernstein polinomları Bézier eğrilerinin temelini oluşturur. Şekil 3'de dört kontrol noktasına sahip Bézier eğrisinin dört Bernstein polinomu gösterilmektedir. Bernstein polinomları,  $u$  parametresinin değişik değerleri için, kontrol noktalarının Bézier eğrisi üzerindeki etkilerini belirtir. İlk kontrol noktasının ( $\mathbf{p}_0$ ) Bézier eğrisine olan etkisi  $u=0$  olduğu zaman en üst düzeydedir. Bunun yanında, diğer kontrol noktalarının ( $\mathbf{p}_1$ ,  $\mathbf{p}_2$  ve  $\mathbf{p}_3$ )  $u=0$  olduğu zaman eğri üzerinde bir etkisi yoktur çünkü o kontrol noktalarına ait Bernstein polinomlarının ( $B_{1,3}$ ,  $B_{2,3}$  ve  $B_{3,3}$ ) değeri sıfırdır.

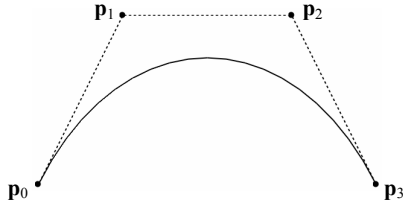


**Şekil 3.** 4 kontrol noktasına sahip ( $n=3$ ) Beziér eğrisine ait Bernstein polinomlarının grafikleri

<sup>1</sup> Bézier eğrileri, Paul de Casteljau ve Pierre Bézier tarafından birbirlerinden bağımsız olarak geliştirilmiştir. Paul de Casteljau, geliştirdiği yöntemle ilgili teknik raporunu [5] Pierre Bézier'den önce yazmış olmasına rağmen, Pierre Bézier konuyla ilgili çalışmalarını daha önce halka açık hale getirdiği için bu yöntem Bézier eğrileri olarak adlandırılmıştır [6].

Tüm kontrol noktalarının az veya çok eğri üzerinde bir etkisi vardır. Çünkü Bernstein polinomlarının değeri sadece  $u=0$  ve  $u=1$  dışında sıfırdan farklıdır. Kontrol noktalarının sayısı ne olursa olsun bu durum değişmez. Kontrolün geniş çaplı olup, kontrol noktalarının tüm eğri üzerinde etkili olması Bézier eğrilerinin dezavantajlarından birisi olarak kabul edilebilir.

Ayrıntılı Bézier eğrilerini tanımlamak için birçok kontrol noktası kullanmak mümkündür. Fakat Bézier eğrilerinde kontrolün geniş çaplı olması ve polinomun derecesi arttıkça matematiksel hesaplamalarının karmaşıklaşması yüzünden kontrol noktalarının sayısının fazla olması tercih edilmez. Bunun yerine dört kontrol noktasına sahip Bézier eğrilerinin arka arkaya birleştirilmesiyle daha ayrıntılı eğriler oluşturulur. Bu nedenden dolayı, kübik Bézier eğrileri (4 kontrol noktasına sahip Bézier eğrileri) diğer dereceden Bézier eğrilerine göre daha yaygın olarak kullanılır.



Şekil 4. Bir kübik Bézier eğrisi ve onu tanımlamak için kullanılan dört kontrol noktası

Aşağıda kübik Bézier eğrilerine ait Bernstein polinomlarının denklemi gösterilmektedir:

$$B_{0,3} = (1-u)^3$$

$$B_{1,3} = 3u(1-u)^2$$

$$B_{2,3} = 3u^2(1-u)$$

$$B_{3,3} = u^3$$

Buna göre, kübik Bézier eğrisinin denklemi aşağıdaki gibi olacaktır:

$$P(u) = p_0(1-u)^3 + p_1 3u(1-u)^2 + p_2 3u^2(1-u) + p_3 u^3$$

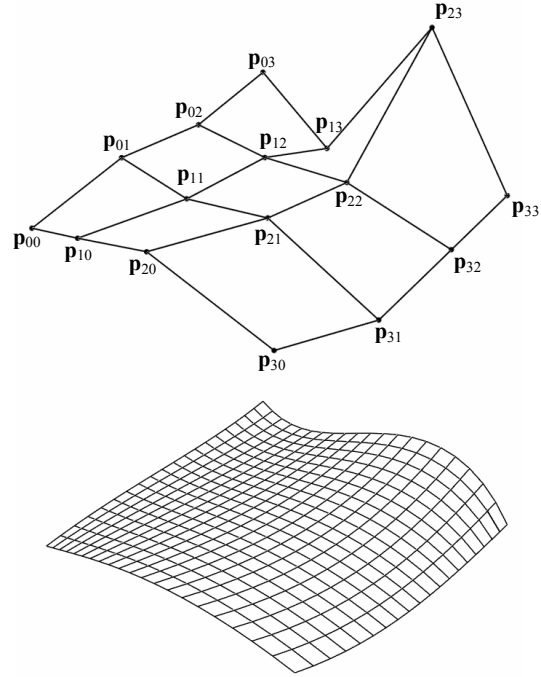
Dört kontrol noktasına sahip Bézier eğrilerinin gösterimi üçüncü dereceden (küçük) bir polinom olmaktadır. Polinomun derecesi her zaman kontrol noktalarının sayısından bir eksiktir.

### 1.2.3 Bézier Yüzeyler

Üç boyutlu Bézier yüzeylerinin (diğer adıyla Bézier yamalarının) tanımlanması için Bézier eğrilerinin gösteriminin genelleştirilmesi mümkündür. Matematiksel olarak, üç boyutlu yüzeyler iki eğrinin Kartezyen çarpımından elde edilebilir. Dolayısıyla,  $(m+1)(n+1)$  kontrol noktası tarafından belirtilen Bézier yüzeyinin gösterimi şöyle olacaktır:

$$p(u, v) = \sum_{i=0}^n \sum_{j=0}^m p_{i,j} B_{i,n}(u) B_{j,m}(v)$$

Şekil 5'de 16 kontrol noktası ile tanımlanmış bir Bézier yüzeyi gösterilmektedir.



