

**LBG ALGORİTMASINDA
GÖRÜNTÜ İSTATİSTİKLERİNDEN
FAYDALANMA**

Kemal ÖZKAN

**Yüksek Lisans Tezi
Elektrik – Elektronik Mühendisliği
Anabilim Dalı**

2000

**TEC YÜKSEK LİSANS TEZİ
DOKÜMANIYON MERKEZİ**

**USE OF IMAGE STATISTICS
in LBG ALGORITHM**

by
Kemal ÖZKAN

**Master Thesis
Elektrik – Elektronik Engineering
Department**

2000

LBG ALGORİTMASINDA GÖRÜNTÜ İSTATİSTİKLERİNDEN FAYDALANMA

Kemal ÖZKAN

**Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik – Elektronik Anabilim Dalı
Telekomünikasyon Bilim Dalında
Yüksek Lisans Tezi
Olarak Hazırlanmıştır.**

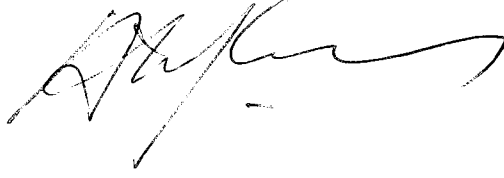
96956

Danışman : Yrd. Doç. Dr. Erol SEKE

Temmuz 2000

Kemal OZKAN'ın YÜKSEK LİSANS tezi olarak hazırladığı “LBG ALGORİTMASINDA GÖRÜNTÜ İSTATİSTİKLERİNDEN FAYDALANMA” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Üye : Y. Doç. Dr. Atıf ÇAY



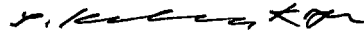
Üye : Y. Doç. Dr. Erol SEKE



Üye : Y. Doç. Dr. Nihat ADAR



Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ..15.09.2000..... gün ve
..2000.14/19.. sayılı kararıyla onaylanmıştır.



Prof. M. Selami KILIÇKAYA
Enstitü Müdürü

ÖZET

Bilinen ve en popüler olan vektör kuantalama algoritması LBG algoritmasıdır. Linde – Buzo – Gray algoritması öz yinelemeli bir yaklaşımdır. Bundan dolayı gerçekleştirilmesi uzun zaman almaktadır.

Görüntü dizilerinin çerçeveleri arasında yüksek oranda benzerlik vardır. Bu çalışma da, görüntü dizilerindeki bu benzerlikten yararlanarak her çerçeve için kod kitabı bulmak yerine ilk çerçeveden elde edilen kod kitabını kullanarak ardından gelen çerçeveler için bu kod kitabı kullanılmıştır. Bunun için görüntü dizisinin ilk çerçevesine LBG algoritması uygulanmış, ardından gelen çerçeveler içinse bu kod kitabı güncellenmiştir. Güncelleme esnasında, yeni çerçevedeki eğitim vektörleri en yakın sınıfa konmuş ve her kod sınıfındaki eğitim vektör sayıları hesaplanmıştır. Yeni kod kitabındaki kod vektörlerden sıfır eğitim vektörüne sahip kod vektörü varsa, o kod vektörün yerine sınıflardaki en uzak vektörler yerleştirilerek toplam hata azaltılmaya çalışılmıştır.

Güncelleme algoritması kullanıldığı zaman yaklaşık olarak %1 'in altında bozulma olmasına karşılık 3.5:1 oranı civarında zamandan kazanılmıştır.

Anahtar Sözcükler

Vektör kuantalama

Kod kitabı

Kod vektör

LBG algoritması

Güncelleme algoritması

SUMMARY

One of the most popular (and well known) vector quantization algorithm is the LBG algorithm. The Linde-Buzo-Gray algorithm is an iterative process. Implementation of LBG algorithm is more complex, then it takes too much time.

There are highly correlation between the frames of the image sequences. In this research, instead of finding a new code book for each frames, the code book which founded from the first frames is used for the consecutive frames by using correlation between frames. For this, the LBG algorithm is applied the first frame, and this code book updated for consecutive frames. During the updating, additional training vectors in new frame is place the nearest code vector in the first code book and number of training vectors are calculated in each code vector. If any code vector has zero training vector, the total error is try to minimise by replacing code vector which has zero training vector the farthest vectors in the code vectors in the code book.

By using the updating algorithm, applied time is decreasing ratio of 3.5:1 with about %1 distortion.

Key Words

Vector quantization

Code vector

Code book

LBG algorithm

Updating algorithm

TEŐEKKÜR

Bu alıřmada yardımını ve desteęini her zaman hissettięim danıřman hocam Yrd. Do. Dr. Erol SEKE ' ye en iten teŐekkürlerimi sunarım. Bu alıřma süresince evi benimle paylaşan ve ayrıca iř arkadaşlarım olan Ahmet YAZICI ve İsmail UZUN ' a ve beni bugünlere getiren aileme teŐekkürü bir bor bilirim.



İÇİNDEKİLER DİZİNİ

Sayfa

ÖZET	iv
SUMMARY.....	v
TEŞEKKÜR	vi
ŞEKİLLER DİZİNİ	ix
TABLolar DİZİNİ	xii
1 GİRİŞ	1
2 SİNYAL, KODLAMA, SIKIŞTIRMA	5
2.1 Temel Terminoloji	5
2.2 Sayısal Görüntünün Gösterilmesi	5
2.3 Basit bir Görüntü Modeli	6
2.4 Niçin Sayısal	7
2.5 Örnekleme ve Kuantalama	7
3 İMGE SIKIŞTIRMA	9
3.1 Doğruluk Kriteri	12
3.2 Sıkıştırma Modeli	12
3.3 Kayıpsız Sıkıştırma Modeli	15
3.3.1 Huffman kodlama	17
3.3.2 Run-Length kodlama	17
3.3.3 Lempel-Ziv-Welch kodlama	17
3.3.4 Aritmetik kodlama	17
3.4 Kayıplı Sıkıştırma Modeli	18
3.4.1 Gri seviyeli run-length kodlama	18
3.4.2 Blok esrime kodlama	18
3.4.3 Vektör kuantalama	18
3.4.4 Farkları öngörücü kodlama	19
3.4.5 Dönüşüm kodlama	19
3.4.5.1 Kesikli kosinüs dönüşümü	20

İÇİNDEKİLER DİZİNİ (Devamı)

Sayfa

3.4.6 JPEG	21
3.4.7 MPEG	23
4 SKALAR KUANTALAMA	26
4.1.1 Tek boyutlu kuantalayıcı	28
4.1.2 Merdiven kuantalayıcı	28
4.2 Skalar Kuantalayıcı Dizaynı	28
5 VEKTÖR KUANTALAMA	31
5.1 Vektör Kuantalamanın Tanımı	32
5.2 Bozuşma Ölçümleri	35
5.3 Vektör Kuantalama için Optimum Olma Şartları	36
5.3.1 En yakın komşu şartı	37
5.3.2 Merkezleme şartı	38
5.4 Vektör Kuantalama Dizaynı	40
5.5 Vektör Kuantalama Çeşitleri	42
5.5.1 Ağaç yapılı vektör kuantalama	42
5.5.2 Sınıflandırılmış vektör kuantalama	42
5.5.3 Dönüşüm vektör kuantalama	44
5.5.4 Bölümlemeli vektör kuantalama	45
5.5.5 Ortalaması çıkartılmış vektör kuantalama	46
5.5.6 Öngörücülü vektör kuantalama	47
5.5.7 Adres vektör kuantalama	48
6. LBG ALGORİTMASI	49
6.1 Ayırma İşlemi	53
6.2 LBG Algoritmasını Görüntü Dizileri için Güncellenmesi	56
7. UYGULAMA	59
8. SONUÇLAR	104
KAYNAKLAR DİZİNİ	112

ŞEKİLLER DİZİNİ

Sayfa

Şekil 3.1 Sıkıştırma katı	12
Şekil 3.2 Sıkıştırmayı çözücü katı	13
Şekil 3.3 Sıkıştırıcı katının ayrıntıları	14
Şekil 3.4 Sıkıştırmayı çözücü katının ayrıntıları	15
Şekil 3.5 JPEG kodlayıcı	23
Şekil 3.6 JPEG kod çözücü	23
Şekil 3.7 MPEG bloklama yapısı	24
Şekil 3.8 MPEG kodlayıcı ve kod çözücü blok yapısı	25
Şekil 4.1 Tek boyutlu kuantalayıcı	28
Şekil 4.2 Merdiven kuantalayıcı	29
Şekil 4.3 Lloyd algoritmasının akış diyagramı	30
Şekil 5.1 Kuantalama hücre örneği	33
Şekil 5.2 Vektör kuantalama kodlayıcı	34
Şekil 5.3 Vektör kuantalama kod çözücü	34
Şekil 5.4 Vektör kuantalama örneği	39
Şekil 5.5 Vektör kuantalama örneği	40
Şekil 5.6 Ağaç yapılı vektör kuantalama kodlayıcının temel yapısı	43
Şekil 5.7 Sınıflandırılmış vektör kuantalama	44
Şekil 5.8 Dönüşüm vektör kuantalama	45
Şekil 5.9 Ortalaması çıkartılmış vektör kuantalama	47
Şekil 6.1 LBG algoritması akış diyagramı	52
Şekil 6.2 Ayırma örneği	53
Şekil 6.3 Güncelleme algoritması	58
Şekil 7.1.a Orijinal görüntü (Lena 512x512)	60
Şekil 7.1.b LBG algoritması sonucu	60
Şekil 7.2.a Orijinal görüntü (Uçak 256x256)	61
Şekil 7.2.b LBG algoritması sonucu	61
Şekil 7.3.a Orijinal görüntü (Tank 512x512)	61

Şekil 7.3.b LBG algoritması sonucu	61
Şekil 7.4.a Orijinal görüntü (savaş uçağı 512x512)	62
Şekil 7.4.b LBG algoritması sonucu	62
Şekil 7.5.a Orijinal görüntü (Boat 512x512)	62
Şekil 7.5.b LBG algoritması sonucu	62
Şekil 7.6.a Orijinal görüntü (Elaina 512x512)	63
Şekil 7.6.b LBG algoritması sonucu	63
Şekil 7.7.a Orijinal lena omuzu.....	63
Şekil 7.7.b LBG algoritması sonucu	63
Şekil 7.8.a Orijinal lena gözü.....	64
Şekil 7.8.b LBG algoritması sonucu	64
Şekil 7.9.a Orijinal 'boat'ın orta kısmı	64
Şekil 7.9.b LBG algoritması sonucu	64
Şekil 7.10.a 1.çerçeve orijinal görüntü (Tenis).....	71
Şekil 7.10.b LBG algoritması sonucu	71
Şekil 7.10.c Güncelleme algoritması sonucu	71
Şekil 7.11.a 10.çerçeve orijinal görüntü (Tenis).....	71
Şekil 7.11.b LBG algoritması sonucu	71
Şekil 7.11.c Güncelleme algoritması sonucu	71
Şekil 7.12.a 20.çerçeve orijinal görüntü (Tenis).....	72
Şekil 7.12.b LBG algoritması sonucu	72
Şekil 7.12.c Güncelleme algoritması sonucu	72
Şekil 7.13.a 30.çerçeve orijinal görüntü (Tenis).....	72
Şekil 7.13.b LBG algoritması sonucu	72
Şekil 7.13.c Güncelleme algoritması sonucu	72
Şekil 7.14.a 39.çerçeve orijinal görüntü (Tenis).....	73
Şekil 7.14.b LBG algoritması sonucu	73
Şekil 7.14.c Güncelleme algoritması sonucu	73
Şekil 7.15.a 1.çerçeve orijinal görüntü (Futbol).....	83
Şekil 7.15.b LBG algoritması sonucu	83
Şekil 7.15.c Güncelleme algoritması sonucu	83
Şekil 7.16.a 15.çerçeve orijinal görüntü (Futbol).....	83
Şekil 7.16.b LBG algoritması sonucu	83
Şekil 7.16.c Güncelleme algoritması sonucu	83

Şekil 7.17.a 30.çerçeve orijinal görüntü (Futbol).....	84
Şekil 7.17.b LBG algoritması sonucu	84
Şekil 7.17.c Güncelleme algoritması sonucu	84
Şekil 7.18.a 45.çerçeve orijinal görüntü (Futbol).....	84
Şekil 7.18.b LBG algoritması sonucu	84
Şekil 7.18.c Güncelleme algoritması sonucu	84
Şekil 7.19.a 15.çerçeve orijinal görüntü (Futbol).....	85
Şekil 7.19.b LBG algoritması sonucu	85
Şekil 7.19.c Güncelleme algoritması sonucu	85
Şekil 7.20.a 1.çerçeve orijinal görüntü (Bahçe).....	100
Şekil 7.20.b LBG algoritması sonucu	100
Şekil 7.20.c Güncelleme algoritması sonucu	100
Şekil 7.21.a 25.çerçeve orijinal görüntü (Bahçe).....	100
Şekil 7.21.b LBG algoritması sonucu	100
Şekil 7.21.c Güncelleme algoritması sonucu	100
Şekil 7.22.a 50.çerçeve orijinal görüntü (Bahçe).....	101
Şekil 7.22.b LBG algoritması sonucu	101
Şekil 7.22.c Güncelleme algoritması sonucu	101
Şekil 7.23.a 75.çerçeve orijinal görüntü (Bahçe).....	101
Şekil 7.23.b LBG algoritması sonucu	101
Şekil 7.23.c Güncelleme algoritması sonucu	101
Şekil 7.24.a 99.çerçeve orijinal görüntü (Bahçe).....	102
Şekil 7.24.b LBG algoritması sonucu	102
Şekil 7.24.c Güncelleme algoritması sonucu	102

TABLolar DİZİNİ

Sayfa

Tablo 3.1 Kayıplı ve kayıpsız sıkıştırmanın karşılaştırılması	11
Tablo 7.1 Tenis görüntü dizisinin sinyal gürültü oranları	65
Tablo 7.2 Futbol görüntü dizisinin sinyal gürültü oranları	75
Tablo 7.3 Bahçe görüntü dizisinin sinyal gürültü oranları	87
Tablo 8.1 Görüntü dizilerinde bulunan toplam sinyal gürültü oranı	104
Tablo 8.1 Görüntü dizilerinde bulunan ortalama sinyal gürültü oranı	104
Tablo 8.1 Görüntü dizilerinin gerçekleştirme zamanları	105



BÖLÜM 1

GİRİŞ

Sayısal görüntü işleme son yıllarda önemi giderek artmakta ve buna paralel olarak da gelişmekte olan bir teknolojidir. Her ne kadar yer bilim ve hava fotoğraflarının yorumu çok öncelere gitmekteyse de, uydu görüntüleri, yoğun bilgisayar kullanımı ile sivil alanlara yönelik yaygın çalışmaların gelişmesi 1970' li yıllardan sonrasına dayanmaktadır. Özellikle uzaktan algılama teknolojisinin gelişmesi büyük ölçüde bilgisayar alanındaki "sayısal görüntü işleme" (digital image processing) tekniklerin gelişmesine bağlı olmuştur.

Sayısal görüntü işleme, görüntü verilerinin bilgisayar kullanarak işlenmesi anlamına geldiği için literatürde bilgisayarla görüntü işleme (computer image processing) diye de adlandırılır. Görüntülerin sayısal işlenmesi için genel amaçlı bilgisayarlar yanında dizi işlemci (array processor) ve gösterge işlemci (display processor) denilen özel sayısal elektronik cihazlarda geniş ölçüde kullanılmakta hatta gerekli olmaktadır.

Sayısal görüntü işleme metotlarının iki temel uygulama alanı vardır. Birincisi görüntü bilgilerinin gözle yoruma daha uygun hale getirilmesi, ikincisi ise bu bilgilerin makineler tarafından algılanıp çeşitli işleri otomatik yapmak için kullanılmasını sağlamaktır.

İlk uygulama alanı olarak tıpi örnek verebiliriz. Tıpta, x-ışınlarının veya diğer biyomedikal görüntülerinin gri tonlarını veya renk ışık seviyelerini bilgisayar işlemlerine sokarak iyileştirilmekte ve bu bilgilerin daha anlaşılır hale getirilmektedir. Jeologlar ise benzer teknikleri kullanarak havadan veya uzaydan çekilmiş görüntüler üzerinde çalışırlar. Görüntü pekiştirme ve onarımı işlemleri tekrar görüntüsü elde edilemeyen objeler için veya deneyin tekrar edilmesinin pahalı olacağı görüntülerde çok kullanılır. Fizik veya ona yakın alanlarda ise bilgisayar teknikleri yüksek enerjili plazma ve elektron mikroskobu deneylerinden elde edilmiş görüntülerinin iyileştirilmesi için kullanılır. Görüntü işlemlerinin

yaygın uygulama alanları içinde astronomi, biyoloji, nükleer tıp, savunma sanayi ve endüstri uygulamaları sayılabilir.

İkinci uygulama alanı olarak yukarıda da belirtildiği gibi görüntü işleme teknikleri bilgilerin makineler tarafından algılanıp çeşitli işleri otomatik yapmak için kullanılmasını sağlamaktır. Görüntü bilgilerinin makineler tarafından algılanması problemlerine ise otomatik harf tanıma, ürünlerin toplanması ve kontrolünün otomatik makinelerle yapılması, otomatik parmak ve yüz tanınması, kan örneklerinin incelenmesi örnek gösterilebilir. Her iki uygulama alanı için örneklerin artırılması mümkündür.

Eğlence endüstrisinde görüntü işleme özel efekt yapmak, suni görüntüler üretmek ve bilgisayar animasyonları için kullanılır. Görüntü işleme insanların dış görünüşlerini değiştirmek istediklerinde de kullanılmaya başlanmıştır. Mesela yeni saç stili, yeni bir çift gözlük, veya yeni bir burun istediği zaman bunu gerçekleştirmeden yeni stillerini bilgisayar ortamında görebilmektedirler. Görüntü işleme ve bilgisayar grafiklerini kullanarak yeni bir bina veya uzay aracı yapılabilmekte ve insanların bunların içine girip çıkmaları, gezmeleri sağlanabilmektedir. Bu kullanım yetenekleri sayesinde insanlar evlerinin dekorasyonlarını zaman harcamadan bilgisayar karşısında düzenleyebilmektedirler. Gelecekte sanal ortamların ortaya çıkacağı ve insan hayatını bizim şu an düşünemeyeceğimiz kadar etkileyeceği kesindir. Çünkü bu çalışmaların bir sınırı yoktur.

Bugünkü endüstri ve araştırmalarda görüntülerin saklanma ve iletim ihtiyacı giderek artmaktadır. Bugünlerde popüler olan sayısal televizyon bunun en güzel örneğidir. Sayısal TV yayınlarının gerçekleştirilmesi, iletimi ve saklanması için çok büyük bit oranları gerekmektedir. Şöyle ki, 1000 x 1000 boyutlarında bir çerçevelik TV görüntüsünü sayısallaştırdığımız zaman $1000 \times 1000 = 10^6$ görüntü elemanı (image elements, pixels veya pels) oluşur. Ayrıca her pikselin renk / şiddet bilgileri (color / intensity information) içinde 24 bit ayırdığımızı düşünelim. Böyle bir veriyi iletebilmek için 1 / 30 saniyede 2.4×10^7 bit veya bir saniyede 7.2×10^8 bit, buna bağlı olarak 1.44×10^9 Hz'lik bir bant aralığı gerekmektedir. Eğer sinyal sıkıştırma tekniklerini (signal compression techniques) 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 bit miktarını kabul edilebilir bir düzeye indirgememiz gerekmektedir. Bu iletim oranını kabul edilebilir düzeye getirirken çerçeve başına düşen piksel sayısını, saniyedeki çerçeve sayısını veya renk başına düşen bit sayısını azaltmayı istemeyiz. Çünkü bizim amacımız resmin kalitesini düşürmek değil aynı kalitedeki resmi çeşitli sıkıştırma teknikleri kullanarak daha az bit ile temsil etmektir.

Gerçek hayattaki bir resimde birbirine yakın olan pikseller arasında çok büyük bir oranda benzerlik veya tekrarlılık vardır. Bant genişliğini azaltabilmek için bu tekrarlılıktan mümkün olduğu kadar faydalanılabilir. Görüntü sıkıştırmada genel olarak bu özellik kullanılır. Ayrıca gerçek hayattaki görüntü dizilerinde bir önceki çerçeve ile onu takip eden bir sonraki çerçeve arasında da tıpkı birbirlerine yakın olan pikseller arasındaki gibi benzerlik, tekrarlık vardır. Yani birbirlerini takip eden çerçeveler arasında çok az bir değişim gözlenir. Hareketli görüntü sıkıştırmada çerçeveler arasındaki bu tekrarlılıktan da faydalanılmaktadır.

Bu çalışmada veri sıkıştırmada kullanılan yöntemlerden biri olan vektör kuantalama ve vektör kuantalamada en popüler olan LBG algoritmasının geliştirilmesine çalışılmıştır. Bilindiği üzere optimum kuantalama seviyesinin belirlenmesi için LBG algoritması kullanılmaktadır. Ancak LBG algoritması öz yinelemeli bir yaklaşımdır. Bundan dolayı gerçekleştirmesi uzun zaman almaktadır. Bu çalışmada bu zaman azaltılmaya çalışılmıştır. Görüntü dizilerinin ayrı ayrı her çerçevesine LBG algoritması uygulanarak bir sıkıştırma işlemi gerçekleştirilir. LBG algoritması başlangıç kod kitabı ile başlar ve kod kitabını öz yinelemeli olarak düzenleyerek optimum sonucu bulmaya çalışır. LBG algoritmasında bütün görüntü vektörlerini tek bir kod vektör ile temsil edilerek başlanır ve belirlenen kod vektör sayısına ulaşıncaya kadar devam eder. Başlangıçta bütün eğitim dizileri tek bir sınıfta olduğu düşünülür. Her bir öz yinelemede sınıf sayısı ve kod vektör sayısı ikiye katlanır ve eğitim vektörleri hangi sınıfa ait olduğu ve sınıfları temsil edecek kod vektörleri ise o sınıfın ağırlık merkezi olarak hesaplanır. Tekrardan eğitim vektörleri hangi kod vektöre yakınsa o kod vektörün temsil ettiği sınıfa konular ve tekrar sınıfların ağırlık merkezleri hesaplanarak kod vektörlerinin uzaydaki yerleri değiştirilir. Bu işlem herhangi bir eğitim vektörünün sınıf değiştirmemesine kadar devam eder. Daha öncede 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 çerçeveden elde edilen kod kitabındaki vektörlerin uzaydaki yeri ikinci çerçeve için hesaplandığında çok az yer değişmesi beklenmektedir. Bu tez çalışmasında ise görüntü dizisinin ilk çerçevesine klasik LBG algoritması uygulanmış, ardından gelen çerçeveler ile ilk çerçeve arasında bu tekrarlıktan faydalanarak tekrar kod kitabı (kodbook) oluşturmak yerine ilk kod kitabı güncellemeye çalışılmıştır. Çünkü tekrar kod kitabı oluşturmak yukarıda da belirtildiği gibi çok zaman almaktadır.

İkinci bölümde görüntünün ne olduğunu ve tarihçesinden bahsedilmiştir. Örnekleme (sampling) ve kuantalamanın (quantization) nasıl yapıldığı, aralarında nasıl bir ilişki olduğunu ve niçin sayısallaştırmaya gerek duyulduğu anlatılmıştır.

Üçüncü bölümde ise veri sıkıştırmanın temelleri verilmiş, kayıpsız (lossless) ve kayıplı (lossy) sıkıştırma örnekler verilerek arasındaki farklar anlatılmaya çalışılmıştır.

Dördüncü bölümde ise kuantalama, özellikle skalar kuantalamanın ne olduğu ve Shanon'un oran bozulma kuramından bahsedilmiştir.

Beşinci bölümde ise vektör kuantalama hakkında ayrıntılı bilgi verilmiş, şu ana kadar geliştirilmiş olan yöntemler sırayla anlatılmıştır.

Altıncı ve son bölümde ise tez çalışmasını büyük bölümünü alan LBG algoritması ayrıntılı bir şekilde anlatılmış, ve önerilen yöntemle klasik LBG algoritmasını sonuçları deneysel olarak karşılaştırılmıştır.

BÖLÜM 2

SİNYAL, KODLAMA VE SIKIŞTIRMA

2.1) Temel Terminoloji

Sinyal kelimesinin anlamı genellikle kesintisiz zaman, kesintisiz genlikte (continuous time, continuous amplitude) bir dalgadır. Genel anlamda sinyal ise zamana bağımlı (zaman kesintili veya kesintisiz olabilir) genliği veya değeri kesintili veya kesintisiz (continuous or discrete) olan bir fonksiyon, bir skalar veya vektördür. Yani sinyal herhangi bir anda değeri gerçel bir sayı veya gerçel bir vektör olan bir dizi veya dalgadır. Bazen sinyal değeri zamana değil de iki koordinata bağımlı bir resim veya görüntü veya hareketli görüntülerde olduğu gibi hem zamana hem de iki koordinata bağlı olabilir. Veriyi ise sinyalle benzer anlamda kullanırız, ancak veri kesintili zaman sinyallerinde görürüz. Sinyal herhangi bir kaynaktan elde edilebilir; ses olarak bir müzik enstrümanından, insanın ses tellerinden, termokopullardan elde edilen elektrik dalgasından, mikrofön çıkışından, bir modemden, bir bilgisayardan, bir kameradan vs.

Kodlama elimizde kaynak olarak bulunan sembol veya sembol gruplarını yeni sembol veya sembol gruplarına dönüştürme işleminden başka bir şey değildir. İşte bu yeni oluşturulan sembollere kod veya kod sistemi adı verilir.

2.2) Sayısal Görüntünün Gösterilmesi

Görüntü kelimesi iki boyutlu ışık yoğunluk fonksiyonu, $f(x, y)$, (light intensity function) ile ifade edilir. Burada x ve y koordinat eksenlerini, f değeri ise x ve y koordinatındaki parlaklık veya gri seviyeleri (brightness, gray level) göstermektedir. Sayısal görüntü ise bir görüntünün hem koordinatının hem de parlaklık seviyesinin sayılaştırılması ile elde edilir. Sayısal görüntü bir matris şeklinde ifade edilir. Satır ve sütun numaraları görüntü

noktalarının koordinatlarını, matris eleman değerleri ise o noktadaki piksel değerini (görüntü elemanı, resim elemanı veya pel) gösterir. Sayısal görüntünün boyutlarının, uygulama alanlarına göre çeşitlilik gösterse de 2' nin katları şeklinde seçilmesinin avantajları vardır.

2.3) Basit bir Görüntü Modeli

Görüntünün iki boyutlu ışık yoğunluk fonksiyonu, $f(x, y)$, ile ifade edildiğini daha önce belirtmiştik. Işık bir enerji formudur ve sıfırdan büyük ve sınırlı değerler alır. Yani ;

$$0 < f(x, y) < \infty, \quad (2.1)$$

Günlük yaşamda insanlar objelerden yansıyan ışıkları görüntü diye algılar. Doğal olarak $f(x, y)$ iki bileşen ile karakterize edilir; ışık kaynağının şiddeti ve yansıma katsayısı. Bunlar aydınlanma (illumination)ve yansıma (reflectance) diye adlandırılır ve kısaca $i(x, y)$ ve $r(x, y)$ ile gösterilir. Bir noktadaki ışık yoğunluk fonksiyonu ise o noktadaki aydınlanma ve yansıma katsayıların çarpımı ile bulunur.

$$f(x, y) = i(x, y) * r(x, y) \quad (2.2)$$

Burada $0 < i(x) < \infty$,
 $0 < r(x) < 1$ dir.

Monokrom bir görüntünün (x, y) noktasındaki ışık yoğunluğu o noktadaki gri seviyesi, L , (gray level) ile ifade edilir. $L_{min} = i_{min} r_{min}$ ve $L_{max} = i_{max} r_{max}$ dir. L_{min} ile L_{max} arasındaki değerlere ise gri ölçekleri (gray scale) denir. Pratikte ise L_{min} sıfıra kaydırılır ve bu aralık $[0, L]$ arasında olur. Buradaki sıfır siyah renge, L ise beyaz renge karşılık gelir. Aradaki değerler ise siyahtan beyaza doğru gri seviyeleri gösterir.

2.4) Niçin Sayısal

Bir sinyali sayısal olarak işlemek istiyorsak, ilk önce analog sinyalden sayısal sinyale dönüştürmemiz gerekmektedir. Sinyali işledikten sonra tekrar gerçek hayata yani analog sinyale çevirmemiz gerekir. Çünkü gerçek hayat analogtur. Sayısal işlemenin daha mı iyi olduğu sorusuna kesin bir cevap vermek mümkün değildir. Bazı uygulamalar analog dizaynları bazıları ise sayısal dizaynları gerektirmektedir. Buna karşın sinyal işleme uygulamalarında sayısal işlemenin analog işlemeye göre avantajları vardır. Bu avantajları aşağıdaki sıralayabiliriz:

- Programlanabilmesi
- Kararlı olması
- Özel uygulamalar

2.5) Örnekleme ve Kuantalama

Bilgisayar işlemleri yapabilmek için, görüntü fonksiyonunu mutlaka sayısalaya çevirmek gerekir. Koordinatlarını sayısalaya çevirme işlemine görüntü örnekleme, değerlerinin sayısalaya çevrilme işlemine de gri seviyeye kuantalama adı verilir. Kesintisiz bir görüntüyü yaklaşık eşit aralıklı parçalara bölerek $M \times N$ boyutlarında bir matris formunda düzenlendiğini kabul edelim. Bu matrisin her elemanı sayısal bir değerdir ve o noktadaki ışıklandırma seviyesine karşılık gelmektedir.

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdot & \cdot & \cdot & f(0,M-1) \\ f(1,0) & f(1,1) & \cdot & \cdot & \cdot & f(1,M-1) \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ f(N-1,0) & f(N-1,1) & \cdot & \cdot & \cdot & f(N-1,M-1) \end{bmatrix}$$

Sayısalılaştırma işleminde M, N değerleri ile her piksel için ne kadar gri seviyesi ayrılması gerektiği önemlidir. Pratikte kullanılan ikinin katları şeklindedir, yani ;

$$N = 2^n , M = 2^k , \text{ ve } G = 2^m$$

Burada G gri seviye sayısıdır. Gri seviyeleri daha öncede belirlediğimiz gibi 0 ile L arasındadır. Bir resmi saklamak için gerekli bit sayısı ise

$$b = N \times M \times m \quad (2.3)$$

formülü ile bulunur. Örneğin 128 x 128 boyutlarındaki bir görüntüyü 64 gri seviyesi ile göstermek istersek 98.304 bit gerekmektedir. İyi bir görüntü için ne kadarlık örnek ve gri seviyesi ayırmamız gerektiği önemlidir. Çünkü görüntünün kalitesi ise bu iki parametreye bağlıdır. Aslında iyi bir görüntünün tanımını yapmak zordur, çünkü uygulama alanlarına göre çeşitlilik göstermektedir. Gri seviye sayısını sabit tutup örnek sayısını artırırsak görüntü çözünürlüğü artar. Eğer örnek sayısını sabit tutup gri seviyesi sayısını artırırsak görüntünün izlenebilirlik kalitesi artar.

BÖLÜM 3

İMGE SIKIŞTIRMA

İmge sıkıştırma görüntü işlemenin en önemli alanlarından biridir. İmge sıkıştırma bilgisayarların gelişmesi, çoklu medya uygulamaların giderek artması, World Wide Web'in ortaya çıkması, internetin herkes tarafından kullanılma isteğinin artmasına bağlı olarak önemi giderek artmaktadır. Bunlara ek olarak gelişen video teknolojisi yeni, daha iyi, ve daha hızlı görüntü sıkıştırma algoritmalarının geliştirilme ihtiyacını ortaya çıkarmıştır. Sıkıştırma algoritmaları iki boyutlu (2-D) durağan resimlerle (still) başlamıştır. Çünkü video ve TV görüntüleri çerçevelere bölünerek 2-D' li görüntü verilerine çevrilebilir. Durağan resmin sıkıştırılması alanındaki gelişmeler aynı zamanda hareketli resimlerinde (video / motion picture) sıkıştırılması uygulamaların geliştirilmesine yol açar. Bundan dolayı şu an görüntü sıkıştırma algoritmaları iki boyutlu tek çerçevelik görüntüler üzerinde yoğunlaşırlar.

Görüntü sıkıştırma görüntü dosyalarındaki önemsiz bilgileri atarak boyutunu küçültmektir. Boyutu azaltılmış dosyaya sıkıştırılmış dosya (compressed file), bu dosyayı kullanarak geri elde edilmiş dosyaya yeni elde edilmiş dosya (decompressed file) adı verilir. Herhangi bir sıkıştırma algoritması uygulanmamış orijinal görüntüye ise sıkıştırılmamış (uncompressed file) denir. Orijinal, sıkıştırılmamış görüntü, ile sıkıştırılmış görüntü arasındaki orana da sıkıştırma oranı adı verilir. Sıkıştırma oranı aşağıdaki gibi formüle edilir :

$$\text{Sıkıştırma oranı} = \frac{\text{Orijinal Görüntü Büyüklüğü}}{\text{Sıkıştırılmış Görüntü Büyüklüğü}} \quad (3.1)$$

Genellikle *Büyükük* U : *Büyükük* C şeklinde yazılır. Diğer bir gösteri şekli ise piksel başına düşen bit (bits per pixel) sayıdır. $N \times N$ boyutlarındaki bir görüntüde

$$\text{Piksel başına düşen bit sayısı} = \frac{\text{Bit Sayısı}}{\text{Piksel Sayısı}} = \frac{(8)\text{Bayt Sayısı}}{N \times N} \quad (3.2)$$

Veri sıkıştırma terminolojide piksel veya çerçeveler arasındaki tekrarlıkları atıp sadece önemli olan bilgilerin saklanması ve verinin sayılı bir kısmının iletilmesi şeklinde tanımlanır. Şu anki iletim sistemlerinin bant genişliği ve verilerin bilgisayarda saklanabilmesi için verilerin sıkıştırma işlemi gerçekleştirerek boyutlarının azaltılması gerekmektedir. Başarılı bir sıkıştırmanın anahtarı tanımında da belirtildiği gibi önemsiz bilgilerin atılma olayıdır. İki temel görüntü sıkıştırma yöntemi vardır. İlk yöntemde ,geri elde edilen yöntemde, veri kaybı yoktur. Bundan dolayı bu sıkıştırma yöntemine kayıpsız (lossless) sıkıştırma denir. Kompleks görüntülerde bu yöntemle elde edilen sıkıştırma oranı sınırlı olup genellikle 2:1 ile 3:1 arasındadır. Ancak sadece metinden oluşan görüntülerde kayıpsız sıkıştırma yöntemi daha fazla bir sıkıştırma oranı sağlar. İkinci sıkıştırma yöntemi ise kayıplı (lossy) sıkıştırma olarak isimlendirilir. Çünkü tekrar elde edilen görüntü orijinal görüntünün tamamen aynı değildir. Kompleks görüntülerde sıkıştırma oranı 10 ila 20 arasında değişir. Basit görüntülerde veya düşük kaliteli sonuçlarda sıkıştırma oranı 100 ila 200 arasında olabilmektedir [1, sayfa 239] .

