

**KOCAELİ ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ**

**VEKTÖR NİCEMLEME İLE VIDEO GÖRÜNTÜ SIKIŞTIRMA**

**YÜKSEK LİSANS TEZİ**

**Elektronik ve Haberleşme Müh. Erkan KOCAKAYA**

**Anabilim Dalı: Elektronik ve Haberleşme Mühendisliği**

**MAYIS 2005**

**168823**

**KOCAELİ ÜNİVERSİTESİ \* FEN BİLİMLERİ ENSTİTÜSÜ**

**VEKTÖR NİCEMLEME İLE VIDEO GÖRÜNTÜ SIKIŞTIRMA**

**YÜKSEK LİSANS TEZİ**

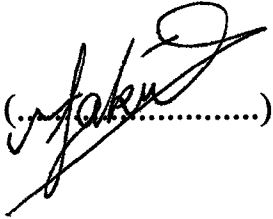
**Elektronik ve Haberleşme Müh. Erkan KOCAKAYA**

**Tezin Enstitüye Verildiği Tarih : 25 Mayıs 2005**

**Tezin Savunulduğu Tarih : 14 Eylül 2005**

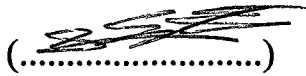
**Danışman**

Yrd. Doç Dr. Mehmet YAKUT

  
(.....)

**Üye**

Doç Dr. Sarp ERTÜRK

  
(.....)

**Üye**

Yrd. Doç Dr. Feyzullah TEMURTAŞ

  
(.....)

**MAYIS 2005**

# VEKTÖR NİCEMLEME İLE VIDEO GÖRÜNTÜ SIKIŞTIRMA

Erkan KOCAKAYA

**Anahtar Kelimeler:** Vektör Nicemleme, Video Sıkıştırma, Görüntü Nicemleme, Görüntü Kodlama, Görüntü Sıkıştırma, MPEG Sıkıştırma

**Özet:** Günlük hayatımızın bir parçası olan görüntü teknolojileri gün geçtikte ilerlemekte ve buna bağlı olarak da ses ve görüntü kalitesindeki ihtiyaçlar artmaktadır. Analog olarak kullanılmaya başlanılan bu görüntüleme sistemi teknolojinin ilerlemesi ile birlikte artık sayısal olarak kaydedilmekte ve işlenmektedir.

Görüntülerin sayısal bilgi haline dönüştürülmesi ile birlikte bu bilgilerin iletimi için sıkıştırma teknikleri geliştirilmiştir. Bu sayede büyük boyutlardaki görüntü bilgileri daha küçük boyutlarda daha hızlı bir şekilde iletimi sağlanır. Bu görüntü sıkıştırma formatları Hareketli Resim Uzman Grubu (MPEG) tarafından belirlenmiştir. MPEG-1, MPEG-2 vb. çeşitli görüntü sıkıştırma formatları belirlenmiş ve ihtiyaca göre kullanılmaktadır.

İmge ve İmge Dizileri şuan birçok teknoloji de kullanılmaktadır. İnternet ve 3. Nesil mobil telefonlar üzerinden görüntü iletimi ise giderek yaygınlaşmakta buna bağlı olarak ta daha iyi sıkıştırma ve daha iyi görüntü kaliteleri gerekmektedir. Bu amaçla Vektör Nicemleme ile Görüntü kodlama yöntemi geliştirilmiş olup görüntü kalitesinden en az oranda kayıp vererek en iyi sıkıştırma elde edilmeye çalışılmaktadır.

Bu çalışmada bir imge dizisinin vektör nicemleme yöntemi kullanılarak kodlanması ve kodlanan bilginin bir çözücü sistemi ile imge dizisinin yeniden elde edilmesi amaçlanmıştır. Bu doğrultuda gelen imge dizisindeki ilk imge vektör gruplarına ayrılarak GLA Algoritması ile eğitime tabi tutulmaktadır. Bu eğitimde amaç, istenen kod vektör sayısına ulaşıncaya kadar imge vektör gruplarını temsil edecek kod vektörlerinin oluşturulması, tespit edilmesi ve indekslenmesidir. Kod kitabının oluşturulmasından sonra imgedeki vektör gruplarını en iyi temsil eden kod vektörlerin indeks numaraları belirlenir ve bu bilgi kod çözücü kısma ulaştırılarak belirlenen indeks numaralarına ait olan kod vektörlerinin birleştirilmesi ile imgenin tekrar elde edilmesi gerçekleştirilmiştir. İlk imgeden sonra gelen imgeler bu kod kitabına bağlı olarak kodlanmış her yeni imge öncesinde PSNR değeri tespit edilerek eşik değerinin altında olması durumunda ise yeni bir kod kitabı oluşturulmuştur. Bu imgeden sonra gelenler ise yeni bir kod kitabına gereksinim duyuluncaya kadar bu yeni kod kitabından kodlanmışlardır.

Yapılan bu çalışma ile kod kitabı ve indeks bilgilerinden oluşan bir kodlanmış bilgi elde edilmiştir. Sıkıştırma oranı seçilen vektör boyutunun büyük seçilmesi ile artmış bununla beraber görüntü kalitesinde azalma meydana gelmiştir. Kod kitabını büyüklüğüne bağlı olarak görüntü kalitesi artmakta fakat sıkıştırma oranı azalmakta ve bir imgenin kodlanması için geçen süre uzamaktadır.

Sonuç olarak alıcı taraftaki basit bir kod çözücüsünden oluşan vektör nicemleme sistemi imge dizileri için önerilmiş ve gerçekleştirilmiştir.

# VIDEO COMPRESSION WITH VECTOR QUANTIZATION

Erkan KOCAKAYA

**Keywords:** Vector Quantization, Video Compression, Image Quantization, Image Coding, Image Compression, MPEG

**Summary:** As a part of our lives, image technology has gradually developed and depending on this development, the needs for image and sound quality is also rising. Having been used as analogue, as the result of growing technology, the images are now stored and processed digitally.

As a result of conversion of images into digital data, compression techniques have been improved for the transmission of the data. By compression, the big sized data is transmitted in smaller sizes and faster. Still image and video formats were designated by Moving Pictures Experts Group. The image compression formats such as MPEG1 and MPEG 2 are used according to the requirements.