Şekil 5. 16 kontrol noktasıyla tanımlanmış üç boyutlu Bézier yüzeyi

Bézier eğrileri dışında yaygın olarak kullanılan diğer yöntemler arasında Hermite, B-Spline, Cardinal, Kochanek-Bartels ve Catmull-Rom eğrileri [7] sayılabilir. Ayrıca Loop [8], Butterfly [9], Catmull-Clark [10], Doo-Sabin [11] ve  $\sqrt{3}$  [12] gibi yaygın olarak kullanılan yüzey bölümlendirme yöntemleri de vardır.

## 2. Görünür Yüzey Tespit Algoritmaları (Visible Surface Detection / Hidden Surface Removal Algorithms)

Üç boyutlu grafik uygulamalarında, sahnenin ve sahnedeki nesnelerin hangi kısımlarının görüldüğü, gerçekleştirilmesi gereken önemli saptamalardan birisidir. Nesnelerin hangi kısımlarının görüldüğünün (dolayısıyla ekranda görüntülenmesi gerektiğinin) ve hangi kısımlarının saklı kaldığının (dolayısıyla elenmesi gerektiğinin) saptanması amacıyla geliştirilmiş birçok yöntem olup, bu yöntemlere *görünür yüzey tespit algoritmaları* adı verilir.

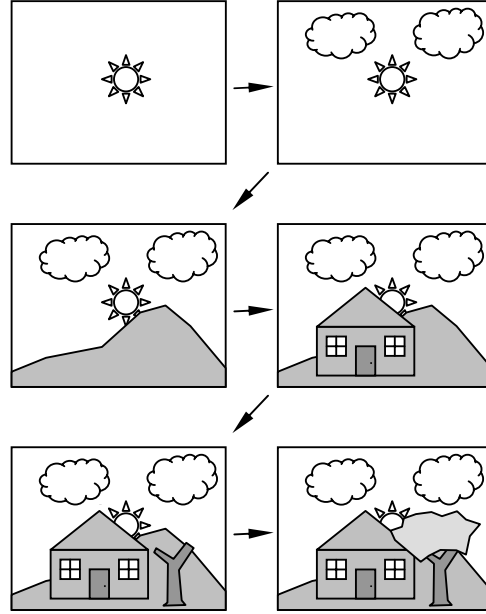
Şu ana kadar geliştirilmiş birçok yöntem olmasına rağmen görünür yüzey tespit problemine tek bir mükemmel çözüm yoktur. Tüm görünür yüzey tespit algoritmalarının birbirlerine göre birçok avantaj ve dezavantajları vardır: hız, bellek ihtiyacı, yöntemin geliştirme zorlukları veya sadece belirli nesnelere üzerine uygulanabilirlik gibi. Bunun yanında gerçek zamanlı görüntüler elde etmek amacıyla kullanılan yöntemler ile yüksek derecede gerçekçi görüntüler elde etmek amacıyla kullanılan yöntemler de farklılıklar gösterecektir.

Bu konuda yapılan en önemli çalışmalardan birisi Ivan E. Sutherland, Robert F. Sproull ve Robert A. Schumacker tarafından 1974 yılında tamamlanmış ve *A Characterization of Ten Hidden-Surface Removal Algorithms* adı altında yayımlanmıştır [13]. Bu makale yayınlandığı zamanın en önemli 10 görünür yüzey tespit algoritmasını detaylı olarak incelemiş ve karşılaştırmıştır.

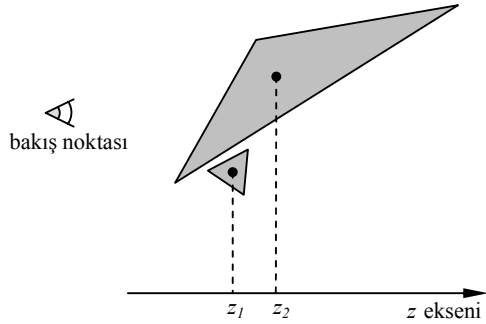
### 2.1 Ressam Algoritması (Painter's Algorithm)

Ressam Algoritması [14], en basit görünür yüzey tespit algoritmalarından birisidir. Bu yöntemde ekranda görüntülenecek yüzeyler derinliklerine göre arkadan öne doğru sıralanır ve bu sıralamaya göre ekrana basılır. Bu yöntem bir bakıma ressamın resim yapmasına benzer. Nasıl bir ressam sıra ile önce en arkadaki dağları, ovaları, sonra gerideki ağaçları, evleri, en sonra da en önde oynayan çocukları boyarsa, bu yöntemde de en arkadaki yüzeyden başlayarak yüzeyler sırayla çizilir (Şekil 6).

Ancak, bu algoritmanın basitliği yanında bazı önemli dezavantajları da vardır. Şekil 7'de göze daha yakın üçgen olan büyük üçgenin önde olması gerekirken, üçgenler, orta noktaların derinliklerine göre sıralandığı zaman küçük üçgenin daha önde olduğu sonucuna varılır ve bu da yanlış sonuçtur.

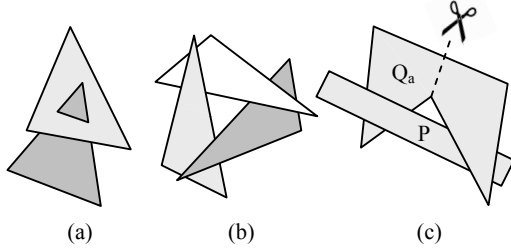


Şekil 6. Ressam Algoritması: yüzeyler arkadan öne doğru sıralanır ve bu sıraya göre ekranda görüntülenir.



Şekil 7. Yüzeyleri orta noktalarına göre sıralamak her zaman düzgün sıralamayı vermez.

