

KONUŞMA SİNYALLERİ SIKIŞTIRMADA BİR VEKTÖR TRANSFORMU UYGULAMASI

Buket Dedeakayoğulları⁺, M. Atif Çay⁺

⁺ Osmangazi Üniversitesi, Mühendislik Mimarlık
Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü.
26090 Batı Meselik/ESKİŞEHİR.

E-mail:dedeakay@ogu.edu.tr

E-mail:acay@ogu.edu.tr

Özet: Veri sıkıştırma, çok yüksek hızdaki veri akışının göreceli olarak, düşük hızdaki veri akışına dönüştürülmesidir. Günümüze kadar, konuşmayı sıkıştırmak için bir çok teknik geliştirilmiştir. Bu çalışmada, ilk olarak, konuşmayı sıkıştırmak için doğrusal vektör kuantalama kullanılmıştır. Daha sonra, görüntü sıkıştırma için kullanılan LBG (Linde-Buzo-Gray), algoritması konuşma sinyallerine uyarlanarak sıkıştırma işlemi yapılmıştır. Ayrıca, bu işlemde önce, konuşma sinyallerine ikinci bölümde tanımlanan vektör transformu (VT) uygulanarak, konuşma vektörleri, vektör transform etki alanına atılmış ve vektörelleştirilmiş LBG algoritmasıyla kodlama yapılmıştır. Buradan elde edilen verilere, ters vektör transformu uygulanarak, konuşma sinyalleri tekrar üretilmiştir. Elde edilen sıkıştırma sonuçları, SNR değerleri ve grafiksel gösterimleri kullanılarak değerlendirilmiştir.

Anahtar Kelimeler: Konuşmayı Kodlama, VT, LBG.

I. GİRİŞ

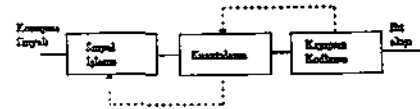
Sürekli zamanda, çok yüksek orandaki veri akış hızının, sayısal haberleşme hattındaki iletim için düşürülmesi veya herhangi bir sinyalin sayısal gösteriminin mümkün olduğunca az bit kullanılarak temsil edilmesi sıkıştırma olarak bilinmektedir. Sayısal ve güvenli haberleşmede veri sıkıştırmanın önemi büyüktür.

Görüntü verisiyle ilgili olarak, sıkıştırma, saklama cihazları veya iletim hatları için çok büyük bit oranları gerekmektedir. Konuşma sinyalleri de skaler kuantalama tarafından oluşturulmuş bir örnekleyiciden oluşan basit bir PCM kullanılarak sayısallaştırılmış ise, konuşma sinyalinin bantgenişliği artar. Bu sebeple, eğer sayısallaştırılmadan önceki haberleşme kanalı kullanılacaksa, bilgi sıkıştırma gerekli olacaktır. Ayrıca, mevcut ses kayıtlarının depolanmasında yer tasarrufu sağlamak amacıyla da sıkıştırma yapılır.

Görüntü sıkıştırma ve konuşma sıkıştırma işlemlerine aynı zamanda, görüntü kodlama veya konuşma kodlama denilmektedir. Yapılan kodlama esnasında, yüksek orandaki verinin düşük oranda

veriye çevriminde, göreceli olarak, bozulmada artış veya doğrulukta azalma olacaktır. Burada, veri sıkıştırmanın temel amacı, verilen sıkıştırma oranı için mümkün olan en iyi doğruluğu bulmaktır. Veya buna eşdeğer olarak, verilen doğruluk için gerekli sıkıştırma oranını minimize etmektir. Göreceli olarak iyi bir doğruluk, darbe kod modülasyonu, transform kodlama, tahminleyici kodlama gibi tekniklerle elde edilebilmektedir. Bu tekniklerin tümünün temel özelliği, kodlamada skaler işlem yapmalarıdır.