Aşağıdaki tabloda iki sıkıştırma şekli karşılaştırılmıştır.

	Kayıpsız Sıkıştırma	Kayıplı Sıkıştırma
Orijinal görüntünün geri elde etme	Orijinal görüntü geri elde edilir	Orijinal görüntü geri elde edilemez
Sıkıştırma oranı	Sıkıştırma oranı azdır	Sıkıştırma oranı fazladır
İşlem karmaşıklığı	Daha azdır	Daha fazladır
Uygulama alanları	Her veriye	Görüntülere, seslere

Tablo 3.1 : Kayıplı ve kayıpsız sıkıştırmanın karşılaştırılması

Sıkıştırma algoritmalarında görüntülerdeki tekrarlık büyük bir avantajdır. Bir görüntüde, sıkıştırmada faydalanabileceğimiz üç tekrarlık vardır; kodlama tekrarlılığı, pikseller arası tekrarlılık, insan gözünün ürettiği tekrarlılık. Kodlama tekrarlılığı görüntüyü temsil eden verilerin optimum olarak kodlanmaması olarak görülür. Örneğin, biz piksel başına 8 bit kullanırsak 256 gri seviye değerleri elde ederiz, ancak gerçek görüntü yalnız 16 gri seviye sahip olabilir, bundan dolayı optimum kodlama yapmış olmayız. Çünkü bu görüntü için piksel başına 4 bit yeterlidir. Bir görüntüde pikseller arasındaki tekrarlılıkta görülür, çünkü gerçek görüntülerde birbirine yakın olan pikseller arasında yüksek oranda ilinti mevcuttur. Bunun sonucu olarak görüntülerde ışıklandırma seviyeleri birden değişme göstermez. Bunun sonucu olarak gerçek görüntülerde birbirine yakın olan piksel değerleri birbirine çok yakındır. Video veya hareketli resimlerinin çerçeveleri arasında aynı pikseller arasında olduğu gibi büyük bir benzerlik mevcuttur. 3. tekrarlılık ise gözün fiziksel yapısıdır. Bazı bilgiler insan gözü için daha önemlidir. Göz örneğin düşük frekanstaki bilgileri yüksek frekanstaki bilgilere göre daha iyi algılar [1, sayfa 239].

Görüntü sıkıştırma algoritmalarını temel amacının gerekli bilgileri minimuma indirilmesi olduğunu daha önce söylemiştik. Hangi bilgilerin daha önemli olduğunu bulmak için görüntü doğruluğunun tanımının yapılması gerekir.

3.1) Doğruluk Kriteri (Fidelity Criteria)

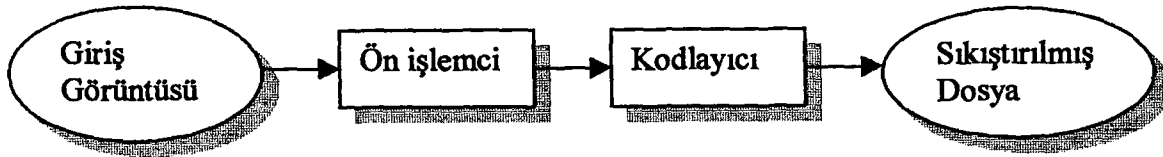
Doğruluk kriteri iki sınıfa ayrılır.

- 1) Nesnel doğruluk kriteri (objective fidelity criteria)
- 2) Öznel doğruluk kriteri (subjective fidelity criteria)

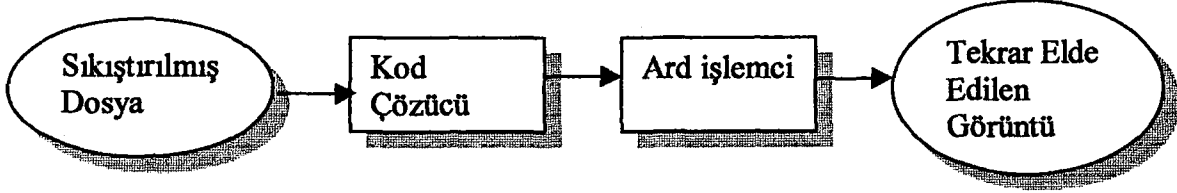
Nesnel doğruluk kriteri, sayısal sinyal işleme ve bilgi teorisinde de olduğu gibi tekrar oluşturulan görüntü ile orijinal görüntü arasında ne kadarlık bir hata olduğunun bulunmasıdır. Nesnel kriter aslında çok kullanılmasına karşın görüntünün kalitesi ile bağlantılı değildir. Bununla birlikte aynı görüntünün iki farklı versiyonunun karşılaştırılmasında göreceli bir ölçüm için kullanılır. Öznel doğruluk kriteri ise görüntü kalite derecesi (qualitative scale) olarak tanımlanır. Bu derecelendirme ise insanlar tarafından yapılan testlerle belirlenir. Öznel ölçümlemede test görüntülerinin ve deney düzeninin çok iyi yapılması gerekir. Öznel testte ise test edilecek görüntülerin bir grup insan tarafından izlenip karar verilmesidir.

3.2) Sıkıştırma Sistem Modeli

Sıkıştırma sistem modeli iki kısımdan oluşur; sıkıştırıcı ve sıkıştırmayı çözücü . Sıkıştırıcı ön işlem katı (preprocessing state) ve kodlayıcı katından (encoding state) oluşur. Sıkıştırmayı çözücü ise kod çözücü katını (decoding state) arka işlemci katı (post processing state) izler [1, sayfa 242-249] .



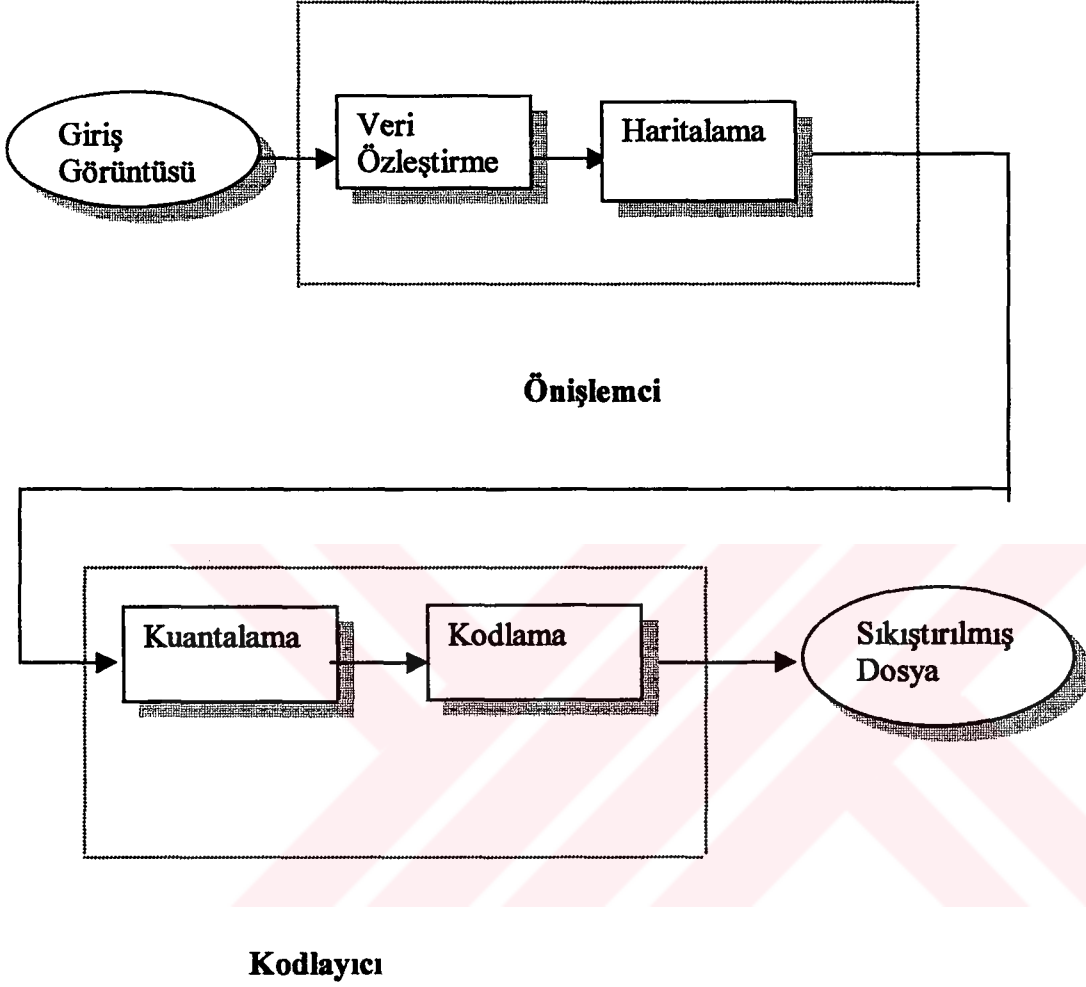
Şekil 3.1) Sıkıştırma katı



Şekil 3.2) Sıkıştırmayı çözme katı

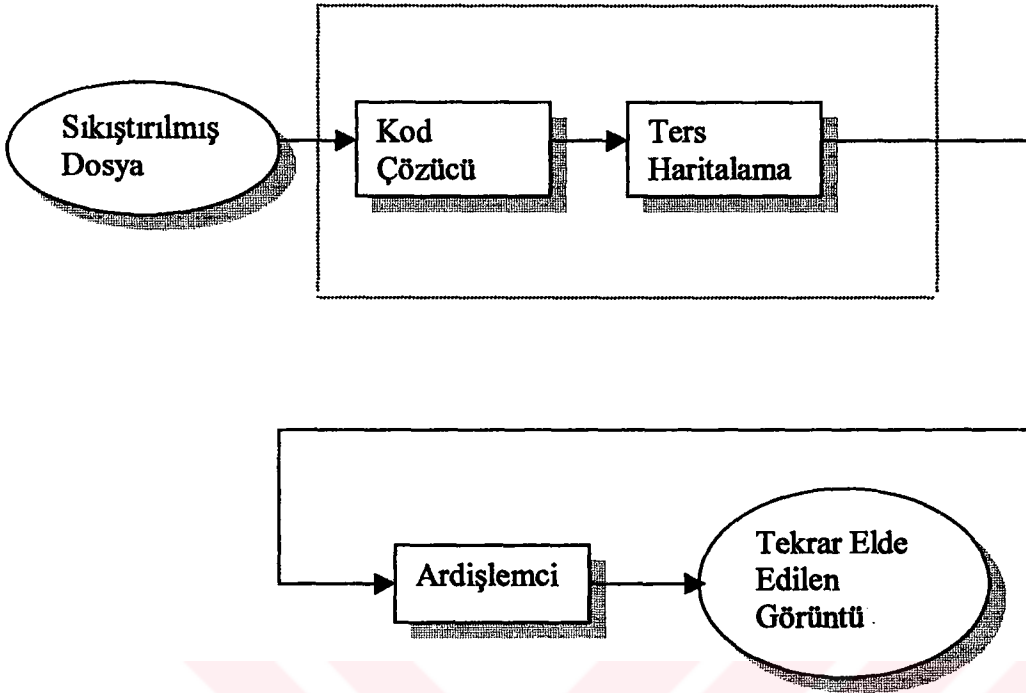
Kodlayıcıdan önce, uygulamalara göre değişmesine rağmen, bir dizi işlemler ile görüntüyü kodlama işlemine hazırlamak gerekir. Sıkıştırılmış görüntünün açılması sırasında ise bu ön işlemler ters şekilde ard işlemler olarak yapılır.

Sıkıştırma kısmı aşağıdaki şekilde görüldüğü gibi kısımlara ayrılır. İlk kat verinin azaltıldığı kattır. Burada görüntü verisinin gri seviyesinin azaltılması ve / veya uzamsal nicemleme (spatial quantization) veya imge pekiştirme (image enhancement) kullanılabilmesi için düzenlenmesi işlemleri yapılır. (örneğin, gürültülerin yok edilmesi) İkinci işlem ise haritalama (mapping) işlemidir. Bu kısımda orijinal görüntü daha iyi bir sıkıştırmanın yapılabildiği başka bir matematiksel boyuta haritalanır. Sonra kodlama işlemlerindeki kuantalama katı ile kesintisiz veriler kesintili verilere çevrilir. Daha sonra kodlamanın yapıldığı kodlama katı gelir. Sıkıştırma algoritmalarında genellikle bütün bu katlar bulunur.



Şekil 3.3) Sıkıştırıcı katının ayrıntıları

Sıkıştırıcıyı çözücü katı da aşağıdaki şekilde görüldüğü gibi çeşitli kısımlara ayrılır. İlk kat, kod çözücü katı, sıkıştırılmış görüntü orijinal görüntüye kod çözme işlemi yardımıyla çevrilir. Sonra bu değerler ters haritalama yöntemiyle orijinal tanım kümesine haritalanır. Son olarak da ardışıklar ile orijinal görüntü elde edilmeye çalışılır.



Kod Çözücü

Şekil 3.4) Sıkıştırmayı çözücü katının ayrıntıları

3.3) Kayıpsız Sıkıştırma Metodu (Lossless Compression Methods)

Kayıpsız sıkıştırma metotları bazı görüntü uygulamaları için gereklidir. Örneğin, medikal görüntülerin herhangi bir bilgi kaybı olmaksızın sıkıştırılması lazımdır. Kayıpsız sıkıştırmanın teorisi haberleşme ve bilişim kuramından gelir. Görüntüdeki ortalama bilginin ölçülmesi gerekir ki buna da entropi (entropy) adı verilir. $N \times N$ boyutlarındaki bir görüntüdeki entropi ise aşağıdaki eşitlik ile bulunur.

$$Entropi = \sum_{p=0}^{L-1} p_i \log_2 (p_i) \quad (3.3)$$

Burada ,

$$P_i : i. gri seviyesinin olasılığı \frac{n_k}{N^2}$$

n_k : k gri seviyesine sahip toplam piksel sayısı

L : gri seviye sayısı (örneğin 8 bit için 256)

Teorik olarak bu ölçüm piksel başına düşen ortalama bit sayısının minimum olması o görüntüye daha iyi bir kodlama yapılabileceğini gösterir. Entropi 0 ile $\log_2(L)$ arasındadır, yani,

$$0 \leq \text{Entropi} \leq \log_2(L) \quad (3.4)$$

Görüntüde çok rasgelelik (more randomness) varsa, yani gri seviyeleri düzgün dağılmamışsa bu görüntüyü temsil etmek için daha fazla bite ihtiyaç vardır. Bu da ayrıca bilginin bağımsız olduğunu, her pikselin birbirine az benzediğini gösterir. Kodlama kuramında bu olay şu şekilde açıklanır. Eğer veriyi daha az bit ile temsil etmek istiyorsak verinin birbirine çok benzemesi gerekir.

Entropi ayrıca kodlamanın performansını ölçmek için kullanılır. Kodlamadaki piksel başına düşen ortalama değer aşağıdaki gibi formüle edilir.

$$L_{ave} = \sum_{p=0}^{L-1} l_i p_i \quad (3.5)$$

Burada ,

l_i = i gri seviyesini kodlamak için kullanılan bit boyutu

p_i = histogram, i. gri seviyesinin olasılığı

L_{ave} ne kadar küçük olursa, o kadar iyi kodlama yapıldığını gösterir [1, sayfa 247-249].

3.3.1) Huffman Kodlama

Gri seviye değerlerinin istatistikleri çıkartılarak Huffman algoritması yardımıyla mümkün olabilen minimum sınıra, (idealde $L_{ave} = Entropi$), sahip kod üretilebilir. Bu metodun sonucu olarak değişken uzunlukta kod (variable length code) elde edilir. Başka bir ifadeyle, kod kelimelerin uzunlukları eşit değildir. Kompleks görüntülerde yalnız Huffman kodlamayı kullanırsak datayı %10 ila %50 (1.1:1 ile 1.5:1) arasında azaltır. Ama basit görüntülerde bu oran 2:1 veya 3:1 olabilmektedir [1, sayfa 249-253], [2, sayfa 343-348], [3, sayfa 34-79] .

3.3.2) Run – Length Kodlama

Run – length kodlama görüntü sıkıştırma metodu aynı gri seviye değerine sahip olan komşu piksellerin sayılmasıyla yapılır. Bu sayma işlemi run-length adı verilir ve bu sayılan değerler kodlanır ve saklanır [1, sayfa 253-257], [2, sayfa 354-358].

3.3.3) Lempel – Ziv – Welch Kodlama

Lempel – Ziv – Welch veri dizgilerini (strings of data) kodlamada kullanır. Görüntüler için bu veri dizgileri arka arkaya gelen piksel değerleridir. Bir tane tablo yapılır. Bu tabloda dizgiler ve onların kodları vardır. Yeni bir dosya okunduğu zaman dizgi tablosu güncellenir. Eğer yeni bir dizgi gelmiş ise ona yeni bir kod üretilir. Eğer yeni dizgi tabloda hali hazırda var ise bu dizginin kodu sıkıştırılmış dosyaya yazılır [1, sayfa 257], [2, bölüm 9] .

3.3.4) Aritmetik Kodlama

Aritmetik kodlamada kod ile piksel değeri arasında direkt bir bağlantı yoktur. Aritmetik kodlamada input verileri 0 ila 1 arasında kayan noktalı (floating point) sayıya transfer edilir. Bu kodlamada verinin olasılık dağılımını (probabily distribution) kullanır, bunda dolayı teorik olarak entropi ile sınırlandırılmış maksimum sıkıştırma oranını sağlar [1, sayfa 258], [2, sayfa 348-349], [3, bölüm 5].

3.4) Kayıplı Sıkıştırma Metodu (Lossy Compression Methods)

Kompleks görüntülerde yüksek bir sıkıştırma oranı istiyorsak kayıplı sıkıştırma metodunu kullanmamız gerekir. Kayıplı sıkıştırma da görüntü kalitesi ile sıkıştırma derecesi arasında ters orantı vardır. Sıkıştırma oranı artıkça görüntü kalitesi düşer. İmge pekiştirme ve onarımı işlemleri ile tekrar elde edilen görüntü kalitesi artırılır. Kayıplı sıkıştırma metotlarında parametreler kullanıcı tarafından ayarlanabilmekte, bu da sıkıştırma oranını ve görüntünün kalitesini değiştirmektedir. Basit görüntülerde yüksek bir sıkıştırma oranı elde edilse de bu oran görüntüye bağlıdır. Kayıplı sıkıştırma hem uzamsal hem de dönüşüm alanında (spatial / transform domain) yapılabilir.

3.4.1) Gri Seviyeli Run Length Kodlama(Gray Level Run Length Coding)

Run length kodlama tekniği kayıpsız sıkıştırmada olduğu gibi kayıplı sıkıştırmada da kullanılır. Kayıplı sıkıştırmada gri seviye sayısını azaltıp kayıpsız sıkıştırmada ki kullanılan tekniğin aynısı uygulanır [1, sayfa 260-261].

3.4.2) Blok Esrime Kodlama (Block Truncation Coding)

Blok esrime kodlama görüntüyü bloklara bölerek, her bloktaki gri seviyesini azaltır. Bu azaltma işlemi her bloktaki alt görüntülerin istatistiklerine bakarak kuantalama işlemi ile gerçekleştirilir. Kuantalamanın derecesi ise belirtilmiş hata kriterini minimum yapacak şekilde seçilir [1, sayfa 261-265].

3.4.3) Vektör Kuantalama (Vector Quantization)

Vektör kuantalama vektörlerin daha az boyuta sahip olan başka vektörlere haritalanma işlemidir. Görüntü kodlamada vektörler daha küçük alt görüntülere veya bloklara ayrılır. Vektör kuantalama ile ilgili daha ayrıntılı bilgi ilerideki bölümlerde verilecektir.

3.4.4) Farkları Öngörücü Kodlama (Differential Predictive Coding)

Farkları öngörücü kodlama bir sonraki piksel değerini önceki piksel değerine bakarak öngörme olayıdır. Gerçek değerle öngörülen değer arasındaki fark kodlanarak gerçekleştirilir. Yüksek bağımlı görüntülerde bu tekniğin avantajları çoktur. Çünkü bu tür görüntülerde komşu pikseller arasındaki fark az olmakta, dolayısıyla daha az sayıda bit gerekmektedir [1, sayfa 270-276], [2,sayfa 362-374].

3.4.5) Dönüşüm Kodlama (Transform Coding)

Bilindiği üzere sayısal görüntü pikselleri birbirine yüksek oranda bağımlıdır. Diğer bir ifadeyle, görüntü pikselleri arasında yüksek bir oranda tekrarlılık mevcuttur. Görüntü sıkıştırmanın amacı da bu tekrarlılıkları azaltıp görüntüyü daha az bit sayısı ile temsil etmektir. Bu tekrarlılığı azaltmanın bir yolu öngörme yöntemidir. Öngörmeli kodlamada asıl piksel ile öngörülen piksel arasındaki fark kodlanmaktadır. Bu fark pikselin değerinden daha azdır.

Dönüşüm kodlama ise tekrarlılığı azaltmanın bir başka yoludur. Dönüşüm kodlamada görüntü pikselleri bloklara ayrılarak dikgen dönüşüm uygulanır. Dönüşüm alanındaki katsayılar piksellerden daha az bağımlıdır. Dönüşümün performansı ise bağımlılığı ne kadar azalttığı ve enerjiyi ne kadar sıkıştırabildiği ile ölçülür. Birçok dönüşüm yöntemi olmasına rağmen imge sıkıştırmada popüler olan kesikli kosinüs dönüşümüdür. Çünkü kesikli kosinüs dönüşümü bağımlılığı azaltma özelliğine yaklaşır. Bu dönüşüm görüntü ve video sıkıştırma standartları olan JPEG, MPEG de kullanılmaktadır.

Vektör transformu kodlama klasik kodlama tekniklerinin (örneğin skalar dönüşüm kodlama) vektörselleştirilmiş halidir. Vektörleştirme işlemi ise piksel yerine piksel bloklarının yerleştirilmesidir. Buradan da anlaşılacağı gibi vektör dönüşüm kodlama klasik skalar dönüşüm kodlamasının geliştirilmiş halidir. Diğer bir ifadeyle skalar dönüşüm kodlama vektör dönüşüm kodlamasının özel bir halidir.

Vektör dönüşümünün iki özelliği vardır :

Özellik 1) Vektörler arası bağımlılığı azaltmak (Reduction of inter-vector correlation)

Vektör dönüşümü diğer koşullar sabit kalmak şartıyla, minimum kodlama bozulması elde etmek için vektör dönüşüm alanındaki vektörler arasındaki bağımlılığı azaltmak için kullanılır. Minimum bağımlılık maksimum enerji sıkıştırmasına sebep olur.

Özellik 2) Vektör içi bağımlılığı korumak (Preservation of intra-vector correlation)

Vektör dönüşümü diğer koşullar sabit kalmak şartıyla, minimum kodlama bozulması elde etmek için vektör dönüşüm alanındaki vektör elemanları arasındaki bağımlılığı koruması da gereklidir.

Bundan dolayı optimum vektör dönüşümü bu iki özeliğe sahip olması gerekir [4], [5], [6].

3.4.5.1) Kesikli Kosinüs Dönüşümü

Sinyalleri zaman bölgesinde gösterildiği zaman her zaman noktasına karşılık gelen voltaj değerlerini görünür. Bu sinyale hızlı fourier dönüşümü uygulandığı zaman aynı sinyali frekans örnekleri olarak görülür. Eğer sinyal farklı frekans bileşenleri toplamından oluşuyorsa zaman bölgesindeki gösterimde bu frekans bileşenlerini görülmez. Aynı sinyale hızlı fourier dönüşümü uygulandığı zaman bu frekans bileşenleri ayrıntılı olarak görülebilmektedir. Hemen hemen bütün dönüşümlerde olduğu gibi hızlı fourier dönüşümü de tersinirdir. Kesikli kosinüs dönüşümü hızlı fourier dönüşümüne benzer ve aynı sonucu verir. Bununla birlikte orijinal sinyal 2-boyulu ise kesikli kosinüs dönüşümü uygulandığında 3-boyutlu sinyal elde edilir. Bu durumda sinyal grafiksel bir görüntüdür ve X,Y eksenleri verinin boyutlarını, Z eksenini ise sinyalin genliğini ekranda gri seviye değerleri olarak gösterir.

$$KKD(i, j) = \frac{1}{\sqrt{2N}} C(i)C(j) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} piksel(x, y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (3.6)$$

$$piksel(x, y) = \frac{1}{\sqrt{2N}} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} C(i)C(j) KKD(i, j) \cos\left[\frac{(2x+1)i\pi}{2N}\right] \cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (3.7)$$

$$C(x) = \frac{1}{\sqrt{2}} \quad \text{eğer } x = 0, \text{ değilse } 1$$

3.4.6) JPEG (Joint Photographic Expert Group)

JPEG hem gri seviyeli hem de renkli görüntülere uygulanabilir. JPEG kesintisiz tonlu görüntülerde en iyi sonucu vermektedir. Eğer renkli görüntülerdeki renk değerleri ani değişiyorsa sıkıştırma sonuçları kötü olmaktadır. Sıkıştırma oranı ve görüntü kalitesi parametrelerin kullanıcı tarafından ayarlanabilmesiyle değişiklik göstermektedir.

JPEG algoritması:

Adım 1) Gri seviyeli görüntülerde dönüşüm işlemi yapılmaz, ancak renkli görüntüler RGB uzayından ışık/renk uzayına (YUV uzayına) dönüştürülür. Işık elemanları gri seviyelerini, diğer iki eleman da renk bilgilerini içerir. Bunun amacında insan gözünün ışık bilgilerinin renk bilgilerine göre daha duyarlı olması dolayısıyla renk bilgilerinde daha fazla sıkıştırma yapılabilmesidir. Eğer renk uzayı değiştirilmek istenmezse algoritmanın geri kalan kısmı her renk elemanına ayrı olarak uygulanır. Bununla birlikte bütün elemanların ışık kalitesinde sıkıştırma yapıldığından sıkıştırma oranı daha az olmaktadır.

Adım 2) Pikselin her elemanına örnek indirgemesi yapılır. Işık elemanları tüm çözünürlükte, renk elemanları ise genellikle dikey yönde 2:1 oranında ve yatay yönde 2:1 veya 1:1

oranında azaltılır. Bu adım sayesinde görüntü bilgileri 2:1 veya 3:1 oranında azalmış olur. Nümerik olarak yüksek kayıp olsa da insan gözünün renk çözünürlüğü zayıf olduğundan izleme kalitesini etkilememektedir. Örnek indirgemesi gri seviyeli görüntülere uygulanmaz. Bundan dolayı renkli görüntüler gri seviyeli görüntülere göre daha fazla oranda sıkıştırılabilmektedir.

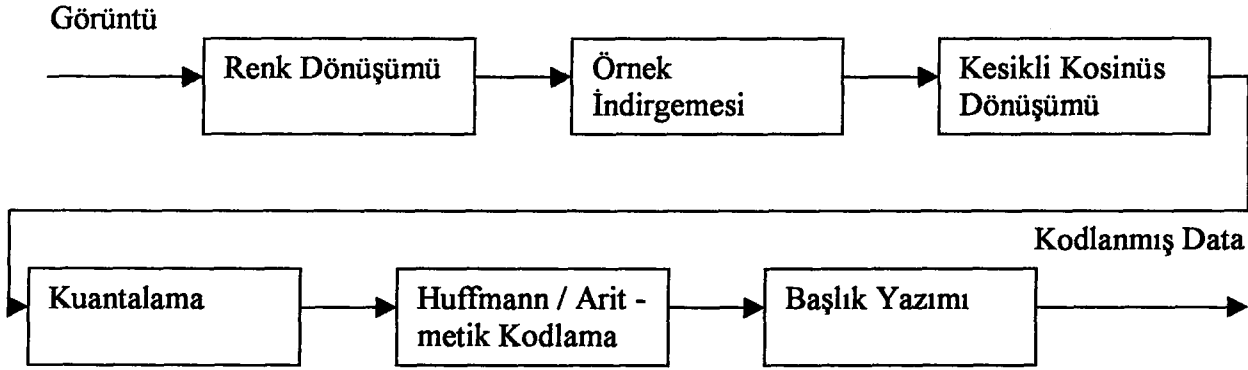
Adım 3) Her görüntü elemanı 8x8 bloklara bölünür. Her bloğa kesikli kosinüs dönüşümü uygulanır. Kesikli kosinüs dönüşümünün kendisi yuvarlama hataları dışında tersinirdir.

Adım 4) Her bloktaki , 64 frekans elemanına kuantalama katsayılarına ayrılır ve sonuçlar tam sayıya yuvarlanır. Burası bilginin asıl kaybolduğu adımdır. Kesikli kosinüs dönüşümü çıkışı tam sayı değildir. Yüksek frekans bilgileri düşük frekans bilgilerine göre daha düşük duyarlıktadır, ancak göz tarafından daha az hissedilir. Ayrıca ışık bilgileri renk bilgilerine göre daha duyarlı kuantalanır.

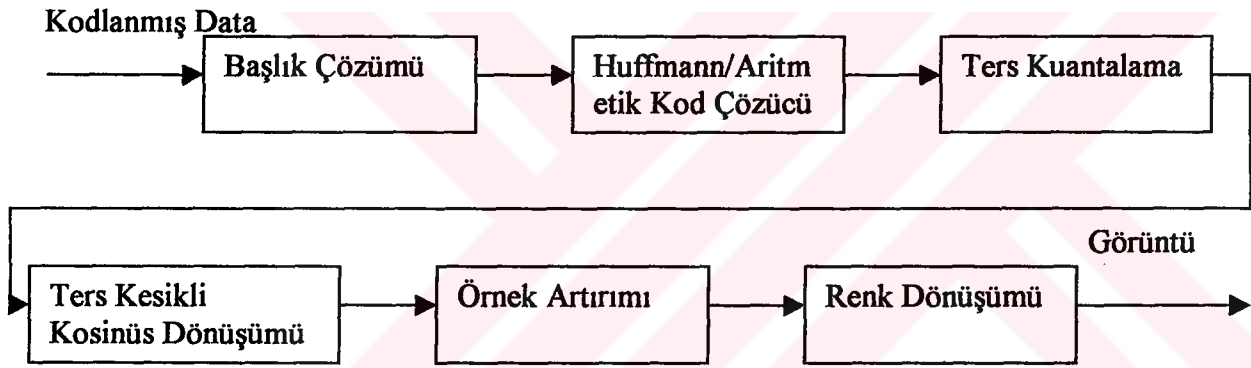
Adım 5) Azaltılmış katsayılar Huffmann veya aritmetik kodlama ile kodlanır. Bu adım kayıpsızdır ve görüntü kalitesini etkilemez.

Adım 6) Uygun başlık yazılır. Başlıkta kodu çözebilmek için bütün sıkıştırma parametreleri bulunur. Bu parametreler ayrıca kuantalama tablosunu ve Huffmann kodlama tablosunu içerir. Eğer kod çözücüde kuantalama ve Huffmann kodlama tablosu var ise kodlayıcıda bu tablolar ayrıca yazılmaz.

Kod çözücü işlemi ise kodlayıcının tam tersidir. Kodlayıcı kuantalama tablosundaki katsayıları çarparak kesikli kosinüs dönüşüm katsayılarına yaklaştırır. Yaklaştırılmış katsayılar ters kesikli kosinüs dönüşümünden geçirilerek tekrar oluşturulmuş görüntü elde edilir. Yüksek kaliteli kod çözme için ekstra adımlar yapılabilir. Bunlar pikseller arasındaki kesikliliği giderir [7], [3, bölüm 11] .



Şekil 3.5) JPEG kodlayıcı



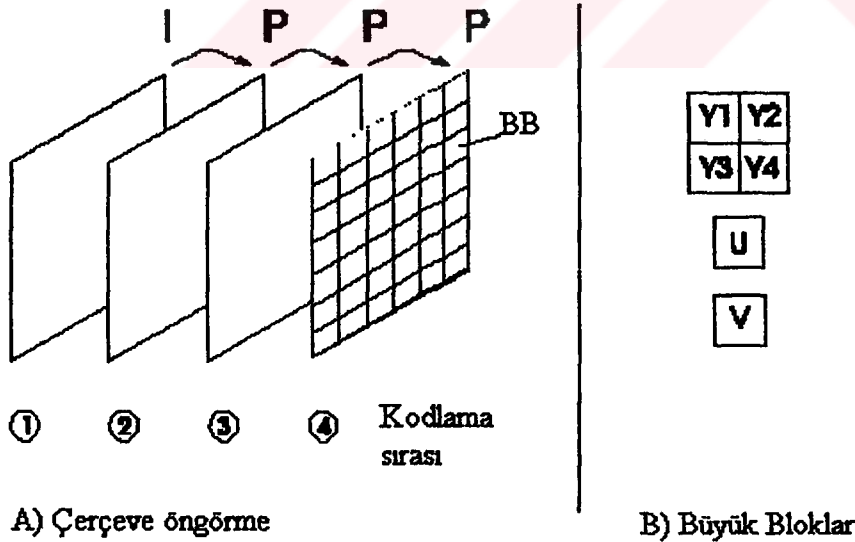
Şekil 3.6) JPEG kod çözücü

3.4.7) MPEG (Moving Picture Expert Group)

Video dizilerinde genellikle geçici ve uzamsal boyutta istatistiksel ve öznel tekrarlılık vardır. MPEG aynı çerçeve içindeki pikseli komşu piksellerden (çerçeve içi kodlama tekniği kullanılır) veya komşu çerçevedeki piksellerden (çerçeveler arası kodlama tekniği kullanılır) yararlanarak tahmin eder. Çerçeve içi kodlama tekniğinde görüntü 8x8 bloklara bölünüp, her bloğa kesikli kosinüs dönüşümü uygulanır. Bununla birlikte , eğer çerçeveler arası ilinti yüksekse, bu durumda çerçeveler arası kodlama tekniği olarak farkları öngörücü kodlama kullanılır (çerçeveler arasında devimin dengelemeli öngörme yapılır.) Hemen hemen bütün

video kodlama tekniğinde olduğu gibi ve JPEG’de de anlatıldığı gibi MPEG’de örnek indirgemesi yapılır. Devimin dengelemeli öngörme çerçeveler arasındaki geçici boyuttaki tekrarlılığı azaltmak için etkili bir yoldur. Devimin dengelemeli öngörmenin temeli çerçeveler arasındaki hareketi tahmin etmektir. Eğer video dizilerindeki pikseller uzamsal boyutta yaklaşık aynıysa, çerçeveler arasındaki hareket az sayıdaki hareket parametreleri ile tanımlanabilir. En iyi öngörme bir pikselin bir önceki çerçevedekine göre hangi yönde değiştiğinin öngörülmesidir. Hareket vektörleri arasında da yüksek ilinti olduğu için piksel yerine piksel bloklarının hareketi öngörülür.