Bunun yanında, birbiriyle kesişen yüzeyler (Şekil 8.a) ve sırayla birbirlerinin üzerine örten yüzeyler (Şekil 8.b) için ressam algoritması düzgün sonuçlar vermez. Çünkü ressam algoritmasına göre yüzeylerin birbirlerine göre ya bütünüyle önünde, ya da bütünüyle arkasında olduğu kabul edilir. Yapılacak bazı testler ve yüzeylerin alt yüzeylere bölünmesiyle hataların önüne geçilmesi mümkündür. Örneğin, Şekil 8.c'de gösterilen P ve Q yüzeyleri hangi sırayla gösterilirse gösterilsin yanlış sonuç elde edilecektir. Ancak Q yüzeyi Q<sub>a</sub> ve Q<sub>b</sub> olmak üzere iki parçaya bölünürse Q<sub>a</sub>, P, Q<sub>b</sub> görüntülenme sırası düzgün sonucu sağlar.



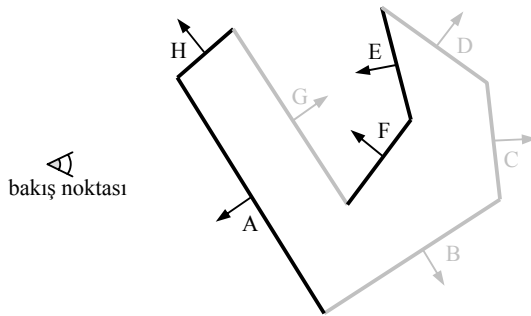
**Şekil 8.** Birbirleriyle kesişen ya da sırayla birbirlerinin üzerine örten yüzeyler için ressam algoritması düzgün sonuçlar vermez.

### 2.2 Arka Yüz Tespiti (Back-Face Detection)

*Arka yüz tespiti*, poligon tabanlı grafik uygulamalarında görüntülenmesi gereken poligon sayısını yaklaşık olarak yarıya indiren basit ve hızlı bir yöntemdir. Ancak arka yüz tespitinin düzgün sonuçlar verebilmesi için bazı şartların sağlanması gereklidir.

1. Nesneler kapalı olmalıdır. Bir başka deyişle, nesneler içi görülebilen boşluklar içermemelidir.
2. Nesneler şeffaf ya da yarı geçirgen olmamalıdır.
3. Nesneyi oluşturan yüzeylere ait normal vektörleri dışarıya doğru tanımlanmış olmalıdır.

Yukarıda bulunan koşulların sağlanması durumunda, normal vektörleri bakış noktasına yönelmemiş yüzeyler diğer yüzeyler tarafından bütünüyle örtülür (Şekil 9). Bu nedenle, *arka yüz* adı verilen ve görünmeyecek olan bu yüzeyler gözardı edilebilir.



**Şekil 9.** Gri tonlarda gösterilen arka yüzler (B, C, D ve G), diğer yüzeyler tarafından bütünüyle örtüldüğünden dolayı gözardı edilir. Bu nesnenin

ekranda düzgün görüntülenebilmesi için yalnızca ön yüzlerin (A, E, F, H) dikkate alınması yeterlidir.

Arka yüz tespiti tek başına kullanıldığı zaman görünür yüzey tespit yöntemi olarak yeterli değildir. Bu nedenle başka bir görünür yüzey yöntemi ile desteklenmelidir. Ancak basitliği ve hızı nedeniyle, diğer görünür yüzey tespit yöntemlerinin uygulanmasından önce bir ön algoritma olarak işleme konulur.

### 2.2 Z-tamponu (Z-buffer)

Ed Catmull tarafından 1974 yılında ortaya atılan Z-tamponu algoritması [15], basit bir görünür yüzey tespit yöntemi olmasına rağmen diğer birçok görünür yüzey tespit algoritmasının dezavantajlarına sahip değildir. Bu özellikleri Z-tamponu algoritmasını bilgisayar grafiklerinde en yaygın kullanılan görünür yüzey tespit algoritması olmasını sağlamıştır. Birçok grafik uygulaması ve grafik hızlandırıcı donanım görünür yüzey tespit algoritması olarak Z-tamponu veya Z-tamponuna benzeyen yöntemler kullanır.

Z-tamponu algoritmasının arkasında yatan temel fikir ekranda gösterilecek olan piksellerden (görüntü elemanı) bakış noktasına en yakın olanları (en küçük  $z$  koordinatına sahip olanları) bulmaktır. Bunun için, ekrandaki tüm pikseller için iki adet iki boyutlu dizi kullanılır. İlk dizi noktaların rengini, ikincisi de noktaların derinliğini saklar. Renk dizisi arka plan rengiyle, derinlik dizisi de mümkün olan en büyük değer ile doldurulur. Ekranda görüntülenecek tüm pikseller için pikselin  $z$  koordinatı hesaplanır. Hesaplanan  $z$  değeri derinlik dizisinde aynı piksele karşılık gelen değerden daha küçükse (bir başka deyişle, piksel daha yakındaysa), derinlik dizisindeki pikselin eski  $z$  değeri, hesaplanan yeni  $z$  değerine, renk dizisindeki pikselin değeri de görüntülenecek pikselin rengine eşitlenir.

Çoğunlukla, Z-tamponu yöntemi, poligonlar kullanılarak oluşturulmuş sahneler için kullanılır. Çünkü, doğrusal interpolasyon yöntemleri, poligonu oluşturan piksellerin  $z$  koordinatlarının hızlıca bulunmasını sağlar. Ancak Z-tamponu algoritmasının önemli avantajlarından bir diğeri de düzlemsel olmayan yüzeyler için de uygulanabilir olmasıdır. Üç boyutlu nesnenin tanımlanma biçimi ne olursa olsun, nesne üzerinde bulunan noktaların  $z$  koordinatı hesaplanabildiği sürece, Z-tamponu algoritması kullanılabilir.

