

LBG ALGORİTMASININ GÖRÜNTÜ DİZİLERİ İÇİN GÜNCELLENMESİ

Kemal ÖZKAN¹

e-posta¹ : kozkan@ogu.edu.tr

Erol SEKE²

e-posta² : eseke@ogu.edu.tr

^{1,2} Eskişehir Osmangazi Üniversitesi Elektrik-Elektronik Mühendisliği Bölümü,
Batı Meşelik, Eskişehir

ABSTRACT

Linda-Buzo-Gray algorithm is one of the well known codebook creation algorithms for vector quantization. LBG algorithm is an iterative process with very high complexity. However it attracts real time video researchers because of high compression possibilities. In this study, instead of trying to find a new code book for a new frame group, the previous codebook is used as initial codebook before iterations, exploiting the high correlation between consecutive frames. Updating the codebook reduces the time by 3.5:1 with a penalty of only about 1% additional distortion.

Anahtar sözcükler : vector nicemleme, LBG algoritması

I. GİRİŞ

Bugünkü endüstri ve araştırmalarda yüksek miktarda verinin saklanma ve iletim ihtiyacı giderek artmaktadır. Sayısal televizyon bunun en güzel örneğidir. Sayısal TV yayınlarının gerçekleştirilmesi, iletimi ve saklanması için çok büyük veri oranları gerekmektedir. Şöyle ki, 1000 x 1000 boyutlarında bir çerçevelik TV görüntüsünü sayısallaştırdığımız zaman $1000 \times 1000 = 10^6$ görüntü elemanı (piksel) oluşur. Ayrıca her pikselin renk / şiddet bilgileri içinde 24 bit ayırdığımızı düşünürsek bu çerçeveyi iletebilmek için 1 / 30 saniyede 2.4×10^7 bit veya bir saniyede 7.2×10^8 bit, buna bağlı olarak yaklaşık 1.44×10^9 Hz'lik bir bant aralığı gerekmektedir. Eğer veri sıkıştırma tekniklerini kullanmazsak normal yayın sistemlerinde bu bilgiyi iletmemiz mümkün olmaz. Çünkü normal TV yayınları VHF ve UHF bantlarından yapılmaktadır. VHF (very high frequency) bandı 30 –300 MHz, UHF (ultra high frequency) bandı 300 – 3000 MHz dir. Bundan dolayı veri miktarını gerçekleştirilebilir düzeylere indirgememiz gerekmektedir [1]. Veri miktarını azaltırken çerçeve başına düşen piksel sayısının, saniyedeki çerçeve sayısının veya renk başına düşen bit sayısının azalması istenmez. Çünkü bizim amacımız resmin kalitesini düşürmek değil aynı kalitedeki resmi çeşitli sıkıştırma teknikleri kullanarak daha az bit ile temsil etmektir [2].

Gerçek hayattaki bir resimde birbirine yakın olan pikseller arasında çok büyük bir oranda benzerlik veya tekrarlılık vardır. Bant genişliğini azaltabilmek için

bu özellikten mümkün olduğu kadar faydalanılabilir. Görüntü sıkıştırmada genel olarak bu özellik kullanılır. Ayrıca gerçek hayattaki görüntü dizilerinde bir önceki çerçeve ile onu takip eden çerçeve arasında da tıpkı birbirlerine yakın olan pikseller arasındaki gibi benzerlik, tekrarlılık vardır. Yani birbirlerini takip eden çerçeveler arasında çok az bir değişim gözlenir. Hareketli görüntü sıkıştırmada çerçeveler arasındaki bu tekrarlılıktan da faydalanılmaktadır [1].