Herhangi bir sıkıştırma tekniğinde modelleme üç aşamadan oluşmaktadır ve Şekil 1'de verilmiştir [1]. Birinci aşama genellikle konuşma sinyalini kendi orijinal formundan farklı etki alanına çevirmek için yapılan sinyal işlemedir. İkinci aşama, bilgide kayıpların olduğu kuantalamadır. Üçüncü aşama kayıpsız kodlamadır. Burada sıkıştırılmış bit akışı üretilmektedir. Kuantalama aşaması sıkıştırmanın yapıldığı ve bilgi kayıplarının meydana geldiği yerdir. Bu aşama, her sıkıştırma tekniğinin çekirdeğini oluşturmaktadır. Kuantalama aşamasında skaler kuantalama veya vektörel kuantalama kullanılmaktadır. Birinci aşama olan sinyal işleme aşamasında yapılan transformlar, skaler veya vektörel temelli olabilir. [2-3].



Şekil 1. Üç aşamadan oluşan sıkıştırma tekniği modeli.

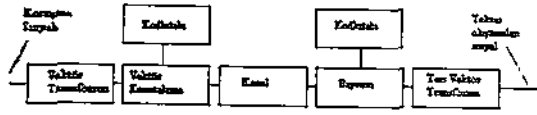
Bu çalışmada, sinyal işleme aşamasında konuşma verilerini ikinci aşamaya en iyi şekilde hazırlamak için, ikinci bölümde tanımlanan, vektör transformu (VT) kullanılmıştır [4-5]. Konuşma verileri bloklara bölünerek VT uygulanmış, vektör transform etki alanında, vektörler kuantalama işlemine tabi tutularak sıkıştırma gerçekleştirilmiştir.

II. VEKTÖR TRANSFORMU VE KONUŞMAYI KODLAMA

II.1. Vektör Transformu

Kuantalamada iyi bir performans elde etmek için kuantalamadan önce sinyal işleme aşamasında konuşma sinyallerine vektör transformu uygulanmaktadır. Şekil 2'de blok diyagramı verilmiştir. Vektör transformu ile bloklar arasındaki ilişki azaltılıp, blok içindeki sinyal örneklerinin birbirine benzemesi sağlanmaktadır. Böylece vektör içi sinyal örneklerinin bağımlılığı artırılıp vektör kuantalamada minimum bozulma elde edilmeye çalışılmaktadır. Konuşma verileri bloklara bölünüp, her blok bir vektör gibi düşünülmekte ve her vektör,

kodkitabında bir kodvektör ile temsil edilerek vektör kuantalama yapılmıştır. Sıkıştırma, kod vektörünün içeriği verilerek gerçekleştirilmektedir.



Şekil 2. Vektör transformu kullanılarak vektör kuantalama.

Kodkitabının içeriği alıcıya da verilmektedir. Başvuru sonunda kod vektörlerin her setinde ters vektör transformu almarak konuşma sinyali tekrar oluşturulur [5].

Öncelikle, konuşma sinyal örnekleri bloklara bölünerek her blok bir vektör gibi düşünülmektedir. Oluşturulan vektörlere vektör transformu uygulanarak bilgiler kuantalamaya hazır hale getirilmektedir. Vektör transformunda bilgilerde herhangi bir kayıp söz konusu olmamaktadır. Başlangıçta vektör transformu uygulandığında çıkışta üretilen kod vektörlerine de ters vektör transformu uygulanmalıdır.

x_n ve x_k vektörleri $\frac{N}{2}$ bileşene sahip olmak üzere vektör transformu ve tersi sırasıyla (1) ve (2) nolu denklemler ile tanımlanır [5].

$$x_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n^T W^{nk} \quad (1)$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} x_k^T W^{-nk} \quad (2)$$

(1) ve (2) nolu denklemlerde W skew-circulant matrisinin boyutu $\frac{N}{2} \times \frac{N}{2}$ dir ve (3) nolu denklem ile verilmiştir.

$$W = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -1 & 0 & 0 & \dots & 0 \end{bmatrix}_{\frac{N}{2} \times \frac{N}{2}} \quad (3)$$

Orjinal veri vektörlerinin yerine transform vektörleri kodlanarak sıkıştırma gerçekleştirir.