### 3. Işık ve Aydınlanma Modelleri

Işık-nesne etkileşimleri, gerçekçi görüntülerin elde edilmesinin arkasında yatan en önemli etkenlerden birisidir. Bu nedenle doğada gerçekleşen ışık-nesne etkileşimleriyle ilgili birçok araştırma yapılmıştır. *Aydınlanma modeli* belirli özelliklere sahip nesne ve ışık kaynağının birbirleriyle olan etkileşiminin tanımlandığı matematiksel modeldir. Aydınlanma modelinin sunduğu matematiksel denklemler, yüzey üzerinde bulunan bir noktanın aydınlanma değerinin doğru ve gerçekçi olarak hesaplanmasına olanak sağlar.

Bu konunun detayına inildikçe çok karmaşık ve matematiksel modellere oturtulması bir hayli güç ışık-nesne etkileşimleriyle karşı karşıya kalınır. Dolayısıyla, olası tüm ışık-nesne etkileşimlerini yakalayabilen mükemmel bir matematiksel modelin oluşturulması hiç de kolay değildir. Karmaşık ışık-nesne etkileşimlerinin çözülebileceğini ve matematik ile tanımlanabilir hale getirilmesinin olası olduğunu kabul etsek bile, hesaplamalar için gereken sürenin makul sınırlar içerisinde olacağını söylemek oldukça zor olacaktır. Ayrıca deneyimsiz bir kullanıcı karmaşık bir aydınlanma modelini anlamak ve istediği görsel amaçlar doğrultusunda kontrol etmekte zorlanacaktır. Bu tür nedenlerden dolayı, araştırmacılar ya bazı varsayımlar yaparak, ya da bazı ışık etkileşimlerini gözardı ederek daha basit ve hesaplanması imkânlar dahilinde olan aydınlanma modelleri geliştirmişlerdir. Şu ana kadar geliştirilen birçok aydınlanma modeli vardır ve kullanılacak aydınlanma modeli üç boyutlu uygulamanın amaç ve gereksinimlerine göre seçilir. Ulaşılmak istenilen gerçekçilik düzeyi arttıkça, daha karmaşık ve daha uzun işlemler gerektiren aydınlanma modellerinin kullanılması gerekir.

Örneğin, bir uçuş simülasyonunda çok gerçekçi ama yavaş görüntülerin yerine, hızlı ve akıcı bir görüntü tercih edilir. Bu sebepten dolayı, fazla işlemci gücü gerektirmeyen basit aydınlanma modelleri kullanılır. Bununla birlikte, bilgisayarla kullanılarak yapılmış film hilelerinde en gerçekçi görüntüye ulaşmak birincil amaçtır. Bu tür uygulamalarda ileri düzeyde aydınlanma modelleri kullanılır ve görüntülerinin oluşturulması uzun zaman alır.

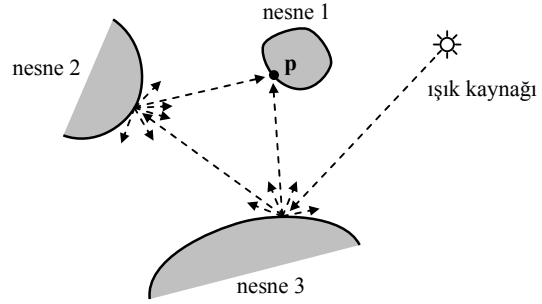
#### 3.1 Phong Aydınlanma Modeli

Ne halen üzerinde araştırmaların devam ettiği aydınlanma modelleri kadar karışık, ne de uçuş simülasyonlarında kullanılan aydınlanma modelleri kadar basit bir aydınlanma modeli Bui-Tuong Phong tarafından 1975 yılında geliştirilmiştir [4]. Phong aydınlanma modeli geliştirildiği yıllardan beri bilgisayar grafiklerinde çok yaygın olarak kullanılmıştır. Phong aydınlanma modeli *ortam ışığı*,

*dağınık yansımaya* ve *düzgün yansımaya* adı verilen üç temel bileşenden oluşur.

##### 3.1.1 Ortam Işığı (Ambient Light)

Bir ışık kaynağından direkt olarak ışık almayan yüzeyler, sahnede bulunan diğer yüzeylerden yansıyan fotonlar tarafından aydınlatılabilir. Bir başka deyişle, sahnede bulunan yüzeyler hem ışık kaynakları tarafından doğrudan (direct lighting) hem de diğer yüzeyler tarafından yansıtılan fotonlar tarafından dolaylı olarak (indirect lighting) aydınlatılır.



Şekil 10. p noktası, nesne 2 ve nesne 3'ten yansıyan fotonlar tarafından aydınlatılmaktadır.

Şekil 10'de nesne 1'in yüzeyinde bulunan p noktası ışık kaynağından direkt olarak ışık almamasına rağmen, nesne 2 ve nesne 3'ten yansıyan fotonlar p noktasının aydınlatılmasına yol açmaktadır. Bu aydınlanma değerinin tam olarak hesaplanması, sahnede bulunan nesnelerin birbirleri arasındaki ışık etkileşimlerinin hesaplanmasını gerektirdiğinden, zor ve uzun zaman alan bir süreçtir<sup>2</sup>. Ayrıca Phong aydınlanma modeli yerel bir modeldir. Yerel aydınlanma modellerinde, aydınlanma hesaplamaları yapılırken ışık kaynakları ve aydınlatılan noktanın özellikleri dışında hiçbir etken göz önünde bulundurulmaz.

Phong aydınlanma modelinde, nesneler arası ışık etkileşimleri nedeniyle oluşan aydınlanma miktarı ortam ışığı (sahne ışığı, arkaplan ışığı) adı verilen sabit bir bileşenle taklit edilmeye çalışılır. Ortam ışığının yüzeyin her noktasına eşit miktarda düştüğü kabul edilir ve hiç bir yön özelliği yoktur.  $I_a$  ortam ışık kaynağının şiddeti ve  $k_a$  yüzeyin ortam ışık katsayısı olmak üzere bir yüzeyin ortam ışığından dolayı kaynaklanan aydınlanma miktarı

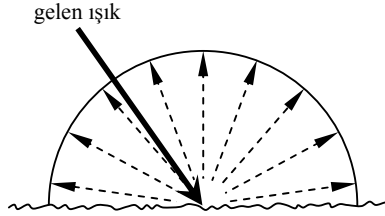
<sup>2</sup> İlk olarak 1984 yılında Goral, Torrance, Greenberg ve Battaile tarafından ortaya atılan *yayılım* (radiosity) yöntemi tüm yüzeyler arasındaki ışık etkileşimlerinin hesaplanıp son derece gerçekçi görüntülerin oluşturulmasına olanak sağlamıştır [16].

$$I_{ortam} = I_a k_a$$

olarak hesaplanır.