MPEG video sıkıştırmanın temeli blok yapısına dayanır. MPEG kodlayıcıda ilk çerçeve, çerçeve içi kodlama tekniği kullanarak kodlanır (I – resmi). Daha sonra gelen çerçeveler ise çerçeveler arası kodlama tekniği kullanılarak kodlanır (P – resmi). Çerçeveler arası öngörmede (P – resimleri) önceki kodlanmış I veya P çerçevesinden öngörülür. Giriş çerçevesindeki piksellerin her renk elemanı şekil 3.7’ de görüldüğü gibi gruplanarak büyük bloklar oluşturulur. Her blok 8x8 pikselden oluşur.



Şekil 3.7) MPEG bloklama yapısı

MPEG'in blok diyagramı şekil 3.8'de görülmektedir. İlk çerçeve (I –resmi) bir önceki veya sonraki çerçeveden öngörülmeden kodlanır. Işık ve renk bloklarına kesikli kosinüs dönüşümü uygulanır (KKD). Kesikli kosinüs değişiminden elde edilen bu katsayı kuantalanır (K). Sıfırdan farklı kuantalama değerleri değişken uzunlukla kodlayıcı tarafından kodlanır (DUK). Sıfırdan farklı kuantalama değerleri değişken uzunlukla kodlayıcı tarafından kodlanır (DUK).

Kod çözücü işlemi ise kodlayıcının tersidir. İlk önce değişken uzunluktaki kod çözücüyle kod kelimeleri çözülür, ters kuantalama yapılır. Daha sonra ters kesikli kosinüs dönüşümden geçirilerek orijinal görüntüye benzetilen görüntü elde edilir [8].

KKD : Kesikli Kosinüs Dönüşümü

DUK : Değişken Uzunluktaki Kodlama

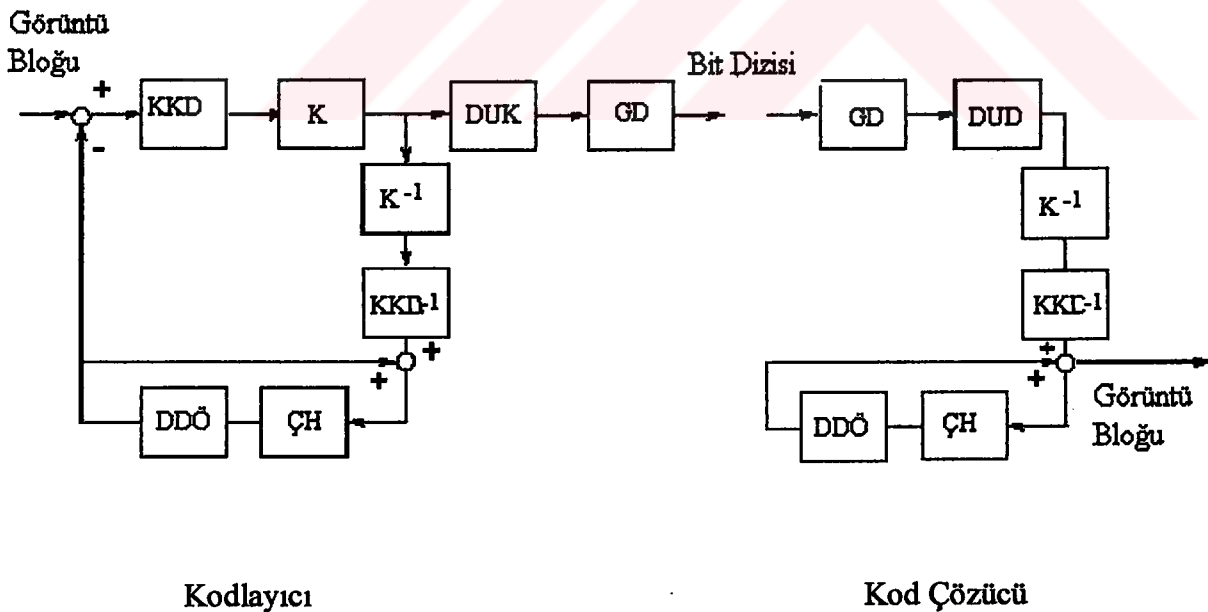
K : Kuantalama

DUD : Değişken Uzunluktaki Kod Çözücü

ÇH : Çerçeve Hafızalayıcı

GD : Görüntü Depolayıcı

DDÖ : Devinim Dengelemeli Öngörme



Şekil 3.8) MPEG kodlayıcı ve kod çözücü blok yapısı

BÖLÜM 4

SKALER KUANTALAMA

Kuantalama, analog sinyalin dijital sinyallere çevrilmesi olarak bilinmektedir. En basit haliyle kuantalama daha önceden belirlenmiş nümerik değerlerden en yakın olan nümerik değerlere atama işlemidir. Çoğunlukla giriş değerleri analogtur, yani kesintisiz genlikte herhangi bir değer olabilir. Bununla birlikte çıkış değerleri ise dijitaldir ve bir set şeklinde, $\{ 1, 2, 3, \dots, N \}$, ifade edilir. Burada N değeri çıkış setinin büyüklüğüdür. Kuantalama veri sıkıştırmanın veya kodlamanın önemli bir kısmıdır. Analog sinyallerin etkin bir kodlamayla iletim kanallarında daha etkili bant genişliğine ulaşması sağlanabilmektedir. Fakat sinyal eninde sonunda kullanıcı için analog sinyale çevrileceğinden, kuantalama esnasında ortaya çıkan bozulma sinyali kodlamada istenmeyen bir durumdur.

Analog kaynağın (sürekli zaman, sürekli genlik) dijital kaynağa (kesikli zaman, kesikli genlik) çevrimi iki kısım içermektedir. Bunlar örnekleme ve kuantalamadır. Örnekleme, sürekli zaman sinyalini, düzenli zaman aralıklarında sinyal değerlerini ölçerek kesikli zaman sinyallerine çevirmektir. Kuantalama sonucunda elde edilen kesikli genlikteki sinyal kuantalama hatası veya gürültü nedeniyle giriş sinyalinden farklıdır.

Sinyal örneklerinin her birinin ayrı olarak kuantalanmasına “ skalar kuantalama veya kodlama “ denir. Sinyal örneklerinin bir bloğu veya sinyal parametrelerin bir bloğunun kuantalanmasına ise “ blok kuantalama veya vektör kuantalama “ denir.

Çok kısaca, N-noktalı skalar (tek boyutlu) kuantalamanın tanımı :

$$K : R \rightarrow C$$

Burada ;

R reel doğru ve

$C \equiv \{ y_1, y_2, y_3, \dots, y_N \} \subset R$ çıkış seti veya kodkitabıdır.

Boyutu ise $|C| = N$ dir. Çıkış değerleri, y_i , ise çıkış seviyesi, çıkış noktaları veya yeni üretilen değerler diye adlandırılır.

Ayırma duyarlılığı (resolution) veya kod oranı (code rate) tanımı ise, $r, r = \log_2 N$ dir. Bu eşitlik belirlenmiş kuantalama değeri için ne kadar bit gerektiğini ölçer ve orijinal analog genliğinin doğruluk derecesini belirler. Eğer r değeri tamsayı (integer) ise her y_i tek ikili r-sıralı (unique binary r-tuple) olur. Bu durumda kuantalama değerini belirtmek için sabit oranlı kod (fixed rate code) kullanılır. Alternatif olarak, farklı kuantalama değerleri için değişen uzunlukta basamağa (digit) sahip kelimeler kullanılabilir ki buna da değişken oranlı kod (variable rate code) ismi verilir.

Haberleşme teknolojisinde örnek dizileri belli zaman aralıklarında örneklenir ve sabit bit miktarı ile kuantalanır. İletim oranının (transmission rate) tanımı ise, R, her örnek için ne kadarlık bit iletileceğidir ve $R = \log_2 N$ eşitliği ile bulunur. Kod oranı ile iletim oranı nümerik olarak eşittir. Bundan dolayı her iki orana birden tek kelimeyle oran ismi verilir.

Eğer örnekleme dönemi (sampling period) T ise iletim oranı normalize edilerek bir saniyede iletilebilecek bit miktarı ile ifade edilir ve saniyedeki bit miktarı (bits per second) diye adlandırılır.

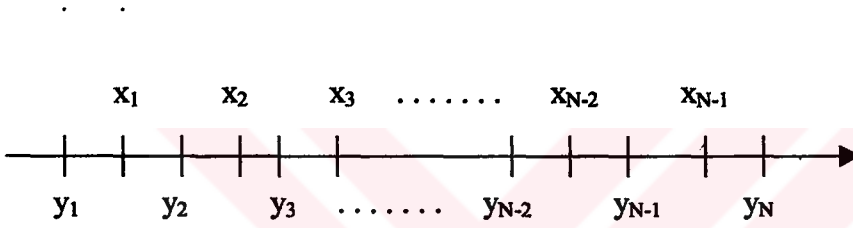
Skalar kuantalama yapabilmek için çok sayıda teknik geliştirilmiştir. Bunlardan en çok kullanılan iki tanesi ise şöyledir.

4.1.1) Tek boyutlu kuantalayıcı (One-dimensional quantizer)

Bu kuantalayıcıda aralık son noktaları (interval endpoints) aşağıdaki gibi sıralanır.

$$x_0 < y_1 < x_1 < y_2 < x_2 < \dots < y_N < x_N$$

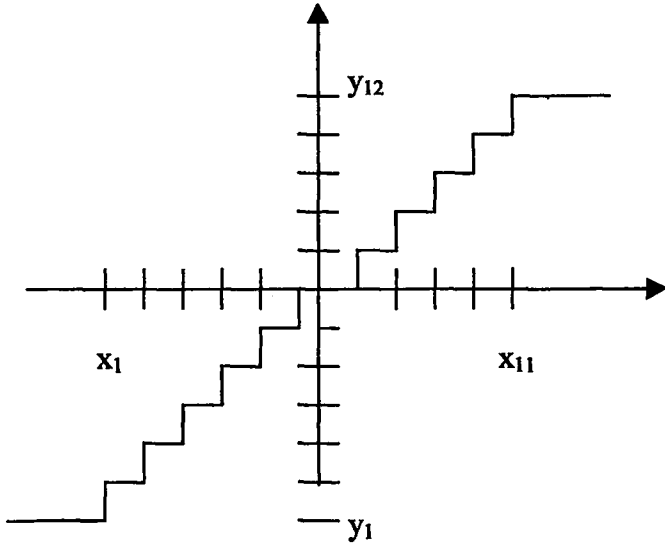
Aralık son noktaları ve çıkış değerleri kuantalayıcı tarafından belirlenir [9, sayfa 133-135] .



Şekil 4.1) Tek Boyutlu kuantalayıcı. Aralık son noktaları ve çıkış seviyeleri yatay çizgide gösterilir.

4.1.2) Merdiven Kuantalayıcı (Staircase Quantizer)

Merdiven kuantalayıcının grafiği şekil 4.2 de gösterilmiştir. Giriş çıkış karakteristiği N basamaklı merdiven şeklindedir. Yatay adım aralıkları giriş vektör aralıklarını, dikey adımlar ise çıkış sınır noktalarını gösterir. Bu kuantalayıcıya orta basamaklı kuantalayıcı adı da verilmektedir [9, sayfa 135-138] .



Şekil 4.2) Merdiven kuantalayıcının grafiği

4.2) Skalar Kuantalayıcı Dizayn Algoritması

Skalar kuantalayıcı algoritması ilk olarak Lloyd tarafından geliştirilmiştir. Lloyd algoritması skalar kuantalama için geniş ölçüde kullanılmaktadır. Bu algoritma başlangıç kod kitabı ile başlar ve bitme kriteri sağlanasıya kadar öz yinelemeli olarak devam eder [9, bölüm 6].

Kod kitabı geliştirmek için kullanılan Lloyd öz yinelemesi ise şöyledir:

- a) Verilen kod kitabı, $C_m = \{ y_i \}$, için optimum kuantalama hücreleri bulunur. Bu bulma işleminde en yakın komşu şartı aranır. En yakın komşu şartı ise ;

$$R_i = \{ x: d(x, y_i) \leq d(x, y_j) ; \text{bütün } j \neq i \} \quad (4.1)$$

- b) Merkezleme şartı kullanılarak optimum yeni üretilmiş kod kitabı bulunur.

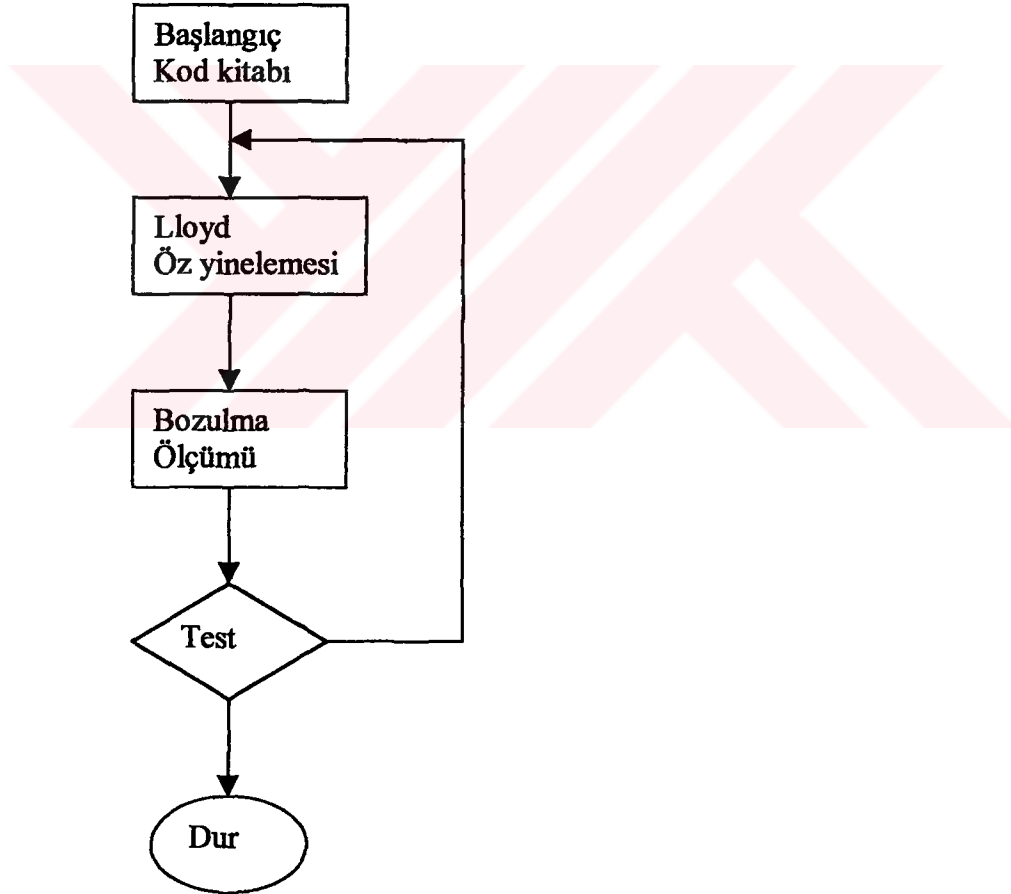
$$\text{Merkez } (R) = \frac{1}{\|R\|} \sum_{i=1}^{\|R\|} X_i \quad (4.2)$$

Lloyd Algoritması :

Adım 1) Öz yineleme sayısı, m , bire eşitlenir. Başlangıç kod kitabı ile başlanır.

Adım 2) Verilen kod kitabı, C_m , Lloyd öz yinelemesine tabi tutularak yeni kod kitabı, C_{m+1} , üretilir.

Adım 3) Yeni kod kitabının, C_{m+1} , ortalama bozulması hesaplanır. Eğer bozulma bir önceki bozulmaya göre çok az bir değişim gösteriyorsa algoritma durdurulur. Diğer şartta öz yineleme sayısı artırılır ve ikinci adıma dönülür.



Şekil 4.3) Lloyd algoritmasının akış diyagramı

BÖLÜM 5

VEKTÖR KUANTALAMA

Veri sıkıştırmanın temel amacı kabul edilebilir doğruluk veya görüntü kalitesinde iletim veya veri saklamak için bit sayısını azaltmaktır. Uyarlanır farksal vurumlu kod kişlenimi (differential pulse code modulation), dönüşüm kodlaması (transform coding), hybrid kodlama ve bunların uyarlanır varyasyonları (adaptive variations) gibi bir çok bant genişliği azaltma teknikleri geliştirilmiştir. Bu teknikler bit oranını azaltırken görüntüdeki tekrarlıklardan yararlanılır. Örneğin dönüşüm kodlamada, ilk olarak vektörler bloklara ayrılır ve her blok dönüşüme uğratılır. Dönüşüm örnekleri daha sonra skalar olarak kuantalanır. Öngörücü kodlamada şimdiki örnekle öngörülen örnek arasındaki fark kodlanır.

Shannon'un oran bozulma kuramına göre teorik olarak daha iyi bir performans elde edebilmek için skalerler yerine vektörler kullanmak gerekir [9, sayfa 471]. Eğer sinyal örnekleri veya sinyal parametreleri istatistiksel olarak birbiriyle bağımlı ise örnekler veya parametreler bloklar halinde alınarak bu bağımlılık kullanabilmektedir. Böylece vektör kuantalama kullanarak ulaşılan bit miktarı skaler kuantalamayla ulaşılan bit miktarı ile karşılaştırıldığında daha düşük olduğu görülmektedir. Görüntü sinyal örneklerinde bu bağımlılık mevcuttur. Bu imge kodlamada vektör kuantalamayı kullanmak için geçerli bir sebeptir.

Bir kuantalayıcıya optimum diyebilmek için çıkıştaki bozulmanın minimum olması gerekmektedir. Bir kuantalayıcıda bozulma yerel olarak (blok içinde) minimum ise yapılan kuantalama genel olarak da minimumdur. Böylece, orijinal görüntü sinyal bloklarının en iyi temsil edileceği bir kod kitabı oluşturulabilecektir.

5.1) Vektör Kuantalamanın Tanımı

Birçok araştırmacı vektör kuantalama veya çok boyutlu kuantalama üzerine kapsamlı çalışmalar yapmışlardır. Görgül veriden (emprical data) optimum vektör kuantalama dizaynı Linde, Buzo, Gray'in çalışmalarında sınıflandırma yaklaşımı ile amaçlanmıştır. Bu algoritma LBG algoritması olarak da bilinmektedir.

Vektör kuantalama K Boyutlu Euclidian uzayının, R^k , aynı uzaydaki sınırlı alt sete, Y, haritalanması diye tanımlanır. Yani,

$$K : R^k \rightarrow Y$$

Burada;

$$Y = \{y_i ; i=1,2, \dots, L\} \quad \text{tekrar üretilmiş vektör seti}$$

$$L \quad \text{Y uzayındaki vektör sayısıdır.}$$

Vektör kuantalama, N boyutlu giriş vektörleri, x, ve yine N boyutlu daha az sayıdaki çıkış vektörlerine, y, haritalamaktır. y vektörleri, x vektörünün kuantalanmış değeri olmaktadır. Şu şekilde yazılabilir.

$$y = K(x) \quad (5.1)$$

$K(\cdot)$, kuantalama operatörüdür.

y, tekrar oluşturulan vektör veya x ile ilişkili çıkış vektörü olarak tanımlanabilir. Tipik olarak;

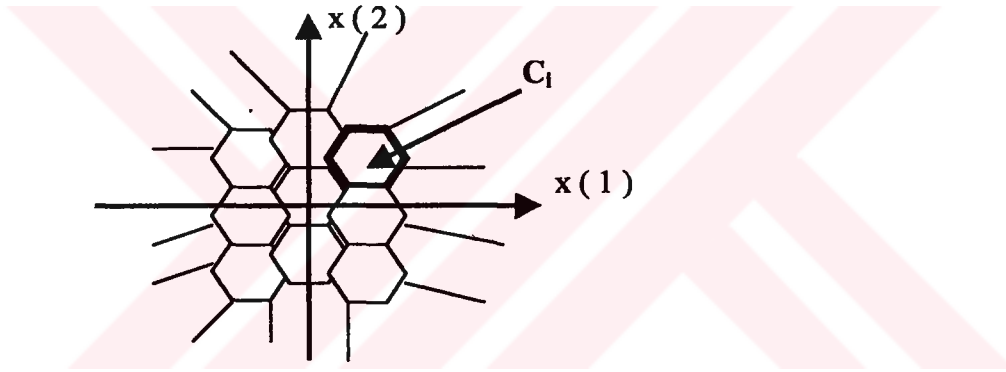
$$Y = \{y_i ; 1 < i < L\}, \text{ ve } y_i = (y_{i1}, y_{i2}, \dots, y_{iN})$$

yazılabilir. Y seti, tekrar oluşturulan kod kitabı veya basit olarak kod kitabı olarak tanımlanır. L kod kitabının büyüklüğüdür ve kuantalama seviyesinin derecesini belirler. Bu açıklamaların sonucu olarak L-seviyeli kod kitabı ve L-seviyeli kuantalayıcı aynı anlama gelmektedir. Kod kitabı dizaynı için x giriş vektörünün N-boyutlu uzayı, L parçaya veya hücreye ayrılmaktadır.

$\{ C_i, 1 < i < L \}$ ile gösterilebilir ve C_i hücresi y_i vektörü ile temsil edilmektedir. Eğer x giriş vektörü C_i hücresinde ise kod vektörü olarak y_i atanır [10], .

$$x \in C_i \quad K(x) = y_i \quad (5.2)$$

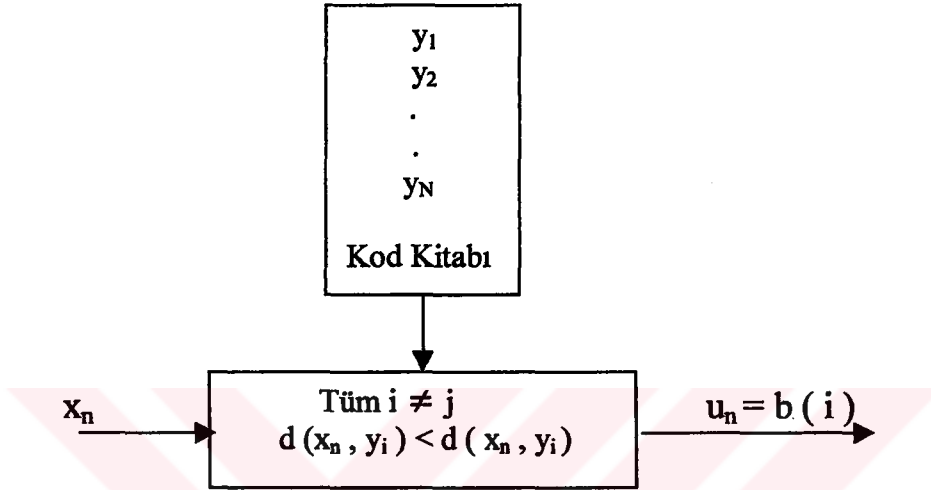
Aşağıdaki şekil vektör kuantalamaya ($N = 2$) iki boyutlu uzayda örnek verilmiştir. İki boyutlu uzay 22 hücreye ($L = 22$) bölünmüştür. Her giriş vektörü C_i hücresindeki bir vektör ile kuantalanmaktadır. Hücre şekilleri çok farklı olabilmektedir. Kod vektörlerin uzaydaki pozisyonları diğer hücrelerle bağlantılıdır [11, sayfa 26-28] .



Şekil 5.1) Kuantalama hücre örneği

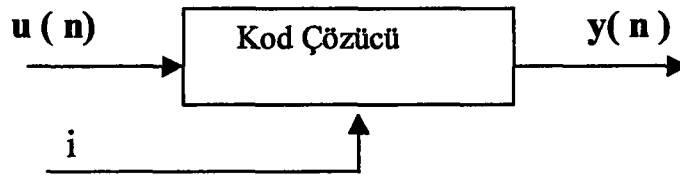
Kuantalama iki fonksiyonun birleştirilmesi ile olur : kodlayıcı ve kod çözücü. Kodlayıcıda, giriş vektörü ve $K(x)$ ile belirlenen tekrar üretilmiş vektörünü indisinin ikili

gösterimi görülür. Giriş vektörü ve her kod vektörü arasındaki mesafe hesaplanarak minimum mesafeye sahip kod vektörünün indisinin ikili gösterimi çıkışta kodlanır.



Şekil 5.2) Vektör kuantalama kodlayıcı

Kod çözücünde ise, ikili indisleri kullanarak giriş vektörüne benzeyen vektörler, y , üretilir.



Şekil 5.3) Vektör kuantalama kod çözücü

Orijinal giriş vektörü, x , tekrar üretilmiş vektör, y , arasında ne kadarlık bozulma olduğunun ölçülmesi gerekir ki bu da $d(x, y)$ ile gösterilir. Daha iyi bir haritalama için $d(x, y)$ minimize etmek gerekir.

5.2) Bozulma Ölçümleri

Literatürde çok sayıda bozulmayı ölçme yöntemleri vardır. Bozulma ölçümü kolayca uygulanabilir olmak zorundadır. Bugün kullanılan bozulma ölçümleri kolayca uygulanabilir olmasına rağmen biraz uzun sürmektedir. Kodlamada matematiğin kolay olmasından dolayı en çok kullanılan yöntem hataların karesi (square error) yöntemidir.

$$d(x, \hat{x}) = \|x - \hat{x}\|^2 = \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^2 \quad (5.3)$$

Diğer yaygın olarak kullanılan bozulma ölçümü ise L_v veya Holder normu'dur.

$$d(x, \hat{x}) = \left\{ \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^v \right\}^{1/v} = \|x - \hat{x}\|_v \quad (5.4)$$

Bir diğeri ise v^{th} - law bozulma ölçümüdür.

$$d(x, \hat{x}) = \sum_{i=0}^{k-1} |x_i - \hat{x}_i|^v = \|x - \hat{x}\|_v^v \quad (5.5)$$

Ağırlaştırılmış hataların karelerin toplamı ise;

$$d_w(x, \hat{x}) = (x - \hat{x})^T W (x - \hat{x}) \quad (5.6)$$

şeklindedir.

Burada W , pozitif tanımlı (positive-definite) ağırlıklı matristir. $W = N^{-1} I$ alındığında $d_w = d_2$ olmakta, bu da hataların karesi yöntemine eşittir. $W = \Gamma^{-1}$ olarak seçilmektedir. Γ , x vektörünün ortak değişimi (covariance) matrisidir ve şu şekilde hesaplanmaktadır.

$$\Gamma = B \left[(x - \hat{x})(x - \hat{x})^T \right] \quad (5.7)$$

Burada;

B_0 = Beklenen deęer

Bu durumda d_w řu řekile dđner,

$$d_w(x, \hat{x}) = (x - \hat{x})^T \Gamma^{-1} (x - \hat{x}) \quad (5.8)$$

Bu ölçüm Mahalonobis mesafesi olarak bilinmektedir[11, sayfa 43-47], [12], [13, sayfa 11-12] .

Vektör kuantalamanın en büyük problemi ise giriş vektörlerinin kod kitabındaki en yakın vektöre eşleyebilmek için kod kitabındaki bütün vektörlerle bozulma ölçümünün hesaplanarak minimum bozulma ölçümüne sahip olan vektörün araştırılmasıdır. Bu araştırma işlemi Buzo tarafından önerilmiş olan ağaç arařtırımalı vektör kuantalama (tree search vector quantization) yöntemi ile azaltılmıştır. Ağaç arařtırımalı vektör kuantalama da kod kitabının hepsinde arařtırmak yerine küçük kod kitabı dizilerinde arařtırma diye tanımlanabilir. Bu yöntemin dezavantajı ise tam arařtırımalı (full search) vektör kuantalamaya göre daha fazla hafıza istemesidir. Diğer vektör kuantalama yöntemleri ise hafızalama problemlerini azaltmaya çalışmışlardır. Bunlara Fisher tarafından dizayn edilen piramit vektör kuantalama (Pyramid VQ) örnek verilebilir.

5.3) Vektör Kuantalama İçin Optimum Olma Şartları

Vektör kuantalayıcının temel amacı optimum kod kitabını bulmaktır. Toplam performansı maksimum yapacak kodlayıcı ve kod çözücünün dizaynının yapılması gerekmektedir. Toplam performansın ölçülmesi ise ;

$$D = B \{ d(x, K(x)) \} = \sum_i d(x_i, K(x_i)) p_x(x_i) \quad (5.9)$$

Burada ;

$p_x(x)$ olasılık kütle fonksiyonu (probability mass function)
 $d(x, K(x))$ x vektörü ile $K(x)$ vektörü arasındaki bozulma miktarı

Büyüklüğü N olan bir kod kitabı, k -boyutlu giriş vektörleri verildiğini ve bozulma ölçümü $d(.,.)$ için herhangi bir yöntemin seçildiğini düşünelim. Bu şartlar altında minimum ortalama bozulmayı sağlayacak bir kuantalayıcı tasarlamak isteyelim. Kodlayıcı, R^k uzayını R_1, R_2, \dots, R_N gibi hücrelere böler ve her hücre için o hücreyi temsil edecek olan kod vektörünü belirler. Böyle bir kuantalama tasarımında verilen kod kitabı için optimum bölmeyi ve verilen bölünmüş hücreler için de optimum kod vektörünün bulunma şartlarının belirlenmesi gerekir.

İlk önce sabit kod çözücü için optimum kodlayıcı şartını ele alalım. Verilen kod kitabı için, optimum bölmeyi sağlama koşulu en yakın komşu (nearest neighbor condition) koşuludur. Bütün giriş vektörleri kod kitabındaki kendisine en yakın olan kod vektörü ile temsil edilmelidir.

5.3.1) En yakın komşu şartı:

Verilen çıkış setinin optimum bölünme hücreleri ;

$$R_i \subset \{ x : d(x, y_i) \leq d(x, y_j) ; \text{bütün } j\text{'ler için} \} \quad (5.10)$$

sağlamalıdır. Bu da ;

$K(x) = y_i$ yalnız ve yalnız $d(x, y_i) \leq d(x, y_j)$ bütün j değerleri için

şartını sağlar. Buradan da kodlayıcı da minimum bozulmayı sağlayacak veya en yakın komşuya haritalanır. Yani ;

$$d(x, K(x)) = \min_{y_i \in C} d(x, y_i) \quad (5.11)$$

İkinci durum ise merkezleme şartıdır. Bu da her bölme hücresi için optimum kod vektörünün belirlenmesidir. Merkezlemenin tanımı ise herhangi bir setteki, $R \in R^k$, bütün vektörlerin ortalamasıdır. Yani;

$$\text{Merkez}(R) = \frac{1}{\|R\|} \sum_{i=1}^{\|R\|} X_i \quad (5.12)$$

Burada;

$$R = \{ x_i ; i= 1,2,3, \dots , \|R\| \}$$

$\|R\|$ R setindeki eleman sayısıdır.

5.3.2) Merkezleme Şartı

Verilen hücreler için optimum kod vektörü için

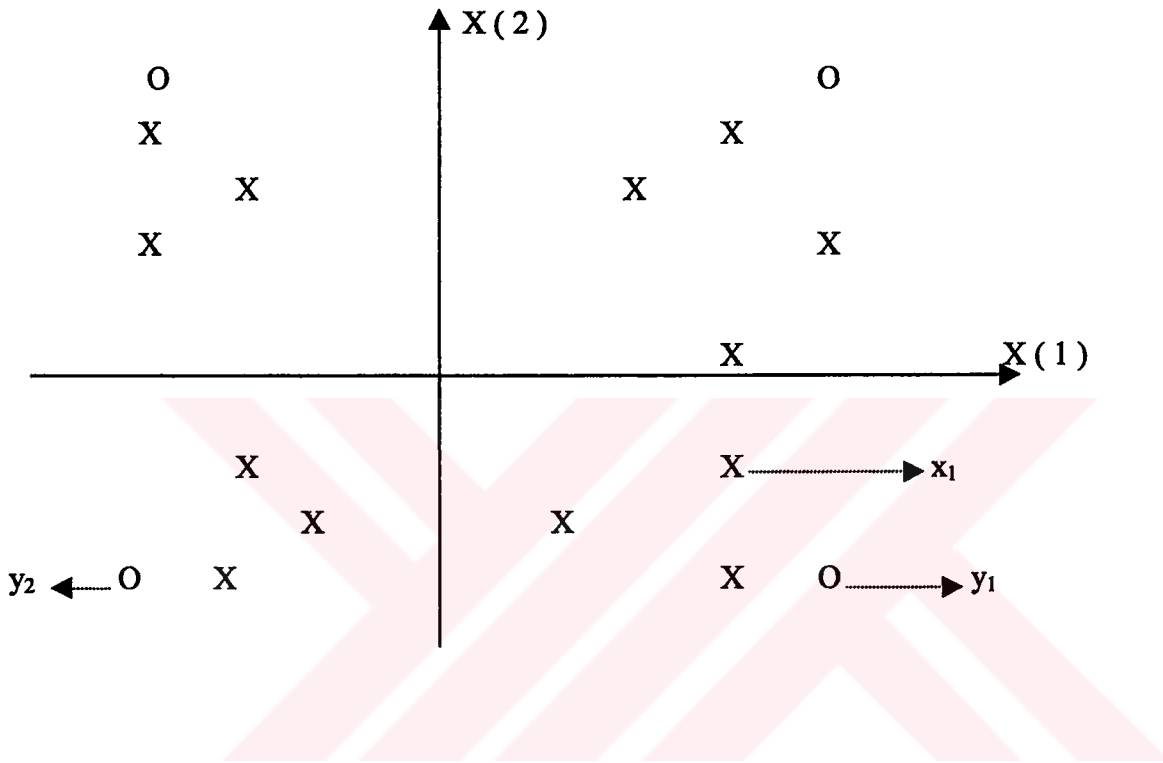
$$y_i = \text{merkez} (R_i) \quad (5.13)$$

eşitliğinin sağlanması gerekir[9, sayfa 349-358].

Minimum bozulmayı sağlayan kod vektörünün seçilmesi aşağıdaki örnek ile açıklanabilir. Örnekte 2 – boyutlu 12 eğitim dizisi alınmış ve 4 tane kod vektörü üretilerek 4 bölgeye yerleştirilmiştir. Burada 12 eğitim vektörü kendisine en yakın kod vektör ile temsil edilerek minimum bozulma elde edilmiştir.

X : eğitme vektörü

O : kod vektörleri



Şekil 5.4) Vektör kuantalama örneği

$d(x_1, y_2) > d(x_1, y_1)$ olduğundan x_1 vektörü y_1 vektörü ile temsil edilmesi gerekir. Benzer şekilde her eğitme vektörü için bozulma ölçümleri hesaplandıktan sonra aynı kod vektörünü kullanan eğitme verileri kullandıkları kod vektörünü kendi ağırlık merkezlerine çekerek, aşağıdaki şekilde optimum kodlamayı gerçekleştirilmiş olur.

Adım 2) Verilen $A_m = (y_i ; i = 1, 2, \dots, N)$ için eğitim dizisinde minimum bozulmayı sağlayacak bölümlenme $P(A_m) = (S_i ; i = 1, 2, \dots, N)$ bulunur. $x_j \in S_i$ eğer $d(x_j, y_i) \leq d(x_j, y_l)$ bütün değerleri için ortalama bozulma hesaplanır.

$$D_m = D[A_m, P(A_m)] = (n-1) \sum_{y \in A_m} \min d(x_j, y)$$

Adım 3) Eğer $(D_{m-1} - D_m) / D_m \leq \varepsilon$ öz yineleme durdurulur ve A_m sonuç tekrar üretilen alfabe olur. Diğer şartta öz yinelemeye devam edilir.