Image transmission is now used in many fields i.e. Internet and 3G mobile phone has been spreading so better image quality and image compression are required. For this reason, vector quantization and image coding methods have been improved and it is attempted to get the better compression with the minimum loss in image quality.

In this study, it is aimed to an image sequences are coded by vector quantization and video sequences are reconstructed from coded data by decoding system. First image sequence is divided into groups and it is trained using GLA algorithm. Purpose of this training, code vectors that represent group of image vector are performed detecting and indexing, until achieving number of desired code vector. After performing the codebook, it is determined number of index code vectors that representing better vector groups in the image. By this information that is sent to decoder it is realized to reconstructed the image by combining code vectors belong to numbers of index. The images following the first one are coded depend on the code book and the PSNR value of each new image was determined and a new codebook was formed if it is below the threshold value. The images following this image are coded using the new codebook till a new requirement for a new codebook.

By this study, we obtained the coded data formed by the code book and index particulars. The compression rate increased by setting the vector size as large, although the quality of image is reduced. The quality of image is increased relating to the size of the code book but compression rate decreased and the time for the image coding is increased.

Conclusively, the vector quantization system formed by a decoder on the side of receiver was suggested for image series and fulfilled

## ÖNSÖZ ve TEŞEKKÜR

Günümüzün vazgeçilmez ihtiyaçlarından olan görüntü teknolojileri hızla gelişmekte ve hayatımızın her alanında kullanılmaktadır. Görüntü işleme yöntemleri; mobil iletişimden tıbbi cihazlara askeri uygulamalardan görsel medyaya kadar teknolojinin her alanında kullanılmaktadır. Bundan dolayı görüntünün iletilmesi ve saklanması konusu giderek önem kazanmaktadır. Giderek artan bilgi miktarına bağlı olarak bu bilgilerin saklanması ve iletilmesi konusunda sıkıştırma teknikleri ortaya çıkmıştır.

Yapılan araştırmalar ile birlikte geliştirilen görüntü sıkıştırma yöntemleri ile bu ihtiyaçlar karşılanmaya çalışılmaktadır. Vektör Nicemleme ile sıkıştırma ise bu yöntemlerden biridir. Bir örneği nicemlemek yerine bir örnek bloğunu nicemleme prensibine dayanan bu yöntem ile yüksek sıkıştırma oranlarına düşük bozulmalar ile ulaşılabilmektedir. Vektör nicemleme’de sıkıştırma işlemi uzun zaman alırken, Sıkıştırılmış verinin tekrar elde edilmesi basittir ve kısadır. Bu çalışma ile görüntü sıkıştırma yöntemleri üzerine araştırma yapanlara katkısı olmasını dilerim

Bana bu konuda çalışma yapmama imkân sağlayan, çalışmalarımnda gösterdiği destek ve yönlendirmeleri ile bana her konuda yardımcı olan değerli hocam Yrd. Doç. Dr. Mehmet YAKUT’a, yaşamım boyunca bana katkı ve desteklerini hiç eksik etmeyen aileme ve bu çalışmayı gerçekleştirirken destek, sevgi ve huzuru hiç eksik etmeyen eşim Aybike’ye teşekkür ederim.

Mayıs 2005, KOCAELİ

Erkan KOCAKAYA

## İÇİNDEKİLER

ÖZET .....	ii
ABSTRACT.....	iii
ÖNSÖZ ve TEŞEKKÜR .....	iv
İÇİNDEKİLER.....	v
SİMGELER DİZİNİ VE KISALTMALAR .....	vii
ŞEKİLLER LİSTESİ .....	ix
TABLolar LİSTESİ.....	xi
BÖLÜM 1. GİRİŞ.....	1
BÖLÜM 2. VERİ SIKIŞTIRMA VE VİDEO KODLAMA .....	3
2.1 Kayıpsız Sıkıştırma Yöntemleri.....	3
2.1.1 Entropi kodlama.....	3
2.1.2 Huffman kodlama .....	4
2.1.3 Aritmetik kodlama .....	7
2.1.4 Run-Length kodlama .....	10
2.2 Kayıplı Sıkıştırma Yöntemleri.....	11
2.2.1 Skaler nicemleme.....	11
2.2.2 Dönüşüm kodlaması .....	12
2.2.3 Ayrık kosinüs dönüşümü .....	13
2.2.4 Dalgacık dönüşümü .....	16
2.2.5 MPEG sıkıştırma.....	21
2.2.5.1 MPEG-1 video formatı .....	23
2.2.5.1.1 Çerçeve kodlama.....	24
2.2.5.1.2 MPEG-1 sıkıştırma yapısı.....	27
2.2.5.2 MPEG-2 sıkıştırma yapısı.....	28
2.2.5.3 MPEG-4 sıkıştırma yapısı.....	30
2.2.5.3.1 MPEG-4'ün özellikleri .....	30
2.2.5.4 MPEG-7 sıkıştırma yapısı.....	35
BÖLÜM 3. VEKTÖR NİCEMLEME.....	36
3.1 Temel Tanımlamalar ve Özellikler .....	37

3.2. Kod Kitabı Tasarımı .....	40
3.2.1 Genelleştirilmiş lloyd algoritması.....	40
3.2.2. Kod kitabı bölümlenme .....	42
3.3 Yapısal Vektör Nicemleme.....	44
3.3.1 Ağaç yapısında vektör nicemleme.....	44
3.3.2 Ortalaması atılmış vektör nicemeleme .....	45
3.3.3 Çarpım kod vektör nicemeleme .....	47
3.4 Adaptif Vektör Nicemleme.....	48
BÖLÜM 4. VEKTÖR NİCEMLEME İLE MPEG.....	49
SONUÇLAR VE ÖNERİLER.....	55
KAYNAKLAR.....	61
ÖZGEÇMİŞ.....	63