Bu çalışmada veri sıkıştırmada kullanılan yöntemlerden biri olan vektör nicemlemenin temel elemanlarından biri olan kod kitabı üretme algoritmalarından LBG algoritmasının geliştirilmesi hedeflenmiştir. Bilindiği üzere optimum nicemleme bölgelerinin belirlenmesi için LBG algoritması kullanılmaktadır [3]. Ancak LBG algoritması özinelemeli bir yaklaşımdır ve zorluk seviyesi oldukça yüksektir. Araştırmada algoritmanın çalışma zamanının azaltılması hedeflenmiştir. En basit hali ile vektör nicemlemenin görüntü dizilerinin sıkıştırılmasında kullanılması öncelikle geçmiş çerçevelerden kod kitabı üretilmesi ve kod kitabının daha sonraki çerçevelerin sıkıştırılmasında kullanılması ile gerçekleşmektedir. LBG algoritması bir başlangıç kod kitabı ile başlar ve kod kitabını özinelemeli olarak iyileştirerek optimum kod kitabını bulmaya çalışır. LBG algoritmasında bütün görüntü vektörleri tek bir kod vektörü ile temsil edilerek başlanır ve belirlenen kod vektörü sayısına ulaşıncaya kadar devam eder. Başlangıçta bütün eğitim vektörlerinin tek bir sınıfta olduğu düşünülür. Her bir öz yinelemede sınıf sayısı ve kod vektör sayısı ikiye katlanır ve sınıfları temsil edecek kod vektörleri o sınıfın ağırlık merkezine yerleştirilir. Eğitim vektörleri en yakın sınıfa merkezine atanır ve ağırlık merkezi yeniden hesaplanır. Bu işlem eğitim vektörlerinin sınıf değiştirmemesine kadar yada toplam hatanın küçülme oranı belli bir değerin altına ininceye kadar devam eder [4,5,6,6,7].

Daha önce de belirtildiği gibi görüntü dizilerin çerçeveler arasında yüksek oranda benzerlik vardır. Yani birbirini takip eden iki çerçeve arasındaki fark çok azdır. Bundan dolayı bu iki çerçeveden elde edilen görüntü vektörleri de birbirine benzeyecektir. Dolayısıyla ilk çerçeve grubundan elde edilen kod kitabındaki vektörlerin uzaydaki yeri ikinci çerçeve grubu için hesaplandığında çok az yer değişmesi beklenmektedir. Bu çalışmada ise görüntü dizisinin

ilk çerçevesine klasik LBG algoritması uygulanmış, ardından gelen çerçeveler ile ilk çerçeve arasında bu tekrarlıktan faydalanarak tekrar kod kitabı oluşturmak yerine ilk kod kitabı güncellemeye çalışılmıştır. Çünkü tekrar kod kitabı oluşturmak yukarıda da belirtildiği gibi çok zaman almaktadır

2. LBG ALGORİTMASI

İyi bir bloklama veya kod kitabı oluşturma için etkili algoritmalar geliştirilmiştir. LBG algoritması en çok bilinen ve çalışılan kod kitabı oluşturma algoritmalarından biridir. Linde–Buzo–Gray uzaydaki bir grup noktanın tek bir noktaya uzaklıkları toplamının minimum olması için o tek noktanın grup merkezinde olması gerektiğini söylemektedirler. Kod kitabı oluşturmada amaç M sayıdaki vektörü N sayıdaki vektör ile temsil etmektir. Temsil eden vektörlere kod vektörü denilmektedir. Her kod vektörü temsil ettiği vektör gruplarının merkezinde seçilmektedir. LBG algoritması öz yinelemeli bir algoritmadır. Başlangıçta tüm vektörlerin merkezi hesaplanır ve ilk kodvektör bu merkeze yerleştirilir. Daha sonra bu kodvektör çok yakınında yer alan iki adet kodvektör ile değiştirilir. Vektörler daha yakın oldukları kodvektörün grubuna dahil edilir ve yeni ağırlık merkezleri hesaplanır. Merkezlerin yeri değişmiş olacağı için tekrar hesaplanırlar. Daha sonra bu kodvektörler yine ikiye bölünürler. Böylelikle herbir öz yinelemede sınıf sayısı ve kodvektör sayısı ikiye katlanır. Bu işlem herhangi bir vektörün grup değiştirmemesine kadar devam eder. Bütün vektörlerin en yakın sınıfta olması algoritma sonucunun veya her adımdaki sonucun optimum olmasını garanti etmez. Bu sonuçlar başlangıç sınıf vektörünün ağırlık merkezinin farklı yerde olmasına veya kod vektörlerinin ikiye ayrılması sırasında eklenen veya çıkarılan ϵ değerine bağlı olarak değişir [7,8,9,10].

