

# ÇOK ÇÖZÜNÜRLÜKLÜ MODELLEME İÇİN POLİGONAL BASİTLEŞTİRME

**Uğur GÜDÜKBAY**  
**Bilkent Üniversitesi**  
**Bilgisayar ve Enformatik Mühendisliği Bölümü**  
**gudukbay@bilkent.edu.tr**

## Özetçe

Poligonlar, bilgisayar grafiğinde kullanılan en popüler çizim primitifleridir. Özel grafik donanımı ile çok çabuk boyanabilir ve herhangi bir modeli kolayca tanımlayabilirler. Bununla birlikte, üç boyutlu bir modeli doğru bir şekilde tanımlamak için çok sayıda poligon kullanmak gerekebilir. Poligonal basitleştirme, modelin orjinal şekil ve görüntüsüne sadık kalarak modeli tanımlamak için gerekli poligon sayısını azaltır. Bu makalede mevcut poligonal basitleştirme algoritmaları anlatılmış ve algoritmaların performans ve kalite açısından karşılaştırılması yapılmıştır. Ayrıca, modellerin ve kameranın hareketli olduğu dinamik ortamlar için uygun algoritmalar da anlatılmıştır.

## 1. Giriş

Poligonlar, bilgisayar grafiğinde kullanılan en popüler çizim primitifleridir. Özel grafik donanımı ile çok çabuk boyanabilir ve herhangi bir modeli kolayca tanımlayabilirler. Bununla birlikte, üç boyutlu bir modeli doğru bir şekilde tanımlamak için çok sayıda poligon kullanmak gerekebilir. Örneğin, bir küreyi modellemek için birkaç bin poligon kullanmak gerekebilir. Halen mevcut modelleme sistemlerinde detaylı modeller sofistike modelleme teknikleri (serbest form deformasyon, konstrüktif katı geometri vb.) geometrik primitiflere uygulanarak elde edilmektedir. Detaylı modeller, gerçek fiziksel nesnelere üç boyutlu tarayıcılar kullanılarak da elde edilmektedir. Her iki durumda da bu detaylı modellerin depolanması transfer edilmesi ve grafik ekranlarda gösterilmesi zorlaşmaktadır.

Halen mevcut bilgisayar grafiği amaçlı bilgisayarlarla saniyede yaklaşık olarak bir milyon poligon boyanabilmektedir. Bir animasyon uygulaması saniyede 30 kare göstermesi gerekiyorsa, yaklaşık 30,000 den az poligonu bir kare için boyayabilir. Bu sayı ilk bakışta çok gibi görünse de üç boyutlu bir poligonal modelin bu kadar veya daha fazla poligon içermesi büyük bir olasılıktır. Bu günlerde, bir milyon poligon içeren modeller sıklıkla kullanılmaktadır. Bilgisayar grafiği donanımının model karmaşıklığı ile başbaşa gitmesi zor gözükmemektedir. Poligonal basitleştirme model karmaşıklığı (poligon sayısı) ile donanım performansı arasındaki bu farklılığı azaltmayı amaçlamaktadır. Böylece, modeller çok çabuk çizilebilecek, değişik gereksinimlere göre modellerin değişik detaylardaki versiyonları kalite ve zaman kıstasları gözönüne alınarak kullanılabilirler.

Makalenin organizasyonu şöyledir. İkinci kısımda uygulama alanları anlatılmaktadır. Üçüncü kısımda poligonal basitleştirme algoritmalarının temel özelliklerine dayalı bir sınıflandırması yapılmakta ve kalite ve performans açısından karşılaştırması yapılmaktadır.

Dördüncü kısımda bu algoritmaların gerçekleştirme detaylarından bahsedilmektedir. Beşinci kısımda basitleştirme algoritmaları ile elde edilmiş modellerden örnekler verilmektedir.