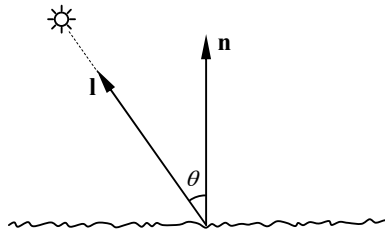
### 3.1.2 Dağınık Yansımaya (Diffuse Reflection)

Kumaş, kağıt, tahta, tebeşir gibi mat ve pürüzlü yüzeylere hangi yönden bakılırsa bakılsın aynı aydınlıkta gözlenir. Phong aydınlanma modelinin ikinci bileşeni olan dağınık yansımaya, yüzeylerin üzerine gelen ışığın belirli bir miktarını her yöne eşit şiddette yansıtmasından dolayı ortaya çıkar (Şekil 11). Bu tür yüzeylerin aydınlanma değeri, üzerine gelen ışık enerjisinin miktarına ve yüzeyin yansıtma özelliklerine bağlıdır. Hesaplamalarda bakış noktasının konumu dikkate alınmaz.



Şekil 11. Kumaş, kağıt, tahta, tebeşir gibi mat ve pürüzlü yüzeyler, üzerine gelen ışığı her yöne eşit şiddette yansıtırlar. Bu nedenle bu tür yüzeylere hangi doğrultudan bakılırsa bakılsın aynı aydınlıkta gözlemlenir.

Dağınık yansımaya bileşeni Lambert teoremine göre hesaplanır. Bu teoreme göre bir yüzeyin dağınık yansımaya nedeniyle kaynaklanan aydınlanma miktarı, yüzeyin normal vektörü  $\mathbf{n}$  ile ışık kaynağına doğru olan  $\mathbf{l}$  (light) vektörü arasındaki açının kosinüsü ile orantılıdır (Şekil 12).



Şekil 12. Dağınık yansımaya bileşeni Lambert teoremine göre hesaplanır.

$I$  ışık kaynağının şiddeti,  $k_d$  yüzeyin dağınık yansımaya katsayısı ve  $\theta$  yüzey normal vektörü ile ışık kaynağına

doğru olan vektör arasındaki açı olmak üzere dağınık yansımaya miktarı

$$I_{dağınık} = I k_d \cos \theta$$

denkleminde göre hesaplanır.

### 3.1.3 Düzgün Yansımaya (Specular Reflection)

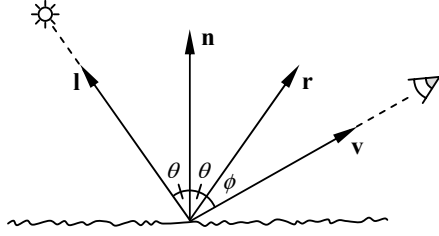
Birçok durumda gerçekçi aydınlanma sonuçları elde etmek için ortam ışığı ve dağınık yansımaya bileşenleri yetersiz kalır. Bunun başlıca nedeni çoğu yüzeyin üzerlerine gelen ışığı her yöne eşit şiddette yansıtmasıdır. Örneğin, düzgün, pürüzsüz ve cilalı yüzeyler söz konusu olduğu zaman, bakış doğrultusunun algılanan ışık şiddeti üzerinde etkisi vardır ve bakış noktasının konumunun değişmesiyle yüzey üzerinden algılanan ışık şiddeti de değişir. Çünkü düzgün yüzeyli nesnelere bir bakıma ayna gibi davranır ve üzerlerine gelen ışığın büyük çoğunluğunu yansımaya doğrultusunda yansıtır.

Phong, bu olguyu gözledikten sonra aydınlanma modeline düzgün yansımaya adı verilen üçüncü bir bileşen eklemiştir. Phong aydınlanma modelinin temelini düzgün yansımaya bileşeni oluşturur. Zaten sadece ortam ışığı ve dağınık yansımaya bileşenleri ile oluşturulan basit aydınlanma modelleri, Phong aydınlanma modeli ortaya atılmadan önce de kullanılmaktaydı [17].

Phong aydınlanma modelinin dayandığı hiçbir matematiksel ya da fiziksel kuram yoktur. Bütünüyle Bui-Tuong Phong tarafından yapılan gözlemler sonucu elde edilmiş ampirik (bir kurama değil de yalnızca deneye, gözleme dayanan) bir modeldir. Ancak gerçekçi sonuçlar elde edilmesinin yanında hesaplanmasının kolay olması, Phong aydınlanma modelinin bilgisayar grafiklerinde çok geniş bir yer edinmesine olanak sağlamıştır. Ayrıca bilgisayar grafiklerinde kullanılan birçok geleneksel aydınlanma modeli teoride yeri olmayan çok sayıda varsayım, hile ve basitleştirme içerir. Ancak, bunlar pratikte oldukça düzgün sonuçlar verir.

Düzgün yansımaya bileşeninin değeri, bakış noktasına doğru olan  $\mathbf{v}$  (viewing) vektörü ile yansımaya doğrultusunu belirten  $\mathbf{r}$  (reflection) vektörü arasındaki açıya bağlıdır (Şekil 13).

## Nesne Tanımlama Yöntemleri



Şekil 13. Düzgün yansıma bileşeni,  $r$  ve  $v$  vektörleri arasındaki açıya bağlıdır.