## SİMGELER DİZİNİ ve KISALTMALAR

$X$	: Değişken Kümesi
$N$	: Değişken Sayısı
$H$	: Entropi Kodlayıcı
$p_x$	: Olasılık Ağırlık Fonksiyonu
$C$	: Kod Kitabı
$I$	: İndeks Dizisi
$\alpha$	: Skaler ve Vektörel Nicemleme Kodlayıcısı
$\beta$	: Skaler ve Vektörel Nicemleme Kod Çözücüsü
$Q$	: Nicelenmiş Bilgi Fonksiyonu
$N$	: Piksel Sayısı
$M$	: Blok Sayısı
$W$	: Dalgacık Dönüşüm Fonksiyonu
$\psi$	: Ana Dalgacık
$\mathcal{T}$	: Dönüşüm Parametresi
$H_0$	: Yüksek Geçiren Filtre
$G_0$	: Alçak Geçiren Filtre
$\hat{x}$	: Vektör Nicemlenmiş Bilgi
$\gamma$	: İndeks Kodlayıcı
$D$	: Ortalama Bozluma
$R$	: Ortalama Oranı
$E$	: Beklenti Operatörü
$\mu$	: Skaler Nicemleyici Ortalaması
AK	: Aritmetik Kodlama
RLC	: Run-Length Kodlama
SN	: Skaler Nicemeleme
KLD	: Karhunen-Loeve Dönüşümü
AKD	: Ayrık Kosinüs Dönüşümü (DCT)



KZFD : Kısa zamanlı fourier dönüştürümü  
SDD : Sürekli Dalgacık dönüştürümü  
ADD : Ayrık Dalgacık dönüştürümü  
MPEG : Moving Picture Expert Group  
YUV : Y, Cr ve Cb Çerçeve Grubu  
RGB : Kırmızı, Yeşil ve Mavi Renk Formatı  
VS : Video Oturumu  
VO : Video Nesnesi  
VOL : Video Nesne Katmanı  
GOV : Video Nesne Grubu  
GOP : Resim Grubu  
VOP : Video Nesne Uzayı  
DS : Tasarı Açıklaması  
AVN : Adaptif Vektör Nicemleme  
VN : Vektör Nicemleme  
GLA : Genelleştirilmiş Lloyd Algoritması  
AYVN: Ağaç Yapısında Vektör Nicemleme  
OAVN: Ortalaması Atılmış Vektör Nicemleme (MRVQ)  
ÇKVN: Çarpım Kod Vektör Nicemleme(PCVQ)  
AVN : Adaptif Vektör Nicemleme(AVQ)

## ŞEKİLLER DİZİNİ

Şekil 2.1	Ağaç Formunda Huffman Kod Gösterimi .....	6
Şekil 2.2	Run-Length Kodlama Örneği .....	10
Şekil 2.3	Dönüşüm Kodlama Ve Dönüşüm Kod Çözücü Blokları.....	12
Şekil 2.4	Akd Gösterimi.....	13
Şekil 2.5	Akd Katsayılarının Gösterimi.....	14
Şekil 2.6	İmge Ve Akd Örnekleri .....	15
Şekil 2.7	Zig-Zag Tarama Gösterimi .....	16
Şekil 2.8	Bir Sinyalin Dalgacık Gösterimi.....	16
Şekil 2.9	Bir İşaretin Filtre Grubuna Uygulanması .....	17
Şekil 2.10	Üç Katlı Dalgacık Bölünme Ağacı .....	19
Şekil 2.11	İki Katmanlı Dalgacık Kodlama .....	19
Şekil 2.12	Çapraz Bant Kodlama İçin Sıfır Ağaç Yapısı.....	20
Şekil 2.13	Üç Katlı Dalgacık‘Tan Tekrar Elde Etme Ağacı.....	20
Şekil 2.14	1.Aşama Add Görüntü Sonuçları.....	21
Şekil 2.15	MPEG-1 İçin 4:2:0 Renk Alt Örnekleme Formatı.....	24
Şekil 2.16	Bir Görüntü Dizisinin Piksel Ve Makro Blok Temelli Gösterimi .....	25
Şekil 2.17	4:2:0 Örnekleme Blok Formatı .....	25
Şekil 2.18	I Çerçeve Kodlanması.....	26
Şekil 2.19	Çerçeveler Arasında Geçişler .....	26
Şekil 2.20	Zig-Zag Tarama Örneği.....	27
Şekil 2.21	Mpeg Dönüşüm Blokları .....	28
Şekil 2.22	Mpeg-2 Kodlama Özelliği .....	29
Şekli 2.23	Mpeg-4 Yapısı .....	31
Şekil 2.24	Mpeg-4 ‘De Video Dizisi Nesneleri .....	33
Şekil 2.25	Nesne Tabanlı Kodlama.....	34
Şekil 2.26	Vop Kodlama Yapısı .....	34
Şekil 2.27	Mpeg-7 Yapısı .....	35
Şekil 3.1	Vektör Nicemleme Blok Diyagramı .....	36

Şekil 3.2	Uyarlamalı Vektör Nicemleme Blok Diyagramı .....	37
Şekil 3.3	Vektör Nicemleme Yapısı .....	38
Şekil 3.4	Vektör Nicemleme’de Vektör ve Kod Kelimeleri .....	41
Şekil 3.5	Bölümleme Algoritması.....	43
Şekil 3.6	Ağaç Yapısında Vektör Nicemleme .....	45
Şekil 3.7	Ortalaması Atılmış Vektör Nicemleme Yapısı.....	46
Şekil 3.8	Bağımsız Nicemleyiciler İçin Çarpım Kod Vektör Nicemleme .....	48
Şekil 4.1	Vektör Nicemleme Örneği.....	49
Şekil 5.1	Lab. 38. ve 202 çerçevelerinin orijinal ve kodlanmış görüntüleri .....	55
Şekil 5.2	Miss America 36. çerçevenin orijinal ve kodlanmış görüntüsü.....	56
Şekil 5.3	Miss America 57. çerçevenin orijinal ve kodlanmış görüntüsü.....	57
Şekil 5.4	Miss America 43. çerçevenin orijinal ve kodlanmış görüntüsü.....	58
Şekil 5.5	Laboratuvar 35. çerçevenin orijinal ve kodlanmış görüntüsü.....	59

## TABLULAR DİZİNİ

Tablo 2.1	Huffman Kod Oluşturma Tablosu .....	5
Tablo 2.2	Karakter Olasılık Tablosu.....	7
Tablo 2.3	Karakter Aralık Tablosu .....	8
Tablo 2.4	Kod Tablosu.....	9
Tablo 2.5	Tekrar Elde Etme Tablosu .....	10
Tablo 4.1	Vektör Nicemleme ile MPEG Yapısının Karşılaştırılması.....	53
Tablo 4.2	Vektör Nicemleme ile MPEG Kodlama Yapılarının işlem adımlarının karşılaştırılması .....	53
Tablo 4.3	Ölçüm Sonuçları .....	54

## BÖLÜM 1. GİRİŞ

Günlük hayatımızın bir parçası olan görüntü ve video sistemlerinin gelişmesiyle sunulan görüntüler daha kaliteli hale gelmektedir. Buna bağlı olarak bilgi miktarı artmakta ve iletimi için daha büyük bant genişliklerine ihtiyaç duyulmaktadır. Bu amaçla veri sıkıştırma konusunda birçok alanda çalışılmakta ve başarılı sonuçlar elde edilmektedir ( N. Gregory 2001).