## 2. Uygulama Alanları

Poligon basitleştirmenin getirdiği en büyük yarar, boyama performansının artmasıdır. Bu özellikle dinamik ortamlarda (animasyon), boyama işleminin hızlandırılması açısından önemlidir. Ayrıca, bilgisayar destekli tasarım, reklamcılık ve tıp alanlarında faydalanılabilecektir. Tıp alanında bilgisayarlı tomografi (CT) araçları ile alınan görüntülerden elde edilen üç boyutlu modellerin basitleştirilmesi, bu modellerin hızlı bir şekilde gösterilebilmeleri açısından çok önemlidir. Ayrıca, basitleştirme işlemi bir modelin saklanması için gerekli bellek miktarını da azaltmaktadır. Böylece bu modellerin bilgisayar ağları üzerinde bir noktadan başka bir noktaya transferi de çabuklaştırılmaktadır. Ayrıca, poligon basitleştirme bilgisayar grafiğinin yoğun hesaplamalar gerektiren (ışın izleme, ışınalık ve çarpışma tespiti gibi poligon sayısı ile verimliliği doğrudan ilişkili) problemlerinin çözümlerinde de verimlilik sağlamaktadır. Sonuç olarak bu yöntemlerden iletişim, tıp, bilgisayar destekli tasarım, reklamcılık, animasyon ve bilimsel gösterim (scientific visualization) alanlarında yararlanılabilecektir.

Poligonal basitleştirmenin çözümüne yardımcı olacağı problemleri şöyle sıralayabiliriz:

- **Model basitleştirme**  
Modelleme ve tarama sistemleri ile elde edilen modeller boyama performansı açısından optimize edilmediklerinden çoğu zaman görüntü olarak orjinal modelden ayırdedilemeyen fakat çok daha az sayıda poligon içeren basitleştirilmiş halleri bu modellerin yerine kullanılabilirler. Halen bu işlem önemli ölçüde kullanıcı yardımı ile gerçekleştirilmektedir. Model basitleştirme bu işlemi otomatik hale getirmeyi amaçlamaktadır. Böylece bir model değişik performans özellikleri sağlayan ortamlarda kullanılabilir.
- **Değişik detay seviyeleri**  
Bilgisayarların boyama performansını daha da arttırmak için modelin değişik detay seviyelerinde versiyonlarını tanımlamak halen kullanılan bir yöntemdir. Nesne bakış noktasına (kameraya) yakın bir yerde ise modelin çok detaylı bir versiyonu kullanılabilir. Nesne uzaklaştıkça daha düşük sayıda poligon içeren versiyonlar kullanılabilir. Dinamik ortamlarda (animasyon vb.) nesnelerin değişik versiyonları arasındaki geçiş işlemi görsel olarak göze hoş gelmeyeceğinden (popping) bu versiyonlar arasındaki geçişi düzgün bir şekilde yapacak yöntemler geliştirilebilir.
- **Aşamalı transfer**  
Bir model bilgisayar ağları üzerinde bir noktadan bir noktaya taşınacağı zaman, alıcı veriler geldikçe aşamalı bir şekilde modeli daha iyi bir şekilde göstermek isteyebilir. Bu amaçla eğer modelin en basit hali ile birlikte modeli detaylı halinden en basit haline getirmek için kullanılan operasyonlar saklanırsa modelin basit hali transfer edildikten sonra geriye doğru bu operasyonlar birbir uygulanarak modelin değişik versiyonları ve en sonunda da en detaylı versiyonu elde edilebilir. Modelin en basit hali ile birlikte uygulanan operasyonların toplamına aşamalı model yapısı (progressive mesh) denilmektedir [4].
- **Model sıkıştırma**  
Bir modeli depolamak için gerekli hafızayı en aza indirmek için model basitleştirme ile modeldeki poligon sayısı azaltılabileceği gibi, modeli depolamak için gerekli hafıza aşamalı model yapısı kullanılarak azaltılabilir.

- **Seçici Basitleştirme**

Bir modelin değişik detay seviyelerindeki versiyonları, modelin bakış açısından bağımsız modelin her tarafında aynı oranda bir basitleştirmeye elde edilmiş halidir. Bazen, modelin bazı bölgelerinde (bakış noktasına yakın bölgeler) daha detaylandırma yapılabilir.