LBG algoritmasının adımları şunlardır;

1. Bütün eğitime vektörlerinin ağırlık merkezi bulunur.
2. Bütün eğitime vektörleri en yakın sınıfa konur. (Başlangıçta tek sınıf olduğundan bütün vektörler o sınıfa konur.)
3. Her sınıfın ağırlık merkezi hesaplanır. Bu ağırlık merkezi o sınıfı temsil edecek kod vektördür.
4. Toplam bozulma hesaplanır.
5. Toplam bozulma miktarı ile bir önceki yinelemedeki bozulma miktarı arasındaki fark büyükse 2. adıma dönülür.
6. Kod vektör sayısı yeterli ise öz yineleme durdurulur.
7. Her kodvektör ikiye bölünür.

3. GÜNCELLEME ALGORİTMASI

Görüntü çerçeve dizilerinin küçük grupları üzerinde LBG algoritması uygulanarak elde edilen kod kitabı sonraki çerçevelerin vektör nicemlemesi için

kullanılır. LBG algoritması başlangıç kod kitabı ile başlar ve kod kitabını öz yinelemeler bölerek optimum kod kitabını bulmaya çalışır. Daha önce de belirtildiği gibi görüntü dizilerinin ardışıl çerçeveleri arasında yüksek oranda benzerlik vardır. Yani birbirini takip eden iki çerçeve arasındaki fark çok azdır. Bundan dolayı bu iki çerçeveden elde edilen görüntü vektörleri de birbirine benzeyecektir. Dolayısıyla ilk çerçeve grubundan (ardışıl birkaç çerçeve) elde edilen kod kitabındaki vektörlerin uzaydaki yeri sonraki çerçeve grubu için hesaplandığında çok az değişecektir [11]. Sonraki çerçeve grupları ile yeni kod kitabı bulmak için daha önce bulunmuş olan kod kitabı kullanılabilir. Bunun için görüntü dizisinin ilk çerçeve grubundan gelen vektörlere klasik LBG algoritması uygulanmış, ardından gelen çerçeveler ile ilk çerçeve grubu arasında bu tekrarlıktan faydalanarak yeni kod kitabı oluşturmak yerine halihazırdaki kod kitabı güncellenmiştir. Çünkü, tekrar kod kitabı oluşturmak yukarıda da belirtildiği gibi çok zaman almaktadır. Güncelleme algoritmasının adımları şöyledir.

1. LBG algoritması kullanarak kod kitabı bulunur.
2. Yeni eğitime vektörleri en yakın sınıfa konur. Eğitime dizisi $\{x_j ; j = 0, 1, \dots, n-1\}$ olmak üzere;
 $x_j \in C_i$ eğer $d(x_j, y_i) < d(x_j, y_N)$
3. Her sınıftaki eğitime vektör sayısı bulunur. s_i her sınıftaki eğitime vektör sayısı olmak üzere;
 $s_i = s_i + 1$ eğer $x_j \in C_i$
4. Her sınıfta en az bir tane eğitime vektörü varsa durulur. Yeni kod kitabı bulunmuş olur. $s_i = 0$ algoritma durdurulur. (bütün i 'ler için). sonuç = C olur.
5. Her sınıftaki en uzak eğitime vektörleri bulunur.
 $z_i = x_j, \max(d(x_j, y_i)) \in C_i$
 z_i vektörleri uzaklık sırasına göre büyükten küçüğe sıralanır.
6. Sıfır eğitime vektörüne sahip sınıf sayısı bulunur. k : sıfır eğitime vektörüne sahip sınıf sayısı
 $k = k+1$ eğer $s_i = 0$ bütün i 'ler için
 $v_i = z_i, 0 \leq i \leq k$
7. k sayısı kadar, en uzak vektör kendi sınıflarından çıkartılır. Çıkartılan her vektör yeni bir sınıf oluşturulur.
 $C_i = C_i$ eğer $s_i \neq 0$
 $C_i = C_i - v_k$ eğer $s_i \neq 0, 0 \leq i \leq N - k$
ve
 $C_{i+k} = v_k \quad i < k \leq N$
8. Her sınıfın ağırlık merkezi hesaplanır. Bu ağırlık merkezi o sınıfı temsil edecek kod vektördür. 2. adıma dönülür.

