Model Animasyonlarının PCA Kullanılarak Harekete Bağlı Gruplandırılması

Motion based Clustering of Model Animations using PCA

Kıvanç Köse, A. Enis Çetin

Elektrik Elektronik Mühendisliği Bölümü, Bilkent Üniversitesi, Ankara, Türkiye

kkivanc@ee.bilkent.edu.tr, cetin@bilkent.edu.tr

Özetçe

Gerçek 3-Boyutlu (3B) sahnelerin 3B model animasyonları kullanılarak ifade ve kayıt edilmelerinde son yıllarda büyük bir artış yaşamaktadır. Bu kaydedilen 3B sinyaller daha sonra sıkıştırmakta, kullanıcı tarafına iletilmekte ve burada geri çatılarak kullanıcıya yeniden sunulmaktadır. Burada bahsi geçen her adım aslında sinyal işleme alanından yeni bir konu açmıştır. Bu tip sahnelerde kaydedilen modelleri bir bütün halinde aynı şekilde işlemek çoğu zaman en iyi sonucu vermeyebilmektedir. Bu nedenle modelleri oluşturan köşe noktalarını gruplandırmak sıkça başvurulan bir method haline gelmiştir. Örnek olarak model animasyonlarının sıkıştırılmasında, harekete bağlı gruplandırma sıkça kullanılmaktadır. Bu makalede animasyon modellerinin harekete bağlı gruplandırılması üzerine yeni bir algoritma önerilmektedir. Model animasyonları ilk olarak Ana Bileşen Analizi (PCA) kullanılarak özvektör ve özdeğerlerine ayrılırlar. Önerilen yöntem kullanılarak elde edilen özvektörlere aynı zamanda özgezinge demekte mümkündür. Animasyon köşe noktalarının gezingeleri ile bu özgezingeler arasındaki nokta çarpımı alınarak bulunan özgezinge katsayıları animasyonun köşe noktaları gruplandırılmasında kullanılır. Önerilen methodun sonuçları ve benzer bir algoritma ile karşılaştırılması göstermektedir ki methodun gruplandırması başarılı olmuştur.

Abstract

In the last few years, there is great increase in capture and representation of real 3-Dimensional scenes using 3D animation models. The 3D signals are then compressed, transmitted to the client side and reconstructed for the user view. Each step mentioned here opened a new subject in the field of signal processing. While processing these models, using the model as a whole is not the best approach. Therefore clustering the model vertices became a very common method. For example, it is very common to use motion based clustering in animation compression. In this paper a new dynamic model clustering algorithm is proposed. Animation vertices are first put through PCA and partitioned into their eigenvalues and eigenvectors. The eigenvectors found using the proposed method can be called eigen-trajectories. Then the dot product of the these eigentrajectories with the trajectories of the animation vertice are found. These coefficients are used to cluster the animation model. The results and the comparisons with a similar approach show that the proposed algorithm is successful.

1. Giriş

Üç boyutlu (3B) sinyallerin kullanımı son yıllarda giderek artmıştır. Bilgisayar-destekli çizim ve tasarım programları, oyun endüstrisi gibi alanlarda 3B grafikleri uzun zamandır kullanılmaktadır. Bunların yanına bir de hayatımıza giderek daha fazla girmeye başlayan 3-boyutlu filmler, insanlara daha gerçekçi bir görsel deneyim sunmaktadır. Son yıllarda 3-boyutlu televizyon konusunda ülkemizin de başı çektiği [1] birçok araştırma yapılmaya başlanmıştır.

3-Boyutlu televizyon denildiğinde insanların aklına ilk olarak, Yıldız Savaşları filminde olduğu gibi holografik bir görüntünün oluşturulması gelmektedir [2]. Fakat bu alandaki çalışma konuları sadece görüntü elde etmekten ibaret olmayıp, aynı zamanda bu görüntüyü meydana getirebilecek sinyallerin kaydedilmesi, iletilmesi ve en sonunda gösterilmesidir.

Problem bu açıdan bakıldığında aslında kendi içinde irili ufaklı ama birbirine bağlı birçok araştırma konularını içermektedir. Burada bahsi geçen problem aslında birçok bakımdan görüntü işleme ve aktarımı problemlerine benzetmekle beraber, verilerin genellikle iki yerine üç boyutlu olmaları, farklı yaklaşımların kullanılmasını gerekli kılmaktadır.