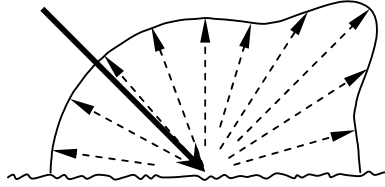
Bu iki vektör arasındaki açı  $\phi$  ile belirtilirse, Bui-Tuong Phong, modelinin en önemli kısmını oluşturan düzgün yansıma bileşenini

$$W(\theta) \cos^n \phi$$

fonksiyonunu temel alarak modellemiştir. Fonksiyonda bulunan  $n_s$  katsayısı yüzeyin düzgünlüğüyle bağlantılıdır: Bu katsayı büyüdükçe daha keskin, küçük alanlı ve odaklanmış yansımalar elde edilir.  $W(\theta)$  ise yüzeyin kendine has optik özelliklerine bağlı yansıma fonksiyondur. Birçok durumda bu fonksiyonun sabit bir  $k_s$  katsayısına eşit olduğunun kabul edilmesi istenilen görsel sonuçları elde etmek için yeterlidir.

Phong aydınlanma modeli, kendisini oluşturan üç bileşenin (ortam ışığı, dağınık yansıma ve düzgün yansıma) toplamına eşittir:

$$I = I_a k_a + I(k_d \cos \theta + k_s \cos^n \phi)$$



Şekil 14. Phong aydınlanma modeli ortam ışığı, dağınık yansıma ve düzgün yansıma bileşenlerinin toplamına eşittir.

Phong aydınlanma modeli dışında yaygın olarak kullanılan diğer bazı aydınlanma modelleri Cook ve Torrance [18], Blinn [19], Oren ve Nayar [20], Ward [21] ve Lafortune [22] tarafından geliştirilmiştir.

## 3.2 Poligonalsal Nesnelere İçin Tonlandırma Yöntemleri

Yüzey üzerinde bulunan noktalara aydınlanma modelinin uygulanması sürecine *tonlandırma* (shading) adı verilir. Poligonalsal nesnelere için kullanılan üç temel tonlandırma yöntemi vardır: düz, Gouraud ve Phong.

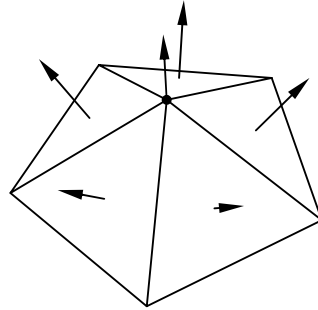
### 3.2.1 Düz Tonlandırma (Flat Shading / Constant Shading)

En hızlı ve basit tonlandırma yöntemi olan düz tonlandırma (flat shading, constant shading) yönteminde ekranda görüntülenecek poligon için bir kere aydınlanma modeli hesaplanır ve poligon elde edilen aydınlanma değeri ile doldurulur.

### 3.2.2 Gouraud Tonlandırma (Gouraud Shading)

Poligonalsal tanımlama yöntemleri eğri yüzeyleri belirtmek için yetersiz kalır. Nesneyi oluşturan poligon sayısını artırmak soruna bir çözüm oluşturmaya rağmen, bu tür bir çözüme gitmek daha fazla bellek ve işlemci gücü gerektirecektir. 1971 yılında Gouraud tarafından ortaya atılan Gouraud tonlandırma yöntemi, poligonalsal nesnelere ait dezavantajları görsel olarak azaltmak amacıyla geliştirilmiştir [3].

Gouraud tonlandırma algoritmasının ilk adımı köşe normal vektörlerinin hesaplanmasıdır. Gouraud, köşe normal vektörlerini hesaplamak için yüzeyin analitik tanımından yararlanmışır. Küre, silindir ya da parametrik yüzeyler gibi matematiksel denklemleri bilinen yüzeylerde, yüzey üzerinde bulunan her noktanın normal vektörünün hesaplanması mümkündür. Ancak ayrıntılı poligonalsal nesnelere bu tür bir yaklaşım ya çok zor ya da imkansızdır. Bu nedenle Gouraud, herhangi bir poligonalsal nesnenin köşe normal vektörlerini yaklaşık olarak hesaplamak için daha kolay bir çözüm sunar: Bu yöntemde, normal vektörünün bulunması istenilen köşeyi paylaşan poligonların normal vektörlerinin ortalaması hesaplanır (Şekil 15).



Şekil 15. Bir köşeye ait normal vektörü, köşeyi paylaşan poligonların normal vektörlerinin ortalamasının hesaplanmasıyla elde edilir.

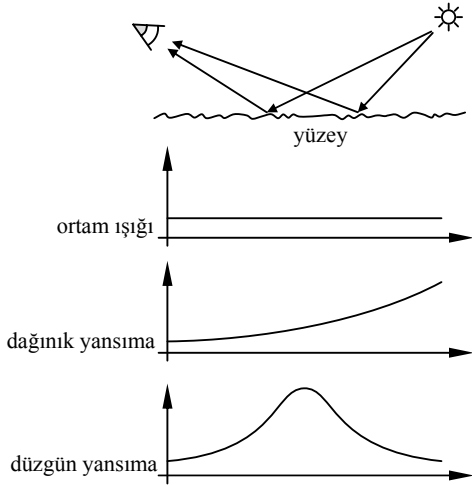


Köşe normal vektörlerinin bulunmasının ardından, her köşeye ait aydınlanma değeri hesaplanır ve poligonu oluşturan piksellerin aydınlanma değeri doğrusal interpolasyon yöntemi kullanılarak elde edilir.