4. UYGULAMA

Hareketli görüntüler için 39 çerçeveden oluşan tenis, 60 çerçeveden oluşan amerikan futbolu, ve 99 çerçeveden oluşan bahçe görüntüleri kullanılmıştır. Üç görüntü dizisinin her çerçevesine LBG algoritması uygulanmış ve kod kitabı büyüklükleri olarak da 2, 4, 8, 16, 32, 64, 128, 512, 1024 seçilmiştir. Güncelleme

algoritmasında ise her üç görüntü dizisinin ilk çerçevesinden elde edilen kod kitabı kullanılarak sonraki kod kitapları bulunmuştur. LBG algoritması ile karşılaştırabilmesi için sinyal gürültü oranları tablo halinde verilmiştir. Hareketli görüntü sayısı fazla olması nedeniyle sadece tenis görüntüsünün 15. çerçevedeki görüntüsü verilmiştir. En iyi sinyal gürültü oranı doğal olarak kod kitabı büyüklüğü 1024 olduğu zaman bulunmuştur. Kod kitabı büyüklüğü artırıldığı zaman SNR artmakta, ancak sıkıştırma oranı azalmaktadır. Ayrıca kod kitabı büyüklüğü ile gerçekleştirme zamanı arasında ters orantı vardır. Kod kitabı büyüklüğü 1024 olduğu zaman LBG algoritması ve güncelleme güncelle algoritması sonucu elde edilen SNR değerleri Tablo 1'de verilmiştir. Tablo 2'de ise algoritmaların gerçekleştirme zamanları saniye cinsinden verilmiştir. Güncelleme algoritması kullanıldığı zaman Amerikan futbolu dizisi için ortalama olarak 1.19 dB (%0.96), tenis dizisi için 0.52 dB (%0.49) ve bahçe dizisi içinde 1.16 db (% 1.21) bozulma olmuştur. Bu bozulma miktarlarına karşın her çerçeve için ortalama olarak sırasıyla 668, 602 ve 534 sn kazanılmıştır. Yani gerçekleştirme zamanı futbol dizisi için 3.61:1, tenis dizisi için 3.78:1 ve bahçe dizisi için 3.39:1 oranında azalmıştır.

Ortalama	Sinyal Gürültü Oranı (SNR) db		
	Her Çerçeveye	İlk Çerçeveye	Güncelle Algoritması
Futbol	122.78	93.37	121.59
Tenis	104.83	90.66	104.32
Bahçe	96.18	81.12	95.00

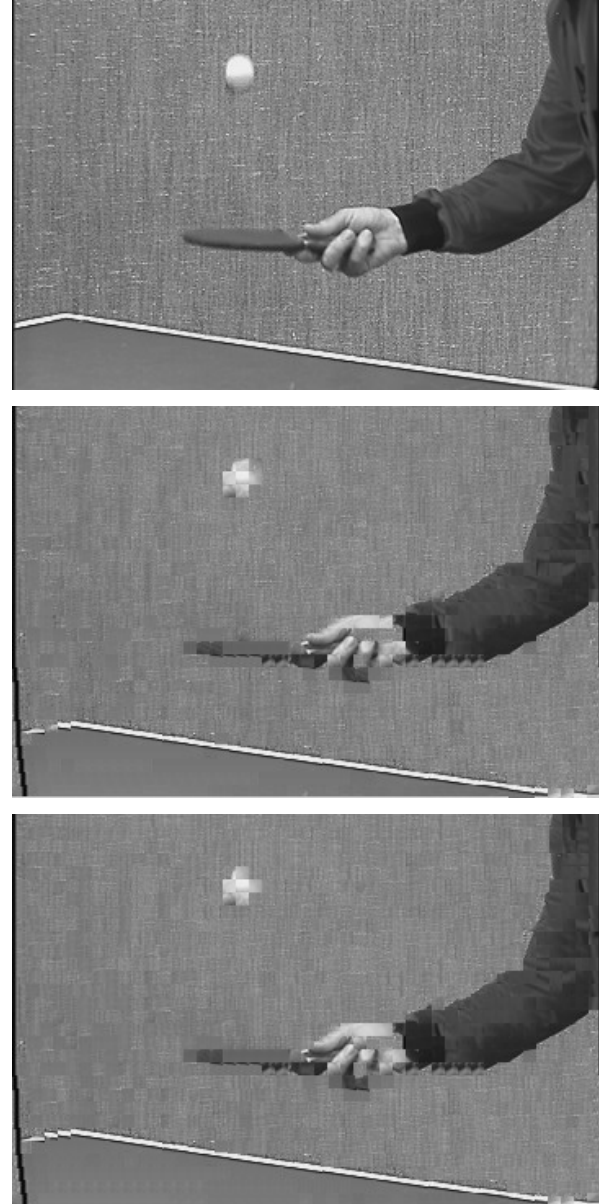
Tablo 1: Görüntü dizilerinde bulunan ortalama SNR