Yine de 2 boyutlu sinyaller için kullanılan birçok yöntem aslında ya veride ya da yöntemlerde yapılacak küçük değişiklikler sayesinde 3-boyutlu sinyallerinde işlenmesinde de kullanılabilir. 3-boyutlu sinyallerin sıkıştırılması için önerilen *Geometry Images* [3], *Geometry Videos* [4] ve CODDYAC [5] gibi yöntemler bu tip algoritmalarla örnek teşkil etmektedir.

[5]'de anlatılan CODDYAC algoritması 3-boyutlu bir animasyonu ilk olarak 2-boyutlu bir matris şeklinde ifade etmekte ve daha sonra da bu matrise Ana Bileşen Analizi (PCA) uygulamaktadır. Bu analiz sonucunda animasyonu oluşturan köşe noktalarının özgezingeleri elde edilmektedir. Farklı sayıda özgezinge kullanılarak animasyonun farklı bozulumlarda gericiyatımı yapılabilir.

Bu makalede bahsi geçen yöntem, CODDYAC algoritmasında anlatılan yöntem kullanılarak bulunan özgezinge katsayılarının K-Means ya da Fuzzy Logic algoritmaları ile gruplandırılmasını amaçlamaktadır. Bu gruplama sayesinde animasyon oluşturan köşe noktalarının harekete bağlı gruplandırılması yapılmıştır.

Makalede CODDYAC algoritması bölüm 2'de, bu algoritmadan çıkan katsayılar kullanılarak yapılan harekete bağlı gruplandırma bölüm 3'de anlatılmaktadır. Gruplandırma sonuçları ve bunların [6] ile karşılaştırılması bölüm 4'te verilmiştir.

Burada bahsi geçen yöntemin ileride ne gibi amaçlarla kullanılması planlandığından bölüm 5'te bahsedilmektedir.

2. CODDYAC

PCA tabanlı sinyal sıkıştırma algoritmaları sinyal işleme literatüründe yaygın olarak kullanılmaktadır. Özellikle veri gruplandırmasında da PCA çok sıkça kullanılan bir yöntemdir. PCA sinyalleri değişimindeki artışa göre sıralayan doğrusal ortogonal bir dönüşümdür [7]. Ayrıca en küçük kare hatası gözönüne alındığında teorik olarak optimum olan dönüşümdür.

PCA sinyal literatüründe *Karhunen-Loeve* dönüşümü olarak anılır. Burada yapılan işlem sinyal verisini bir matris gösterimi olarak yeniden şekillendirmek ve bu matrisin özilinti matrisinin üzerinde özdeğer ve özvektör analizi yapmaktır.

CODDYAC algoritmasının temelinde yukarıda bahsi geçen PCA algoritması bulunmaktadır. İlk olarak animasyon;

$$A = \begin{pmatrix} x_1^1 & y_1^1 & z_1^1 & \dots & z_1^F \\ x_2^1 & y_2^1 & z_2^1 & \dots & z_2^F \\ x_3^1 & y_3^1 & z_3^1 & \dots & z_3^F \\ x_4^1 & y_4^1 & z_4^1 & \dots & z_4^F \\ \dots & \dots & \dots & \dots & \dots \\ x_V^1 & y_V^1 & z_V^1 & \dots & z_V^F \end{pmatrix} \quad (1)$$

şeklinde yerleştirilerek 2-boyutlu bir matris haline getirilir. Özvektör analizinin kullanılması sayesinde, sinyalin içindeki verilerin matris gösteriminde nasıl dizildikleri önemli değildir. Daha sonra bu animasyon matrisinin özilinti matrisi oluşturulur. Bu işlem sırasında CODDYAC algoritması animasyonun vertexleri arasındaki özilintiyi ($A \times A^T$) kullanmak yerine, gezin-geleri arasındaki özilintiyi ($A^T \times A$) kullanılmaktadır. Bu sayede algoritmanın hesaplama karmaşıklığında büyük oranda azalım sağlanmıştır [5]. V tane köşe noktasına ve F tane mesh karesine sahip bir animasyonda, $V \times V$ yerine $3F \times 3F$ büyüklüğünde bir özilinti matrisinin kullanılması sağlanmıştır. Bir animasyonda resim karesi sayısı genellikle köşe noktası sayısından çok olduğundan dolayı ($V \gg F$), bu durum özvektörlerin bulunmasında büyük bir kolaylık sağlar.