II.2. Vektör Transform Etki alanında Bit-Tahsisi

Vektör transform etki alanındaki vektörler farklı enerji yoğunluklarına sahiptirler ve bu nedenle kuantalamada vektörler için farklı büyüklükteki farklı kodkitapları kullanılmaktadır. Yüksek enerji yoğunluklarına sahip vektörler daha büyük kodkitapları kullanılarak nicemlenirken, düşük enerji seviyeli vektörler küçük kodkitabı kullanılarak nicemlenmektedir. Vektör transform kodlamasında, kodkitabı büyüklüklerinin tayini ve kodkitabının dizaynı, vektör transform etki alanında vektör nicemleyiciler için önemlidir.

Transform etki alanı vektörleri için doğrudan bitleri tayin etmek üzere "vektör bit-tahsis etme algoritması" kullanılmaktadır [5]. Aşağıdaki eşitliklerde kullanılan D_k , k ncı transform vektörü

x_k 'nın nicemlenmesi sonucunda oluşan ortalama karesel bozulmayı ifade etmektedir.

$$D_k = A e^{-\frac{4}{N} b_k}, \quad k = 0, 1, 2, \dots, N-1 \quad (4)$$

(4) nolu denklemde kullanılan A sabittir. Vektör transform etki alanı üzerindeki ortalama bozulma (4) nolu denklem kullanılarak (5) nolu denklem ile verilebilir. σ_k^2 , x_k 'nin varyansdır.

$$D = \sum_{k=0}^{N-1} \frac{\sigma_k^2}{N} D_k = \frac{A}{N} \sum_{k=0}^{N-1} \sigma_k^2 e^{-\frac{4}{N} b_k} \quad (5)$$

Vektör Transform etki alanındaki bit tahsisinde, her k ncı vektör için tahsis edilen bitler, b_k , (6) nolu denklem ile hesaplanmaktadır. B , konuşma bloğuna ayrılan toplam bit sayısı olmak üzere, $R = \frac{B}{N}$ (bit/boyut) her boyuttaki bitlerin ortalama sayısıdır.

$$b_k = \frac{N}{2} R + \frac{N}{4} \left(\log_2 \sigma_k^2 - \frac{1}{N} \sum_{k=0}^{N-1} \log_2 \sigma_k^2 \right) \quad (6)$$

Hesaplanan bitler tamsayı olmadığından, en yakın tamsayıya yuvarlatılır. Bit-tahsis etme formülündeki ilk terim olan $\frac{N}{2} \times R$, her vektördeki bitlerin ortalama sayısıdır. Parantez içerisindeki terim, varyans değerine bağlı olarak pozitif veya negatiftir. Bit tahsis-etme formülünde üç önemli nokta vardır. Bunlar:

1. b_k genellikle tamsayı değildir.
2. σ_k^2 çok küçük ise, b_k için formülden negatif bulunabilir.

3. σ_k^2 çok büyük ise, b_k büyük bir pozitif tamsayı olabilir [5].

Bu çalışmada aynı zamanda, Weping Li'nin vektörelleştirilmiş LBG algoritması [5] kullanılarak kodkitabı tasarlanmıştır. Böylelikle her vektöre önemi dahilinde bit tahsis edilerek, doğrusal olmayan kodlama yapılmaktadır. Konuşma verilerine Linde-Buzo-Gray (LBG) algoritması [6] uygulanmadan önce vektör transformu uygulanmıştır. Burada, vektör transformu ile daha önemli görülen bazı vektörlerde, enerji yoğunluğunun diğer vektörlere göre daha fazla olması ve LBG algoritması ile yüksek enerji yoğunluğuna sahip vektörlerin, düşük enerji yoğunluğuna sahip vektörlerden daha büyük kodkitabı ile temsil edilmesi amaçlanmaktadır. Vektör transform etki alanındaki vektörler, veri etki alanındaki vektörlere göre birbiriyle daha az ilişkilidir. Enerji, transform etki alanında birkaç vektörde yoğunlaşmaktadır. Böylece, kodkitabı dizaynında, veri vektörlerinin yerine transform vektörleri kodlanarak sıkıştırma gerçekleştirilmektedir.