	Gerçekleştirme Zamanı sn			
	Toplam Zaman		Ortalama Zaman	
	LBG	Güncelleme	LBG	Güncelleme
Futbol	55391	15327	923.18	255.45
Tenis	31937	8451	818.90	216.69
Bahçe	75030	22138	757.88	223.62

Tablo 2: Görüntü dizilerinin gerçekleştirme zamanı

SONUÇLAR

Önerilen güncelleme algoritması ile yüksek sıkıştırma oranı elde edilebilen (11:512) vektör nicemlemede, çok kullanılan LBG kod kitabı bulma algoritmasının en büyük problemi olan gerçekleştirme zamanı düşük bozulma oranları ile birlikte azaltılabilmektedir. LBG algoritması bloklar arasındaki ilintiyi azaltır, ancak bloklar arasındaki ilintiyi dikkate almaz. Dolayısıyla tekrar oluşturulan görüntülerde blok etkileri belirgin olarak görülmektedir. Blok etkileri çeşitli filtrelerle azaltılabilmekte, ancak bu sadece görüntünün izlenebilirliğini artırmakta fakat sinyal gürültü oranı azalmaktadır. LBG algoritmasının bir başka etkisi de yumuşatma etkisidir. Çalışmalar kod kitabı güncellemesi sırasında eklenen bozulmanın azaltılmasına yönelmiştir.



Şekil 1. a) Orijinal tenis resmi (15. çerçeve)
b) Klasik LBG algoritması
c) Güncelleme algoritması

KAYNAKLAR

- [1]. Umbaugh, S.E., Computer Vision and Image Processing, Prentice-Hall International Inc., 1998.
- [2]. Gonzalez, R., Woods R., Digital Image Processing, Addison-Wesley Publishing Company, 1992.
- [3]. Gerso, A., Gray R., Vector Quantization and Signal Processing, Kluwer Academic Publishers, 1992.
- [4]. Li W., Zhang Y., A framework of Transform Domain VQ for Image Coding, Proc. of IEEE, Vol 83, No.2, February, 1995.

- [5]. Cay A., Li W., Zhang Y., On the Optimal Transform for Vector Quantization of Images, IEEE, ISCAS 93, Chicago.
- [6]. Gray A., Markel J., Quantization and Bit Allocation in Speech Processing, IEEE, Trans. on Acoustic, Speech and Signal Processing, Vol. ASSP-24, December, 1976.
- [7]. Linde Y, Buzo A., Gray R., An Algorithm for Vector Quantizer Design, IEEE Trans. on Communications, Vol.COM-28, No.1, January, 1980.
- [8]. Gray R., Vector Quantization, IEEE ASSP Magazin, April, 1984.
- [9]. Barlaud M., Pyramidal Lattice Vector Quantization for Multiscale Image Coding., IEEE Trans. on Image Processing, Vol.3, No.4, July, 1994.
- [10]. Nasrabadi N., Feng Y., Image Compression Using Address-Vector Quantization, IEEE Trans. on Communications, Vol.38, No.12, December, 1980.
- [11]. Nasrabadi N., King R., Image Coding Using Vector Quantization: A Review, IEEE Trans. on Communications, Vol 36., No.8, August, 1998.