Adım 4) Optimum tekrar üretilen alfabe $X(P(\hat{A}_m)) = (\hat{X}(S_i) ; i = 1, \dots, N)$ bulunur

$$\hat{x}(S_i) = \frac{1}{\|S_i\|} \sum_{j, x_j \in S_i} x_j$$

Adım 5) $\hat{x}(S_i)$, \hat{A}_{m+1} set edilir, $\hat{A}_{m+1} = \hat{x}(S_i)$ ve öz yineleme sayısı artırılır, $m \rightarrow m+1$ ve adım 2'ye dönlür.

Yukarıdaki öz yinelemeli algoritmada başlangıç alfabeti \hat{A}_0 (kod kitabı) algoritmanın başladığı yerdir. Başlangıç kod kitabının belirlenmesi için çok sayıda teknik vardır. En basit teknik eğitim dizisinden kod kitabı boyutu kadar vektör almaktır. Linde bölümlenme (splitting) tekniğini kullanır. Bölümlenme tekniğinde dizinin merkezi (ortalama) hesaplanır ve iki kod vektöre bölünür. Bu vektörler tekrar ikiye bölünür. Bu bölme işlemi N seviyeli başlangıç kod kitabına kadar devam eder. Bölme işlemi her vektöre ε değeri eklenmesi ve çıkarılması ile bulunur. Yani $y_i + \varepsilon$ ve $y_i - \varepsilon$. ε değeri ise eğitim dizisindeki vektörlerin büyüklüğü ile belirlenir [14], [15].

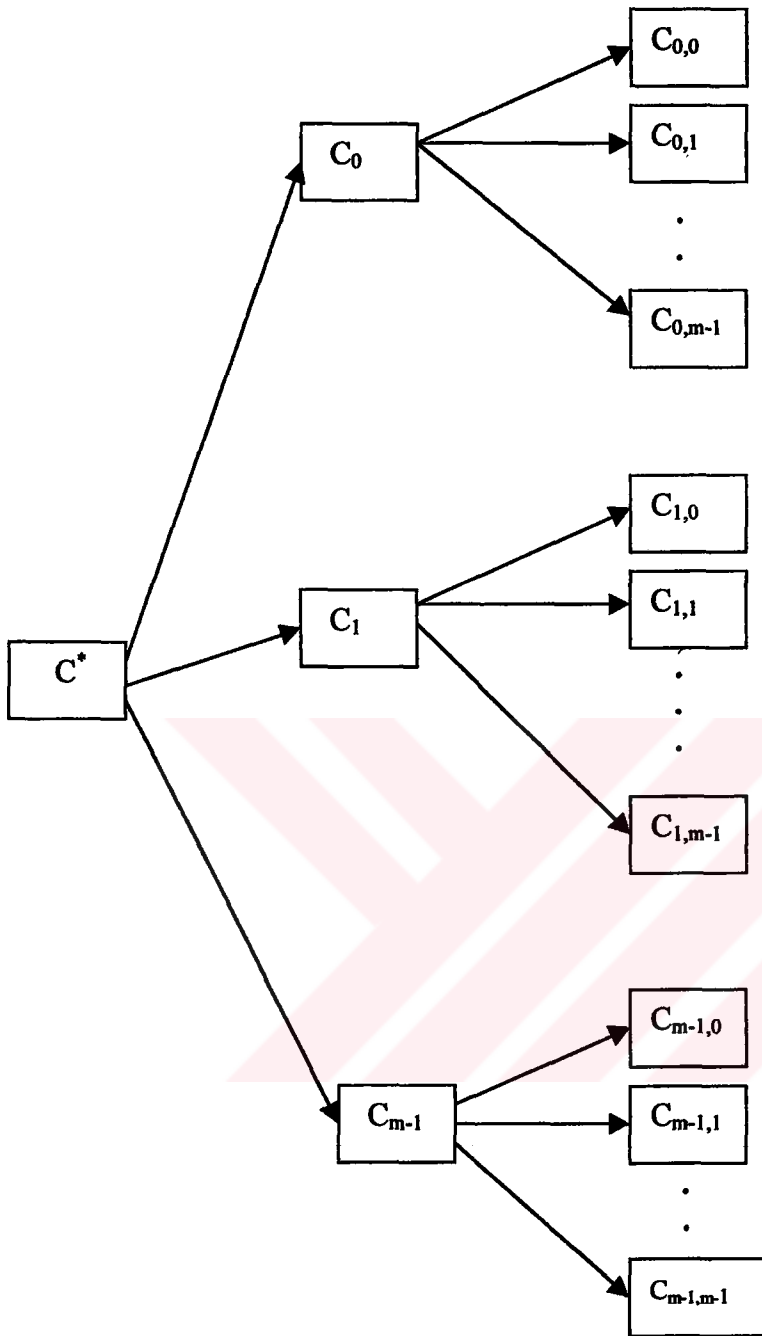
5.5) Vektör Kuantalama Çeşitleri

5.5.1) Ağaç yapılı vektör kuantalama

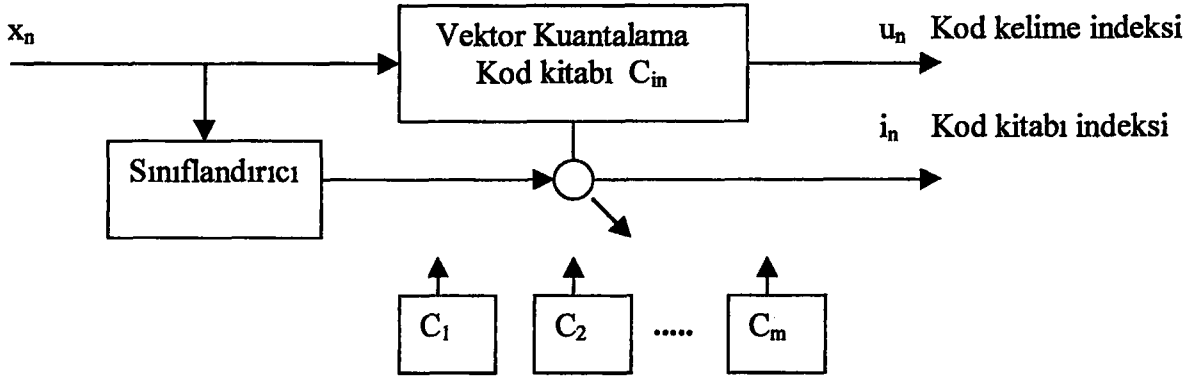
Vektör kuantalamadaki araştırma karmaşıklığını azaltmak için geniş ölçüde kullanılan ve etkili olan bir tekniktir. Ağaç yapılı vektör kuantalama (AYVK) araştırma işini katmanlara bölerek yapar. Her katmanda giriş kod vektörü daha önceden dizayn edilmiş m adet test vektörü (düğüm vektörü) ile karşılaştırarak alt kod vektörlere böler. Giriş kod vektörü en yakın (minimum bozulma) test vektörünün bulunduğu dala konur. Her katmanda vektörler bir önceki katmana göre $1/m$ oranında azalır. Ağaç araştırmanın temel yapısı şekil 5.6 da görülmektedir. Araştırma işlemi uç düğüm vektörüne kadar devam eder. Kodlayıcıdaki ilk araştırma kod kitabı, C^* , ile başlar ve minimum bozulmaya sahip düğüm kod vektörü bulunur. Eğer düğüm kod vektörünün indeksi i ise, ilk sembol i olur ve araştırma işlemi ağacın i . dalında devam eder. C_i kod vektörü ağacın i . dalındaki düğüm kod vektörleri ile karşılaştırarak minimum bozulmaya sahip düğüm kod vektörünün bulunduğu alt dala gider. Araştırma işlemi böyle devam eder [7],[9, sayfa 407-423].

5.5.2) Sınıflandırılmış vektör kuantalama

Sınıflandırılmış vektör kuantalama (SVK) tek katmalı ağaç yapılı vektör kuantalamaya benzemektedir. Sınıflandırılmış vektör kuantalama alt kod kitaplarına bölmek için test vektörleri yerine sınıflandırıcılar kullanmaktadır. Alt küme sayısı ve her kümedeki kod kitabı büyüklükleri değişken olabilmektedir. Şekil 5.7 de sınıflandırılmış vektör kuantalamasının temel yapısı görülmektedir. Sınıflandırıcının seçilmesi için bir çok olasılık vardır. En basit olarak giriş uzayı m bölgeye bölünür. Bu durumda sınıflandırılmış vektör kuantalama ikinci katmanındaki kod kitabı büyüklükleri farklı olan iki katmanlı ağaç yapılı vektör kuantalamadır. Sınıflandırıcı giriş vektörlerinin ortalama (örnek değerlerinin ortalaması), enerjisi (örneklerin kareleri toplamı) ve aralık (örneklerin maksimum ve minimum değerleri arasındaki fark) gibi bir veya daha fazla özelliğine göre sınıflandırır [9, sayfa 423-424] .



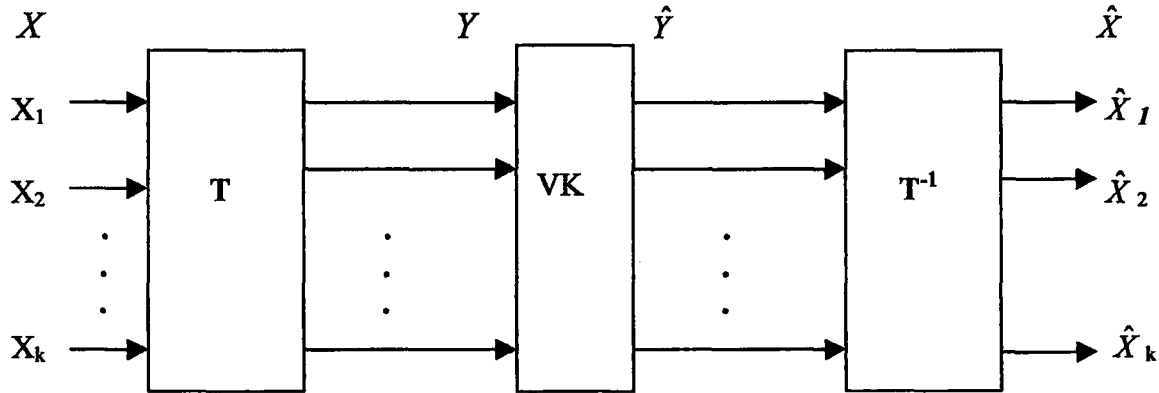
Şekil 5.6) Ağaç yapılı vektör kuantalama kodlayıcının temel yapısı



Şekil 5.7) Sınıflandırılmış vektör kuantalama

5.5.3) Dönüşüm vektör kuantalama

Dönüşüm vektör kuantalama direk giriş vektörünü, X , kuantalamak yerine onu birim dik lineer bir dönüşümden, T , geçirir. Dönüştürmüş vektör, Y , kuantalanır. Kuantalama çıkışına ters dönüşüm uygulanarak orijinal giriş vektörüne benzetilen \hat{X} vektörü elde edilir. Dönüşüm vektör kuantalama şekil 5.8 de görülmektedir. Lineer dönüşümün ana fikri orijinal vektör bilgilerini alt vektör kümelerine sıkıştırmaktır. Örneğin, kesikli kosinüs dönüşümü giriş vektörlerini frekans bölgesine haritalar. Genellikle izgel bantın yüksek frekans bölgesinde daha az enerji vardır. Bu özellikten dolayı giriş vektörlerine kesikli kosinüs dönüşümü uygulandığında katsayılar sıfır frekans etrafında yoğunlaşır ve yüksek frekanstaki bilgiler önemsizlenebilir. Buda kodlanacak vektör boyutlarını azaltmakta ve dolayısıyla vektör kuantalama karmaşıklığı azalmaktadır [5], [9, sayfa 425-434], [16].



Şekil 5.8) Dönüşüm vektör kuantalama

5.5.4) Bölümlenmeli vektör kuantalama

Yüksek boyutlu vektörlerin araştırma ve hafızalama problemini azaltmanın en etkili yöntemi giriş vektörünü alt vektörlere bölünmesidir.

Giriş vektörü,

$$X = (X_1, X_2, \dots, X_k) \quad \text{ise}$$

$$X_a = (X_1, X_2, \dots, X_m) \quad k < m \text{ olmak üzere } m\text{-boyutlu}$$

$$X_b = (X_{m+1}, X_{m+2}, \dots, X_k) \quad k - m \text{ boyutlu vektörlere bölünür.}$$

Hataların kareside;

$$\|X - \hat{X}\|^2 = \|X_a - \hat{X}_a\|^2 + \|X_b - \hat{X}_b\|^2$$

formülü ile bulunur.

Kodlayıcı X_a giriş vektörü için C_a ve X_b giriş vektörü içinde C_b kod kitabı oluşturur. Optimum kod kitabı dizaynı için eğitim seti de T_a (m -boyutlu) ve T_b ($k-m$ boyutlu) bölünür. Bölümleme işlemi 2 veya daha fazla bölümlenmede yapılabilir [9, sayfa 434].

5.5.5) Ortalaması çıkartılmış vektör kuantalama

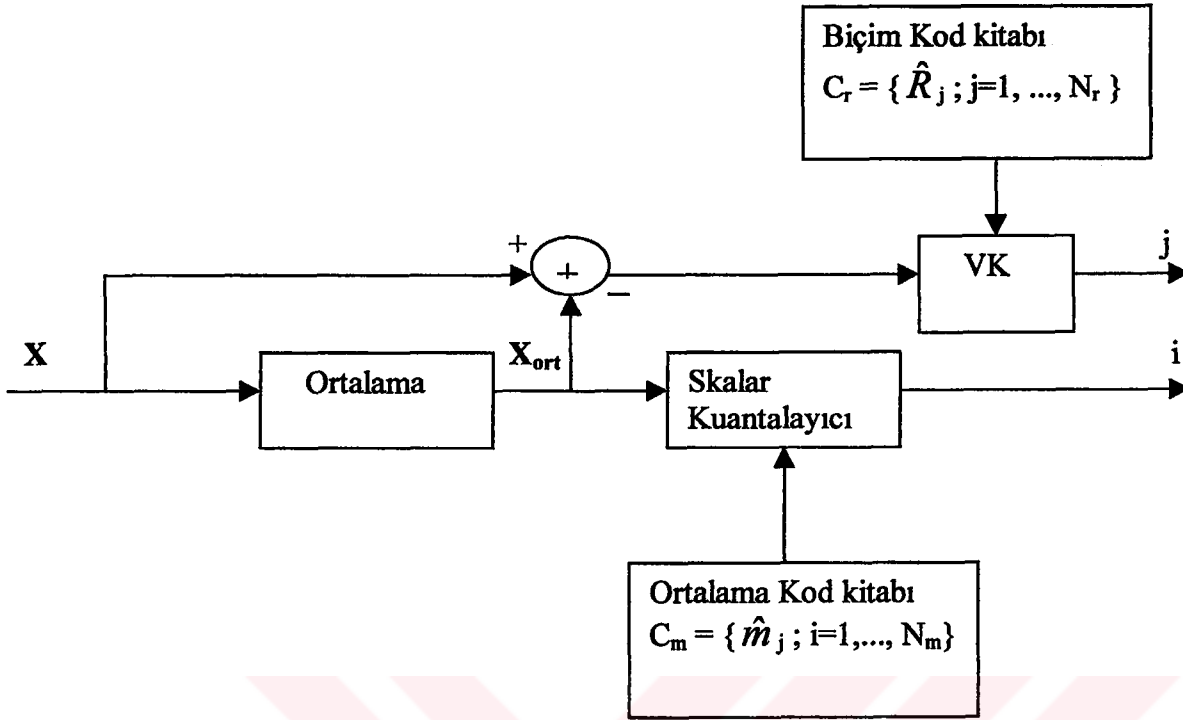
Her elemanın beklenen değeri sifıra eşit olan vektöre istatistiksel sıfır ortalamaya sahip vektör denir. Bununla birlikte, vektör elemanların ortalama değerleri vektörden vektöre çeşitlilik göstermektedir. Ayrıca, görüntüden elde edilen vektör elemanlarının değeri pozitif olmakla birlikte ortalaması sıfırdan farklı çıkmaktadır. Her bloğun lokal ortalaması da görüntü boyunca değişmektedir. Bir vektörün ortalama vektörü de,

$$X_{ort} = \frac{1}{k} \sum_{l=1}^k X_l \quad (5.14)$$

ile bulunur. Ortalaması çıkarılan artık vektör de,

$$R = X - X_{ort} \quad (5.15)$$

ile bulunmaktadır. Artık vektör sıfır ortalamaya sahiptir. Ortalama çıkarma işlemi görüntü vektörlerinin iki özelliğini ayırma işlemidir. Ortalama görüntüdeki arka plan seviyesini, artık vektörde görüntünün biçimini göstermektedir. İki özelliğin ayrı olarak kuantalanmasına da ortalaması çıkartılmış vektör kuantalama denir ve şekil 5.9 da görülmektedir. Ortalama skalar, artık vektör de vektörel kuantalanır [9, sayfa 435-411].



Şekil 5.9) Ortalaması çıkartılmış vektör kuantalama

5.5.6) Öngörücülü vektör kuantalama

Hafızasız vektör kuantalamada her giriş vektörü önceki (veya sonraki) vektörle bağıntısına bakılmaksızın kuantalanır. Bu vektör sinyalin sınırlı bir bölümünden veya birbirine yakın örnek kümelerinden oluşur. Her vektör aynı olasılık yoğunluk fonksiyonuna sahip olduğu kabul edilir, ancak gerçek görüntülerdeki vektörler birbirleriyle istatistiksel olarak bağımlıdır. Vektörler veya bloklar arası bağımlılığının kullanılması ile aynı bit oranında daha iyi performans elde edilmekle birlikte araştırma karmaşıklığını azaltır. Öngörücülü vektör kuantalamada skalar öngörücülü kuantalamadaki skalar kuantalama ve skalar öngörücü yerine vektör kuantalama ve vektör öngörücü kullanılır. Genel şekli 5.10 daki gibidir. Skalar kuantalamada olduğu gibi giriş vektörü ile öngörülen vektör arasındaki fark kuantalanır [9, bölüm 13], [17].

5.5.7) Adres vektör kuantalama

Hafızasız vektör kuantalama komşu pikseller arasındaki ilintiyi azaltır, ancak komşu bloklar arası ilintiyi dikkate almaz. Adres vektör kuantalamanın temeli komşu blokları gruplayıp büyük grup blokları oluşturmaktır. Pratik uygulamalar için 4 blok gruplanır. Adres kod kitabı ve kod vektörlerin adresi (indeksi) , kod kitabının ilk kısmında ufak bloklara LBG algoritmasıyla üretilmiş olan kod kitapları bulunmaktadır. Kod kitabının ikinci kısmında ise bu kod kitabındaki kod vektörlerinin adresleri bulunur. İkinci kısmında eğitime sırasında bulunan bütün olası adres gruplamaları bulunur. Komşu bloklar birbirleri ile yüksek ilintili olduğu için eğitime sırasında olası blok kombinasyonları çok fazladır, ancak sınırlıdır. Kodlayıcıda her ufak bloğa bit atamak yerini 4'ü bir araya getirilmiş büyük bloklara bit ataması yapılarak gerekli olan bit sayısını azaltır.

Adres kod kitabını bulmak için eğitime setindeki görüntüler küçük bloklara bölünür. Her blok LBG algoritması ile kuantalanır, ve 4 komşu bloğun bütün olası adres kombinasyonları bulunur. Adres kod kitabı genellikle çok büyüktür, bundan dolayı iki bölgeye ayrılır; aktif ve aktif olmayan bölge. Eğitime sırasında yan yana gelme olasılığı fazla olan blokların adresi aktif bölgede, yan yana gelme olasılığı az olan blokların adresi aktif olmayan bölgede tutulur. Yalnızca aktif bölge kodlayıcı ve kod çözücüde bulunur. Eğer blok kombinasyonu aktif bölgede ise yalnızca indeksi, değilse o bloktaki kod vektörlerde indeksle beraber gönderilir [18].

BÖLÜM 6

LBG ALGORİTMASI

İyi bir bloklama veya vektör kuantalama için etkili algoritmalar geliştirilmiştir. En popüler ve bilinen vektör kuantalama algoritması LBG algoritmasıdır. Linde – Buzo – Gray uzaydaki bir grup noktanın tek bir noktaya uzaklıkları toplamının minimum olması için o tek noktanın grup noktaların merkezinde olması gerektiğini söylemektedirler. Bundan dolayı, görüntü vektörleri ile bu görüntü vektörlerini temsil edecek merkez vektörü arasındaki geometrik mesafenin tanımlanması gerekir. İki görüntü vektörü arasındaki geometrik mesafenin tanımı ise;

$$H_{kn} = \sqrt{(x_k[i] - x_n[i])^2} \quad (6.1)$$

Burada;

n vektörlerin boyutudur.

Görüntü vektörleri ile bu vektörleri temsil eden vektörün arasındaki toplam mesafe ise

$$H_{ktop} = \sum_{j=0}^{M_k} H_{kj} \quad (6.2)$$

Burada;

M_k k grubundaki vektör sayısı

Vektör kuantalamadaki 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. Bundan dolayı toplam bozulma ise;

$$H_{Top} = \sum_{k=0}^{N-1} \sum_{j=0}^{M_{k-1}} \left(\sum_{i=0}^{k-1} (x_j[i] - \hat{x}_k[i])^2 \right)^{1/2} \quad (6.3)$$

Burada;

x_k : k. Kod vektörü

x_j : k. Gruptaki j. Vektör

Vektör gruplarının merkezi ise ;

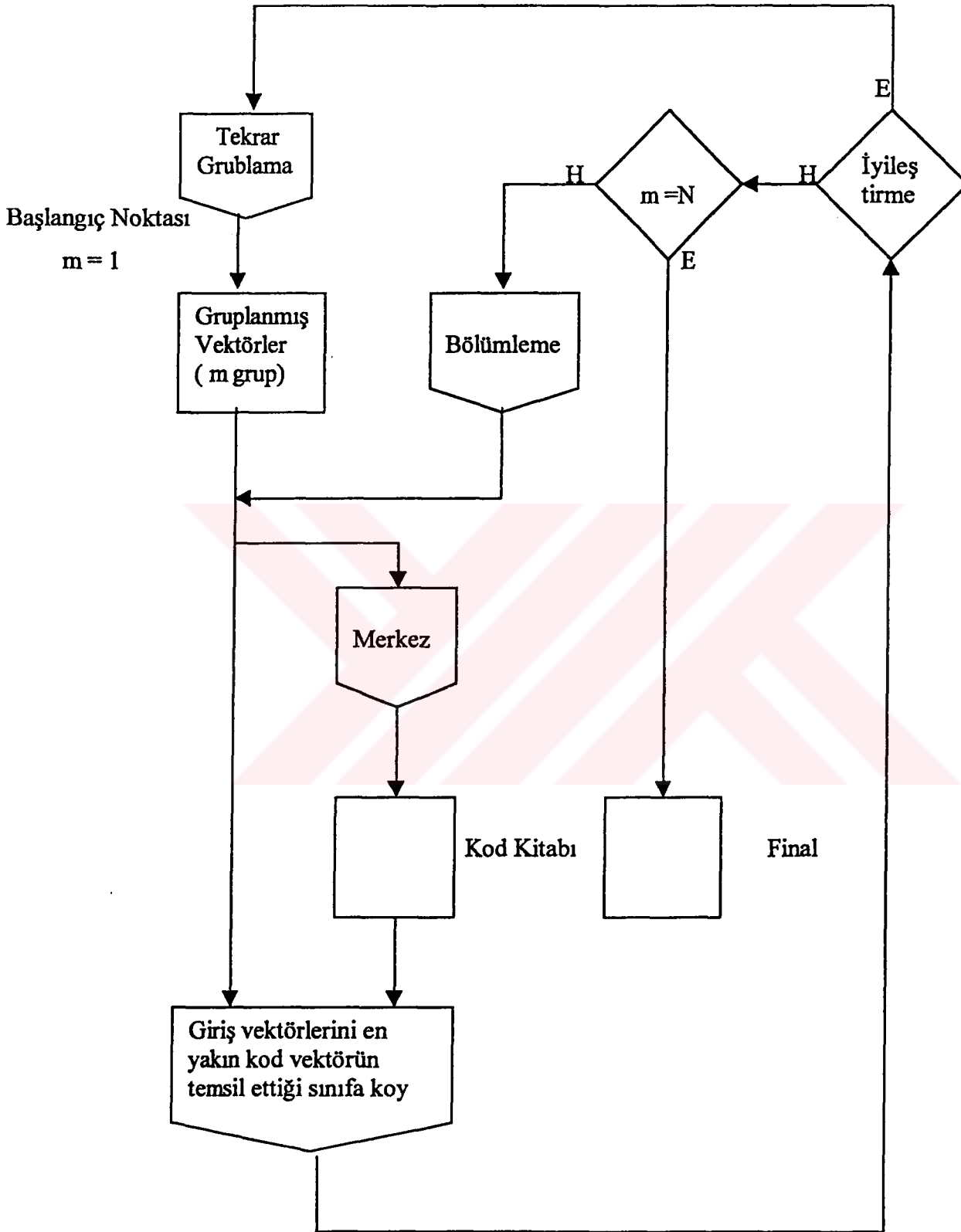
$$x_k[i] = \frac{1}{M_k} \sum_{j=0}^{M_{k-1}} x_j[i] \quad (6.4)$$

ile hesaplanır. LBG algoritması eşitlik 6.3' ü vektörleri düzenleyerek minimum yapmaya çalışır. LBG algoritması öz yinelemeli bir algoritmadır. Bütün görüntü vektörlerini tek bir kod vektör ile temsil edilerek başlanır ve belirlenen kod vektör sayısına ulaşıncaya kadar devam eder. Başlangıçta bütün eğitime dizileri tek bir sınıfta olduğu düşünülür. Her bir öz yinelemede sınıf sayısı ve kod vektör sayısı ikiye katlanır ve eğitime vektörleri hangi sınıfa ait olduğu ve sınıfları temsil edecek kod vektörleri ise o sınıfın ağırlık merkezi olarak hesaplanır. Tekrardan eğitime vektörleri hangi kod vektöre yakınsa o kod vektörün temsil ettiği sınıfa konular ve tekrar sınıfların ağırlık merkezleri hesaplanarak kod vektörlerinin uzaydaki yerleri değiştirilir. Bu işlem herhangi bir eğitime vektörünün sınıf değiştirmemesine kadar devam eder. Bütün vektörlerin en yakın sınıfta olması final 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 [13, sayfa 16-20], [15].

LBG algoritmasının adımları;

- Adım 1)** Bütün eğitim vektörleri ağırlık merkezi bulunur.
- Adım 2)** Bütün eğitim vektörleri en yakın sınıfa konur.
- Adım 3)** Her sınıfın ağırlık merkezi hesaplanır. Bu ağırlık merkezi o sınıfı temsil edecek kod vektördür.
- Adım 4)** Toplam bozulma hesaplanır.
- Adım 5)** Bu öz yinelemedeki toplam bozulma miktarı ile bir önceki öz yinelemedeki bozulma miktarı arasındaki fark büyükse 2. adıma dönülür.
- Adım 6)** Kod vektör sayısı yeterli ise öz yineleme durdurulur.
- Adım 7)** Her kod vektör ikiye bölünür.

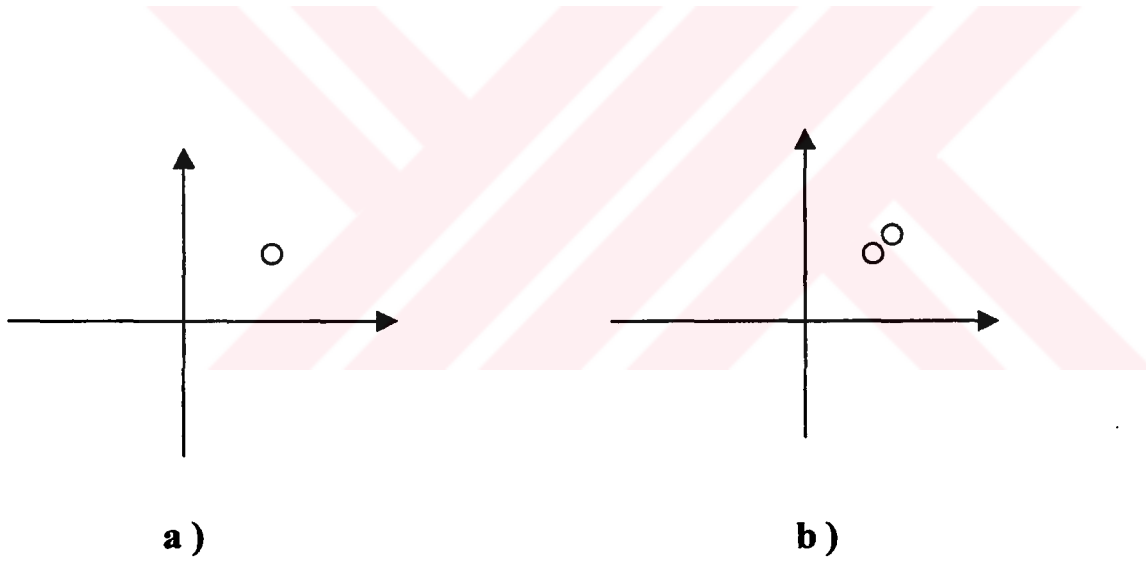


Şekil 6.1) LBG algoritmasının akış diyagramı

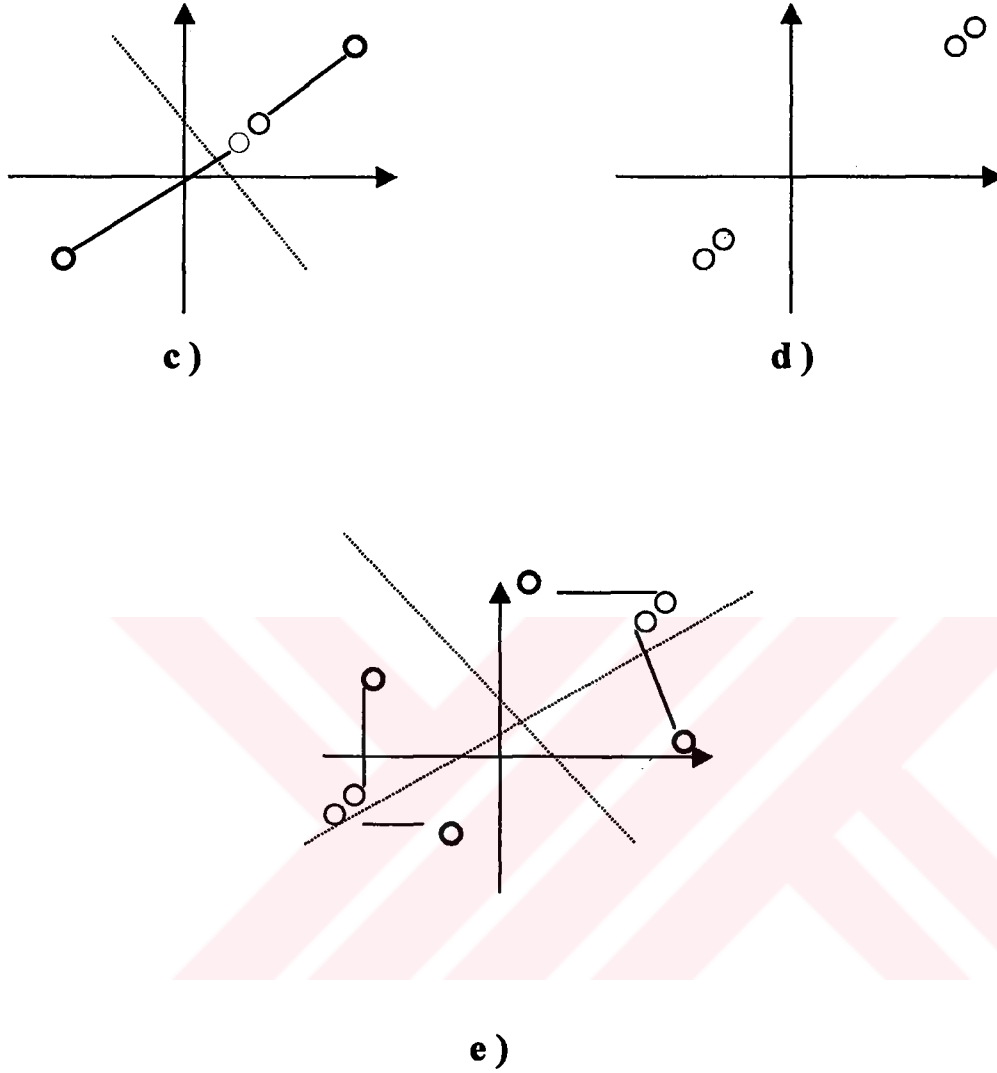
6.1) Ayırma İşlemi

LBG algoritmasının 7. adımında her kod vektörün ikiye ayrıldığından bahsedilmişti. Bu ayırma işleminde her y_i ($i = 0, 1, \dots, N$) kod vektörü $y_i + \epsilon$ ve $y_i - \epsilon$ olmak üzere ikiye ayrılarak, y_i vektörünün yerine iki tane kod vektörü elde edilmektedir. Buradaki ϵ sabit bir vektördür ve veri büyüklüklerine göre uygun olarak seçilmektedir.

Eğitme vektörlerinin $\{ x_j ; j = 0, 1, \dots, k - 1 \}$, üretilecek kod vektörlerinde $\{ y_i ; i = 0, 1, \dots, N - 1 \}$ şeklinde verildiğini düşünelim. Burada ilk kod vektör eğitim vektörlerinin ağırlık merkezi olarak alınmaktadır. Her ayırma adımında $\{ y_i + \epsilon, y_i - \epsilon ; i = 0, 1, \dots, N - 1 \}$ olacak şekilde ikiye ayrılmaktadır. Şekil 6.2 ayırma işleminin aşamaları gösterilmektedir.



Şekil 6.2) Ayırma işlemi örneği (a ve b şekli)



Şekil 6.2) Ayırma işlemi örneği (c, d ve e şekli)

Şekil 6.2 a' da ilk kod vektör eğitme vektörlerinin ağırlık merkezi olarak alınmaktadır. Şekil 6.2 b' de ise başlangıç kod kelimesi ayrılarak iki tane kod kelimesi elde edilmektedir. Şekil 6.2 c' de LBG algoritması ile iki kod kelimesini optimize ederek iyi bir kodlama üretmektedir. 6.2 d' de optimize edilen iki kod kelimesi ayrılarak dört kod kelimesi elde edilmektedir. 6.2 e' de algoritma dört kod kelimesini optimize ederek sonuç kod kelimesini üretmektedir [15], [19].