### 3. Poligonal Basitleştirme Algoritmaları

Poligonal basitleştirme, üç boyutlu poligonal modellerin daha basit bir hale dönüştürülmesi işlemidir. Bu işlem modelin orjinal şekil ve görüntüsüne sadık kalarak modeli tanımlamak için gerekli poligon sayısını azaltır. Poligonal basitleştirme algoritmaları dört guruba ayrılabilir:

1. Poligon köşelerinin iteratif olarak atılmasına dayalı algoritmalar (vertex decimation) [8]
2. Bir grup poligon köşesini birleştirmeye dayalı algoritmalar (vertex clustering) [7]
3. İteratif olarak kenarları yok etmeye dayalı algoritmalar (iterative edge contraction) [2], [3],[6]
4. Örnekleme yolu ile orijinal modele benzeyen yeni bir model oluşturma (sampling) [9]

#### 3.1. Poligon Köşelerinin Yok edilmesi (Vertex Decimation)

Bu gurupdaki algoritmalar, poligonal modelde her safhada iteratif olarak bir poligon köşesini attıktan sonra bu köşeyi kullanan poligonal yüzeyleri de atar ve daha sonra geriye kalan deliği tekrar üçgenlere böler [8]. Bu algoritmalar çokkatlı (manifold) yüzeylere özgü (her kenar en fazla iki poligon tarafından paylaşılabilir) köşelerin sınıflandırılması ve tekrar üçgenleştirme yöntemlerini kullandıklarından modellerin topolojisini koruyan algoritmalarlardır. Bazı alanlarda bu aranan bir özellik olmakla birlikte çok çözünürlüklü boyama sistemleri için bir kısıtlama teşkil etmektedir. Çünkü, bazı modeller yukarıdaki kısıtlamaya uymayan (non-manifold) yüzeylerden oluşmaktadır.

#### 3.2. Poligon Köşelerinin Birleştirilmesi (Vertex Clustering)

Bu gurupta yeralan algoritmalar en genel poligonal modelleri basitleştirebilen algoritmalar [7]. Orijinal modelin etrafına bir kutu yerleştirilmekte ve bu kutu küçük parçalara (grid) bölünmektedir. Herbir hücrenin içinde yeralan poligon köşeleri tek bir poligon köşesine birleştirilmekte ve modelin poligon yüzeyleri de buna göre değiştirilmektedir. Bu işlem çok hızlı ve model üzerinde istenilen bütün topolojik değişiklikleri yapabilmektedir. Fakat küçük parçaların büyüklükleri yapılan basitleştirme için geometrik bir hata payı tanımlanabilmesine olanak sağlamaktaysa da elde edilen basitleştirilmiş modelin kalitesi genellikle düşük olmaktadır. Üretilen basitleştirilmiş model aynı zamanda orijinal modelin gride göre pozisyon ve oryantasyonuna da bağlıdır. Bu yöntem uyarlanabilir bir grid yapısına da kolaylıkla dönüştürülebilir. Bu basitleştirme sonuçlarını iyileştirebilir ancak yine de istenilen kontrol ve kaliteyi destekleyememektedir.

#### 3.3. İteratif Kenar Yoketme (Iterative Edge Contraction)

Bu yöntemi kullanan pek çok algoritma geliştirilmiştir. Bu algoritmaları birbirinden ayıran en önemli özellik, yokedilen poligon kenarının nasıl seçildiğidir. Böyle algoritmalara örnek olarak [2],[3],[6] sayılabilir. Bu algoritmalar da çokkatlı yüzeyler için tasarlanmış görünse de kenar yoketme işlemi çokkatlı olmayan yüzeyler için de kullanılabilir. Ardışık kenar yoketme işlemleri ile modellerdeki delikler kapatılabilir, fakat birbirlerine bağlı olmayan bölgeler birleştirilemez. Birbirlerine çok yakın köşeler aralarında bir poligon kenarı yoksa da birleştirilebilir. Böylece birbirlerine bağlı olmayan bölgeler birleştirilebilir. Bu işlem modelin topolojisini değiştirmekte ancak bu durum çok çözünürlüklü çizim için gerek duyulan bir özellik olmaktadır [1].