Oluşturulan bu özilinti matrisi daha sonra özvektörlerine ve özdeğerlerine ayrılır. Elde edilen özdeğerlerin büyüklüklerine göre sıralanmasıyla, bu özdeğerlere karşılık gelen özvektörlerinde önem sırası bulunmuş olur. Bu yöntem ile bulunan özvektörlere aynı zamanda özgeze de denilir. Bunun nedeni $3F \times 3F$ özilinti matrisinin, animasyonun köşelerin arasındaki değil, gezin-gelerin arasındaki özilintiyi göstermesidir. Bu adımdan sonra, bulunan özgezingelerin köşe noktalarının gezin-geleri ile nokta çarpımı alınarak, *izdüşümü katsayıları* bulunur. Her köşe noktasının her eksen (x-y-z) için $3F$ tane izdüşüm katsayısı bulunur. Bunlardan önem sırasına göre sona gelen özgezingelere karşılık gelenler, gözardı edilebilir. Bütün katsayıların kullanılması ile ise orjinal animasyon kayıpsız geriçatılabilir.

3. Köşe noktası gruplandırması

Bulunan bu izdüşüm katsayıları [5]teki gibi animasyonun sıkıştırılması amaçlı kullanılabilir. Bu katsayılar aynı zamanda köşe noktalarının animasyon boyunca nasıl hareket ettikleri hakkında da bilgi taşırlar. Bu nedenle bu katsayılar

aynı zamanda modelin köşe noktalarının harekete bağlı gruplandırılmasında da kullanılabilir.

PCA kullanılarak, köşe noktalarının gruplandırılması daha önce [6]'da ele alınmıştır. [6]'te bahsedilen yöntem ilk olarak rastgele k adet grup merkezi, r_j belirler. Daha sonra her köşe noktası gezin-gesi bu gruplardan kendine en yakın olanına konulur. Burada kullanılan yakınlık ölçütü, ortalama geriçatım hatası;

$$\|x - \tilde{x}_j\|^2 = \|x - r_j - \sum_{i=1}^c \langle x - r_j, e_{i,j} \rangle e_{i,j}\|^2 \quad (2)$$

olarak belirtilmiştir. Burada x orijinal, x_j ise geriçatılmış veri vektörleridir [6]. $e_{i,j}$ ise her grup için atanmış olan temel vektörlerdir ve bunlardan c adet vardır. Daha sonra, bu gruplandırılan gezin-geler kullanılarak yeni grup merkezleri tespit edilir ve her grup üzerinde PCA uygulanarak yeni temel vektörleri $e_{i,j}$ elde edilir. Bu işlemler, ortalama geriçatım hatası belli bir eşik altına düşene kadar tekrarlanır.

Bu makalede bahsedilen yöntem direk olarak gezin-geleri üzerinden gruplandırma yapmak yerine, ilk olarak her gezin-genin izdüşümü katsayılarını hesaplar ve daha sonra bunları kullanarak bir gruplama yapar. Burada belirtilmesi gereken başka bir konu da gruplandırma için bütün izdüşüm katsayılarına ya da bir başka deyişle, bütün özgezingelere ihtiyaç olmamasıdır. Bölüm 2'de bahsedildiği üzere her köşe noktası eksen için $3F$ tane izdüşümü katsayısı hesaplanabilir. Fakat bunlardan bir kısmı küçük özdeğerlere karşılık gelen, özgezingelere ait olduklarından, bunların etkileri azdır.

Varsayalım ki j 'inci köşe noktasının x eksen bileşeninin gezin-gesine ait izdüşümü katsayıları $m_{j,l}$ olsun ($j = 1, \dots, V$, $l = 1, \dots, 3F$). Bu durumda her köşe noktası için $3F$ uzunluğunda bir öznitelik vektörü oluşturulabilir. Ama yukarıda bahsedilen nedenlere dayanarak en etkili birkaç özgezeğe karşılık gelen izdüşüm katsayılarının kullanılması yeterli olmaktadır. Algoritmanın bu kararı otomatik vermesi açısından, özdeğerlerin salt toplamının % 90'ını oluşturan en yüksek değerli özdeğerlere karşılık gelen özgezingelerin kullanılması uygun olduğuna karar verilmiştir.