Kod kitabı elde edebilmek için LBG algoritmasını kullandığımız zaman üç tane sonuç ile karşılaşırız. Bunlardan birincisi, bu algoritmayla üretilen kod kitabı optimum olmayabilir, yani eşitlik 6.3'ü minimum yapacak başka kod vektörler bulunabilir. En son sonuç başlangıç noktasına bağlıdır. Her ne kadar bütün vektörleri temsil eden ilk vektör bütün vektörlerin ağırlık merkezi seçilse de global minimumu sağlayacak başka bir vektör hesaplanabilir. LBG algoritmasının her adımında bir önceki adıma göre optimum sonuca yaklaşır. LBG algoritmasının doğası gereği global optimuma yaklaşılr, çünkü LBG global optimum için bütün olası sınıflamalara bakmaz. Bundan dolayı, LBG algoritması lokal optimumu bulur ve buda çoğu zaman yeterlidir.

Diğer sonuç ise LBG algoritmasının ortalama veya yumuşatma etkisidir. Eşitlik 6.4'de verilen merkez hesaplamasında da görüleceği gibi bütün vektör bileşenlerinin (piksellerin) ortalaması alınır. Bu da imgedeki kenarların yumuşamasına sebep olmaktadır. Aslında gerçek imgelerde bir imgeyi belli eden en önemli kısım kenarlardır. LBG 'nin bu etkisinden dolayı imgenin belirlenebilmesini azalmaktadır.

Üçüncü sonuç ise LBG insan gözünün fiziksel özelliklerini dikkate almaz. Eğer imge insan tarafından izlenecekse insan gözü önemli bir rol oynar. Geometrik veya matematik olarak optimum sonuç çoğu zaman insan gözü için optimum olmaz. Aslında bu etkiyi ölçmek için herhangi bir eşitlik bulmak da kolay değildir [13, sayfa 21-22].

6.2) LBG Algoritmasının Görüntü Dizileri İçin Güncellenmesi:

Veri sıkıştırmada kullanılan yöntemlerden biride vektör kuantalamadır. Bilindiği üzere optimum vektör kuantalama için LBG algoritması kullanılmaktadır. Ancak LBG algoritması öz yinelemeli bir yaklaşımdır. Bundan dolayı gerçekleştirmesi uzun zaman almaktadır. Görüntü dizilerinin ayrı ayrı her çerçevesine LBG algoritması uygulanarak bir sıkıştırma işlemi gerçekleştirilir. LBG algoritması başlangıç kod kitabı ile başlar ve kod kitabını öz yinelemeli olarak düzenleyerek optimum sonucu bulmaya çalışır. Daha öncede 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 çerçeveden elde edilen kod kitabındaki vektörlerin uzaydaki yeri ikinci çerçeve için hesaplandığında çok az yer değişmesi beklenmektedir. Sonraki çerçevelere kod kitabı bulmak için daha önce bulunmuş olan kod kitabı kullanıla bilinir. Bunu için 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. Güncelleme algoritması aşağıda verilmiştir.

Güncelleme algoritması adımları;

Adım 1) LBG algoritması kullanarak kod kitabı bulunur.

Adım 2) Yeni eğitim vektörleri en yakın sınıfa konur. Eğitim dizisi $\{ x_j ; j = 0, 1, \dots, n-1 \}$ olmak üzere;

$$x_j \in C_i \text{ eğer } d(x_j, y_i) < d(x_j, y_N)$$

Adım 3) Her sınıftaki eğitim vektör sayısı bulunur. S_i her sınıftaki eğitim vektör sayısı olmak üzere;

$$s_i = s_i + 1 \text{ eğer } x_j \in C_i$$

Adım 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)

$C^{\text{sonuç}} = C$ olur.

Adım 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.

Adım 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

$$v_i = z_i, 0 \leq i \leq k$$

Adım 7) k sayısı kadar, en uzak vektör kendi sınıflarından çıkartılır. Çıkartılan her vektör yeni bir sınıf oluşturulur.

$$C_i = C_i \quad \text{eğer } s_i \neq 0$$

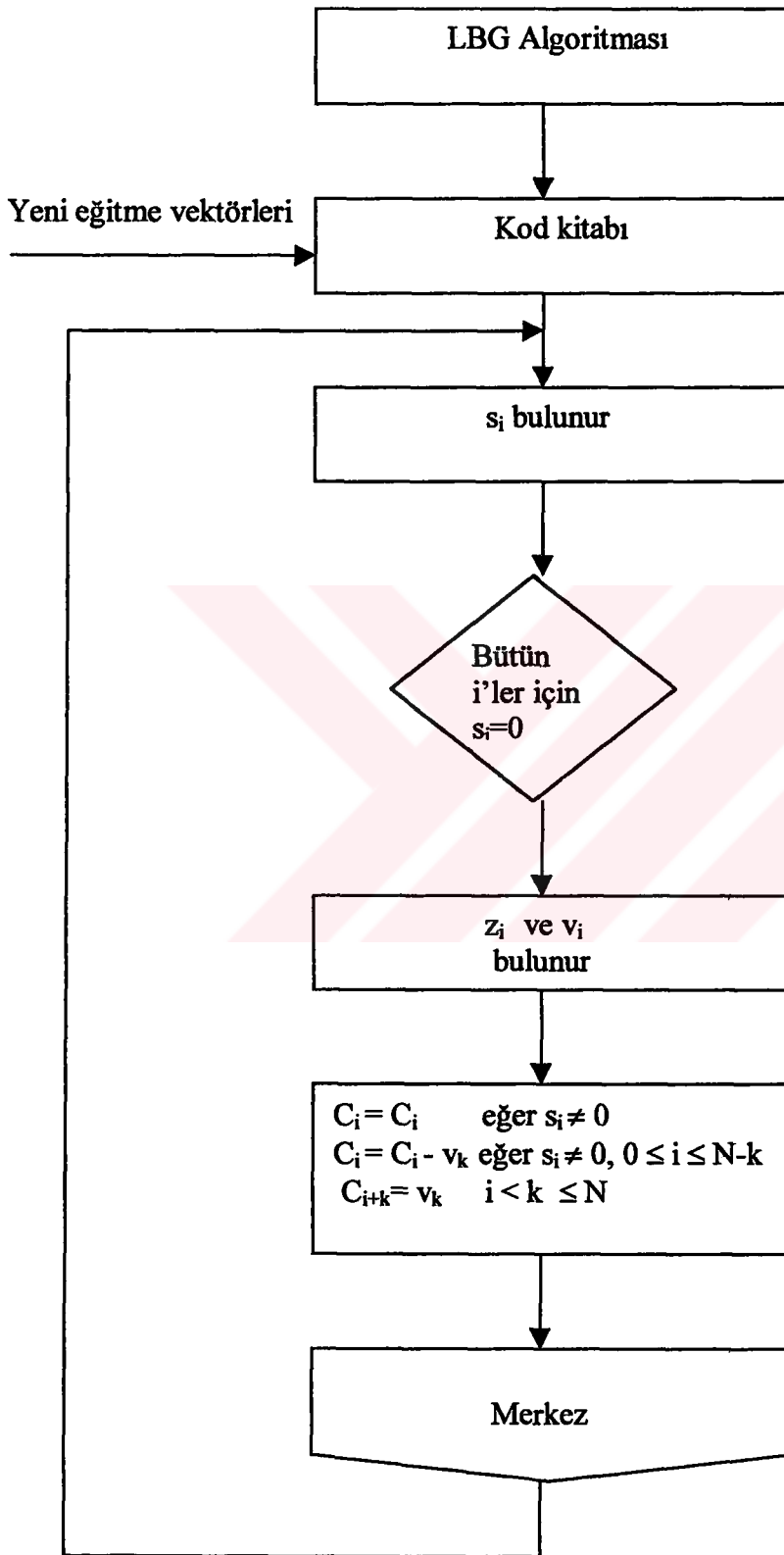
$$C_i = C_i - v_k \quad \text{eğer } s_i \neq 0, 0 \leq i \leq N - k$$

ve

$$C_{i+k} = v_k \quad i < k \leq N$$

Adım 8) Her sınıfın ağırlık merkezi hesaplanır. Bu ağırlık merkezi o sınıfı temsil edecek kod vektörüdür. Adım 2'ye dönülür.

$$\hat{x}(C_i) = \frac{1}{\|C_i\|} \sum_{j, x_j \in C_i} x_j$$



Şekil 6.3) Güncelleme algoritması

BÖLÜM 7

UYGULAMA

Bu çalışmada gri seviyeli durağan görüntüleri LBG algoritması ve gri seviyeli hareketli görüntüleri hem LBG algoritması ve güncelleştirilmiş LBG algoritması kullanarak görüntüler kodlanmıştır. Bütün görüntüler .pgm uzantılıdır. ANSI C programlama dilinde yazılmış programlar Linux işletim sisteminde çalıştırılmıştır.

Durağan görüntüleri LBG algoritması uygulanarak LBG algoritmasının nasıl çalıştığı ve uygulanan görüntülerde nasıl bir etki yaptığına bakılmıştır. 25 adet durağan resmin 16 adeti eğitime görüntüsü, diğerleri ise test görüntüsü olarak ayrılmıştır. Eğitim görüntülerinin boyutları ise 4 tanesi 256x256, 10 tanesi 512x512 ve 2 tane 1024x1024'dür. Her görüntü 8x8 bloklara bölünmüş, dolayısıyla

$$\frac{(4 * 256 * 256 + 10 * 512 * 512 + 2 * 1024 * 1024)}{8 * 8} = 77824$$

adet vektör elde edilmiştir. Kod kitabı büyüklüğü olarak da 1024 seçilmiş, dolayısıyla $\log_2(1024) = 11$ bit ile ifade edilebilmektedir. Sıkıştırma oranı da 11:512 olmaktadır. Çünkü her blok 8x8 = 64 pikselden oluşmakta ve her piksel 8 bit olduğu için her blok 64x8 = 512 bittir. Hataların matematiksel ölçümü içinde sinyal gürültü oranı kullanılmıştır.

$$\text{Sinyal Gürültü Oranı} = 20 \log \sum_{i=0}^N \frac{\text{Orijinal Sinyal}^2}{(\text{Orijinal Sinyal} - \text{Oluşturulmuş Sinyal})^2}$$

Hareketli görüntüler içinde 39 çerçeveden oluşan tenis, 60 çerçeveden oluşan amerikan futbolu ve 99 çerçeveden oluşmuş 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 ve 1024 seçilmiştir. Güncelleme algoritmasında ise her üç görüntü dizisinin ilk çerçevesinden elde edilen kod kitabı kullanılarak sonraki kod kitapları

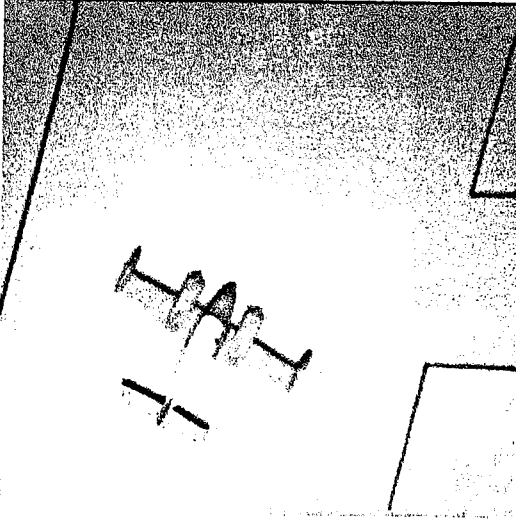
bulunmuştur. LBG algoritması ile güncelleme algoritmasının karşılaştırılabilmesi için sinyal gürültü oranları tablo halinde verilmiş ve her kod kitabı büyüklükleri içinde toplam sinyal gürültü oranlarının grafikleri gösterilmiştir. Hareketli görüntü sayısı fazla olması nedeniyle her çerçevenin görüntüsü yerine belli çerçevelerin görüntüsü verilmiştir.

7.1) Durağan Görüntüler

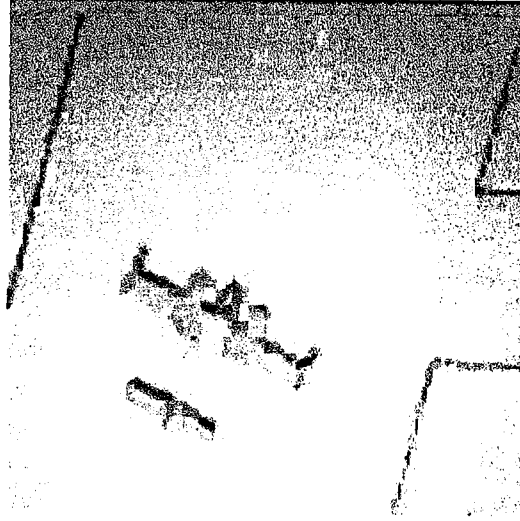


Şekil 7.1.a) Orijinal görüntü (Lena 512x512)

Şekil 7.1.b) LBG algoritması sonucu
(0.17 bit / piksel, snr = 39.98)



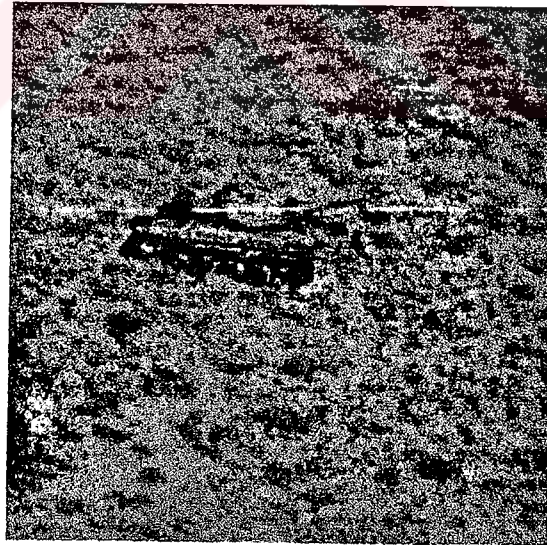
Şekil 7.2.a) Orijinal görüntü (Uçak 256x256)



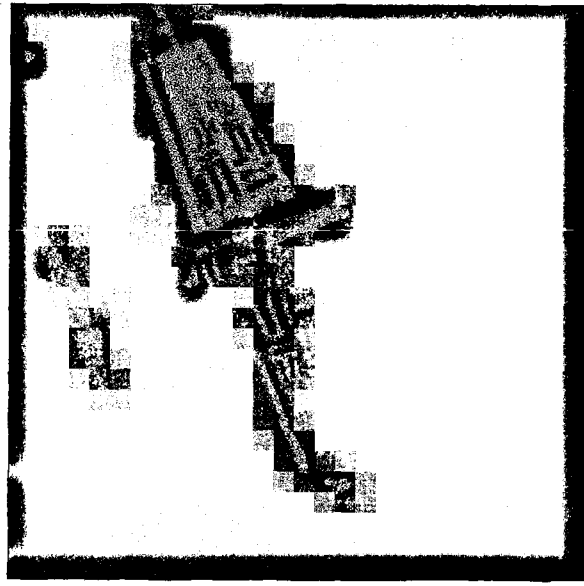
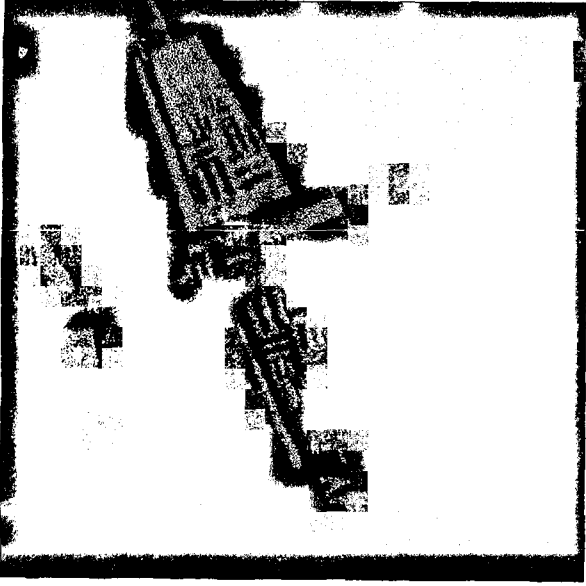
Şekil 7.2.b) LBG algoritması sonucu
(0.17 bit / piksel, snr = 51.12)



Şekil 7.3.a) Orijinal görüntü (Tank 512x512)



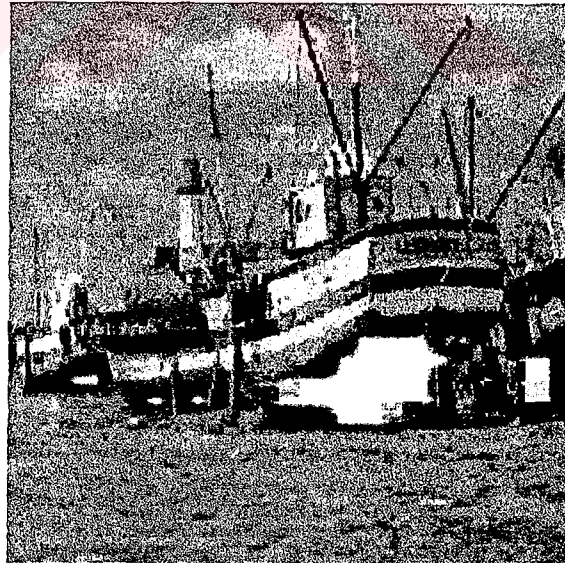
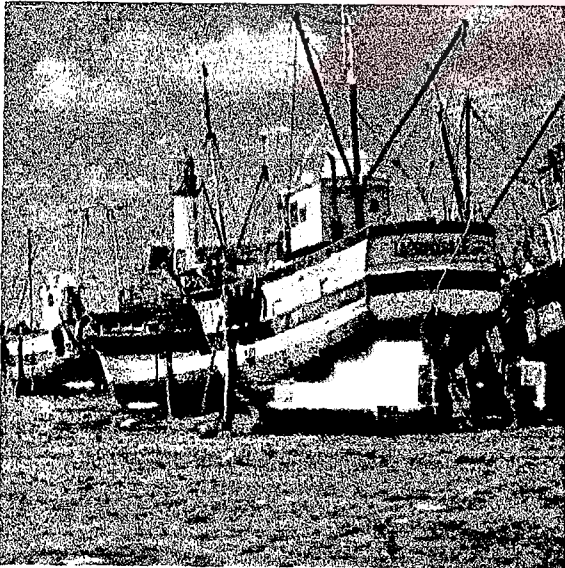
Şekil 7.3.b) LBG algoritması sonucu
(0.17 bit / piksel, snr = 39.58)



Şekil 7.4.a) Orijinal görüntü (SavaşUçağı 1024x1024)

Şekil 7.4. b) LBG algoritması sonucu

(0.17 bit / piksel, snr = 31.58)



Şekil 7.5.a) Orijinal görüntü (Boat 512x512)

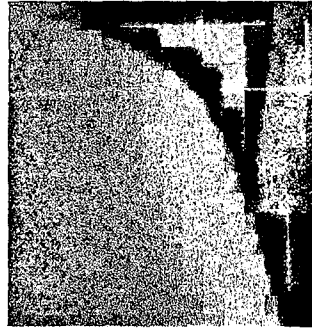
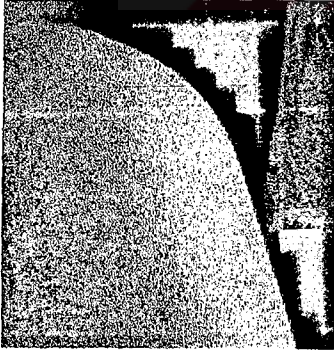
Şekil 7.5.b) LBG algoritması sonucu

(0.17 bit / piksel, snr = 39.98)



Şekil 7.6.a) Orijinal görüntü (Elaine 512x512) **Şekil 7.6.b)** LBG algoritması sonucu
(0.17 bit / piksel, snr = 46.84)

LBG algoritmasının etkilerini görebilmek için Lena görüntüsünün omuz ve göz kısmıyla, boat görüntüsünün orta kısmı büyütülerek verilmiştir.



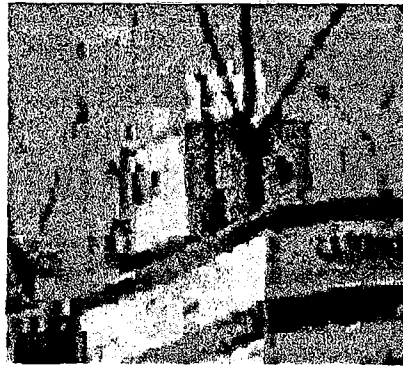
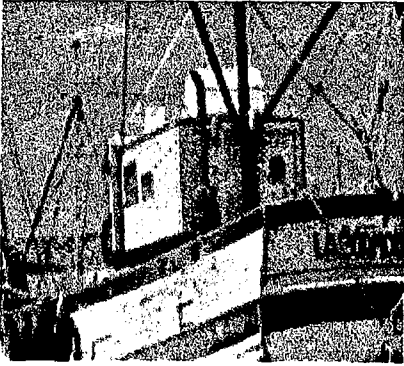
Şekil 7.7.a) Orijinal Lena omuzu

Şekil 7.7.b) LBG Algoritması sonucu



Şekil 7.8.a) Orijinal Lena gözü

Şekil 7.8.b) LBG algoritmasının sonucu



Şekil 7.9.a) Orijinal 'boat'ın orta kısmı

Şekil 7.9.b) LBG algoritmasının sonucu

LBG algoritması bloklar içindeki ilintiyi azaltır, ancak bloklar arası ilintiyi dikkate almaz. Dolayısıyla tekrar oluşturulan görüntülerde blok etkileri belirgin olarak görülmektedir. Şekil 7.7, 7.8, 7.9 da görüldüğü gibi blok etkileri çeşitli filtrelerle azaltılabilmekte, ancak bu sadece görüntünün izlenebilirliği artırmakta fakat sinyal gürültü oranı azalmaktadır. LBG algoritmasının bir başka etkisi de yumuşatma etkisidir. Yumuşatma etkisi yukarıda verilen şekillerde belirgin olarak görülebilmektedir.

7.2) Hareketli Görüntüler

7.2.1) Tenis

39 çerçevelik gri seviyeli tenis dizilerine LBG ve güncellenmiş LBG algoritması uygulanmış ve sonuçları aşağıda verilmiştir.

Çerçeve Sayısı	Sinyal Gürültü Oranları					
	Kod Kitabı 2		Kod Kitabı 4		Kod Kitabı 8	
	Büyüklüğü	Büyüklüğü	Büyüklüğü	Büyüklüğü	Büyüklüğü	Büyüklüğü
	2	2	4	4	8	8
	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme
1	41.7164	41.7164	45.6479	45.6479	60.0330	60.0330
2	41.6161	41.6057	45.3213	45.4984	59.3140	59.3591
3	41.7150	41.6353	45.9293	45.5471	60.2148	59.4821
4	41.6992	41.6761	45.3182	45.6532	59.3442	59.6311
5	41.8661	41.7998	48.3040	45.8331	56.6577	59.5991
6	42.0292	41.8515	48.4388	45.8208	58.7607	59.4398
7	41.9307	41.8284	45.1344	45.8221	58.3289	59.3600
8	42.0476	41.9047	45.2852	45.8990	58.6964	59.5385
9	42.1743	42.0329	48.1748	46.0566	58.5773	59.3545
10	42.1703	42.0977	48.2771	46.2376	58.0390	59.3700
11	42.3710	42.2554	46.9457	46.1260	58.9046	59.0043
12	42.4003	42.3029	46.6545	46.0625	59.5469	58.9122
13	42.4213	42.3018	46.2242	46.0042	59.1204	58.9593
14	42.4262	42.3161	46.5625	46.0993	58.9475	58.9187

15	42.3815	42.3346	47.3796	46.1363	58.9283	58.8300
16	42.3291	42.2914	47.3057	46.0597	55.1564	58.7593
17	42.4918	42.4259	46.5637	46.1378	58.7847	58.6335
18	42.4377	42.3861	47.3344	46.1470	58.7800	58.8825
19	42.5637	42.4743	46.5697	46.1895	57.5723	58.5944
20	42.5738	42.4674	45.8337	46.1823	58.4580	58.4330
21	42.4579	42.4043	45.7727	46.2191	58.2343	58.3606
22	42.7459	42.6185	45.6899	46.2963	58.5044	58.4994
23	42.7939	42.6316	45.7824	46.3550	58.9317	58.4678
24	42.8957	42.8004	47.4557	46.6339	58.7824	58.7050
25	43.1055	42.9385	47.4131	47.0257	59.9240	58.9493
26	43.0594	42.9263	47.1662	47.0897	59.2343	58.3972
27	43.1694	43.1096	47.0129	47.0889	58.9006	58.2565
28	43.1424	43.0665	47.0254	47.2201	59.1588	58.3212
29	42.9438	42.9254	46.6889	47.0554	59.1288	57.7755
30	42.8861	42.8113	46.6683	46.8653	58.4128	57.1763
31	42.9166	42.9060	46.5966	47.0799	58.2995	57.2007
32	42.8251	42.8143	46.4626	46.9936	57.8328	56.9646
33	42.7966	42.7686	46.3884	47.0175	57.8858	56.7287
34	42.8078	42.7792	46.4356	47.0370	57.5706	56.6334
35	42.8697	42.8064	46.3549	47.2116	57.6245	56.4300
36	43.3178	43.2322	47.2524	47.9664	58.9111	57.7990
37	43.4234	43.2774	56.6795	48.5349	59.2500	58.0848
38	43.6868	43.5561	56.6297	49.3780	58.9053	58.1569
39	43.7506	43.5976	56.1433	49.6997	59.4694	57.9664

Tablo 7.1.a) Tenis görüntü dizisinin sinyal gürültü oranları

Çerçeve Sayısı	Sinyal Gürültü Oranları					
	Kod Kitabı	Büyüklüğü	Kod Kitabı	Büyüklüğü	Kod Kitabı	Büyüklüğü
	16	16	32	32	64	64
	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme
1	62.5940	62.5940	68.6048	68.6048	82.4383	82.4383
2	62.2737	62.0995	68.3854	68.4977	82.1065	81.9448
3	62.5096	62.0303	66.7598	68.0730	82.5124	82.4595
4	62.3278	62.2160	67.5821	67.9924	83.1691	82.9068
5	62.9210	62.1269	69.7414	68.6521	83.0207	82.6681
6	62.5379	62.0845	69.2593	68.0169	82.5637	82.1482
7	62.4877	61.7835	67.7388	67.6205	82.2416	81.8824
8	62.5028	62.1636	66.7501	67.3958	82.6089	82.4357
9	62.4451	61.9568	69.1276	67.6239	83.2731	82.7912
10	62.6206	61.9392	68.8986	67.4126	83.2849	82.7757
11	61.7149	61.4644	66.3489	66.1828	82.6222	82.2374
12	62.0358	61.5427	67.5592	66.7090	82.4159	82.4085
13	62.1570	61.6751	66.7926	66.9037	82.1639	82.1599
14	62.1026	61.6393	64.7500	66.2818	83.0579	82.7837
15	62.1886	61.4189	64.4025	66.1097	82.6772	82.5010
16	61.7667	61.2948	67.4205	66.5023	82.5834	82.1328
17	61.8534	61.3192	64.2926	66.2862	81.7985	81.68546
18	61.8441	61.5123	64.1689	65.6172	81.5983	81.5070
19	61.9871	61.0436	66.9080	65.0673	82.3382	81.7517
20	60.2200	60.8364	65.9962	65.5837	82.5458	82.0092
21	61.2393	60.7132	66.4160	64.9551	82.5475	82.1930
22	61.0665	60.9602	65.4372	65.3407	82.3598	81.6585
23	61.2964	60.9090	65.4987	65.3007	82.1053	81.7407
24	62.0133	61.1058	66.6067	64.9790	83.7363	83.6164
25	62.0871	61.2397	66.2092	65.5168	86.6184	86.4372
26	60.2734	60.8976	64.0389	65.4100	88.9325	88.7201

27	59.9065	60.8124	63.5999	65.6552	91.2256	90.9352
28	61.9308	61.1289	65.1364	65.2993	92.8035	91.9291
29	60.9990	60.2793	65.9068	65.4633	93.0077	92.3340
30	61.2904	59.8424	67.1981	64.6104	93.9620	92.6708
31	59.2641	59.8036	64.4114	64.6463	95.1889	94.3032
32	59.8856	59.8913	64.4413	64.6165	98.1656	96.2607
33	60.2649	59.4375	63.7002	64.1712	98.4133	96.8084
34	54.9549	59.8454	64.2228	64.3006	99.5119	98.6401
35	58.6324	59.2559	63.9968	64.1307	100.1009	98.9835
36	61.2769	60.4643	65.1831	64.2754	101.7889	100.8814
37	64.5121	60.5068	66.9111	64.5180	103.5337	101.6898
38	64.4033	60.6672	66.5003	63.6597	104.2417	102.1249
39	64.2190	60.2385	65.7014	64.8840	105.3344	103.4620

Tablo 7.1.b) Tenis görüntü dizisinin sinyal gürültü oranları

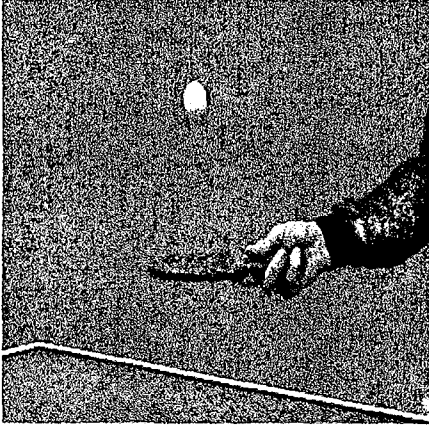
	Sinyal Gürültü Oranları					
	Kod Kitabı		Büyükklüğü		Kod Kitabı	
	Büyükklüğü		Büyükklüğü		Büyükklüğü	
Çerçeve Sayısı	128	128	256	256	512	512
	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme
1	84.9743	84.9743	87.7420	87.7420	97.1276	91.3196
2	84.7345	84.7142	87.5916	87.5393	97.0081	91.2926
3	85.0857	85.0380	87.5818	87.1290	97.2761	92.1310
4	85.9168	85.5541	88.4542	88.4065	97.6301	92.2270
5	85.4810	85.2586	88.0338	87.8130	97.2308	91.8289
6	85.0123	84.8045	87.7774	87.6117	96.9031	91.3854
7	84.8781	84.5007	87.4454	87.4340	97.0504	91.1899
8	85.2497	84.7516	87.8165	87.8057	97.5499	91.5729
9	85.7974	85.5015	88.2053	88.1711	97.9680	92.0176

10	85.9533	85.4973	88.5966	88.4219	98.1938	92.0372
11	85.3628	84.9032	87.9320	87.4456	97.1327	91.3677
12	84.8571	84.0176	87.7690	87.6954	97.0509	91.4686
13	84.6103	84.6032	87.2635	87.1859	97.0784	91.4530
14	85.5083	85.4697	88.2221	88.2156	98.3319	92.0612
15	85.4940	84.2289	88.1649	88.0242	98.3267	91.9197
16	85.1827	84.7309	87.7461	87.4216	97.3992	91.0838
17	84.5397	84.5476	87.3070	87.2213	96.8811	91.1080
18	84.2064	84.1618	86.7782	87.2186	96.6651	90.9879
19	84.7532	84.3999	87.4846	87.3584	97.0700	91.3153
20	85.0599	84.6753	87.9226	87.5380	97.3399	91.3333
21	85.2701	84.9662	87.9489	87.8864	97.4473	91.6337
22	85.0142	84.4135	87.5878	87.3839	96.8773	90.9521
23	84.8940	84.4572	87.4638	87.4264	96.7836	91.2214
24	86.2831	86.2630	88.7938	88.2525	98.7396	92.9618
25	89.1325	88.6308	91.7792	91.6673	101.3098	95.9433
26	91.4159	91.4007	94.3458	94.2844	104.0441	98.5710
27	93.9353	93.0719	96.8590	96.3148	106.9338	101.2851
28	95.5024	95.1615	98.5265	98.4615	109.2353	103.2202
29	96.2347	94.8460	98.9923	98.5427	110.3042	103.5934
30	96.6191	96.2891	99.6263	99.5610	111.0533	104.4842
31	98.3182	97.9963	101.4623	101.0483	113.1190	106.4667
32	101.3474	101.0028	104.8538	104.8168	116.3924	109.5848
33	101.7357	101.7186	105.4789	105.4069	117.4083	110.3040
34	103.0368	102.2908	106.7172	106.1193	119.4394	112.0933
35	103.7669	103.7085	107.6045	107.5273	120.5900	113.0926
36	105.5524	105.4418	109.4101	109.2871	122.7297	115.0622
37	107.6633	106.6645	111.7349	110.7842	124.9784	116.5224
38	108.3871	107.0995	112.8437	111.8811	126.1050	117.1081
39	109.6808	108.1068	114.2662	112.8653	127.7345	118.9376

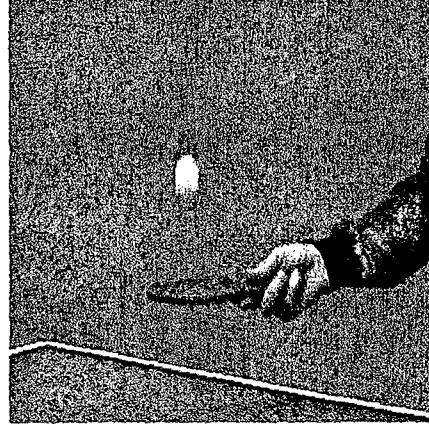
Tablo 7.1.c) Tenis görüntü dizisinin sinyal gürültü oranları

Çerçeve Sayısı	Sinyal Gürültü Oranı				
	Kod Kitabı Büyüklüğü		Çerçeve Sayısı	Kod Kitabı Büyüklüğü	
	1024 LBG	1024 Güncelleme		1024 LBG	1024 Güncelleme
1	97.1276	97.1276	21	97.9006	97.4473
2	97.5120	97.0081	22	97.2134	96.8773
3	98.3601	97.2761	23	97.4255	96.7836
4	98.4866	97.6301	24	99.2102	98.7396
5	98.2664	97.2308	25	102.1282	101.3098
6	97.8255	96.9031	26	105.1924	104.0441
7	97.4766	97.0504	27	108.0433	106.9338
8	97.9075	97.5499	28	109.9531	109.2353
9	98.2534	97.9680	29	110.7314	110.3042
10	98.3658	98.1938	30	111.5144	111.0533
11	97.6119	97.1327	31	113.7862	113.1190
12	97.7245	97.0509	32	117.0652	116.3924
13	97.5812	97.0784	33	118.0899	117.4083
14	98.4204	98.3319	34	120.1294	119.4394
15	98.2909	98.2267	35	121.4084	120.5900
16	97.4809	97.3992	36	123.8084	122.7297
17	97.4189	96.8811	37	124.9788	124.9784
18	97.0752	96.6651	38	125.7629	125.1050
19	97.2637	97.0700	39	128.0935	127.7345
20	97.5756	97.3399			

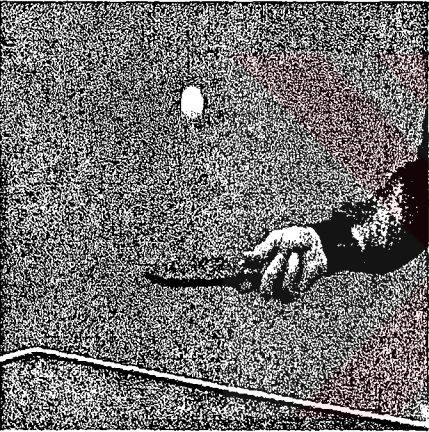
Tablo 7.1.d) Tenis görüntü dizisinin sinyal gürültü oranları