Yapılan çalışmalar ile imge ve imge dizileri üzerine birçok veri sıkıştırma tekniği geliştirilmiştir. Bunları kayıplı/kayıpsız, öngörülü/öngörüsüz ve dönüşüm kodlama algoritmaları olarak gruplandırabilir. Bu çalışmamızda bu algoritmalar ve çalışma yapıları konusunda bilgi verilmiştir.

İmge ve imge dizilerinin kodlaması konusunda birçok çalışma grupları çeşitli standartlar geliştirmiştir. MPEG (Motion Picture Expert Group) ve JPEG (Joint Photographic Experts Group) bunlardan bazılarıdır. MPEG hareketli imge dizileri konusunda MPEG-1, MPEG-2 ve MPEG-4 gibi standartlar oluşturmuştur. Bu algoritmalar kayıplı bir yapıya sahiptir.

Bu çalışmada ikinci bölümde veri sıkıştırma algoritma yapıları anlatılmıştır. İlk olarak entropi kodlama yönteminden bahsedilmiş, temeli sık karşılaşılan karakterlerin daha kısa bit uzunluğu ile kodlanarak gönderilmesi temeline dayanan bir yöntemdir. Huffman, aritmetik kodlama ve run-length algoritmaları bu yapıya örnek olarak verilebilir. Kayıplı sıkıştırma yöntemleri ise bir veri dizisi içerisinde örneklemeler yapılarak gerçekleştirilen tekniklerdir. Ayrık kosinüs dönüşümü (AKD) bu yöntemlerden biridir. İki boyutlu görüntü işleme uygulamalarında önemli bir dönüşüm tekniğidir. Hızlı ve yüksek sıkıştırma oranına sahip bir uygulamadır. AKD uluslararası iki imge/video standardında kullanılmaktadır. Bunlar Joint Photographic Experts Group (JPEG) ve Motion Picture Experts Group (MPEG)'dir (David S. Taubman2002).

Üçüncü bölümde çalışmanın temeline dayanan vektör nicemleme yöntemi hakkında bilgi verilmiştir. İlk olarak 1980'li yıllarda Linde, Buzo ve Gray tarafından çalışılmaya başlanmıştır (Gray 1980). Bu teoreminin temeli bir kaynak içerisinde ard arda gelen ayrı sembollerin vektörlerden oluşan bir paket hale dönüştürmesidir. Böylece kaynağın daha verimli olarak kullanılması amaçlanmıştır. Bu sistemde bir imge belirlenen boyutta bloklara ayrılır ve genelde 8x8 lik bloklar kullanılmaktadır. Bloklara ayrılan imgeye bağlı olarak bir kod kitabı oluşturulur. Bu kod kitabı kod çözücü kısımda da bulunur ve oluşturulurken en iyi kod vektörlerini elde etme etmek için bir eğitimden geçirilir. Gelen bilginin içeriği kod kitabına en yakın vektörlere bağlı olarak indekslenir ve bu indeks bilgisi alıcı tarafa gönderilerek indeks 'in gösterdiği kod kitabı içerisindeki bilgi alınarak görüntü oluşturulur.

Vektör nicemlemenin en önemli özelliklerinden birisi kod çözücü yapısının basit olmasıdır. Bu sayede düşük işlemcili sistemlerde kolaylıkla uygulanabilir bir yapıdadır. Yapılan bu çalışma ile bir imge dizisinin sıkıştırılarak bir kod çözücü sistem tarafından çözümlenmesi gerçekleştirilmiştir. Bu yapının mobil sistemlerde ve internet gibi iletişim ortamlarında imge ve imge dizilerinin daha hızlı ve kaliteli aktarımı yapılabilmesine olanak sağlayabileceği gösterilmiştir.

## BÖLÜM 2. VERİ SIKIŞTIRMA VE VİDEO KODLAMA

Bu bölümde veri ve imge sıkıştırma tekniklerinden bahsedilecektir. Bir verinin sıkıştırılması ve kodlama algoritmaları üç kategoriye ayrılmıştır. Bunlar kayıpsız, öngörülü ve dönüşüm kodlama algoritmalarıdır.

### 2.1 Kayıpsız Sıkıştırma Yöntemleri

#### 2.1.1 Entropi kodlama

Entropi kodlama, kayıplı veri sıkıştırma yöntemlerine göre yüksek sıkıştırma oranlarına ulaşabilen ve bilginin tekrar kayıpsız olarak elde edilebildiği yöntemlerden biridir. Entropi kodlama yöntemlerinden ikisi şöyledir

- Huffman Kodlama
- Aritmetik Kodlamadır.

Ayrık Rasgele  $X$  değişkenlerinden oluşan bir alfabe kümesi ele alalım.

$A = \{a_0, a_1, \dots, a_{N-1}\}$ ,  $N$  adet sembolden oluşan bu kümenin olasılık ağırlık fonksiyonu ise  $p(a) = \Pr\{X = a\}, a \in A$  olsun. Bu kümenin entropisi ;

$$H(X) = -\sum p(a) \log_2 p(a) \quad (2.1)$$

$\{a_0, a_1, \dots, a_{N-1}\}$  kümesinin olasılık kümesi ise  $\{p_0, p_1, \dots, p_{N-1}\}$  dir.

Bu kümenin entropisi

$p_0 = p_1 = \dots = p_{N-1} = 1/n$  ise  $H(a_0, a_1, \dots, a_{N-1}) = \log_2 n$  değerini alarak entropiyi maksimize eder.

$p_0 = 1, p_1 = \dots = p_{N-1} = 0$  olması durumunda ise  $H(a_0, a_1, \dots, a_{N-1}) = 0$  değerini alarak entropiyi minimize eder.