Yukarıda bahsedilen methodla oluşturulan öznitelik vektörleri daha sonra *K-Means* ya da *Fuzzy-Logic* gibi algoritmalarla gruplanmıştır. Farklı özgeze miktarları (öznitelik vektörü uzunlukları) kullanılarak yapılan gruplama sonuçları bölüm 4'te verilmiştir. Ayrıca dans animasyonu ile yapılan örnek bir gruplama Şekil 2'de verilmektedir. Algoritmanın ilk etap çıkışında bazı aykırı değerler bulunmaktadır. Bunların temizlenmesinde ve uygun gruplara yerleştirilmesinde, modelin bağımlılık verisi kullanılmıştır. Her grup kendi içinde taranmış ve küçük yamalar halinde ana gruptan bağımsız olarak bulunan köşe noktaları belirlenmiştir. Daha sonra bu küçük gruplar kendilerine orijinal modelde bağlı olan daha büyük gruplara katılmıştır.

4. Sonuçlar ve Karşılaştırmalar

Önerilen algoritma farklı animasyonlar üzerinde denenmiştir. Bu bölümde dans, inek ve tavuk animasyonları ile yapılan testlere yer verilmektedir. Dans modeli özellikle MPEG-3DGC grubu tarafından kullanılmasından dolayı tercih edilmiştir.

Şekil 1: Önerilen gruplandırma algoritmasının blok diyagramı.

Şekil 2: Dans modeli için gruplandırma sonucu.

Tavuk ve inek animasyonları ise [6]'da bahsi geçen algoritma ile karşılaştırma yapabilmek için kullanılmıştır. Yapılan deneyler sırasında modellerin kaç gruba ayrılacakları kullanıcı tarafından önceden belirlenmiştir. Grup sayısının otomatik olarak belirlenmesi konusunda bir çalışma yapılmamıştır.

Dans animasyonun 6 gruba ayrılmış hali Şekil 2'de verilmiştir. Tavuk animasyonu için bu makalede ve [6]'da önerilen yöntemlerle gruplandırılması Şekil 3'de verilmiştir.

Önerilen yöntemde, gruplandırma sırasında farklı boyutlarda katsayı vektörlerinin kullanılması mümkündür. Daha kısa vektörler, daha basit ve hızlı bir gruplandırma sağlar. Şekil 4'te farklı uzunluklarda katsayı vektörlerinin kullanılması sonucu elde edilen sonuçlar verilmektedir. Ayrıca Şekil 5'te, inek animasyonun aynı uzunlukta katsayı vektörleri kullanılarak 5 (a) ve 10 (b) gruba ayrılmış halleri verilmektedir.

Şekil 3'de verilen karşılaştırma göstermektedir ki [6]'da bahsi geçen algoritma ile önerilen algoritma birbirleriyle tutarlı çalışmaktadır. İki algoritma da birbirlerine yakın gruplandırma sonuçları vermektedir. Methodlar birbirlerine benzemekle beraber, gruplandırmaların yapılışı açısından iki algoritma farklıdır. Bu makale de önerilen yöntem, özvektör katsayılarını direk olarak gruplandırma da kullanmakta, [6]'da önerilen yöntem ise gezinimleri kullanmaktadır.

PCA algoritmasının temelini oluşturan özvektör ve özdeğer analizi en yaygın olarak *tekil değer ayrıştırması (Singular Value*

Decomposition - SVD) ile yapılır. Kare şeklinde bir özilinti matrisi için SVD'nin hesaplama karmaşıklığı $O(N^3)$ 'tür. Burada N özilinti matrisinin bir boyutuna eşittir. Bu makale de önerilen algoritmanın hesaplama karmaşıklığı şöyledir:

$$CC_{\text{önerilen}} = \text{özilinti matrisi hesabı} + SVD = O((3F)^2 \cdot V^2) + O((3F)^3). \quad (3)$$

Burada F animasyon karesi sayısı ve V de modelin köşe noktası sayısıdır. [6]'da önerilen algoritmanın ise hesaplama karmaşıklığı;

$$CC_{[6]} = \text{özilinti matrisi hesabı} + SVD + \text{hata hesabı} = O\left(\frac{(3F)^2 \cdot V^2}{G} \cdot D\right) + O\left(\frac{(3F)^3}{G} \cdot D\right) + O((3F)^2 \cdot V \cdot c \cdot D). \quad (4)$$

Burada G grup sayısı, c ise katsayı vektörü uzunluğu ve D ise gruplandırma algoritmasının döngü sayısıdır. 3 ve 4'üncü denklemlerde görüldüğü üzere, döngü sayısının (D) grup sayısından (G) büyük olduğu her durumda, bu makale de önerilen algoritmanın hesaplama karmaşıklığı, [6]'e göre daha düşüktür.