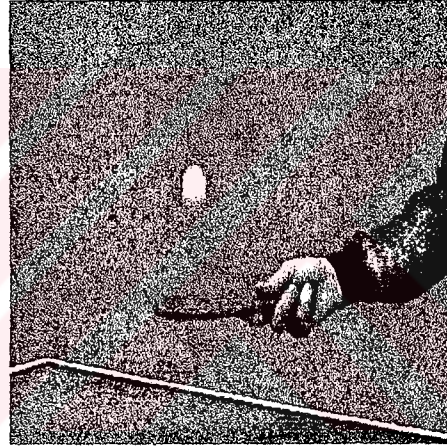
Şekil 7.10.a) 1.çerçeve orijinal görüntü



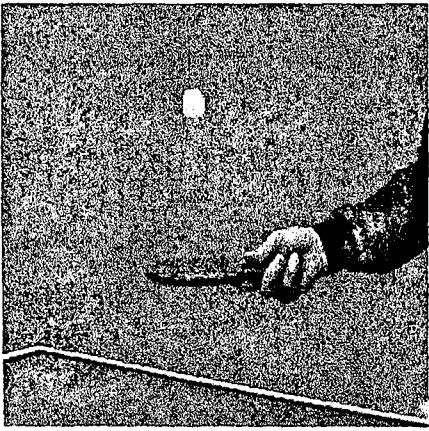
Şekil 7.11.a) 10. çerçeve orijinal görüntü



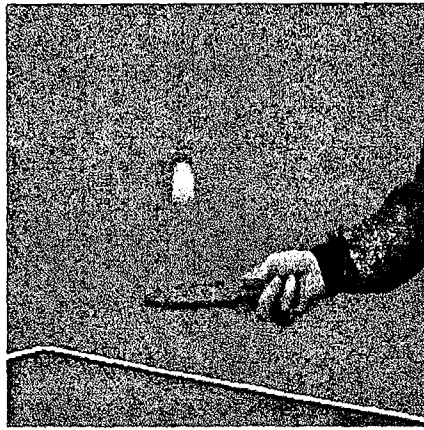
Şekil 7.10.b) LBG algoritması sonucu



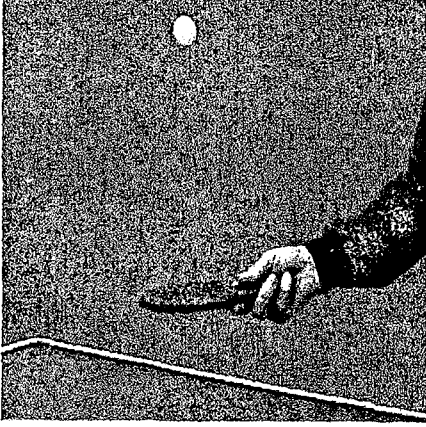
Şekil 7.11.b) LBG algoritması sonucu



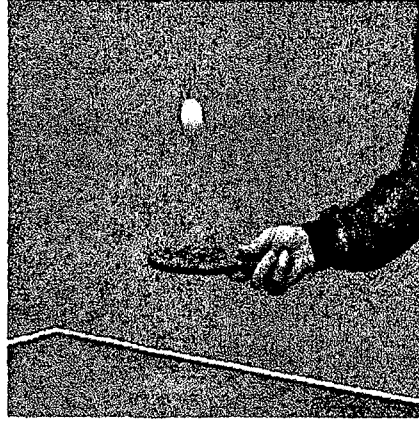
Şekil 7.10.c) Güncelleme algoritması sonucu



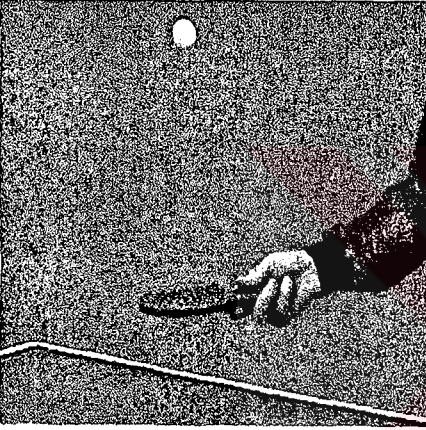
Şekil 7.11.c) Güncelleme algoritması sonucu



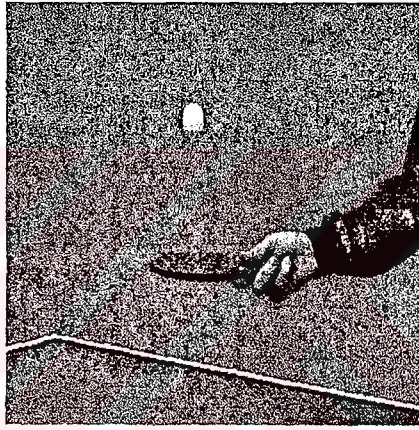
Şekil 7.12.a) 20.çerçeve orijinal görüntü



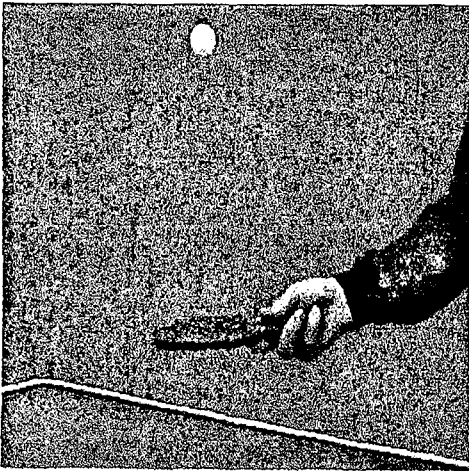
Şekil 7.13.a) 30. çerçeve orijinal görüntü



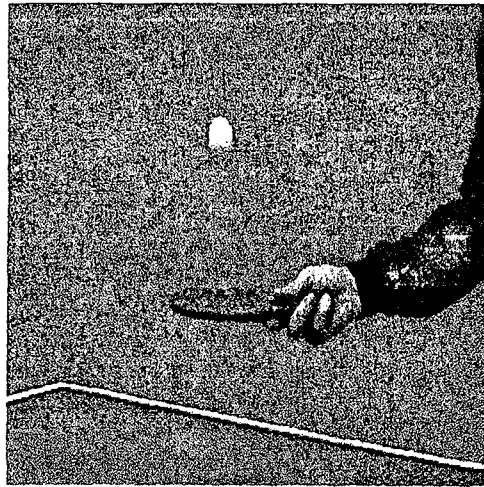
Şekil 7.12.b) LBG algoritması sonucu



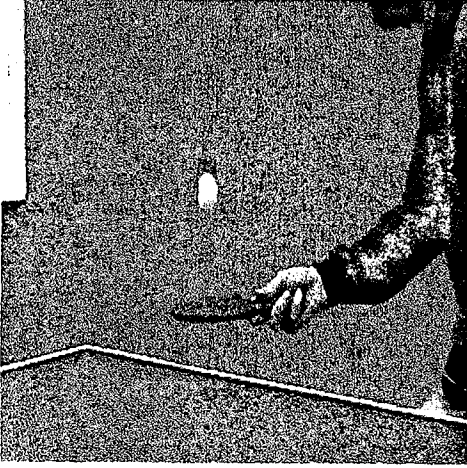
Şekil 7.13.b) LBG algoritması sonucu



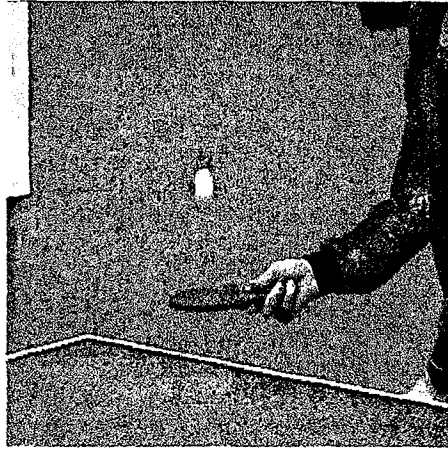
Şekil 7.12.c) Güncelleme algoritması sonucu



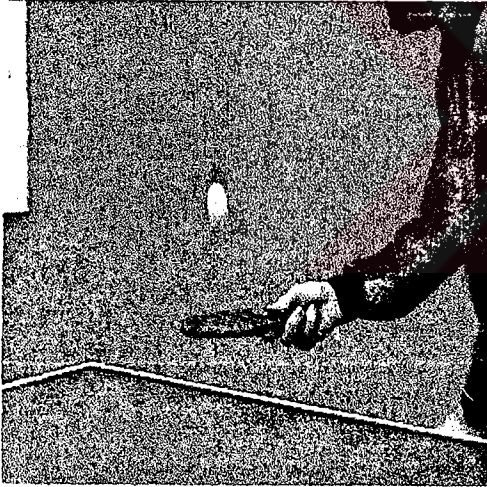
Şekil 7.13.c) Güncelleme algoritması sonucu



Şekil 7.14.a) 39. çerçeve orijinal görüntü

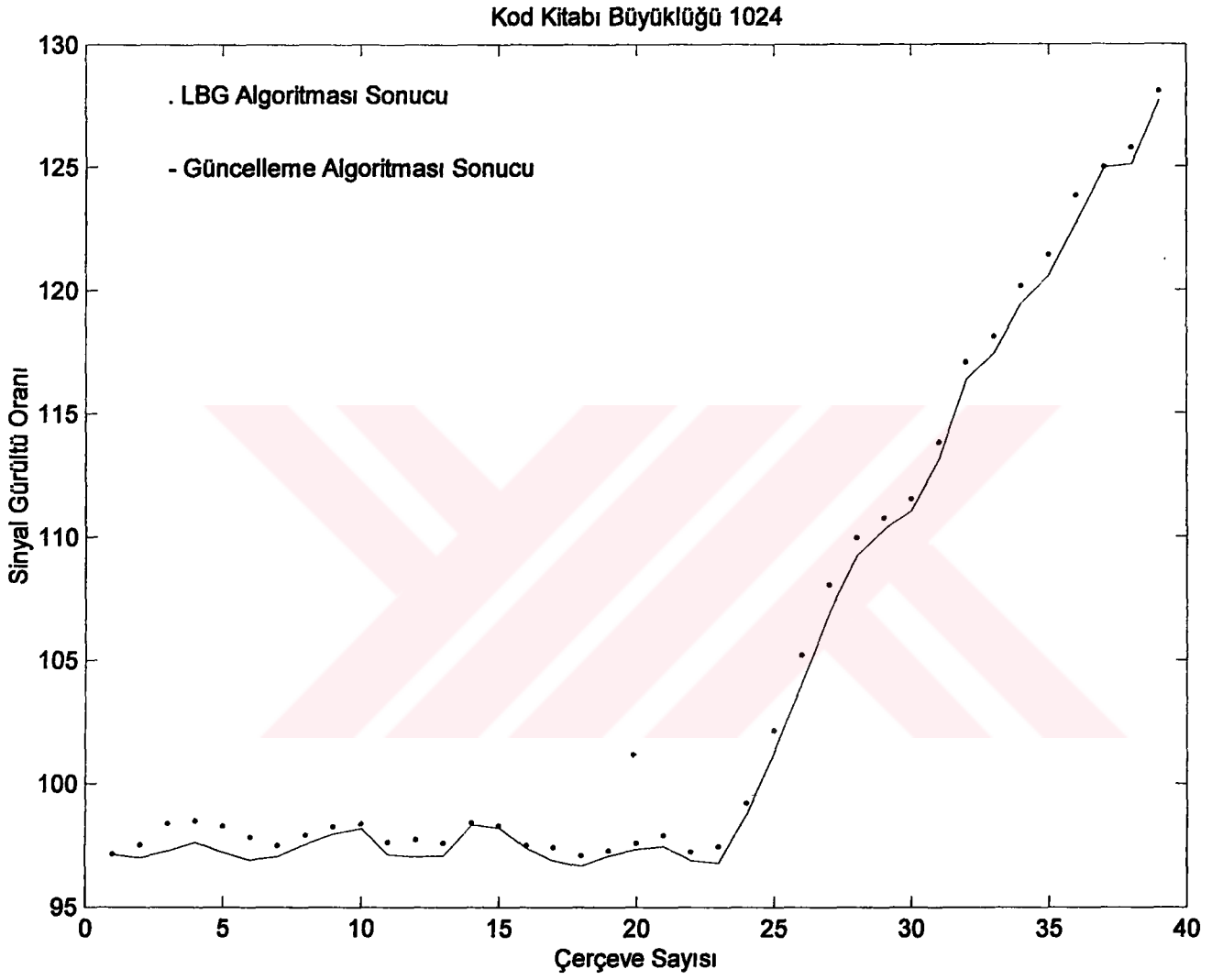


Şekil 7.14.b) LBG algoritması sonucu



Şekil 7.14.c) Güncelleme algoritması sonucu

Yukarıda verilen şekillerde kod kitabı büyüklüğü olarak 1024 alınmıştır.



7.2.2) Futbol

60 çerçevelik gri seviyeli futbol dizilerine LBG ve güncellenmiş LBG algoritması uygulanmış ve sonuçları aşağıda verilmiştir.

Çerçeve Sayısı	Sinyal Gürültü Oranları							
	Kod Kitabı		Büyüklüğü		Kod Kitabı		Büyüklüğü	
	2	2	4	4	8	8		
	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme		
0	80.9002	80.9002	91.3612	91.3612	97.4588	97.4588		
1	79.3225	79.3266	88.7181	88.7262	94.3252	94.2717		
2	77.6516	77.6508	87.4341	87.3995	92.6579	92.7232		
3	75.1393	75.1123	85.1226	85.0783	90.7393	90.7075		
4	74.0090	74.0021	83.7223	83.6481	89.3417	89.2894		
5	74.2339	74.2310	83.5290	83.5135	88.6703	88.6210		
6	73.5433	73.5311	83.8264	83.7655	89.1523	89.1467		
7	71.9335	71.9256	82.3692	82.3116	87.5134	87.4421		
8	71.7230	71.7141	82.5879	82.5221	87.8984	87.8063		
9	71.4504	71.4477	81.6066	81.5390	87.0002	86.9018		
10	71.4650	71.4525	82.1169	82.0093	87.5833	87.4759		
11	71.9922	71.9812	83.0571	82.9776	88.3857	88.3651		
12	71.9317	71.9039	83.2396	83.1809	88.7079	88.5359		
13	71.8363	71.8063	82.5533	82.4551	87.7407	87.5450		
14	71.8080	71.7633	81.7454	81.6681	87.2747	87.0628		
15	71.7917	71.7690	81.6515	81.5170	86.9650	86.7654		
16	70.5036	70.4678	79.7343	79.5982	85.1242	85.0146		
17	69.0335	68.9756	77.6175	77.4413	82.3010	82.2849		
18	67.9674	67.8806	76.7349	76.5390	81.3676	81.2580		
19	67.9062	67.8265	76.8812	76.7224	81.4196	81.2577		
20	68.2358	68.1640	77.3031	77.0753	81.9963	81.9108		

21	68.1105	68.0675	77.1341	76.9177	81.4053	81.3351
22	68.0789	68.0473	76.9670	76.8076	81.8443	81.6991
23	67.6241	67.5892	76.4534	76.2602	81.1223	81.0507
24	67.8031	67.7793	76.8737	76.6141	81.4589	81.2568
25	67.8339	67.7874	76.9652	76.6596	81.3040	81.1446
26	67.6237	67.5436	76.4606	76.2315	81.4375	81.3351
27	67.2325	67.1758	75.9246	75.6702	80.8047	80.6934
28	67.3372	67.2857	75.7095	75.4570	80.5983	80.5492
29	67.3274	67.3081	76.1935	75.9676	80.9326	80.8584
30	67.4215	67.3957	76.2221	76.0080	80.7499	80.5162
31	67.7167	67.6877	76.5133	76.2851	81.3192	81.2258
32	68.0615	68.0225	76.6853	76.4464	80.7926	80.4983
33	68.1949	68.1834	77.0916	76.8385	81.8247	81.6639
34	67.3998	67.3731	75.7063	75.4638	80.7565	80.5616
35	67.0546	67.0100	75.5188	75.2109	80.2347	80.1279
36	66.4042	66.3839	74.6998	74.4090	78.6731	78.0248
37	66.4656	66.4364	75.0377	74.7736	79.1712	78.3030
38	66.3044	66.2636	74.8332	74.5975	79.0949	78.3963
39	66.0408	66.0235	74.2068	73.9811	78.5375	78.4884
40	66.3819	66.3694	74.5427	74.4076	79.2224	79.0779
41	66.0676	66.0443	74.1669	74.0006	78.4679	78.4311
42	65.8397	65.8074	73.4896	73.3071	77.2663	77.1719
43	66.0920	66.0518	73.9062	73.6906	78.2219	78.1452
44	65.8978	65.8816	73.8972	73.6822	78.7620	78.4195
45	65.2571	65.2148	73.4308	73.1673	77.9228	77.8311
46	64.8600	64.8374	73.5229	73.1973	77.9195	77.7996
47	65.3867	65.3443	74.1840	73.9244	78.7493	78.6409
48	65.6245	65.5661	74.6557	74.4271	79.0631	78.8259
49	65.5526	65.5014	74.4422	74.2404	78.7130	78.5788
50	65.4172	65.3811	74.2424	74.0819	78.6709	78.5535
51	65.2832	65.2008	74.0310	73.8770	78.1713	78.0722

52	65.1639	65.0809	74.1171	73.9782	78.4439	78.4317
53	65.2358	65.1728	74.2068	73.9768	78.6298	78.3690
54	65.2013	65.0984	74.1998	74.0290	78.6285	78.4366
55	64.8328	64.7533	73.7348	73.5400	78.1554	77.9723
56	64.6848	64.6303	73.4016	73.2300	77.7949	77.6127
57	64.9247	64.8281	73.7314	73.5260	77.9390	77.8181
58	65.1300	65.0750	73.7115	73.5602	77.9648	77.7796
59	65.2878	65.2302	73.8128	73.6886	78.2013	78.1178

Tablo 7.2.a) Futbol görüntü dizisinin sinyal gürültü oranları

Çerçeve Sayısı	Sinyal Gürültü Oranları					
	Kod Kitabı 16		Kod Kitabı 32		Kod Kitabı 64	
	Büyüklüğü	Güncelleme	Büyüklüğü	Güncelleme	Büyüklüğü	Güncelleme
0	105.7519	105.7519	109.4715	109.4715	109.4715	109.4715
1	102.6805	101.9909	106.5746	105.3599	106.5746	105.3599
2	100.6567	100.0533	104.5143	103.4408	104.5143	103.4408
3	98.2839	97.8141	101.9723	101.1794	101.9723	101.1794
4	97.0251	96.2362	100.9822	99.8206	100.9822	99.8206
5	96.6758	95.8432	100.7780	99.2953	100.7780	99.2953
6	97.5994	96.6810	101.7317	99.8720	101.7317	99.8720
7	95.5951	94.3790	99.7220	97.7780	99.7220	97.7780
8	95.9772	94.9472	99.8552	98.2674	99.8552	98.2674
9	94.8634	93.8109	98.7287	97.1143	98.7287	97.1143
10	96.0088	94.8319	99.8481	98.0275	99.8481	98.0275
11	97.1654	96.0228	100.8677	99.9403	100.8677	99.9403
12	96.3533	95.8078	100.9302	99.7053	100.9302	99.7053
13	96.0659	94.9162	99.9926	98.1009	99.9926	98.1009
14	95.4464	94.0018	99.4560	97.3637	99.4560	97.3637
15	95.1599	93.9214	99.0441	97.3731	99.0441	97.3731

16	93.6034	92.1224	97.5183	95.4546	97.5183	95.4546
17	90.6986	89.6094	94.7909	92.9823	94.7909	92.9823
18	90.1591	88.6767	94.4295	92.2104	94.4295	92.2104
19	90.6700	88.8960	94.9438	92.2557	94.9438	92.2557
20	90.3448	88.7758	94.7238	92.0523	94.7238	92.0523
21	90.4307	88.7468	94.4046	91.9243	94.4046	91.9243
22	90.3937	88.5638	94.2404	91.9689	94.2404	91.9689
23	89.4870	87.8108	93.3815	91.2388	93.3815	91.2388
24	89.7885	88.2672	93.9655	91.5805	93.9655	91.5805
25	89.9464	88.3257	93.7927	91.9114	93.7927	91.9114
26	89.7460	87.8868	93.5481	91.4133	93.5481	91.4133
27	89.1839	87.5115	93.1517	91.0975	93.1517	91.0975
28	89.3637	87.2430	93.1316	90.5599	93.1316	90.5599
29	89.3937	87.6985	93.3012	91.3380	93.3012	91.3380
30	89.7585	88.0231	93.5030	91.6586	93.5030	91.6586
31	90.1688	88.4559	94.3379	92.0221	94.3379	92.0221
32	90.0807	88.3059	93.9935	91.9909	93.9935	91.9909
33	90.5102	88.3523	94.0790	91.8777	94.0790	91.8777
34	89.6803	87.4819	93.4938	91.0748	93.4938	91.0748
35	88.9085	86.9205	92.6549	90.4754	92.6549	90.4754
36	88.2984	86.5087	92.4716	89.6993	92.4716	89.6993
37	88.5063	86.7467	92.6454	90.0716	92.6454	90.0716
38	88.7755	86.7043	92.8009	90.3882	92.8009	90.3882
39	87.8435	85.8729	91.6979	89.3287	91.6979	89.3287
40	87.7348	86.1702	92.0116	89.5600	92.0116	89.5600
41	87.3622	85.4316	91.4284	88.9222	91.4284	88.9222
42	86.7541	85.0500	90.9928	88.1859	90.9928	88.1859
43	87.3227	85.1453	91.2154	88.6362	91.2154	88.6362
44	87.6528	85.6294	91.4628	89.0133	91.4628	89.0133
45	86.7802	84.9221	90.8630	88.3075	90.8630	88.3075
46	86.6838	84.9180	90.8095	88.3625	90.8095	88.3625

47	87.2382	85.3054	91.2484	88.7355	91.2484	88.7355
48	87.7928	85.7482	91.7841	89.2230	91.7841	89.2230
49	87.6669	85.6229	91.7110	89.0326	91.7110	89.0326
50	87.5169	85.6174	91.3984	88.8027	91.3984	88.8027
51	87.1016	85.1068	91.2584	88.5678	91.2584	88.5678
52	87.2523	85.2453	91.0775	88.7845	91.0775	88.7845
53	87.6549	85.6448	91.8299	89.0730	91.8299	89.0730
54	87.8008	85.4575	91.7844	89.0940	91.7844	89.0940
55	87.6693	85.0474	91.7729	88.5502	91.7729	88.5502
56	87.1328	84.7059	91.0662	88.2023	91.0662	88.2023
57	87.2017	84.6942	91.1900	88.1339	91.1900	88.1339
58	87.1537	84.9575	91.0314	88.4423	91.0314	88.4423
59	87.0281	84.9059	91.0051	88.2335	91.0051	88.2335

Tablo 7.2.b) Futbol görüntü dizisinin sinyal gürültü oranları

	Sinyal Gürültü Oranları					
	Kod Kitabı	Büyüküğü	Kod Kitabı	Büyüküğü	Kod Kitabı	Büyüküğü
Çerçeve Sayısı	128	128	256	256	512	512
	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme
0	113.4262	113.4262	117.8914	117.8914	124.4682	124.4682
1	110.5137	109.9026	115.0882	115.0704	122.2370	122.0521
2	108.6381	107.7500	113.3235	113.2063	120.2368	120.1926
3	105.8772	105.4694	110.4631	110.0838	117.5781	117.4101
4	104.9575	104.0090	109.4854	109.4705	116.6281	116.1409
5	104.8677	103.8720	109.3382	109.0921	116.5879	116.5546
6	105.5788	104.5712	110.2054	110.1062	117.8013	117.0265
7	103.7719	102.7126	108.2636	107.8469	115.5284	115.5093
8	104.2661	102.6916	108.9792	108.2862	115.6929	115.4445
9	102.7699	101.7924	107.4788	106.9498	114.4857	114.1900
10	103.7313	102.8068	108.8245	108.3953	115.6882	115.3521

11	104.7278	104.0284	109.6521	109.2521	116.5626	115.9103
12	104.6239	103.8078	109.2573	109.2416	116.2661	116.1337
13	104.1610	103.0640	108.7318	108.5864	116.2862	116.1656
14	103.7126	102.1188	108.7387	108.1946	116.5162	116.1248
15	103.1676	102.2965	108.1882	108.0205	115.6940	115.5113
16	102.1159	100.2272	106.9417	106.0349	114.8376	114.4354
17	99.2821	97.4271	104.2265	103.2214	111.9625	110.9666
18	98.9207	97.5605	103.5065	101.9409	110.6180	110.0198
19	99.0599	97.3146	104.0099	102.1768	110.9102	110.4653
20	98.6758	96.8349	103.4588	102.5277	110.3256	109.5269
21	98.8655	96.8594	103.8417	101.6097	111.0204	109.6244
22	98.5728	96.8503	103.3389	101.4461	109.9393	109.4608
23	97.6246	95.8889	102.4538	100.8314	109.0930	108.5340
24	98.1980	96.2523	102.9558	101.2499	108.9789	108.2539
25	97.7798	96.1900	102.4393	101.0034	108.6472	108.4998
26	97.9096	95.8688	102.9149	100.6180	109.2418	108.6077
27	97.3099	94.3895	101.8022	99.9532	108.4242	107.7804
28	97.2356	95.3658	102.0083	99.9034	108.7852	107.5270
29	97.8019	94.5371	102.4009	100.4070	109.0963	107.9448
30	97.8831	96.1877	102.6062	100.5834	109.0963	108.1570
31	98.7275	96.5839	103.2986	101.6359	110.0348	109.4400
32	98.4535	96.2460	103.3246	101.4533	109.9663	109.4919
33	98.5845	96.4562	103.4098	102.3096	110.0059	108.9486
34	98.1534	95.9713	103.0977	100.8348	109.5880	109.1052
35	96.9784	94.8314	101.6605	100.1550	108.2885	107.2564
36	96.7859	94.5269	101.1988	99.1105	108.3043	106.9234
37	96.9501	95.0570	101.6905	99.5028	107.8753	107.3302
38	96.9851	95.1064	101.8928	99.7888	108.1989	107.3168
39	95.9972	94.1145	100.6559	98.6922	107.2179	105.8738
40	96.2980	94.2600	101.0692	99.5442	107.3235	106.5097
41	95.5192	92.5193	100.4475	98.4295	106.9386	105.8584

42	95.1633	93.1176	99.8041	97.7636	106.3285	105.1624
43	95.6471	93.4386	100.1809	98.0509	106.5344	105.4789
44	95.3628	93.5950	100.2360	98.2451	106.7315	105.9556
45	95.1906	91.9970	100.1259	97.9010	106.5320	105.4946
46	94.9930	92.7623	100.0123	97.4902	105.9755	105.1663
47	95.2064	93.3291	99.8990	97.8040	106.4084	105.5453
48	95.9554	93.7222	100.9506	98.9101	107.5960	106.5138
49	95.7944	93.9789	100.6268	98.7584	107.1027	106.2848
50	95.8249	93.5774	100.9880	98.6860	107.3232	106.2230
51	95.4444	93.2660	100.4522	98.1272	107.1861	105.8102
52	95.3207	93.4246	100.4777	98.8440	107.1187	105.6290
53	96.3099	93.9529	101.0496	98.6614	107.6870	106.4682
54	96.3665	94.0540	101.0972	99.2578	107.8036	106.7148
55	96.1415	93.7483	101.0749	98.8078	107.5737	106.4426
56	95.4307	91.7604	100.1007	98.7195	106.5057	105.4187
57	95.3373	92.8859	100.0123	97.7317	106.4265	105.3917
58	95.6098	93.1294	100.4360	98.6810	106.9180	105.6803
59	95.4431	92.9699	100.3785	97.6911	106.9059	105.3482

Tablo 7.2.c) Futbol görüntü dizisinin sinyal gürültü oranları

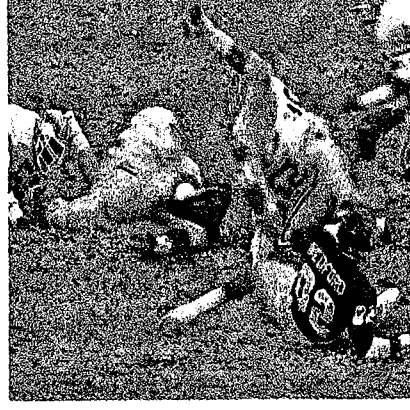
	Sinyal Gürültü		Oranı	Sinyal Gürültü	
	Kod Kitabı	Büyüklüğü		Kod Kitabı	Büyüklüğü
Çerçeve	1024	1024	Çerçeve	1024	1024
Sayısı	LBG	Güncelleme	Sayısı	LBG	Güncelleme
0	132.9419	132.9419	30	121.6718	120.3446
1	131.4389	131.0731	31	122.5345	121.0142
2	128.9553	128.5611	32	122.2818	121.2728
3	127.5920	127.6161	33	122.7290	121.7336
4	127.7192	127.4729	34	121.9152	120.5258

5	127.7031	127.2832	35	120.3356	119.4206
6	128.9999	128.7018	36	120.8945	118.7355
7	127.2152	126.8653	37	120.2682	119.0487
8	127.6445	127.4076	38	120.5119	118.9860
9	126.4041	126.4560	39	119.2785	117.5716
10	127.8113	127.2059	40	119.2032	117.5933
11	127.3849	127.1227	41	119.1982	117.3234
12	127.2959	127.0761	42	118.1922	116.4607
13	127.2878	126.6919	43	118.4464	116.8962
14	127.3512	126.6602	44	118.8066	117.3570
15	127.0766	126.7175	45	118.1862	116.8547
16	126.2113	125.4096	46	117.9779	116.7269
17	124.8883	123.0995	47	119.0104	117.2364
18	123.4128	122.3226	48	119.7421	118.0913
19	123.4111	122.2645	49	119.0376	117.9695
20	122.9057	121.9855	50	119.2820	117.7324
21	123.6134	122.0674	51	119.1405	117.5216
22	123.0809	121.7604	52	119.1739	117.8141
23	122.7461	121.0105	53	119.6584	117.8728
24	122.2502	121.1916	54	120.2027	117.9319
25	121.5798	120.1951	55	119.7597	117.8790
26	121.8350	120.6352	56	119.3938	117.1724
27	121.3225	120.1145	57	119.2082	117.3947
28	121.7358	119.9091	58	119.5428	117.4711
29	121.7009	120.1702	59	119.6548	117.6205

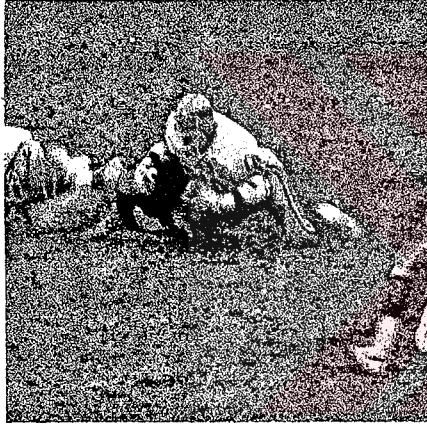
Tablo 7.2.d) Futbol görüntü dizisinin sinyal gürültü oranları



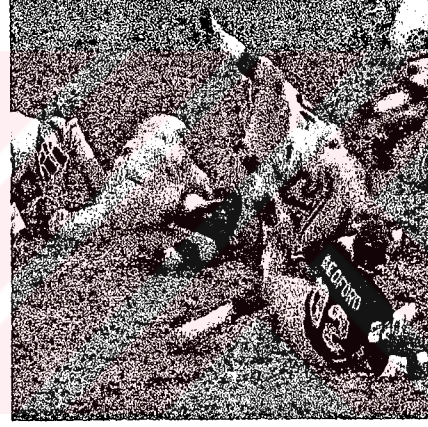
Şekil 7.15.a) 1. çerçeve orijinal görüntü



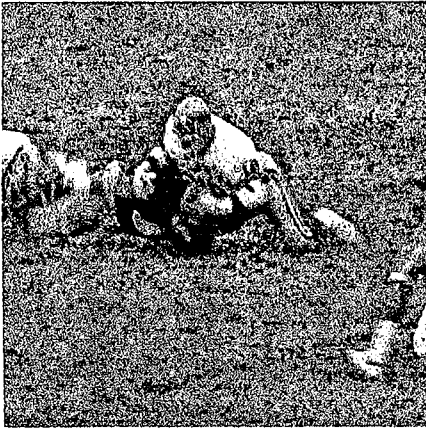
Şekil 7.16.a) 15. çerçeve orijinal görüntü



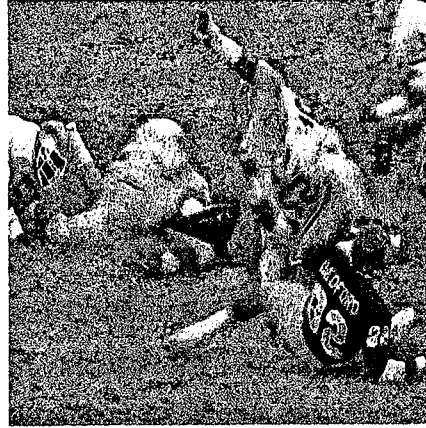
Şekil 7.15.b) LBG algoritması sonucu



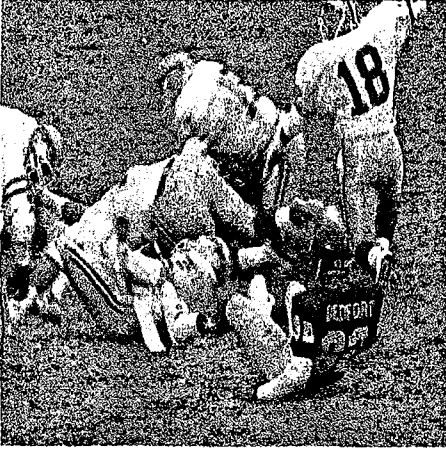
Şekil 7.16.b) LBG algoritması sonucu



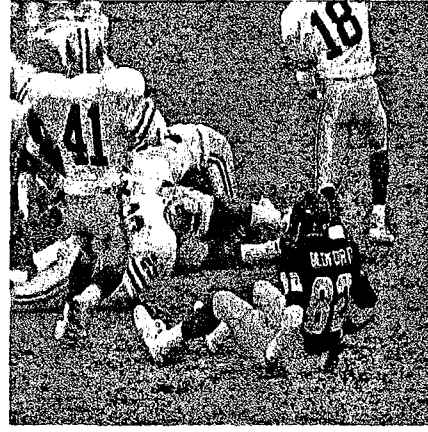
Şekil 7.15.c) Güncelleme algoritması sonucu