### 3.4. Sampling (Örnekleme)

Bu yöntem diğer yöntemlerden farklı olarak mevcut modeli basitleştirme yerine onun yerini alabilecek modeli yeni bir poligon köşeleri seti üzerinde yeniden oluşturmaktadır. Bu amaçla önce model üzerinde ya gelişigüzel olarak alınan ya da model üzerinde oluşturulacak bir ızgaraya (grid) göre alınan poligon köşelerini kullanarak yeni bir poligonal model oluşturmaktadır [9].

### 3.5. Algoritmaların Karşılaştırılması

Bunlardan birinci grupta yeralan algoritmalar kısıtlı modeller için çalışmaktadır. İkinci grupta yeralan algoritmalar genel ve hızlı olmalarına karşın sonuçta çıkacak basit modelin orijinaline yakınlığı (kalitesi) konusunda kullanıcıya istediği kontrolü sağlayamamaktadır. Üçüncü grupta yeralan algoritmalar ise daha genel modeller için çalışmakta ve en iyi sonuçları vermektedir. Ancak, bu algoritmaların en önemli dezavantajı yavaş olmalarıdır.

## 4. Poligonal Basitleştirme Algoritmalarının Gerçekleştirme Detayları

Aşağıda enerji optimizasyonuna dayalı poligonal basitleştirme algoritması ve Garland ve Heckbert'in poligonal basitleştirme algoritmasının gerçekleştirme detayları kısaca anlatılmıştır.

### 4.1. Enerji Optimizasyonuna Dayalı Poligonal Basitleştirme

Bu algoritma bir poligonal modeldeki köşe, kenar ve poligon sayısını bir enerji fonksiyonunu minimize ederek azaltmaya çalışmaktadır. Bu enerji fonksiyonu üç terimden oluşmaktadır [3]:

$$E(K, V) = E_{dist}(K, V) + E_{rep}(K) + E_{spring}(K, V)$$

Bu eşitlikte  $K$  modeldeki yüzeyleri (poligonları) tanımlar ve  $V$  ise poligon köşelerinin kümesidir. Bunlardan ilk terim gerçek poligonal modelin basitleştirilmiş olandan farkını ölçmektedir. Bu terim basitleştirilmiş olan yüzeydeki noktaların orijinal yüzeye olan uzaklıklarının karelerinin toplamını vermektedir. İkinci terim poligonal modeldeki köşe sayısını minimize etmek için kullanılmaktadır. Bu iki terim birbirleri ile ters orantılıdır. Birincisini daha fazla köşe ekleyerek minimize etmek mümkünse de ikinci terim köşe atarak minimize edilebilir. Üçüncü terim, optimizasyonun bir çözüme ulaşmasını (convergence) garanti etmek amacıyla kullanılmaktadır.

Poligonal modele köşe eklemek ya da çıkarmak üç temel transformasyon şeklinde olmaktadır. Bunlar, **kenar bölme**, **kenar yoketme** ve **kenar değiştirme**'dir. Kenar bölme bir kenara bir köşe ekleyerek kenarı ikiye bölme işlemidir. Kenar yoketme kenarı bir köşe haline getirme işlemidir. Kenar değiştirme ise iki üçgen arasındaki bir kenarı silerek diğer iki köşe arasında bir kenar oluşturarak yine iki üçgen oluşturur. Optimizasyon işlemi sırasında modelin topolojisini değiştirmeyen herhangi bir işlem uygulanabilir. Aslında bu üç işlem kenar yoketme işlemine indirgenebilir. Enerji fonksiyonundaki ikinci terimin katsayısı değiştirilerek modelin çok sayıda köşe içeren daha detaylı bir versiyonu ya da az sayıda köşe içeren kaba bir versiyonu elde edilebilir.

### 4.2. Garland ve Heckbert Poligonal Basitleştirme Algoritması

Bu algoritma poligon köşelerini ikişer ikişer alarak iteratif bir şekilde bunları tek bir köşe haline getirir. Bu işlemi yaparken hata matrisleri kullanarak yüzeylerinin orijinalinden ne kadar saptığına bakar. Bu algoritmanın en önemli özelliği sadece poligon kenarları değil, bir kenar oluşturmayan herhangi iki köşe de bir köşe olarak birleştirilebilir. Bu işlem sayesinde modelin topolojisi değişebilir. Çok çözünürlüklü modelleme ve boyama için bu gerekli bir özelliktir [1].