### 3.2.3 Phong Tonlandırma (Phong Shading)

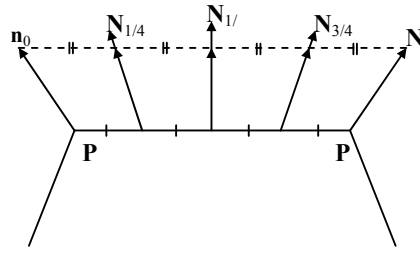
Gouraud tonlandırma yöntemi basit ve hızlı bir yöntemdir. Bundan dolayı, birçok grafik hızlandırıcı donanım Gouraud tonlandırma yöntemini destekler. Ancak, Gouraud tonlandırma yöntemi kullanıldığı zaman karşılaşılan bazı sorunlar vardır. Bu tür sorunların başlıca nedeni doğrusal interpolasyon yöntemlerinin ani aydınlanma değişimlerini yakalayamamasıdır. Örneğin, Gouraud tonlandırma yöntemi kullanıldığı zaman düzgün ve pürüzsüz yüzeylerde gözlenen düzgün yansıma bileşeni çoğunlukla yanlış hesaplanır. Sonuç olarak yanlış görüntüler ve canlandırmalarda rahatsız edici aydınlık değişimleri gözlenir.

Şekil 16'da nokta ışık kaynağı ile aydınlatılan bir yüzeyin ortam ışığı, dağınık yansıma ve düzgün yansıma bileşenlerinin dağılım fonksiyonları görülmektedir. Ortam ışığı ve dağınık yansıma bileşenleri doğrusal fonksiyonlarla yeterli dereceye kadar belirtilebilse de düzgün yansıma bileşeni için aynı şeyleri söylemek olası değildir. Bundan dolayı, Gouraud tonlandırma ancak düzgün yansıma bileşeni göz ardı edilebilecek kadar küçük olduğu zaman kullanışlıdır.



Şekil 16. Ortam ışığı ve dağınık yansıma bileşenleri doğrusal fonksiyonlarla belirli bir doğruluğa kadar ifade edilebilir. Ancak doğrusal interpolasyon yöntemleri düzgün yansıma bileşenini belirtmek için yetersiz kalır.

Phong, 1975 yılında, hem kendine ait aydınlanma modelini geliştirmiş hem de Gouraud tonlandırma yönteminin dezavantajlarının üstesinden gelebilecek yeni bir tonlandırma yöntemi ortaya atmıştır [4]. Phong tonlandırma yönteminin ilk adımını oluşturan köşe normal vektörlerinin hesaplanması Gouraud tonlandırma yöntemi ile aynıdır. Ardından poligonu oluşturan piksellere ait normal vektörleri doğrusal interpolasyon yöntemi kullanılarak elde edilir ve her piksel için aydınlanma modeli uygulanır.



Şekil 17. Phong tonlandırma yöntemi

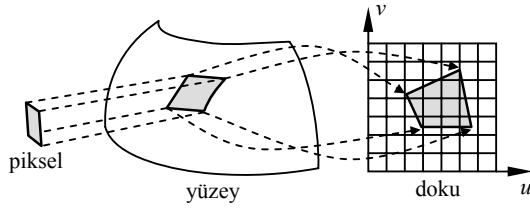
Phong tonlandırma yöntemi Gouraud tonlandırma yöntemine göre çok daha düzgün ve kaliteli sonuçlar verir. Ancak görüntülenen her piksel için aydınlanma modelinin uygulanması, Phong tonlandırma yönteminin çok daha yavaş olmasına neden olur.

### 3.3 Doku Kaplama (Texture Mapping)

Işığın yüzey ile etkileşimini tanımlayan aydınlanma modeli, yüzeyin optik özellikleriyle ilgili olarak birçok parametreye bağlıdır. Örneğin, Phong aydınlanma modelininin davranışı ka ortam ışığı, kd dağınık yansıma ve ks düzgün yansıma katsayıları ile belirtilir. Ancak bu parametrelerin tüm yüzey boyunca sabit olması ayrıntılı yüzeylerin tanımlanmasını güçleştirir. Doku kaplama olarak adlandırılan yöntemle, aydınlanma modelinin bağlı olduğu parametrelerin doku adı verilen yapılardan alınması fikri ortaya atılmıştır. Bu yöntem ilk olarak 1974 yılında Catmull tarafından bulunmuş [15], ardından 1976 yılında Blinn ve Newell tarafından genişletilmiştir [23]. Doku kaplama yöntemi kullanılarak, iki boyutlu resimlerin ya da prosedürel yöntemlerle tanımlanmış dokuların yüzey üzerine kaplanması yoluyla ayrıntılı yüzeylerin oluşturulması mümkündür. Örneğin, tuğlalardan oluşmuş bir duvarın her ayrıntısını birçok poligonla modellemek yerine, bir duvar resmi tek bir poligonun üzerine kaplanabilir. Böylece daha gerçekçi sonuçlar daha hızlı, daha kolay ve daha az bellek kullanılarak elde edilir.

## Nesne Tanımlama Yöntemleri

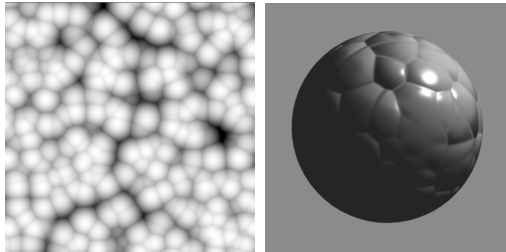
Doku kaplama yöntemi iki temel adımdan oluşur (Şekil 18). İlk adım, ekranda görüntülenecek pikselin dört köşesinin yüzey üzerine izdüşümünün alınmasıdır. Ardından, piksel köşeleri, dokunun tanımlandığı iki boyutlu (u, v) koordinat sistemine taşınır. Dört (u, v) koordinatı doku üzerinde bir dörtgeni belirtir ve bu dörtgenin içinde bulunan doku elemanlarının ortama değeri aydınlanma modelinde kullanılır.