Şekil 7.16.c) Güncelleme algoritması sonucu



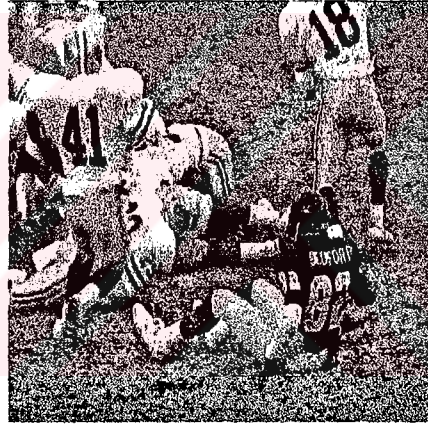
Şekil 7.17.a) 30. çerçeve orijinal görüntü



Şekil 7.18.a) 45.çerçeve orijinal görüntü



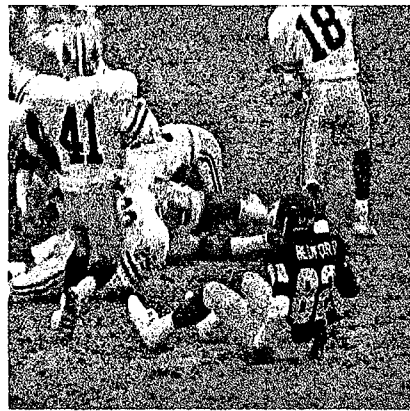
Şekil 7.17.b) LBG algoritması sonucu



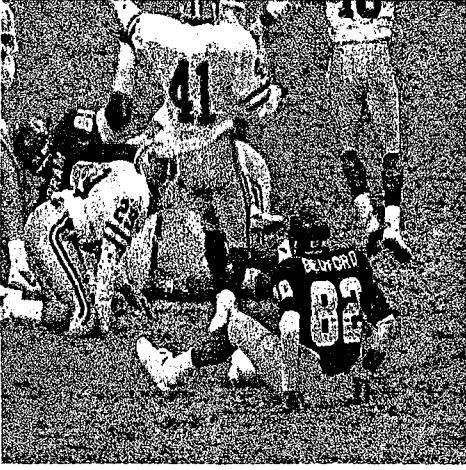
Şekil 7.18.b) LBG algoritması sonucu



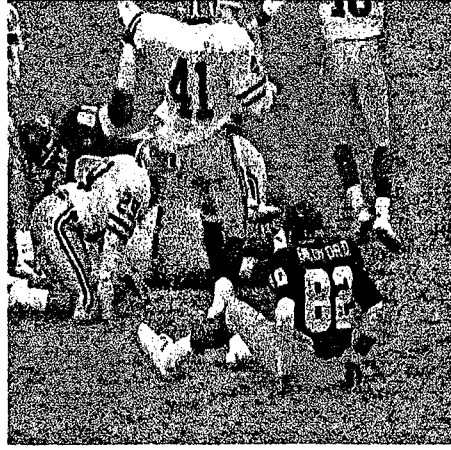
Şekil 7.17.a) Güncelleme algoritması sonucu



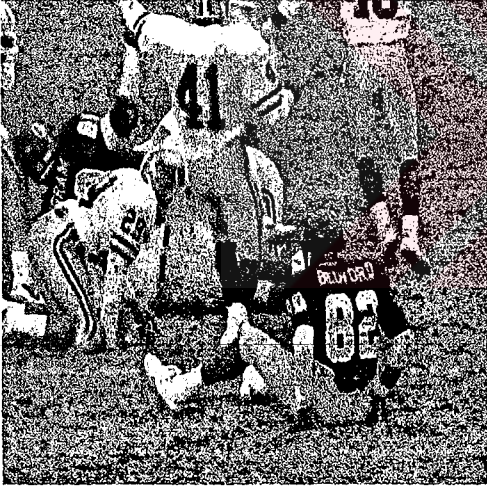
Şekil 7.18.c) Güncelleme algoritması sonucu



Şekil 7.19.a) 59. çerçeve orijinal görüntü

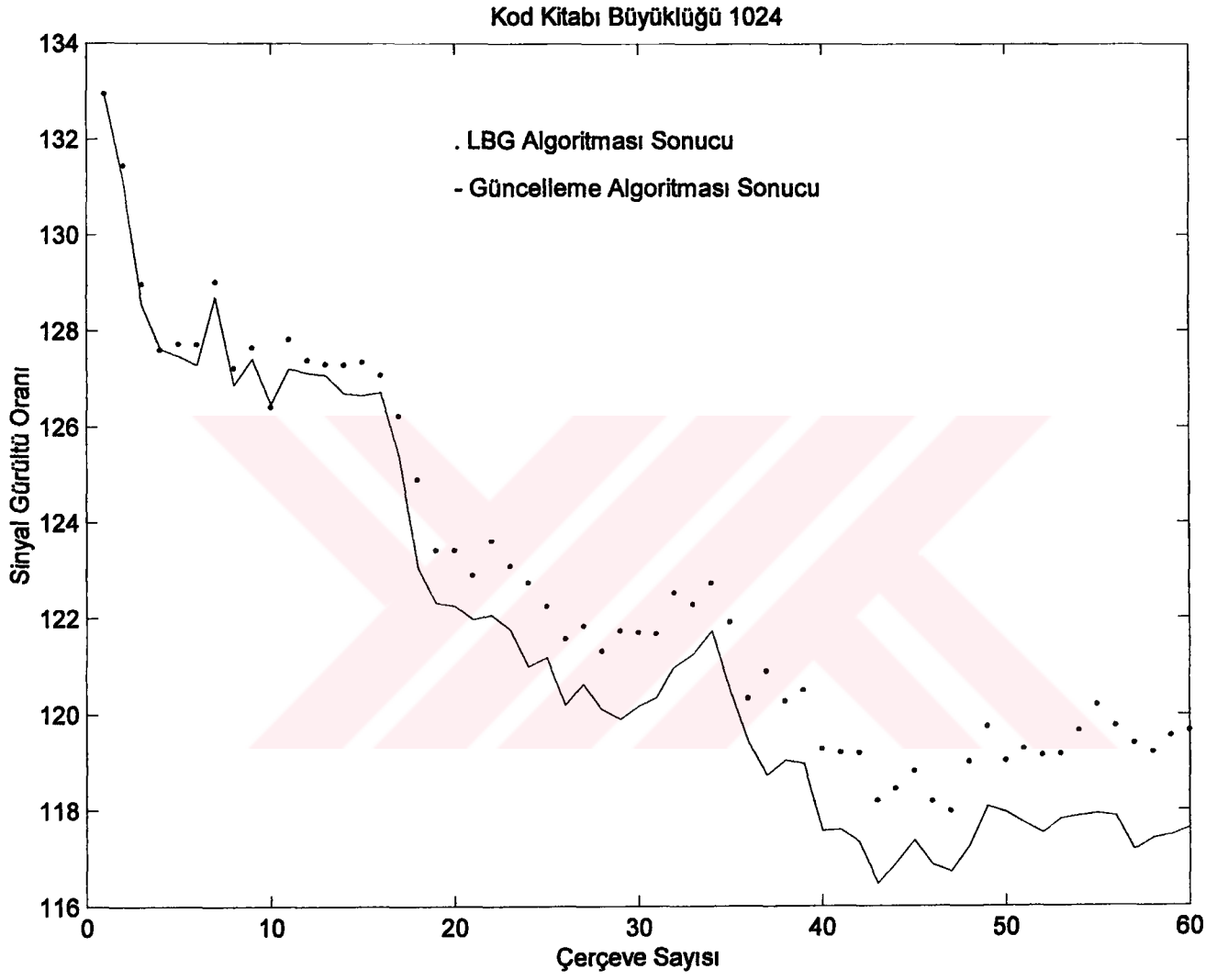


Şekil 7.19.b) LBG algoritması sonucu



Şekil 7.19.c) Güncelleme algoritması sonucu

Yukarıda verilen şekillerde kod kitabı büyüklüğü olarak 1024 alınmıştır.



7.2.3) Bahçe

99 çerçevelik gri seviyeli bahçe dizilerine LBG ve güncellenmiş LBG algoritması uygulanmış ve sonuçları aşağıda verilmiştir.

Çerçeve Sayısı	Sinyal Gürültü Oranları							
	Kod Kitabı		Büyükklüğü		Kod Kitabı		Büyükklüğü	
	2	2	4	4	8	8		
	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme		
1	56.4339	56.4339	65.7166	65.7166	70.4452	70.4452		
2	56.0373	56.0373	65.2374	65.2363	69.4733	69.9783		
3	55.9690	55.9682	65.1390	65.1399	69.9464	69.9219		
4	56.0857	56.0860	65.3342	65.3330	70.0607	70.0515		
5	56.1404	56.1399	65.3616	65.3618	69.7121	70.2636		
6	55.8416	55.8418	65.0367	65.0366	69.8216	69.7866		
7	55.5804	55.5811	64.6449	64.6464	68.9906	69.4012		
8	55.4267	55.4295	64.6109	64.6087	68.8760	69.3782		
9	55.3331	55.3337	64.3712	64.3715	68.6216	69.0965		
10	55.1478	55.1488	64.1591	64.1591	68.8418	68.8233		
11	54.7632	54.7648	63.6304	63.6262	67.7818	68.1749		
12	54.8157	54.8172	63.7642	63.7656	67.8054	68.2380		
13	54.8946	54.8965	63.7895	63.7892	67.7801	68.3200		
14	54.7387	54.7403	63.7503	63.7469	68.2593	68.2508		
15	54.4593	54.4622	63.4059	63.4007	67.4909	67.9201		
16	54.3232	54.3263	63.2785	63.2727	67.9252	67.8547		
17	54.2351	54.2354	63.0948	63.0921	67.6545	67.6537		
18	54.2453	54.2477	62.9977	62.9927	67.3071	67.5834		
19	54.0383	54.0346	62.6210	62.6186	66.6910	67.1576		
20	53.8657	53.8597	62.3510	62.3522	66.3331	66.7916		
21	53.8040	53.7916	62.2273	62.2286	66.4181	66.6893		

22	53.6400	53.6221	62.0899	62.0937	66.1607	66.6316
23	53.5837	53.5683	61.9361	61.9340	66.1157	66.4643
24	53.6351	53.6309	62.0765	62.0762	66.4776	66.4590
25	53.8174	53.8003	62.1831	62.1853	66.6824	66.6495
26	53.6403	53.6178	61.8338	61.8339	66.3469	66.2962
27	53.5881	53.5647	61.8316	61.8307	66.4835	66.4489
28	53.4724	53.4502	61.8050	61.8078	66.4714	66.4446
29	53.3187	53.2896	61.5216	61.5216	66.2209	66.1324
30	53.0541	53.0325	61.1657	61.1680	65.8809	65.7759
31	53.1149	53.0788	61.3721	61.3686	66.1640	66.1238
32	52.9354	52.9131	61.0326	61.0290	65.5881	65.5489
33	52.8807	52.8528	60.8025	60.8013	65.4469	65.3507
34	52.7238	52.6854	60.7609	60.7546	65.4823	65.3895
35	52.5259	52.4769	60.3755	60.3648	64.9795	64.9428
36	52.4457	52.4277	60.3123	60.3134	64.9326	64.9002
37	52.3036	52.2665	59.9823	59.9760	64.1825	64.5502
38	52.3769	52.3488	59.9887	59.9889	64.5367	64.4913
39	52.1352	52.0947	59.5008	59.4978	64.0145	63.9846
40	52.0291	51.9860	59.2438	59.2432	63.3185	63.7167
41	51.9984	51.9329	59.1603	59.1522	63.7108	63.6526
42	52.0400	51.9839	59.0892	59.0742	63.3052	63.7610
43	51.9326	51.8674	58.7999	58.7948	62.9139	63.3444
44	51.8169	51.7334	58.6525	58.6391	63.2827	63.2165
45	51.8215	51.7374	58.7590	58.7397	63.2670	63.1921
46	51.8349	51.7668	58.8752	58.8701	63.4536	63.4166
47	52.1381	52.0762	59.2117	59.2071	63.8951	63.8122
48	52.2555	52.1775	59.4414	59.4251	63.9410	63.8594
49	52.3445	52.2807	59.4894	59.4800	64.0479	63.9849
50	52.6448	52.6132	59.9194	59.9149	64.1356	64.2815
51	52.8263	52.7626	60.1334	60.1139	64.6673	64.6171
52	52.7617	52.7171	60.0862	60.0690	64.6076	64.5408

53	53.0139	52.9879	60.4485	60.4380	64.8398	64.7659
54	53.3747	53.3328	60.9106	60.8950	65.4432	65.3794
55	53.2811	53.2237	60.6872	60.6656	64.9840	64.8898
56	53.2314	53.1891	60.5223	60.5100	64.5341	64.7892
57	53.2393	53.1854	60.5707	60.5548	64.9512	64.8573
58	53.2681	53.2147	60.6711	60.6456	65.1211	65.0149
59	53.4852	53.4456	60.8041	60.7885	65.3354	65.3367
60	53.3361	53.2841	60.5598	60.5383	64.9364	64.8892
61	53.4832	53.4110	60.6212	60.5950	64.9433	65.1665
62	53.3352	53.2866	60.3658	60.3290	64.9053	64.8916
63	53.1529	53.1055	60.1914	60.1683	64.5877	64.4900
64	53.2984	53.2568	60.1978	60.1733	64.7388	64.6484
65	53.4388	53.3761	60.2755	60.2360	64.5685	64.7670
66	53.4100	53.3700	60.2036	60.1857	64.7297	64.6788
67	53.2241	53.1706	59.9964	59.9534	64.3749	64.3469
68	53.1918	53.1059	59.7916	59.7433	64.3157	64.2504
69	53.3458	53.2869	59.9957	59.9409	64.4369	64.4045
70	53.3744	53.3475	60.1758	60.1358	64.5840	64.4798
71	53.3635	53.3016	60.0594	60.0199	64.4214	64.3680
72	53.1630	53.0848	59.6627	59.6223	64.0544	63.9367
73	53.2155	53.1697	59.7699	59.7274	63.7982	64.0570
74	53.3660	53.3312	59.9134	59.8731	63.9171	64.1866
75	53.0779	53.0149	59.6482	59.6005	64.0794	63.9520
76	53.4601	53.4180	60.0356	59.9834	63.9495	64.3033
77	53.3502	53.2949	59.8033	59.7529	63.8689	64.1036
78	53.3510	53.2993	59.9534	59.9083	64.5133	64.4190
79	53.2820	53.2396	59.8043	59.7613	63.8371	64.0564
80	53.3039	53.2431	59.8570	59.7924	63.9263	64.3035
81	53.6111	53.5402	60.1953	60.1355	64.4522	64.6014
82	53.3139	53.2393	59.8951	59.8327	63.9125	64.2856
83	53.2127	53.1387	59.6894	59.6381	64.0878	63.9620

84	53.1864	53.1170	59.6826	59.6287	64.1515	64.0371
85	53.2383	53.1483	59.8114	59.7398	63.9221	64.2277
86	53.3057	53.2259	59.8702	59.7939	64.3858	64.2354
87	53.2033	53.1176	59.7325	59.6795	64.2092	64.1134
88	53.1745	53.0902	59.5983	59.5161	63.5383	63.8713
89	52.9296	52.8450	59.3772	59.3123	63.9024	63.7841
90	52.9233	52.8432	59.2186	59.1662	63.6927	63.6093
91	52.8282	52.7397	59.2334	59.1538	63.6724	63.5553
92	52.6436	52.5544	58.8650	58.7930	63.3030	63.1598
93	52.4568	52.3623	58.7443	58.6800	63.0919	62.9733
94	52.5688	52.4994	58.8826	58.8254	62.8626	63.1242
95	52.4978	52.4145	58.7531	58.6908	63.0792	62.9685
96	52.4582	52.3804	58.7218	58.6482	62.7044	63.0043
97	52.6353	52.5435	59.0100	58.9446	63.4830	63.3101
98	52.7031	52.6303	59.0796	58.9929	63.5166	63.4348
99	52.5323	52.4287	58.8613	58.7752	63.2404	63.0995

Tablo 7.3.a) Bahçe görüntü dizisinin sinyal gürültü oranları

	Sinyal Gürültü Oranları					
	Kod Kitabı	Büyüklüğü	Kod Kitabı	Büyüklüğü	Kod Kitabı	Büyüklüğü
	16	16	32	32	64	64
Çerçeve Sayısı	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme
1	73.9460	73.9460	77.2540	77.2540	80.7573	80.7573
2	73.6047	73.3789	76.8807	76.6022	80.0966	79.6207
3	73.5023	73.2972	76.8811	76.3809	80.1488	79.4995
4	73.7272	73.4755	77.1663	76.8602	80.6964	80.1552
5	73.7107	73.5565	77.1612	76.9024	80.3531	80.0146
6	73.3506	73.2497	76.8167	76.3982	79.9289	79.4722
7	72.8821	72.7663	76.3962	75.8742	79.3923	78.9727
8	72.8814	72.7573	76.1907	75.8951	79.4429	78.8282

9	72.7765	72.4488	75.8549	75.5116	79.0372	78.6099
10	72.6007	72.2241	75.9102	75.4064	79.1303	78.3505
11	71.9677	71.4652	75.3087	74.8331	78.3990	77.8630
12	71.7240	71.5436	74.9932	74.7284	78.3723	77.7864
13	71.7013	71.4746	74.9019	74.6714	78.2155	77.5864
14	72.0272	71.4522	75.3781	74.8095	78.4936	77.8285
15	71.5905	71.0348	74.9191	74.2895	78.0352	77.3654
16	71.7379	71.1762	75.0740	74.5719	78.5151	77.4506
17	71.3858	70.9112	74.6492	74.2108	77.9658	77.0766
18	71.2618	70.6021	74.6407	73.8310	77.7415	76.9012
19	70.8974	70.3032	74.1635	73.6083	77.5539	76.6569
20	70.5161	69.9143	73.8630	73.2288	76.8072	76.1929
21	70.2713	69.9014	73.4594	73.0963	76.7452	75.9230
22	70.0930	69.8335	73.4193	73.1048	76.7961	76.1572
23	70.1600	69.6362	73.7485	73.0357	76.9292	75.9576
24	70.2859	69.6610	73.6554	73.0530	76.9910	76.1144
25	70.3688	69.8773	73.6946	73.0607	76.9487	76.0231
26	69.9650	69.5379	73.3721	72.6670	76.4988	75.6295
27	70.3857	69.7113	73.7632	73.0555	76.8964	76.0827
28	70.2227	69.7757	73.7959	73.0533	77.1867	76.3727
29	69.9921	69.4385	73.6472	72.8358	76.6471	75.9561
30	69.6777	69.1008	73.1749	72.3300	76.3675	75.5172
31	69.8598	69.3997	73.3925	72.6203	76.7106	75.7397
32	69.4833	68.8598	73.1184	72.3010	76.5346	75.6203
33	69.1927	68.6108	72.9084	71.8777	76.1910	75.1573
34	69.1806	68.7281	72.6928	72.1107	75.9153	75.2570
35	68.7763	68.0654	71.9835	71.3525	75.4669	74.5334
36	68.3271	68.0540	72.0365	71.2875	75.5022	74.5471
37	67.9640	67.7811	71.5168	71.2203	74.9198	74.2868
38	67.7503	67.5645	71.5516	70.8641	74.8246	74.0407
39	67.7139	67.1399	71.1707	70.5834	74.4278	73.3629

40	67.5972	66.8887	70.9827	70.2367	74.1973	73.3652
41	67.4419	66.8288	71.1171	70.2593	74.4401	73.3851
42	67.5641	66.9174	70.9938	70.4138	74.4945	73.5508
43	66.7218	66.4116	70.3185	69.6730	73.4533	72.6482
44	66.8405	66.3020	70.3283	69.4594	73.4999	72.6396
45	66.6244	66.3884	70.2223	69.6275	73.5827	72.6995
46	66.8274	66.5829	70.2641	69.6596	73.3870	72.6066
47	67.0416	66.8596	70.6746	69.9906	74.0092	73.1009
48	67.2975	66.9838	70.9651	70.1958	74.3258	73.3271
49	67.6158	67.0324	71.0021	70.2444	74.3573	73.3825
50	67.5965	67.4998	71.3075	70.7245	74.6315	73.7875
51	68.0381	67.7668	71.6330	70.8057	74.8780	73.9719
52	68.0024	67.6640	71.4431	70.8605	74.6354	73.7861
53	68.5811	68.0150	72.2450	71.3933	75.3843	74.6265
54	69.0864	68.6300	72.3097	71.6611	75.6667	74.5693
55	68.4263	68.0771	71.6852	71.1496	74.8700	74.1183
56	68.4186	67.9969	71.9743	71.0628	75.1688	74.0967
57	68.7045	68.0531	71.8762	71.1069	75.0974	74.0364
58	68.4390	68.2431	71.9543	71.3269	75.4184	74.3339
59	69.1527	68.5047	72.5697	71.7107	75.8365	74.7559
60	68.7393	68.1980	72.1079	71.5165	75.3935	74.5144
61	68.7942	68.4257	72.2188	71.6018	75.6089	74.6394
62	68.6601	68.1417	71.9380	71.1312	75.2224	74.1726
63	68.5291	67.7258	71.7225	70.9154	74.9409	74.0237
64	68.5472	68.0115	71.8889	71.0410	75.0545	74.0691
65	68.2731	67.9007	71.8777	71.1111	74.9510	74.1222
66	68.3448	67.9580	71.6986	71.0626	74.9483	73.9236
67	68.0458	67.5275	71.3737	70.4868	74.4612	73.5199
68	68.0146	67.3984	71.0988	70.4563	74.4803	73.4608
69	68.1502	67.6627	71.3587	70.6030	74.6483	73.5935
70	68.0541	67.5828	71.4095	70.6093	74.6121	73.6841

71	68.1206	67.6036	71.3371	70.6560	74.6699	73.6398
72	67.7749	67.2516	70.9896	70.2232	74.0616	73.1714
73	67.7525	67.1623	71.0191	70.1734	74.2700	73.1399
74	67.8415	67.3318	71.1471	70.1739	74.1546	73.1993
75	67.6985	67.1628	70.9182	70.1732	73.9860	73.0669
76	67.7308	67.4502	71.2253	70.3938	74.3788	73.3749
77	67.8628	67.1881	71.2005	70.2627	74.3119	73.2494
78	68.0440	67.4078	71.2743	70.3840	74.6083	73.5030
79	67.5937	67.1838	71.0492	70.2757	74.2952	73.2328
80	67.7495	67.2365	71.1515	70.2731	74.4565	73.2553
81	67.9036	67.6157	71.4324	70.6454	74.6345	73.6669
82	68.1239	67.3696	71.3541	70.4563	74.6624	73.4655
83	67.7505	67.0276	70.9785	70.0111	74.1247	72.9506
84	67.5936	67.1110	70.5837	70.0463	74.0768	72.9610
85	67.6430	67.3477	71.3595	70.3767	74.5746	73.4458
86	67.9314	67.3211	71.1083	70.3391	74.5328	73.8349
87	67.8073	67.3627	71.0915	70.2919	74.4431	73.5739
88	67.3576	67.0674	70.9968	70.0131	74.2395	73.4592
89	67.5517	66.7970	70.8082	69.9731	74.0353	73.3427
90	67.3170	66.7512	70.6082	69.7423	74.0325	72.7708
91	67.2238	66.6416	70.6818	69.6138	73.9593	72.8580
92	66.7958	66.1821	70.1591	69.2691	73.5848	72.3725
93	66.7120	66.0338	69.8495	68.8876	73.0824	71.9636
94	66.6520	66.2057	70.0240	69.1377	73.3832	72.1820
95	66.6232	66.0769	70.0442	68.9829	73.3941	72.2273
96	66.5994	66.0760	70.0134	69.1286	73.3260	72.2029
97	67.0920	66.4457	70.3515	69.4561	73.7298	72.7060
98	67.0267	66.6436	70.5515	69.7530	73.8320	72.8579
99	66.9248	66.3023	70.2832	69.4022	73.6385	72.6302

Tablo 7.3.b) Bahçe görüntü dizisinin sinyal gürültü oranları

Çerçeve Sayısı	Sinyal Gürültü Oranları					
	Kod Kitabı	Büyüklüğü	Kod Kitabı	Büyüklüğü	Kod Kitabı	Büyüklüğü
	128	128	256	256	512	512
	LBG	Güncelleme	LBG	Güncelleme	LBG	Güncelleme
1	84.2645	84.2645	87.9794	87.9794	93.1063	93.1063
2	83.4435	82.5007	87.2984	86.7610	92.0400	92.2906
3	83.5141	83.1349	87.2714	86.7301	92.3660	92.5785
4	84.0844	83.2396	88.1318	87.6275	92.8573	93.0613
5	83.6085	82.8750	87.5266	86.9502	92.2191	92.6818
6	83.3474	82.2535	87.1562	86.3839	91.8507	92.0730
7	82.6254	81.9159	86.6417	86.0368	91.3669	91.5981
8	82.6316	81.7647	86.5851	85.8710	91.1804	91.3652
9	82.5563	81.7905	86.2782	85.1826	90.9293	91.0388
10	82.6150	81.1413	86.1594	85.2808	90.8581	90.8642
11	81.7919	81.1386	85.5725	84.6466	90.0891	90.1617
12	81.6815	81.2221	85.3438	84.7008	90.0543	90.3888
13	81.3931	80.8963	85.2257	84.4612	89.6620	89.9529
14	81.9381	81.2801	85.7068	84.7349	90.2482	90.5548
15	81.3757	80.3501	85.1772	84.3248	89.6966	89.9686
16	81.6987	80.9145	85.6248	84.3242	90.2661	90.3526
17	81.1541	80.4977	84.9793	84.0629	89.8391	89.8414
18	81.2258	79.9865	85.1805	83.8731	89.6006	89.7950
19	81.0754	79.5957	84.7870	83.7177	89.2929	89.2291
20	80.4034	79.0717	84.4039	83.0620	88.7454	88.9360
21	80.1213	78.8989	84.1121	83.2188	88.7482	88.6466
22	80.0901	78.8831	83.8050	82.9108	88.4187	88.6726
23	80.3549	79.5019	84.1591	83.1438	88.8258	88.7679
24	80.4589	79.6885	84.1969	83.2019	88.9425	88.8464
25	80.3724	79.6219	84.0100	83.1268	88.5371	89.0355
26	79.9683	79.1693	83.9476	82.8740	88.6318	88.5550

27	80.4519	79.2195	84.7555	83.3045	89.4313	89.0330
28	80.7084	79.9866	84.6539	83.6924	89.4150	89.3463
29	80.3274	79.0862	84.1132	83.4462	88.8968	89.1198
30	79.9729	78.5565	83.7746	82.5841	88.6264	88.5466
31	80.1561	79.6225	84.0666	83.3149	89.0460	89.0664
32	79.8480	78.6477	83.7463	82.9040	88.5907	88.4587
33	79.6182	78.2515	83.6101	82.4058	88.5169	88.3306
34	79.3824	78.2423	83.4026	82.3136	88.4251	88.0045
35	79.1722	77.9929	83.0721	81.9151	87.6568	87.8093
36	78.7630	78.1237	82.7986	81.7604	87.6595	87.6039
37	78.2574	77.2770	82.1678	81.2710	86.9474	87.0691
38	78.3332	76.8971	82.0103	80.9927	86.7516	86.9621
39	77.8533	76.4671	81.7240	80.5120	86.1238	86.2517
40	77.7908	76.3859	81.4659	80.4240	86.0924	85.9787
41	77.7916	76.4256	81.6088	80.6162	86.4243	86.0605
42	77.8451	76.5054	81.6947	80.6411	86.2854	86.3926
43	76.7818	76.1740	80.7453	79.7326	85.2963	85.3187
44	76.8464	76.1410	80.7974	79.6600	85.5354	85.5746
45	76.7867	76.1863	80.4698	79.7419	85.2866	85.0326
46	76.7827	75.9269	80.3476	79.4452	85.0668	84.9441
47	77.4933	76.5608	81.1895	80.1598	85.8184	85.4771
48	77.7440	76.7519	81.3330	80.2397	85.8748	85.7433
49	77.6494	76.9101	81.4399	80.3661	86.1681	85.9747
50	77.9478	77.2947	81.8436	80.7014	86.4787	86.1532
51	78.3174	77.3202	82.1113	80.9685	86.6659	86.4004
52	78.1031	77.3208	81.9435	80.9406	86.6719	86.3247
53	78.7529	77.9602	82.6639	81.5054	87.4053	87.1512
54	78.9233	77.9809	82.5816	81.4416	87.0556	87.1462
55	78.1456	77.5124	82.0931	81.0627	86.4822	86.5715
56	78.4474	77.6840	82.1214	81.4257	86.8486	86.4883
57	78.3559	77.5430	81.9966	81.1929	86.7048	86.6216

58	78.6248	77.6520	82.4146	81.2253	87.0396	86.7537
59	79.0771	78.2177	82.9960	81.6626	87.4972	87.3889
60	78.9265	78.0340	82.5696	81.6372	87.1969	87.0795
61	78.9397	78.1408	82.7527	81.5722	87.1486	86.9754
62	78.4657	77.6172	82.3288	81.0392	86.8875	86.6712
63	78.2103	77.4855	82.0896	80.9154	86.6969	86.2687
64	78.3733	77.3546	81.9746	80.8515	86.5630	86.2820
65	78.4529	77.5097	82.0375	80.9509	86.3785	85.8023
66	78.2659	77.1991	81.9345	80.8041	86.4816	86.3653
67	77.6881	76.9810	81.5489	80.5076	85.9384	85.6634
68	77.7062	76.8978	81.5757	80.2117	86.0098	85.9094
69	77.8436	77.0766	81.5599	80.5299	86.2602	85.9712
70	78.0851	77.0189	81.8587	80.6549	86.4180	86.1277
71	77.9551	77.0438	81.5065	80.4901	86.1268	85.8849
72	77.5463	76.5947	81.0938	79.9831	85.3106	85.3238
73	77.6217	76.4003	81.2212	80.0070	85.4997	85.1367
74	77.5876	76.5916	81.1386	80.1814	85.8147	85.4505
75	77.4242	76.4082	81.0178	79.9010	85.4697	85.1264
76	77.5781	76.7768	81.2543	80.2670	85.7937	85.2846
77	77.7770	76.7527	81.3458	80.4849	85.8575	85.5494
78	78.0193	76.8384	81.5876	80.2851	86.2557	85.8520
79	77.5361	76.8038	81.3820	80.1770	85.8684	85.4262
80	77.8961	76.7919	81.5513	80.3207	86.0441	85.6624
81	78.1247	77.1693	81.9597	80.7259	86.3314	86.0640
82	77.9932	76.9455	81.7409	80.4984	86.3596	85.9371
83	77.5349	76.4766	81.2205	80.0242	85.7482	85.4153
84	77.5726	76.4206	81.1163	79.9489	85.6106	85.3724
85	78.0236	77.1563	81.7415	80.6409	86.4088	85.9498
86	78.1585	77.1189	81.9455	80.6874	86.4579	86.1719
87	77.8649	76.8489	81.6574	80.5259	86.0887	85.7070
88	77.5444	76.7346	81.1869	80.1266	85.5217	85.2413

89	77.3553	76.4745	81.2173	79.9615	85.7764	85.4325
90	77.4212	76.3522	80.9978	79.8035	85.6473	85.1417
91	77.0062	76.2217	80.9789	79.6746	85.3934	84.9846
92	76.8171	75.8274	80.5089	79.2919	84.7599	84.7415
93	76.4415	75.4169	80.3089	78.9710	84.9035	84.1730
94	76.5401	75.4702	80.2907	78.9399	84.7853	83.9983
95	76.7440	75.7491	80.4104	79.1942	85.0030	84.5440
96	76.7376	75.7250	80.6477	79.2729	85.0892	84.6365
97	77.1560	76.2936	81.0835	79.7496	85.6408	85.1560
98	77.3062	76.3178	81.0833	79.9873	85.7054	85.2755
99	77.1812	76.2343	81.0850	79.8388	85.7030	85.1046

Tablo 7.3.c) Bahçe görüntü dizisinin sinyal gürültü oranları

	Sinyal		Gürültü		Oranı	
	Kod Kitabı	Büyüklüğü		Kod Kitabı	Büyüklüğü	
Çerçeve	1024	1024	Çerçeve	1024	1024	
Sayısı	LBG	Güncelleme	Sayısı	LBG	Güncelleme	
1	101.5275	101.5275	51	95.1197	94.0560	
2	102.0149	100.0485	52	94.8449	94.3133	
3	102.2414	100.3077	53	95.9699	94.6961	
4	102.4762	100.9069	54	96.0380	94.6878	
5	102.6368	100.3872	55	95.1266	93.9491	
6	101.6325	100.0533	56	95.0570	93.9321	
7	100.9737	99.2748	57	95.0243	93.8448	

8	100.5813	98.4963	58	95.3958	94.0339
9	100.3264	98.5641	59	95.9576	94.6679
10	100.3599	98.4666	60	95.6964	94.5219
11	99.3987	97.5689	61	95.7122	94.0977
12	99.2263	97.5524	62	95.3786	94.1037
13	99.2885	97.2848	63	94.5902	93.7745
14	99.8177	98.1032	64	94.8553	94.0369
15	99.3325	97.2488	65	94.8230	93.5804
16	99.2800	98.0107	66	94.7636	93.9821
17	99.1844	97.5498	67	94.0257	93.0332
18	99.2888	97.5726	68	94.1361	93.4966
19	98.6643	97.3172	69	94.2389	93.4342
20	98.2030	96.7213	70	94.6950	93.8974
21	98.0071	96.5554	71	94.1517	93.2319
22	98.0208	96.4047	72	93.2561	92.5393
23	97.9642	97.1192	73	93.4334	92.5252
24	98.1962	96.5060	74	93.5028	92.8209
25	98.2555	96.2859	75	93.2717	92.5362
26	97.8955	96.5759	76	93.9871	93.1114
27	98.4808	97.2261	77	93.7029	93.1065
28	99.1242	97.6564	78	94.2188	93.5523
29	98.6183	97.1028	79	93.9376	93.2222
30	97.7896	96.6110	80	93.7333	93.2870
31	98.6330	97.3126	81	94.6181	93.5345
32	97.9313	96.5042	82	94.3999	93.7242
33	98.1833	96.6794	83	93.6923	92.8703
34	97.7055	96.4325	84	93.6638	93.0121
35	97.3792	95.8576	85	94.4532	93.3158
36	97.2774	95.7315	86	94.7208	93.5746
37	96.5819	95.1495	87	94.2018	93.3677
38	96.4843	94.7722	88	93.8389	92.7383

39	95.3007	94.0557	89	93.7299	92.9413
40	95.2208	93.7182	90	93.4247	92.9067
41	95.7589	94.2553	91	92.9740	92.5111
42	95.4151	94.2745	92	92.9260	92.1164
43	94.0656	93.1008	93	92.3857	91.7764
44	94.3204	93.1346	94	92.4804	92.2464
45	93.9726	93.0751	95	92.6896	92.1964
46	93.6116	92.7127	96	92.9000	92.0928
47	94.4277	93.2963	97	93.8241	92.9409
48	94.2407	93.2017	98	93.6195	92.8490
49	94.5124	93.6761	99	93.5046	92.9740
50	94.8541	94.1695			

Tablo 7.3.d) Bahçe görüntü dizisinin sinyal gürültü oranları



Şekil 7.20.a) 1.çerçeve orijinal görüntü



Şekil 7.21.a) 25. çerçeve orijinal görüntü



Şekil 7.20.b) LBG algoritması sonucu



Şekil 7.21.b) LBG algoritması sonucu



Şekil 7.20.c) Güncelleme algoritması sonucu



Şekil 7.21.c) Güncelleme algoritması sonucu



Şekil 7.22.a) 50. çerçeve orijinal görüntü



Şekil 7.23.a) 75. çerçeve orijinal görüntü



Şekil 7.22.b) LBG algoritması sonucu



Şekil 7.23.b) LBG algoritması sonucu



Şekil 7.22.c) Güncelleme algoritması sonucu



Şekil 7.23.c) Güncelleme algoritması sonucu



Şekil 7.24.a) 99. çerçeve orijinal görüntü



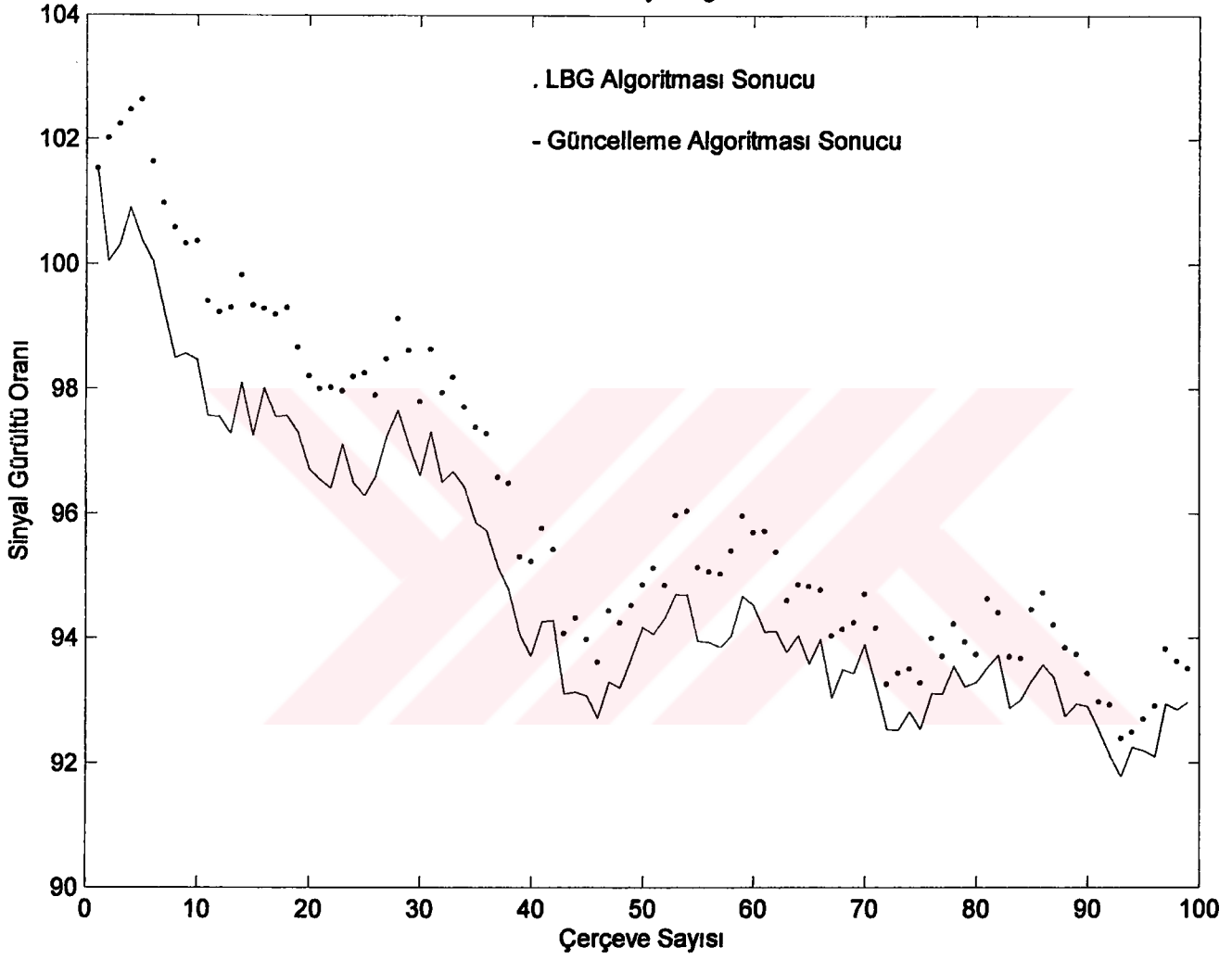
Şekil 7.24.b) LBG algoritması sonucu



Şekil 7.24.c) Güncelleme algoritması sonucu

Yukarıda verilen şekillerde kod kitabı büyüklüğü olarak 1024 alınmıştır.