### 4.3. Hiyerarşik Dinamik Basitleştirme

Yukarıda bahsedilen algoritmalar, modellerin değişik seviyelerdeki versiyonlarını elde etmekte kullanılan statik yöntemlerdir. Dinamik ortamlar için kullanılan basitleştirme yöntemleri, yukarıdaki algoritmaları bir ön işlem olarak kullanıp modellerin değişik versiyonlarını elde ettikten sonra bunları bir ağacın düğümlerine yerleştirmektedir. Modeller ekranda gösterilirken değişik kriterlere göre bir modelin değişik versiyonları ekranda gösterilmektedir. Bu kriterlere örnek olarak modellerin ekranda kapsadığı alan verilebilir. Model ekrana yakın olduğu zaman ekranda daha fazla yer kapsayacağından detaylı bir versiyonu, uzaklaştığı zaman ise kaba bir versiyonu gösterilebilir.

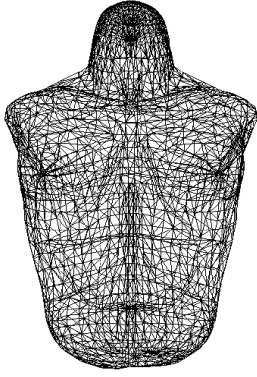
Dinamik ortamlar için diğer bir yöntemde ise, model bir ağaç veri yapısına yerleştirildikten sonra bu yapı dinamik olarak sorgulanarak basitleştirilmiş bir görüntü elde edilmeye çalışılır. Bu yöntem **hiyerarşik dinamik basitleştirme** denilmektedir. Burada önemli bir nokta tüm ortamın tek bir model olarak ele alınmasıdır. Veri yapısında modelin topolojisi ile ilgili bilgi saklanmamakta ve basitleştirme işlemi sırasında topolojinin korunması bir kriter olmaktan çıkmaktadır. Ağacın düğümleri dinamik bir şekilde incelenerek herhangi bir düğümdeki poligon köşelerinin oluşturduğu poligonların ekranda kapsadığı alan belirli bir miktarın üzerine çıktığında o düğümün çocukları kullanılarak detaylı bir hali gösterilir. Bu alan belirli bir miktarın altında ise bu düğüm ve kardeşleri üst seviyedeki düğüme azaltılarak (köşeler birleştirilerek) daha kaba bir hali gösterilir [5].

### 5. Sonuçlar

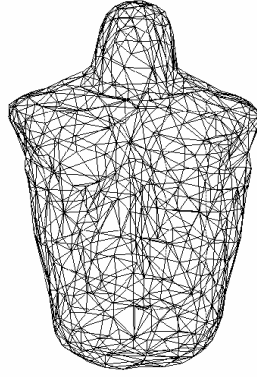
Bu kısımda enerji optimizasyonuna dayalı algoritma ile basitleştirilmiş bir insan gövdesi modelinin görüntüleri verilmiştir. 1. ve 2. şekilde modelin önden görünüşünün iskelet ve boyanmış görüntüleri, 3. ve 4. şekilde ise modelin yandan görünüşünün iskelet ve boyanmış görüntüleri verilmiştir. Orijinal gövde modeli 3716 adet üçgenden oluşmaktadır. Basitleştirilmiş model ise 1080 üçgen içermektedir. Poligon sayısı yaklaşık olarak 1/4 oranına indiği halde modelin şekil ve görüntüsü orijinalinden ayırdedilemeyecek kalitededir.