Entropi kodlamaya örnek vermek gerekirse  $A$  ve  $B$  harflerinden oluşan bir kümenin entropi hesabı şöyledir;

İki elemandan oluşan bu kümede elemanların kullanılma olasılıkları şöyle olur.

$A$  sayısının kullanılma olasılığı  $p_A = 0.5$

B sayısının kullanılma olasılığı  $p_B=0.5$

$$H(A,B) = -p_A \log_2 p_A - p_B \log_2 p_B$$

$$H(A,B) = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1 \text{ 'dir.}$$

Buradan çıkan sonuç her bir sembolün temsili bir bit kullanılması gerektiğidir.

Bilginin kodlanması amacıyla iki çeşit kod yapısı kullanılır. Bunlar;

- Sabit Uzunluklu Kod
- Değişken Yapıdaki Kod'tur

Sabit uzunlukta kodlama yönteminde tüm değişkenler sabit uzunlukta bir bit yapısı ile kodlanırlar. Örneğin;

A - 000 , B - 001 , C - 010 , D - 011 , E - 100 , F - 101

Değişken yapıdaki kodlamada ise değişkenler sabit olmayan bir bit dizisi ile kodlanırlar. Örneğin;

A - 0 , B - 00 , C - 110 , D - 111 , E - 1000 , F - 1011

\* Bu işlem sonucu oluşturulan bir kod diğer bir kodun ön ekli olamaz.

### 2.1.2 Huffman kodlama

Bu kodlama yönteminin temeli sık karşılaşılan karakterlerin daha kısa bit uzunluğu ile kodlanarak gönderilmesi temeline dayanmaktadır. Bit Kod tablosunun oluşturulması ise kodlamanın en önemli kısımlarından biridir. Entropi kodlama yöntemi kullanılarak gönderilen bilgi içerisindeki karakterlerin olasılıklarına bağlı olarak her karakter için değişken uzunluklu bir kod kelimesi oluşturulur. Kodlama işleminin tamamlanması ile beraber bu kod tablosu alıcı tarafa ulaştırılmak sureti ile kod çözme işlemi gerçekleştirilir (William H. Pres 2002).

Huffman kodlamayı bir örnek ile açıklayalım. Beş karakterden oluşan ( $N=5$ ) bir X kümemiz olsun;  $X = \{ A, E, I, O, U \}$  bu harflerin kullanılma olasılıkları ise

$$A = 0,12$$

$$E = 0,42$$

$$I = 0,09$$

$$O = 0,30$$

$$U = 0,07 \text{ olsun.}$$

Bu durumda Huffman Kod Oluşturma Tablosu şu şekildedir.



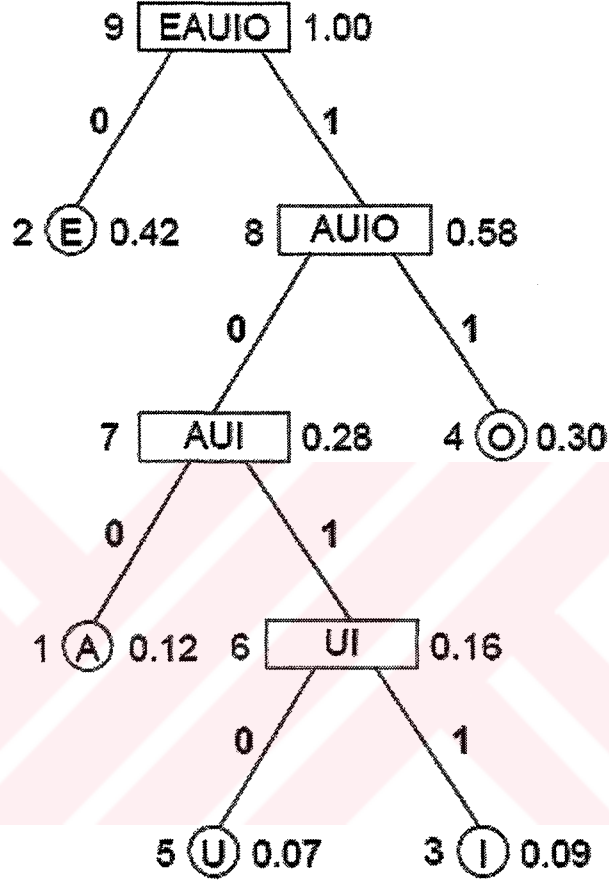
**Tablo 2.1** Huffman Kod Oluşturma Tablosu

Uç	Durum	1	2	3	4	5
1	A:	0.12	0.12■			
2	E:	0.42	0.42	0.42	0.42■	
3	I:	0.09■				
4	O:	0.30	0.30	0.30■		
5	U:	0.07■				
6	UI:		0.16■			
7	AUI:			0.28■		
8	AUIO:				0.58■	
9	EAUIO:					1.00

Tablonun çalışma temeli şu şekildedir. N Karakterli bir küme için tablomuzda N adet durum mevcuttur. Her bir durumda en küçük olasılıklı iki değer yeni bir durum oluşturmak sureti ile toplanırlar. Geriye kalan uç değerleri yeni duruma taşınırlar. Tablodan 1. Duruma bakılırsa burada en küçük olasılıklı iki değer U ve I harflerin olasılıklarıdır. Bu durumda bu iki değer toplanarak UI (0.16) 6. Uç noktayı oluştururlar. Geri kalan diğer olasılıklar 2. duruma aktarılırlar. Aynı şekilde bu durumda da en küçük iki değer toplanır ve yeni değer oluşturulur bu işlem toplam olasılıkların 1 değerine ulaşınca kadar devam eder. Sonuçta her bir harfe bağlı kod numarası bulunmuş olur. Bu tabloyu ağaç grafiğine uyarlırsak Şekil 2-1’de görüldüğü gibi oluşur.

Tablo ‘dan ağaç formuna dönüşüm ise şu şekilde olur.En son durumda başlanarak 1. duruma doğru ilerlenir bu örneğimiz için 5. son durumda olasılık 1.00 ve değişkenimiz EAUIO dur. Bir alt durumda 2 kola ayrılıyor ve bunlar 0.42 olasılık değeri ile E ve 0.58 ile AUIO dur. Sondan başa doğru gidildikçe ağaç yapısına

benzer şekilde yapımız kollara ayrılır ve 1. duruma gelmemizle beraber şeklimiz tamamlanır. Her bir kol üzerine 0 ve 1 değerleri atanarak değişkenlere atanacak kodlar tespit edilir.