Şekil 18. Doku kaplama

### 3.4 Pütür Kaplama (Bump Mapping)

Bilgisayar grafiklerinde gerçekçiliği engelleyen önemli etkenlerden birisi, oluşturulan görüntülerin gereğinden fazla düzgün gözükmesidir. Doğada bulunan nesnelere mükemmel düzgünlükte değildir. Aşınma, çizilme, paslanma ya da yüzeyin kendi doğal yapısı nedeniyle, yüzey üstünde birçok girinti çıkıntı bulunur. Örneğin, portakal gibi pütürlü bir yüzeye sahip bu tür nesnelere, poligonlar kullanarak modellenmesine rağmen, yüzeyin pütürlü yapısını belirtmek için çok fazla sayıda poligon kullanılması gereklidir. Blinn, 1978 yılında, nesnenin geometrik yapısını değiştirmeden nesneye pütürlü, girintili çıkıntılı ya da kırışmış görünümü vermek için bir yöntem geliştirmiştir [24]. Pütür kaplama adı verilen bu yöntemde, aydınlanma hesaplamalarında kullanılan normal vektörü, nesneye kaplanan dokuya bağlı olarak değiştirilir. Pütür kaplama yönteminin önemli bir dezavantajı, nesnenin silüetinin düzgün gözükmesidir. Pütür kaplama nesnenin geometrisini değiştirmedikinden dolayı, nesnenin silüet görüntüsü de değişmez.



Şekil 19. Pütür kaplama

### 3.5 Çevresel Kaplama (Environment Mapping)

Çevresel kaplama, yüzeyler üzerinde oluşan yansımaları elde etmek için kullanılan basit ve etkili bir yöntemdir. Blinn ve Newell tarafından 1978 yılında ortaya atılan bu yöntemde [23], bakış noktasından yüzeye doğru olan ışın yüzey normaline göre yansıtılır. Yansıma yönüne doğru olan ışın çevresel kaplama ile kesiştirilir ve kesişim noktasındaki doku elemanı aydınlanma modelinde kullanılır. Çevresel kaplama basit bir yöntem olduğu için, hemen hemen tüm grafik hızlandırıcı donanımlar çevresel kaplamayı destekler.

### Referanslar

- [1] Brian E. Cripe ve Thomas A. Gaskins. The DirectModel Toolkit: Meeting the 3D Graphics Needs of Technical Applications. Hewlett-Packard Journal. 1998.
- [2] Michael Deering. Data Complexity for Virtual Reality: Where do all the Triangles Go? IEEE Virtual Reality Annual International Symposium (VRAIS), 1993.
- [3] Henry Gouraud. Continuous Shading of Curved Surfaces. IEEE Transactions on Computers, 20(6): 623-629, 1971.
- [4] Bui-Tuong Phong. Illumination for Computer Generated Pictures. Communications of the ACM, 18(6): 311-317, 1975.
- [5] Gerald Farin. Curves and Surfaces for Computer Aided Design – A Practical Guide. Academic Press Inc., 1996.
- [6] Thomas Akenine-Möller ve Eric Haines. Real Time Rendering. A.K. Peters, 2002
- [7] Donald Hearn ve Pauline Baker. Computer Graphics. Prentice-Hall. Second Edition, C Version, 1997.
- [8] Charles Loop. Smooth Subdivision Based on Triangles. Yüksek Lisans Tezi. Matematik Bölümü, Utah Üniversitesi. 1987.
- [9] Nira Dyn, David Levin ve John A. Gregory. A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. ACM Transactions on Graphics, 9(2): 160-169, 1990.
- [10] Ed Catmull ve James Clark. Recursively Generated B-Spline Surfaces on Arbitrary Topological Meshes. Computer Aided Design, 10(6): 350-355, 1978.

- [11] D. Doo ve M. Sabin. Analysis of the Behaviour of Recursive Division Surfaces near Extraordinary Points. *Computer Aided Design*, 10(6): 356-360, 1978.
- [12] Leif Kobbelt.  $\sqrt{3}$ -Subdivision. *SIGGRAPH'2000 Proceedings*. 103-112, 2000.
- [13] Ivan E. Sutherland, Robert F. Sproull ve Robert F. Schumacker. A Characterization of Ten Hidden-Surface Removal Algorithms. *ACM Computing Surveys*, 6(1): 1-55, 1974.
- [14] M. G. Newell, R. G. Newell ve T. L. Sancha. A Solution to the Hidden-Surface Problem. *Proceedings of the 1972 ACM National Conference*. 443-450, 1972.
- [15] Ed Catmull. A Subdivision Algorithm for Computer Display of Curved Surfaces. *Doktora Tezi*. Utah Üniversitesi, 1974.
- [16] C. Goral, K. Torrence, D. Greenberg ve B. Bataille. Modeling the Interaction of Light Between Diffuse Surfaces. *SIGGRAPH'84 Proceedings*, 213-222, 1984.
- [17] W.J. Bouknight ve K.C. Kelly. An Algorithm for Producing Half-tone Computer Graphics Presentations with Shadows and Movable Light Sources. *Proceedings of SJCC*. 1-10, AFIPS Press, Montvale, NJ, 1970.
- [18] Robert L. Cook ve Kenneth E. Torrance. A Reflectance Model for Computer Graphics. *ACM Transactions on Graphics*, 1(1): 7-24, 1982.
- [19] James F. Blinn. Models of Light Reflection for Computer Synthesized Pictures. *SIGGRAPH'77 Proceedings*, 192-198, 1977.
- [20] Michael Oren ve Shree K. Nayar. Generalization of Lambert's Reflectance Model. *SIGGRAPH'94 Proceedings*, 239-246, 1994.
- [21] Gregory Ward. Measuring and Modeling Anisotropic Reflection. *SIGGRAPH'92 Proceedings*. 265-272, 1992.
- [22] Eric P. F. Lafortune, Sing-Choong Foo, Kenneth E. Torrance ve Donald P. Greenberg. Non-Linear Approximation of Reflectance Functions. *SIGGRAPH'97*, 117-126, 1997.
- [23] James F. Blinn ve M. E. Newell. Texture and Reflection in Computer Generated Images. *Communications of the ACM*, 19(10): 542-547, 1976.
- [24] James F. Blinn. Simulation of Wrinkled Surfaces. *SIGGRAPH'78 Proceedings*, 286-292, 1978.

**Atılım Çetin, Uğur Güdükbay**

[Bilkent Üniversitesi, Ankara]