### Kaynakça

- [1] Garland, M. and Heckbert, P.S., "Surface Simplification Using Quadric Error Metrics," *ACM Computer Graphics (SIGGRAPH'96 Proc.)*, Vol. 30, No. 2, August 1996.
- [2] Gueziec, A., "Surface Simplification with Variable Tolerance," in Second International Symposium on Medical Robotics and Computer Assisted Surgery, 132-139, 1995.
- [3] Hoppe, H., "Surface Reconstruction from Unorganized Points," Ph.D. Thesis, University of Washington, Dept. of Computer Science, 1994.
- [4] Hoppe, H., "Progressive Meshes," *ACM Computer Graphics (SIGGRAPH'96 Proc.)*, Vol. 30, No. 2, 99-108, 1996.
- [5] Luebke, D. and Erikson, C., "View-dependent Simplification of Arbitrary Polygonal Environments," *ACM Computer Graphics (SIGGRAPH'96 Proc.)*, Vol. 30, No. 2, 1996.
- [6] Ronfard, R. and Rossignac, J.R., "Full-range Approximation of Triangulated Polyhedra," *Computer Graphics Forum (Proc. Eurographics'96)*, Vol. 15, No. 3, 1996.
- [7] Rossignac, J.R., P. Borrel, "Multiresolution 3D Approximations for Rendering Complex Scenes," in B. Falcidieno and T. Kunii, editors, *Modeling in Computer Graphics: Methods and Applications*, 455-465, 1993.
- [8] Schroeder, W.J., J.A. Zarge, W.E. Lorensen, "Decimation of Triangle Meshes," *ACM Computer Graphics (SIGGRAPH'92 Proc.)*, Vol. 26, No. 2, 65-70, 1992.
- [9] Turk, G., "Re-Tiling Polygonal Surfaces," *ACM Computer Graphics (SIGGRAPH'92 Proc.)*, Vol. 26, No. 2, 55-64, 1992.

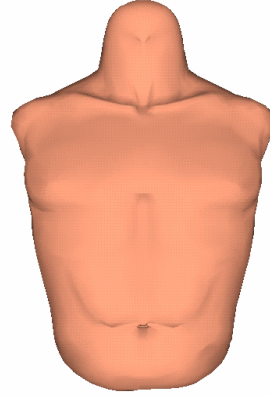


(a)

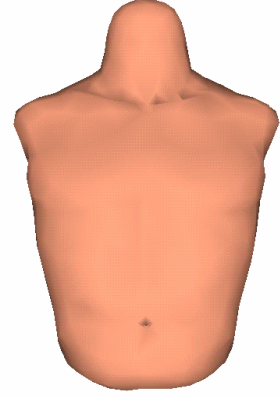


(b)

1. İnsan gövdesinin iskelet (wireframe) önden görünüşü (orijinal (a) basitleştirilmiş (b))

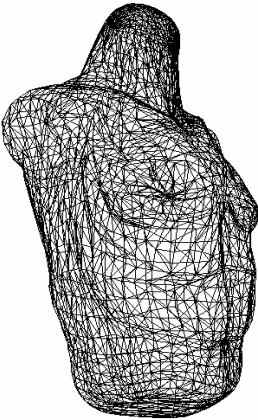


(a)

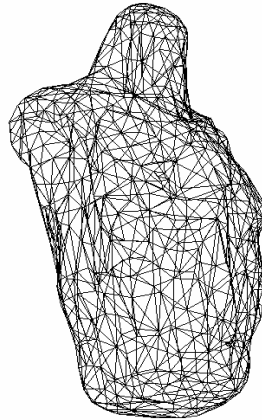


(b)

2. İnsan gövdesinin boyanmış (shaded) önden görünüşü (orijinal (a) basitleştirilmiş (b))

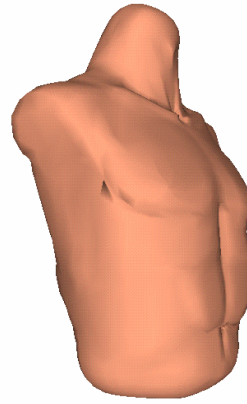


(a)

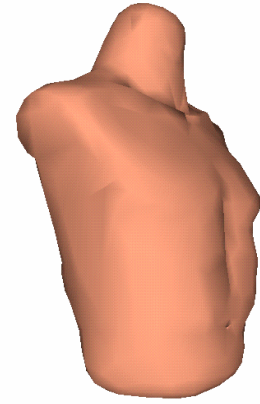


(b)

3. İnsan gövdesinin iskelet (wireframe) yandan görünüşü (orijinal (a) basitleştirilmiş (b))



(a)



(b)

4. İnsan gövdesinin boyanmış (shaded) yandan görünüşü (orijinal (a) basitleştirilmiş (b))