Şekil 2.1 Ağaç Formunda Huffman Kod Gösterimi

Şekil 2-1 den yola çıkarak bu durumda değişkenlerimizin aldığı Huffman kodları ise şöyledir.

E = 0      AUIO = 1  
 A = 100    AUI = 10  
 U = 1010   UI = 101  
 I = 1011  
 O = 11

### 2.1.3 Aritmetik kodlama

Aritmetik kodlamanın temeli bilgi dizisinin 0 ile 1 arasındaki gerçel sayılara dönüştürülerek kodlanmasıdır. Aritmetik kodlama Huffman 'a göre daha düşük sıkıştırma oranına sahip ve aynı zamanda aynı kod kelimelerine karşın tek bir sembol ile kodlanabilmesidir. Tabi bu kodlama yönteminin dezavantajları mevcuttur. Bunlar; alıcı tarafta kod çözme işleminin başlayabilmesi için kod bilgisinin tamamının alıcı tarafa ulaşması gerekmekte ve iletim esnasında bilgide oluşacak bir hata bilginin hatalı olarak gelmesine sebep olmaktadır. İkinci bir dezavantajı ise ondalık hassasiyetinin sınırlı olmasından dolayı sembollerin kodlanmasında oluşan sınırlamadır. Bundan dolayı aynı anda belirli sayıda sembol kodlanarak alıcı tarafa iletilmektedir (Eric Bodden 2004).

Bunu bir örnek ile açıklarsak;

{ BILL GATES } dizisinin aritmetik kodlanarak karşı tarafa gönderilmesi istensin

**Tablo 2.2** Karakter Olasılık Tablosu

Karakter	Kullanılma Olasılığı
Boşluk	1/10
A	1/10
B	1/10
E	1/10
G	1/10
I	1/10
L	2/10
S	1/10
T	1/10

Gönderilecek bilgideki karakterlerin bilgi içerisinde bulunma olasılıkları tablo'da görüldüğü gibidir. Bu tablo oluşturulurken harflerin sırası kullanılma olasılık miktarı önemli değildir. Karakterlerin olasılıklarının bilinmesine bağlı olarak 0 ile 1 arasında olasılık aralıkları tespit edilir. Bu aralıklar tespit edilirken yine harflerin sırası fark etmez. Bu tabloya bağlı olarak olasılık aralıklarını belirleyelim.

Aritmetik kodlamada en önemli parça ilk kodlanan sembole dayanmaktadır. {BILL GATES} mesajının kodlanması durumunda ilk olarak B harfi kodlanmakta ve buna bağlı olarak ta sonuçta ortaya çıkan kodlanmış mesaj 0.20 ile 0.30 arasında bir değere sahip olur.

**Tablo 2.3** Karakter Aralık Tablosu

Karakter	Kullanılma Olasılığı	Aralıklar
Boşluk	1/10	0.00 – 0.10
A	1/10	0.10 – 0.20
B	1/10	0.20 – 0.30
E	1/10	0.30 – 0.40
G	1/10	0.40 – 0.50
I	1/10	0.50 – 0.60
L	2/10	0.60 – 0.80
S	1/10	0.80 – 0.90
T	1/10	0.90 – 1.00

İlk karakterin kod çözümlemesinden sonra aralığımızın 0.20 ile 0.30 arasında olduğu bilinmektedir. Bilindiği üzere diğer karakterlerinde kendilerine ait aralıkları mevcuttur. Bu durumda bu değerler 0.20 ile 0.30 aralığına indirgeneceklerdir. Buda mevcut aralıklarının %50 ile %60 arasında azalması demektir.

Bir bilgi dizisinin aritmetik kod ile kodlanması için kullanılan algoritma şöyledir;

Set Düşük = 0.0

Set Yüksek = 1.0

Son Karaktere ulaşıncaya kadar dön {

Karakter Oku

Kod\_aralığı = Yüksek - Düşük

yüksek = alçak + Aralık\* Yüksek\_Aralığı(Sembol)

alçak = alçak + Aralık\*Düşük\_Aralığı(Sembol)

} Döngü Sonu



Bu algoritmayı karakter dizimize uygularsak bu durumda aşağıdaki değerler elde edilir.

**Tablo 2.4** Kod Tablosu

Karakter	Düşük Değer	Yüksek Değer
	0.0	1.0
B	0.2	0.3
I	0.25	0.26
L	0.256	0.258
L	0.2572	0.2576
Boşluk	0.25720	0.25724
G	0.257216	0.257220
A	0.2572164	0.2572168
T	0.25721676	0.2572168
E	0.257216772	0.257216776
S	0.2572167752	0.2572167756

Sonuç olarak {BILL GATES} karakter dizisinin kodu en son karakterin düşük değeri yani 0,2572167752 olarak oluşur. Bu kodlanmış dizinin çözümü ise çok basittir. Alıcıya ulaşan sayısal kodlanmış değer de ilk değer B karakteri olduğu bilinmektedir. B'nin düşük değerinin 0,2 ve yüksek değerinin 0,3 olduğu bilinmektedir. Buradan 0,2572167752 değerinden 0,2 yi çıkartıp B'nin aralık değeri olan 0,1 e böleriz kalan değer yine bir aralığa denk gelmektedir. Kalan değer yine o karakterin en düşük değerinden çıkartılarak aralığına bölünür bu işlem adımları kalan 0 değerine ulaşınca kadar devam eder. Bu karakter dizisi için çözüm tablosu aşağıdaki gibidir.