Yapılan kodlamalar sonunda, tekrar üretilen konuşma sinyalleri için sinyal gürültü oranı (SNR) (7) nolu denklem yardımıyla hesaplanmıştır.

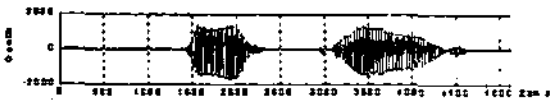
$$SNR(dB) = 10 \log_{10} \frac{\sum_{n=0}^{T-1} x[n]^2}{\sum_{n=0}^{T-1} (x[n] - \hat{x}[n])^2}$$

(7)

$x[n]$ orjinal sinyal, $\hat{x}[n]$ ise tekrar üretilen sinyali temsil etmektedir.

III. YAPILAN SIKIŞTIRMA SONUÇLARI

Bu bölümde, çalışmada sunulan kodlama tekniklerine ait sonuçlar değerlendirilmiş ve karşılaştırılmıştır. Üzerinde çalışılan "sıfır" kelimesine ait, orjinal verilerin oluşturduğu, grafiksel gösterim Şekil 3'te verilmiştir.

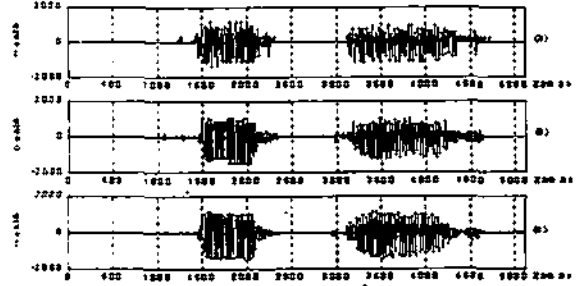


Şekil 3. "sıfır" kelimesine ait konuşma örnekleri.

III.1. Vektör Kuantalama Sonuçları

40 çerçeveden oluşan "sıfır" kelimesi vektör kuantalama metoduyla kodlanmıştır. VQ işlemi ile, 20 elemandan oluşan 256 vektör, 10 elemandan oluşan

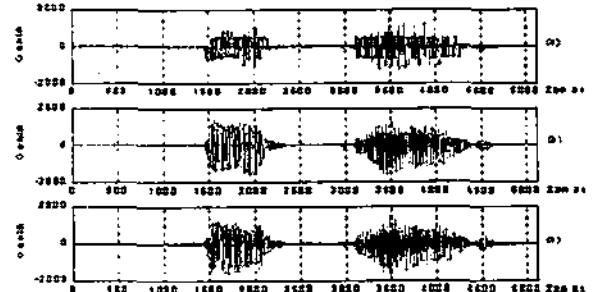
512 vektör ve 5 elemandan oluşan 1024 vektör için, sırasıyla 16, 32 ve 64 vektörden oluşan bir kodkitabı tasarlanarak, 1:16'lık sıkıştırma yapılmıştır. Ayrıca, 2 elemandan oluşan 2560 vektör için 128 vektörden oluşan bir kodkitabı tasarlanarak 1:20'lik sıkıştırma yapılmıştır. Şekil 4.a' da 256 vektör için VQ uygulandıktan sonra tekrar üretilen "sıfır" konuşmasının grafiği verilmiştir. Sinyal gürültü oranı (SNR), 5.3907 olarak hesaplanmıştır. VQ uygulandıktan sonraki konuşma sinyali, orijinal konuşma sinyali ile karşılaştırıldığında sinyal genlik değerlerinin düştüğü görülmüştür. Şekil 4.b' de 512 vektör için VQ uygulandıktan sonra tekrar üretilen "sıfır" konuşmasının grafiği verilmiştir. SNR, 8.7052 olarak hesaplanmıştır. Şekil 4.c' de ise 1024 vektör için VQ uygulandıktan sonra tekrar üretilen "sıfır" konuşmasının grafiği verilmiştir. SNR, 14.1594 olarak hesaplanmıştır.