Kod Kitabı Büyüklüğü 1024



BÖLÜM 8

SONUÇLAR

7. bölümde hareketli görüntülerin her biri için çeşitli kod kitabı büyüklüğünde sinyal gürültü oranları tablolar halinde verilmiştir (Tablo 7.1, Tablo 7.2, Tablo 7.3). Tablolar incelendiğinde en iyi sinyal gürültü oranı doğal olarak kod kitabı büyüklüğü 1024 olduğu zaman bulunmuştur. Kod kitabı büyüklüğü artırıldığı zaman sinyal gürültü oranı artmakta, ancak sıkıştırma oranı azalmaktadır. Ayrıca kod kitabı büyüklüğü ile gerçekleştirme zamanı arasında ters orantı vardır. Kod kitabı büyüklüğü 1024 olduğunda elde edilen sinyal gürültü oranları aşağıdaki tabloda daha genel olarak tekrar verilmiş ve her çerçevede bulunan sinyal gürültü oranları grafiksel olarak verilmiştir. Ayrıca bu kod kitabı büyüklüğünde (1024) LBG algoritması ve güncelleme algoritmasının gerçekleştirme zamanı ayrı bir tabloda ve grafiksel olarak verilmiştir. Tablo 8.1 'de de görüldüğü gibi görüntü dizilerinin her çerçevesine LBG algoritması uygulandığında futbol dizisi için toplam 7.2956×10^3 db, tenis dizisi için 4.0885×10^3 db ve bahçe dizisi içinde 9.5214×10^3 db'lik sinyal gürültü oranları bulunmuştur. İlk çerçeveye LBG algoritması uygulanarak bulunan kod kitabını diğer çerçevelerde de kullanıldığı zaman elde edilen sinyal gürültü oranları sırasıyla 5.9626×10^3 db, 3.5358×10^3 db ve 8.0310×10^3 db olarak bulunmuştur. Güncelleme algoritması kullanıldığı zaman sinyal gürültü oranları 7.2956×10^3 db, 4.0684×10^3 db ve 9.4059×10^3 db olarak bulunmuştur. Buradan da hesaplanacağı gibi bütün çerçeveler toplamında, ilk çerçeveye LBG algoritması uygulanması ile her çerçeveye

Toplam	Sinyal Gürültü Oranı (db)		
	Her Çerçeveye	İlk Çerçeveye	Güncelleme Algoritması
Futbol Dizisi	7.3668×10^3	5.9626×10^3	7.2956×10^3
Tenis Dizisi	4.0885×10^3	3.5358×10^3	4.0684×10^3
Bahçe Dizisi	9.5214×10^3	8.0310×10^3	9.4059×10^3

Tablo 8.1 : Görüntü dizilerinde bulunan toplam sinyal gürültü oranı

Ortalama	Sinyal Gürültü Oranı (db)		
	Her Çerçeveye	İlk Çerçeveye	Güncelleme Algoritması
Futbol Dizisi	122.7800	99.3767	121.5933
Tenis Dizisi	104.8333	90.6615	104.3179
Bahçe Dizisi	96.1757	81.1212	95.0090

Tablo 8.2 : Görüntü dizilerinde bulunan ortalama sinyal gürültü oranı

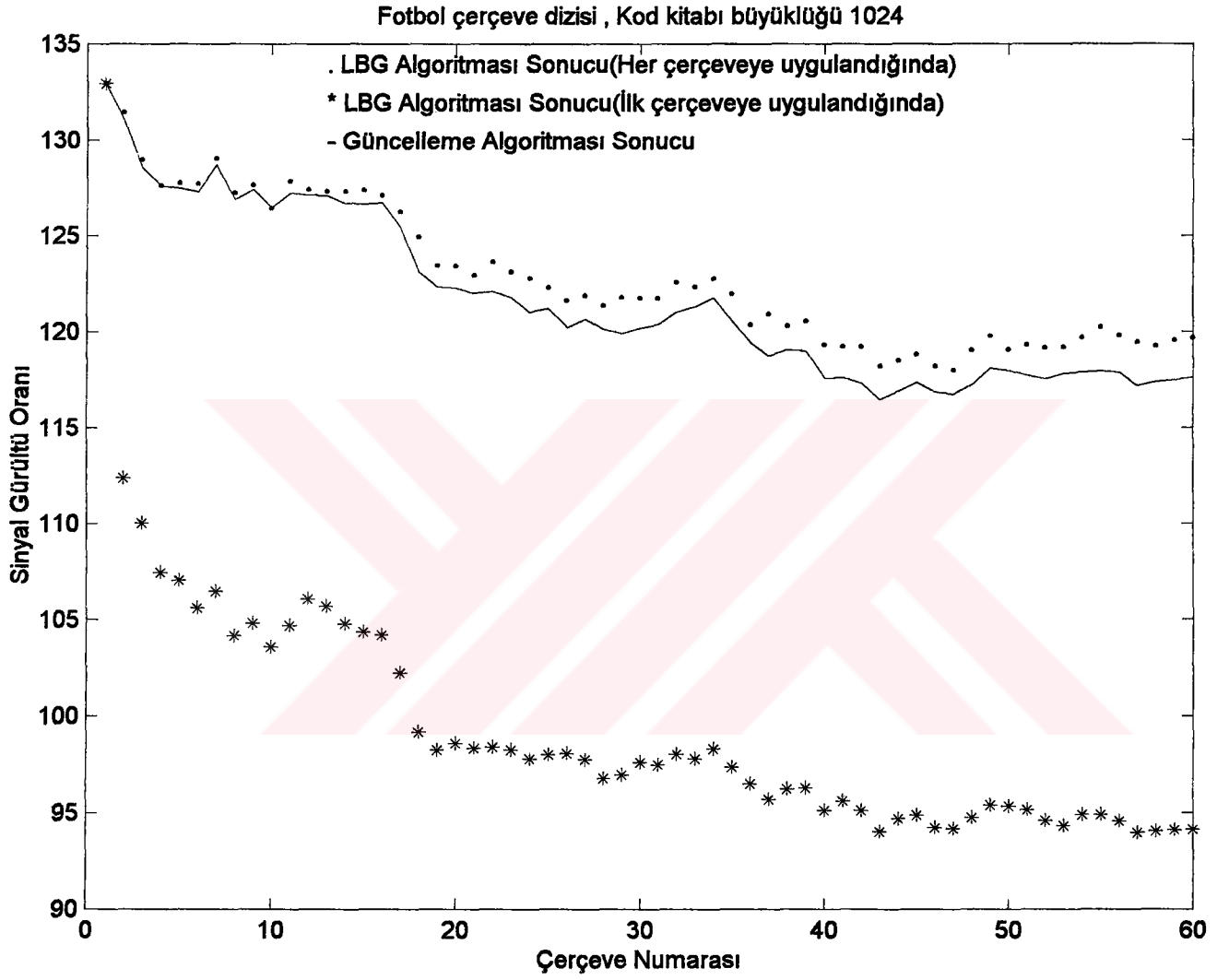
uygulanması arasında futbol dizisi için 1404.2 db, tenis dizisi için 552.7 db ve bahçe dizisi içinde 1490.4 db' lik bir bozulma olmaktadır. Bu da her çerçeve için ortalama olarak sırasıyla 23.4033 db, 15.0545 db ve 14.1718 db bozulmaya sebep olmaktadır. Yani futbol dizisi için %19.06, tenis dizisi için %13.52 ve bahçe dizisi içinde %15.65 oranında bozulma olduğu görülmüştür.

Güncelleme algoritması kullanıldığı zaman bozulma futbol dizisi için 1.1867 db (% 0.96), tenis dizisi için 0.5154 db (% 0.49) ve bahçe dizisi içinde 1.1667 db (% 1.21) olmuştur. Bu bozulma miktarına karşılık futbol dizisi için 40064 sn, tenis dizisi için 23486 sn ve bahçe dizisi içinde 52892 sn kazanılmıştır. Bu da her çerçeve için ortalama olarak sırasıyla 667.73 sn, 602.21 sn ve 534.26 sn kazandırmıştır. Yani gerçekleştirme zamanında futbol dizisi için 3.61:1, tenis dizisi için 3.78:1 ve bahçe dizisi için 3.39:1 oranında zaman kazanım gerçekleşmiştir.

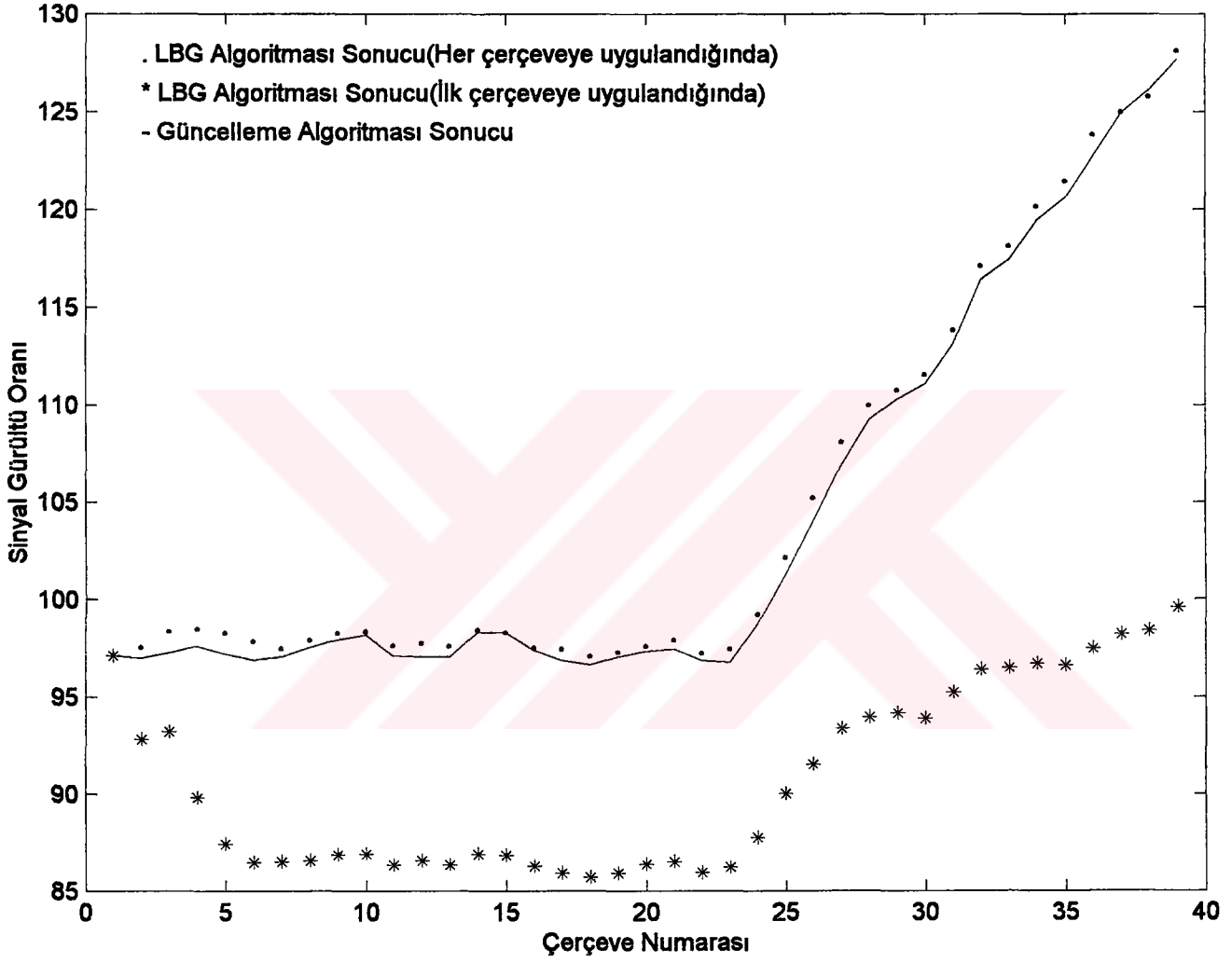
Önerilen güncelleme algoritması ile yüksek sıkıştırma oranı elde edilebilen (11:512) vektör kuantalamada en çok kullanılan LBG algoritmasınınin büyük problemi olan gerçekleştirme zamanı düşük bozulma oranları ile birlikte azaltılabilmektedir.

	Gerçekleme Zamanı (sn)			
	Toplam Zaman		Ortalama Zaman	
	LBG Alg.	GüncellemeAlg.	LBG Alg.	GüncellemeAlg.
Futbol Dizisi	55391	15327	923.18	255.45
Tenis Dizisi	31937	8451	818.90	216.69
Bahçe Dizisi	75030	22138	757.88	223.62

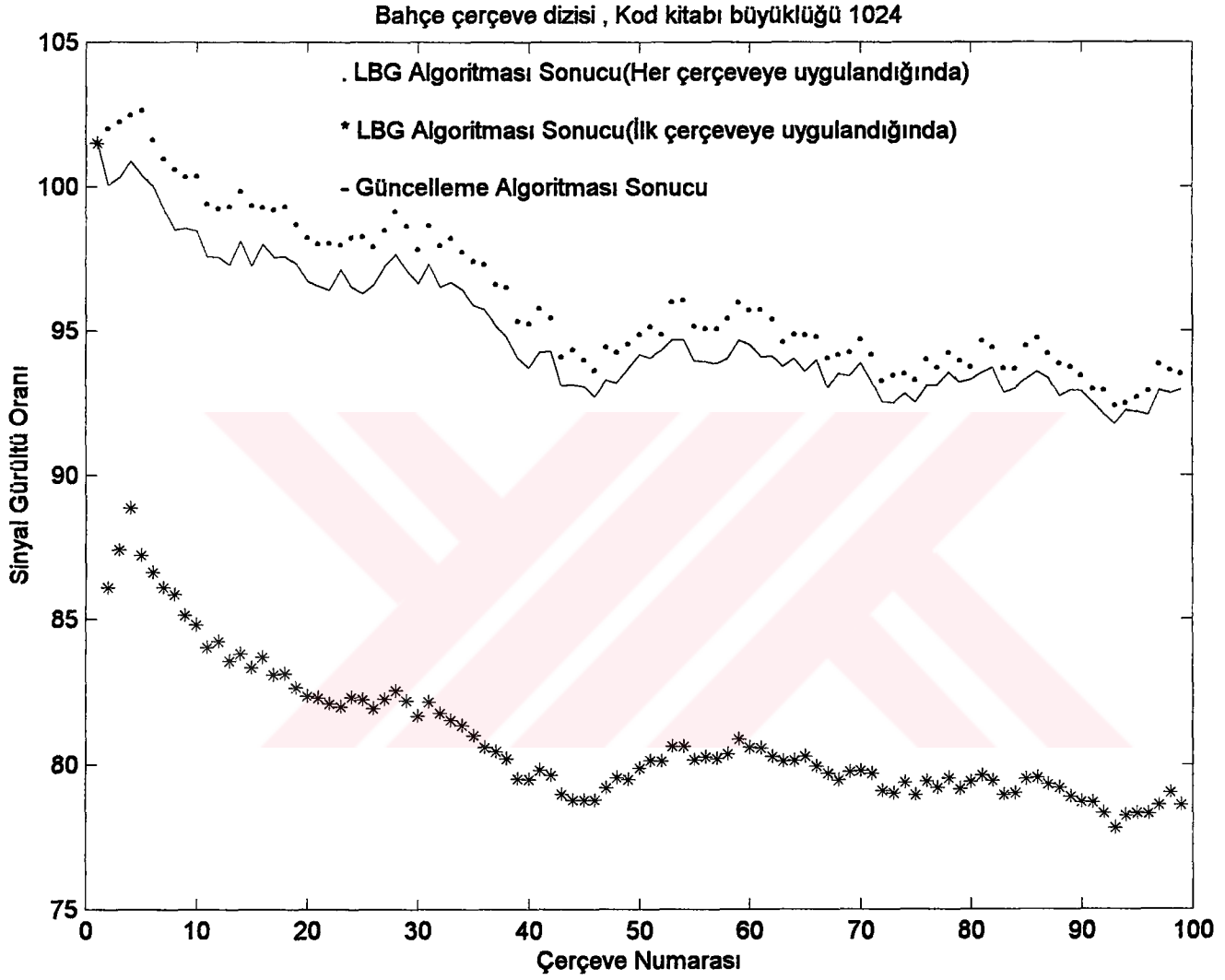
Tablo 8.3 : Görüntü dizilerinin gerçekleştirme zamanları

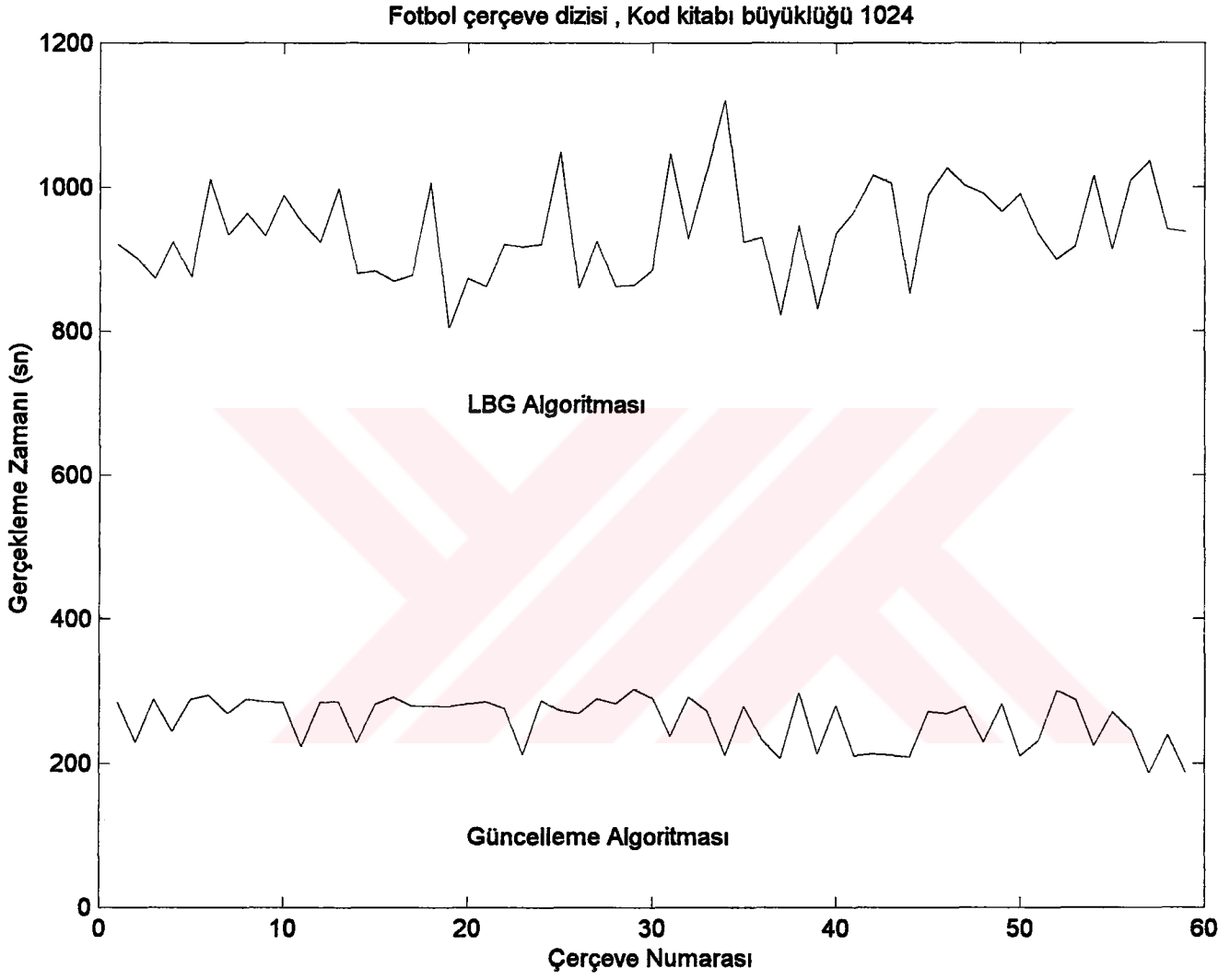


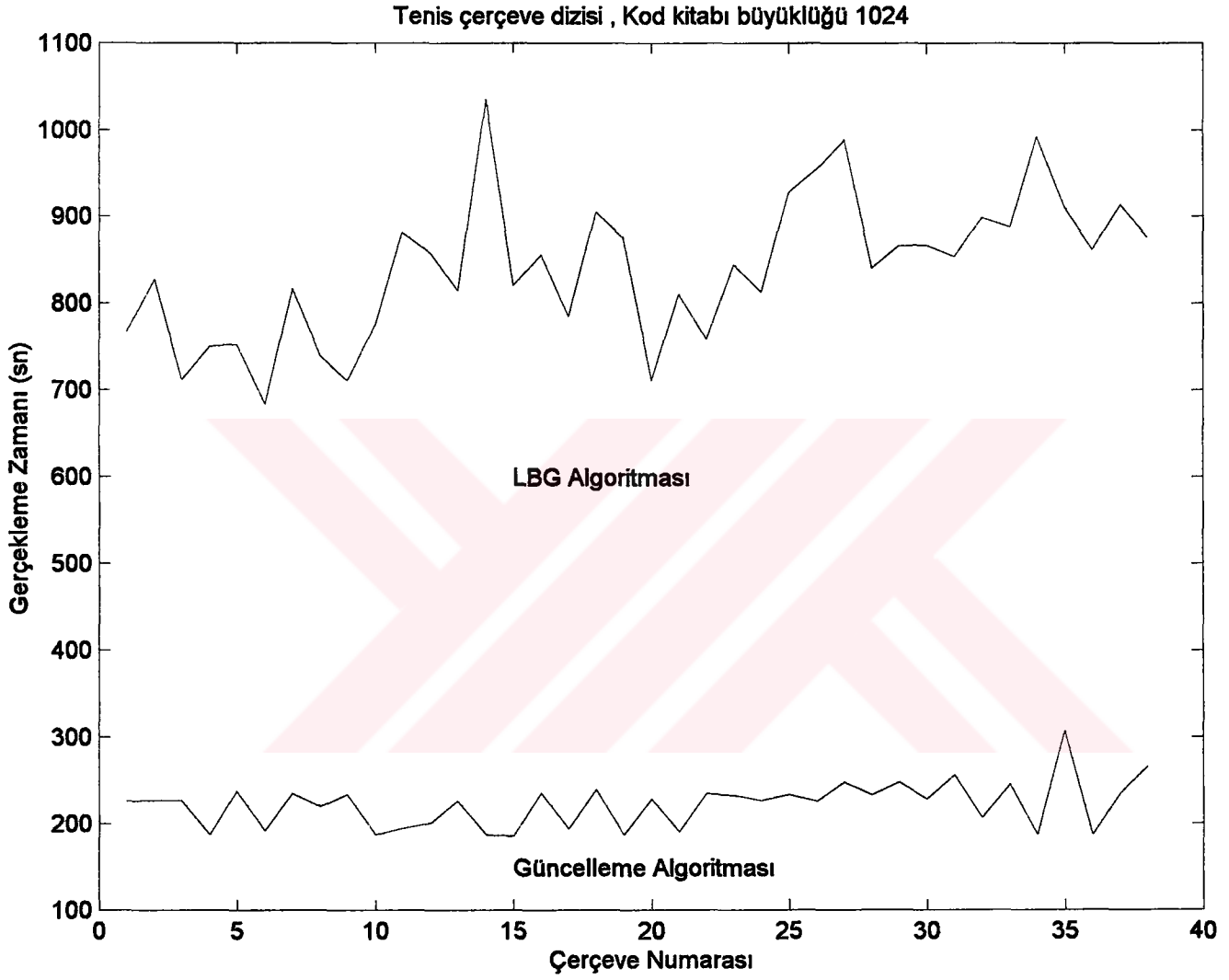
Tenis çerçeve dizisi , Kod kitabı büyüklüğü 1024

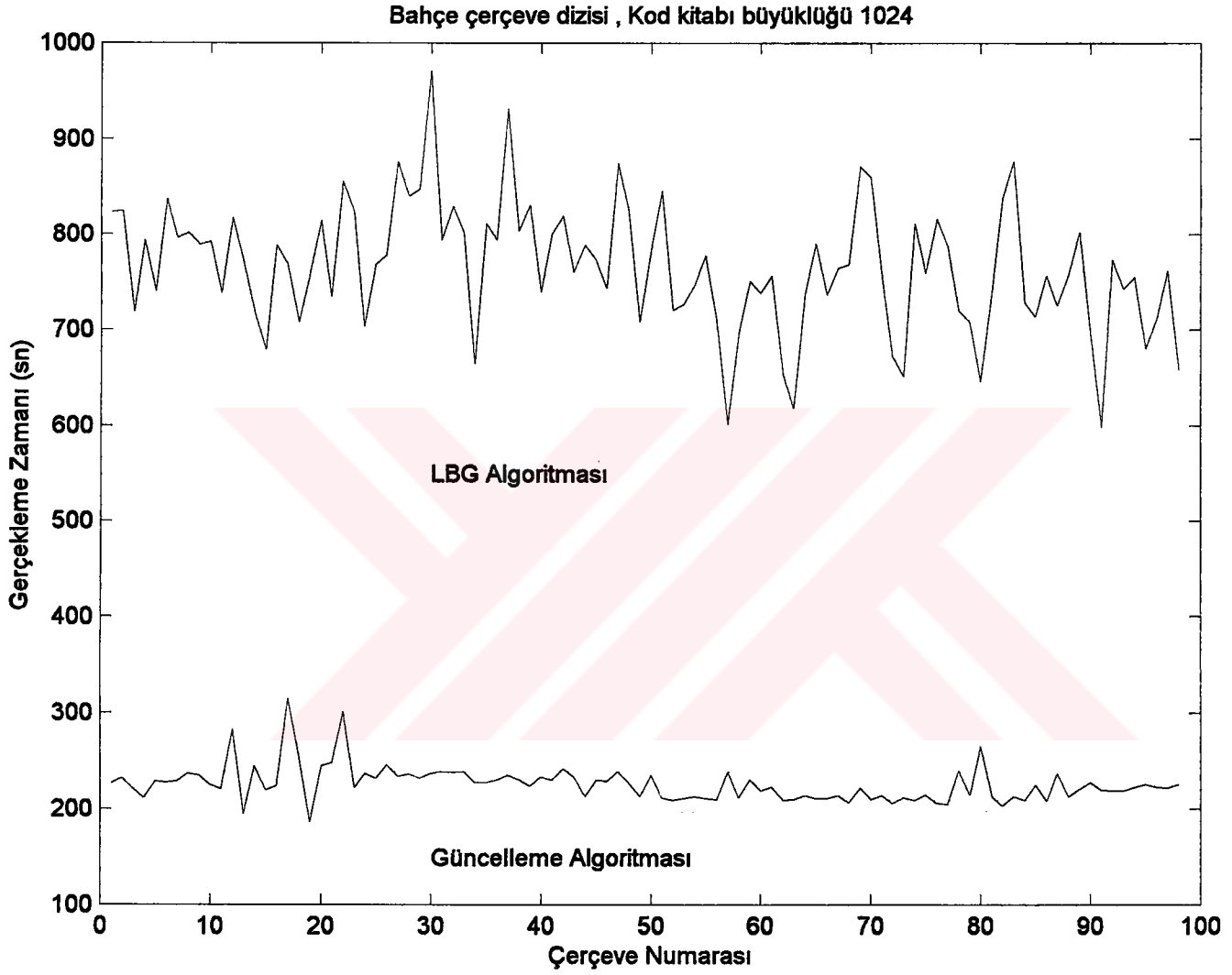


TC YÜKSEKÖĞRETİM KURULU
DOKÜMAN YAYIN MERKEZİ









KAYNAKLAR DİZİNİ

- [1] **Umbaugh, S.E.**, 1998, Computer Vision and Image Processing, Prentice-Hall International, Inc.
- [2] **Gonzalez, R., Woods, R.**, 1992, Digital Image Processing, Addison-Wesley Publishing Company.
- [3] **Nelson, M.**, 1992, The Data Compression Book, M&T Publishing, Inc.
- [4] **Li, W., Zhang, Y.**, “A Framework of Transform Domain VQ for Image Coding”, Processing of the 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, pp 687-690.
- [6] **Li, W.**, “Vector Transform and Image Coding”, IEEE Transactions on Circuits and Systems for Video Tech., Vol.1, No.4, pp 297-307, December 1991.
- [7] **<ftp://ftp.uu.net/graphics/jpeg>**
- [8] **<http://www.am.hhi.de/mpeg-video>**
- [9] **Gerso, A., Gray, R.**, 1992, “Vector Quantization and Signal Processing”, Kluwer Academic Publishers.
- [10] **Nasrabadi, N., King, R.**, “Image Coding Using Vector Quantization: A Review”, IEEE Transactions on Communications, Vol.36, No.8, August 1988.
- [11] **Dedeakyogulları, B.**, 1997, Yüksek Lisans Tezi, Osmangazi Üniversitesi Fen Bilimleri Enstitüsü.
- [12] **Buzo, A., Gray, H., Gray, R., Markel, J.**, “Speech Coding Based upon Vector Quantization”, IEEE Transactions on Acoustics, Speech and Signal Processing”, vol.ASSp-28, No.5, October 1980.
- [13] **Seke, E.**, 1995, Ph.D Thesis , Lehigh University.
- [14] **Gray, A., Markel, J.**, “Quantization and Bit Allocation in Speech Processing”, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol.ASSP-24, pp 459-473, December 1976.
- [15] **Linde, Y., Buzo, A., Gray, R.**, “An Algorithm for Vector Quantizer Design”, IEEE Transactions on Communications, Vol.COM-28, No.1, January 1980.
- [16] **Kim, D., Lee, S.**, “Image Vector Quantizer Based on a Classification in the DCT Domain”, IEEE on Communications, Vol.39, No.4, April 1991.

- [17] **Barlaud, M.**, Pyramidal Lattice Vector Quantization for Multiscale Image Coding”,
IEEE Transactions on Image Processing, Vol.3, No.4, July 1994.
- [18] **Nasrabadi, N., Feng, Y.**, “Image Compression Using Address-Vector quantization”,
IEEE Transactions on Communications, Vol.38, No.12, December 1990.
- [19] **Gray, R.**, “Vector Quantization”, IEEE ASSP Magazine April 1984.