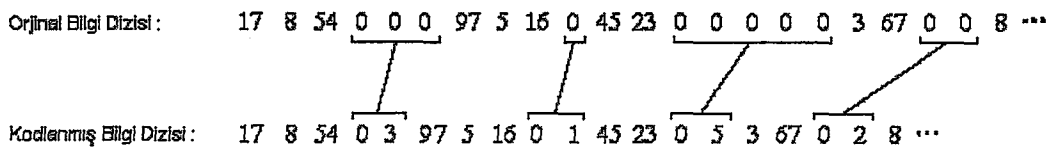
**Tablo 2.5** Tekrar Elde Etme Tablosu

Çözümlemiş Değer	Karakter	Düşük Değer	Yüksek Değer	Aralık
0.2572167752	B	0.2	0.3	0.1
0.572167752	I	0.5	0.6	0.1
0.72167752	L	0.6	0.8	0.2
0.6083876	L	0.6	0.8	0.2
0.041938	Boşluk	0.0	0.1	0.1
0.41938	G	0.4	0.5	0.1
0.1938	A	0.2	0.3	0.1
0.938	T	0.9	1.0	0.1
0.38	E	0.3	0.4	0.1
0.8	S	0.8	0.9	0.1
0.0				

Bu tablonun çözümünü kısaca açıklarsak, ilk gelen karakterin B olduğu biliniyordur. Çünkü ilk gelen değer bulunduğ aralık sadece B aralığına uymaktadır. 0,2572167752 değerinden 0,2 değerini çıkartıp aralık değeri olan 0,1'e bölersek sonuçta 0,572167752 değeri elde edilir bu ise I karakter aralığına uygundur. Aynı şekilde bu değerden 0,5 değerini çıkartıp I 'nın aralık değeri olan 0,1'e böleriz sonuçta kalan 0 oluncaya kadar bu işlemler devam eder böylece bir kodlanmış rakam dizisinden kodlanan bilgi elde edilir.

#### 2.1.4 Run – Length kodlama

Birçok bilgi dosyası aynı satır içerisinde tekrarlanmış karakterleri içerir. Örneğin metin içerisinde satırlar arasında bırakılan boşluklar, sayısallaştırılmış bir bilgide bir değişim olmadığı zamanlarda aynı bilginin tekrarladığı durumlarda veya bir gece görüntü dizisinde bulunan karalık bölgeler tekrar edilen bilgi çeşitlerindedir (David S. Taubman 2002).



**Şekil 2.2** Run Length Kodlama Örneği

Şekil 2.2 de bir bilgi dizisi ve Run-Length uygulanmış hali gözükmektedir. Bir bilgi dizisi içerisinde tekrar eden bilgiler 2 karakter ile ifade edilir. Birincisi tekrar eden bilgi sayısı ikincisi ise tekrar eden değişkendir. Run – Length Kodlama BMP, JPEG ve TIFF gibi resim formatlarında kullanılmaktadır. Bu kodlama tekniği ile değişimin az olduğu durumlarda iyi sıkıştırma oranları elde edilirken hareketli ve değişken resimlerde çok düşük bir sıkıştırma oranı elde edilmektedir.

## 2.2 Kayıplı Sıkıştırma Yöntemleri

### 2.2.1 Skaler nicemleme

Genel olarak bir konuşma sinyali sürekli zaman ve genlik fonksiyonudur. Sayısal olarak işlenebilmesi için sayısal bilgi haline dönüştürülmesi gerekmektedir. Bunun yapılabilmesi için ilk adım sinyalin örneklenmesidir. İkinci adım ise ayrık genlik sinyallerinin elde etmek için nicemlemedir. Bu nicemleme'nin amacı sürekli bir genlik işaretinden ayrık sembol dizisi oluşturmaktır.

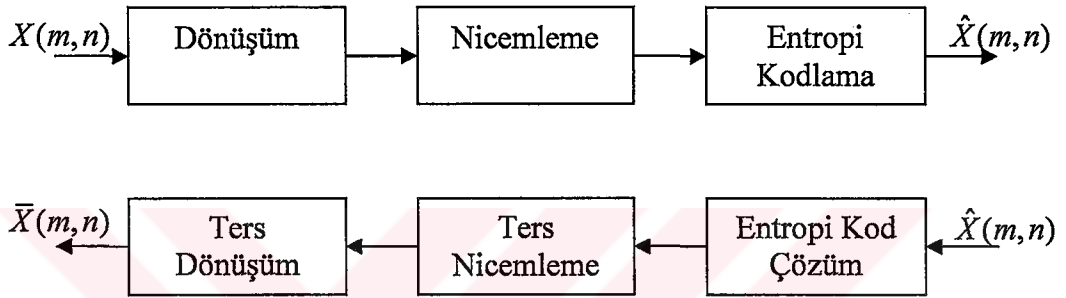
Skaler nicemleme'nin tanımını belirlersek; Skaler kuantalama  $Q: \mathbb{R} \rightarrow C$  kuantalama haritasından ve kod kitabı diğer adı ile N elemanlı  $C = \{c_0, \dots, c_{N-1}\} \subset \mathbb{R}$  tekrar üretim değerlerinden oluşan bir kodlama yöntemidir.

Skaler kodlayıcı  $\alpha: \mathbb{R} \rightarrow I, x \mapsto i$  Eğer  $Q(x) = c_i$  ise  $I = \{0, \dots, N-1\}$  indeks dizisi gerçek çizgisi ile eşlenir. Skaler kod çözücüsü  $\beta: I \rightarrow C$  ise indeks yolu ile üretilecek değeri gösterir. Buradan  $Q = \beta \circ \alpha$  eşitliği bulunur.

Skaler çözücü yeniden elde etme değerleri üzerine bir düzen belirler  $\beta(0), \dots, \beta(N-1)$ . Genellikle Skaler kodlayıcı en yakın komşu kuralı ile (EYKK) öklit normunda  $\alpha(x) = \arg \min_{i \in I} \|x - \beta(i)\|_2^2$  ile gerçekleşir. EYKK kuralı Skaler nicemleyicisi gerçek Hattı ( $\mathbb{R}$ ), N adet bölümlenmiş hücrelere böler  $R_i = \{x \in \mathbb{R} : \alpha(x) = i\}$ . Bundan dolayı bölümlenmiş hücreler  $y_i$ ,  $-1 \leq i \leq N$  sınırlarında  $[y_{i-1}, y_i)$  aralığı ile ifade edilirler.

## 2.2.2 Dönüşüm kodlaması

Dönüşüm kodlaması imge ve video kodlama sistemlerinin önemli parçalarından bir tanesidir. Tipik bir dönüşüm kodlaması blok diyagramı Şekil 2.3-a'da görüldüğü gibidir. Genel olarak üç kısımdan oluşur. İlk kısım kaynağın dönüştürüldüğü kısımdır. Daha sonra dönüşüm alanında kuantalama kullanılır. Son kısım ise kuantalanmış dönüşüm katsayılarının entropi kodlamasının yapıldığı kısımdır (Simon Lucey 1998).