Şekil 4.(a) 20x256 için yapılan 1:16'lık sıkıştırma, (b) 10x512 için yapılan 1:16'lık sıkıştırma, (c) 5x1024 için yapılan 1:16'lık sıkıştırma.

III.2. Vektörelleştirilmiş LBG Algoritması Sonuçları

Vektörelleştirilmiş LBG algoritması kullanılarak tekrar üretilen konuşma sinyallerinin grafiksel gösterimi Şekil 5' te verilmiştir. Şekil 5.a' da N değeri 4 olduğu durumda, 1:365 sıkıştırma oranı için, SNR 4.5778 olarak hesaplanmıştır. Şekil 5.b' de, N değeri 4 iken 1:40 sıkıştırma oranı için, SNR değeri 14.888 olarak bulunmuştur. Şekil 5.c' de N değeri 8 olduğu durumdaki 1:32' lik sıkıştırma oranı için, SNR değeri 7.6898 olmuştur.



Şekil 5. Vektörelleştirilmiş LBG algoritması

kullanılarak (a) $N=4$ alınarak yapılan 1:365 oranındaki sıkıştırma, (b) $N=4$ alınarak yapılan 1:40 oranındaki sıkıştırma, (c) $N=8$ alınarak yapılan 1:32 oranındaki sıkıştırma.

IV. SONUÇ

Doğrusal kuantalamada, mevcut konuşma sinyallerini içeren vektör sayısı artırıldığında, aynı sıkıştırma oranı için daha büyük SNR değerine ulaşıldığı görülmüştür. LBG algoritması kullanılarak yapılan sıkıştırma işleminde, tekrar üretilen konuşma sinyalinin, VQ kullanılarak gerçekleştirilen aynı sıkıştırma oranına göre daha iyi sonuç verdiği gözlenmiştir. Bunun nedeni, vektörelleştirilmiş LBG algoritması ile orijinal konuşma örneklerinden oluşan vektörlere önemleri dahilinde bit tahsis edilmesi ve kodkitabı oluşturulmasıdır. Oysa ki, vektör kuantalama işlemiyle her vektör için aynı büyüklükte kodkitabı oluşturulmaktadır. Bu sebeple, vektörelleştirilmiş LBG algoritması, VQ işlemine göre bir avantaj sağlamaktadır.

V. KAYNAKÇA

- [1] W. Li and Y. Zhang , "Vector-based signal processing and quantization for image and video compression", *Processings of the IEEE*, Vol.83, No.2, February 1995.
- [2] R.M. Gray , "Vector quantization", *IEEE ASSP Mag.*, Vol.1, pp.4-29, April 1984.
- [3] J. Makhoul, S. Roucos ve H. Gish, "Vector quantization in speech coding", *IEEE*, Vol.73, pp.1551-1558, November 1985.
- [4] W. Li , "On vector transformation", *IEEE transactions on signal processing*, Vol.41, No.11, pp.3114-3126, November 1993.
- [5] W. Li , "Vector transform and image coding". *IEEE transactions on circuits and systems for video tech.*, Vol.1, No.4, pp.297-307, December 1991.
- [6] Y. Linde, A. Buzo ve R.M. Gray, "An algorithm for vector quantizer design", *IEEE transactions on communications*, Vol.com-28, No.1, January 1990.