5. Vargılar ve Gelecek İşler

Bu makalede CODDYAC algoritması kullanılarak çıkarılan, katsayı vektörlerinin kullanılmasıyla, animasyon üzerinde bir gruplama yapılabileceği gösterilmiştir. Ayrıca verilen örnekler ve kullanılan animasyonlar karşılaştırıldığında, algoritmanın harekete bağlı bir gruplandırma yaptığı görülmüştür. Bu durum özilinti matrisi bulunurken kullanılan method sayesinde sağlanmıştır. Özilinti matrisinden çıkan özvektörlere, köşe noktalarının hareketlerini temsil ettiklerinden dolayı bunlara özgeçim vektörlere denilebilir.

Şekil 2'de de görüldüğü üzere modelin gövdesi, ayakları ve kafası gibi bölgeleri, bir bütün halinde aynı gruplara yerleştirilmiştir. Diğer bir önemli nokta ise insan modeli animasyonunda birbirinden bağımsız olarak hareket eden iki kolun farklı gruplara atanmış olmasıdır.

Kullanılan katsayı vektörü uzunluğunun, gruplandırmaya etkisi Şekil 4'te görülmektedir. Buradaki sonuçlara göre uzun ya da kısa katsayı vektörleri kullanmak grupların oluşumunda çok büyük farklar yaratmamakla beraber, sadece grupların sınır bölgelerinde değişimlere yol açmaktadır. Bu bölgeler grupların uç bölgeleri olduklarından dolayı bu durum normaldir.

Şekil 5'te verilen sonuçlara bakılarak algoritmanın tutarlı çalıştığı ve daha fazla gruplandırma istendiğinde, büyük grupların daha küçük parçalara ayrıldığı görülmüştür. İnek animasyonunun 5 gruba ayrılmış hali ile 10 gruba ayrılmış halleri

Şekil 3: Önerilen yöntemin (üstteki resim) ve [6]de önerilen yöntemin tavuk modelinin gruplandırılmasındaki sonuçlarının karşılaştırılması. Her iki algoritmada modeli 10 gruba ayırmıştır.

karşılaştırıldığında, örnek olarak kafa kısmını iki farklı gruba ve karın kısmını aynı şekilde 2 farklı gruba ayırdığı verilebilir.

Animasyonların köşe noktalarının gruplandırılması, animasyonların sıkıştırılmasında sıkça kullanılan bir yöntemdir. Bu makale de önerilen algoritma ileriki çalışmalarımızda bu amaçla kullanılacaktır. Ayrıca animasyon modellerinin basitleştirilmesinde de bu gruplandırmanın kullanılmasının olasılıkları üzerinde çalışmalar yapılacaktır.

6. Teşekkürler

Bu çalışma Avrupa Topluluğu 6. Çerçeve Programı tarafından 511568 No'lu proje kapsamında (3DTV: Integrated 3D Television: Capture Transmission and Display) desteklenmiştir. Bu çalışma kapsamında beraber çalıştığım Dr. Libor Vasa'ya bu fikrin bulunması ve geliştirilmesi konusundaki yardımlarından dolayı teşekkür ederim.

7. Kaynakça

- [1] 3DTV-Integrated Three-Dimensional Television Capture, Transmission, and Display, Avrupa Topluluğu 6. Çerçeve Programı, Proje No:511568.

Şekil 4: İnek animasyonunun farklı özvektör sayıları kullanılarak 6 gruba ayrılmış halleri. (a) 30, (b) 60, (c) 90 özvektör kullanılarak yapılan gruplandırma sonucu.

Şekil 5: İnek animasyonunun (a) 5 ve (b) 10 gruba ayrılmış halleri.

- [2] H. M. Özaktaş, L. Onural, Three-Dimensional Television Capture, Transmission, Display Springer, 2007.
- [3] X. Gu, S. J. Gortler, H. Hoppe, "Geometry Images", *Proc. of ACM SIGGRAPH* pp. 355-361, 2002.
- [4] H. M. Brinceno, P. V. Sandler, L. McMillan, S. Gortler, and H. Hoppe, Geometry Videos: A new representation for 3D Animations, *Proceedings of the ACM SIGGRAPH/Eurographics Symposium*, pp. 136-146, 2003.
- [5] Libor Vasa, CODDYAC: Connectivity Driven Dynamic Mesh Compression, *Course at 3DTV Con*, Kos, 2007
- [6] Sattler M., Sarlette R., Klein R., Simple and Efficient Compression of Animation Sequences, *Computer Graphics*, Vol.28, pp.25-34, 2004.
- [7] Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4