Şekil 2.3 Dönüşüm Kodlama Blokları ve Dönüşüm Kod Çözücü Blokları

Dönüşüm kod çözümünün blok diyagramı ise Şekil 2.3-b de gösterildiği gibidir. Dönüşüm katmanının amacı sonraki işlemlere uygun sinyal yapısına ulaşmaktır. Bu tarz dönüşüme örnek olarak Karhunen-Loeve Dönüşümü (KLD) gösterilebilir. Bu dönüşümün özellikleri şöyledir;

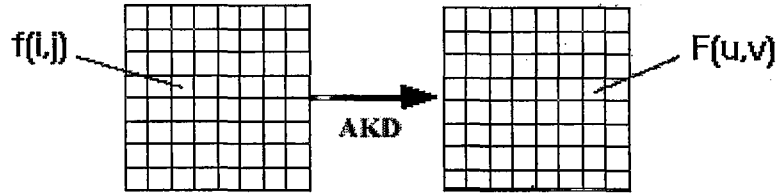
- Dikey bir yapıya sahiptir.
- Dönüşüm domenin de sinyal'in KLD sonucu elde edilen katsayıları tamamen ilişkisizdir.
- En düşük sayıda dönüşüm katsayılarında değişim içermekte ve en az minimum karesel ortalama hata değeri ile boyut azaltışını mümkün kılmaktadır.

Fakat KLD'nin de olumsuz özellikleri bulunmaktadır. İlk olarak ters yönde bir dönüşüm gerçekleştirilmesi için kod çözücü dönüşüm domeninin temelini bilmesi gerekmektedir. Bu da sıkıştırma için gerekli bit miktarını arttırmaktadır. İkinci olarak gerçek uygulamalarda KLD 'yi gerçekleyecek hızlı bir algoritma bulunmamaktadır bundan dolayı imge ve imge dizilerinin kodlanması için daha hızlı ve daha basit ayrık kosinüs dönüşüm algoritması vardır.



### 2.2.3 Ayrık kosinüs dönüşümü (AKD)

Ayrık kosinüs dönüşümü iki boyutlu görüntü işleme uygulamalarında önemli bir dönüşüm tekniğidir. Hızlı ve yüksek sıkıştırma oranına sahip bir uygulamadır. AKD uluslararası iki imge/video standardında kullanılmaktadır. Bunlar Joint Photographic Experts Group (JPEG), ve Motion Picture Experts Group (MPEG)'dir. AKD yapı itibari ile ayrık fourier dönüşümüne (AFD) benzer olmasına karşın bir imge veya sinyali uzamsal domenden frekans domenine dönüştürür (Syed Ali Khayam 2003).



Şekil 2.4 AKD Dönüşümü

AKD, M bloklu  $G_m$ ,  $0 \leq m < M$  görüntüleri  $N \times N$  piksel boyutunda parçalara ayırır. Genel olarak blok boyutu  $8 \times 8$ 'dir ve dönüşüm gerçekleştirir. Bir boyutlu AKD lineer bir dönüşümdür ve şu matrisle gösterilir;

$$(C_N)_{nm} = \left(\frac{2}{N}\right)^{1/2} \left[ k_m \cos\left(\frac{m(n+\frac{1}{2})\pi}{N}\right) \right] m, n = 0, \dots, N-1 \quad (2.2)$$

Burada

$$k_j = \begin{cases} 1, & j \neq 0 \\ \frac{1}{\sqrt{2}}, & 1 \leq j \leq N-1 \end{cases} \quad (2.3)$$

Bir boyutlu bir kodlamada ters dönüşümün gerçekleştirilmesi çok basittir. Bu durumda  $F^{-1}$  şeklinde olur ve bu durumda kabulümüz;

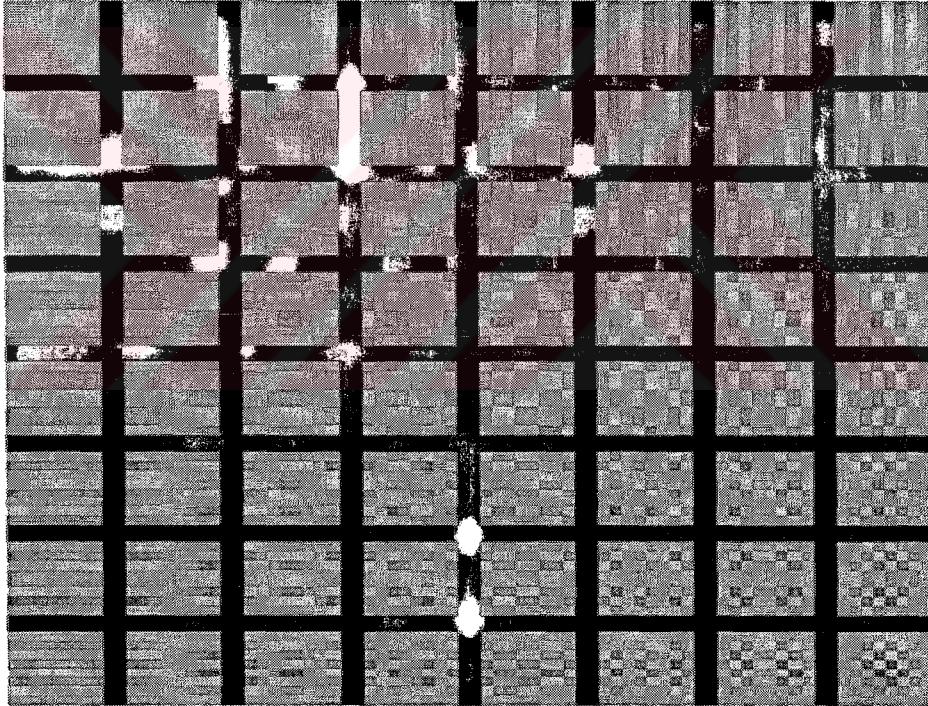
$$k_j = \begin{cases} 1, & 1 \leq j \leq N-1 \\ \frac{1}{\sqrt{2}}, & j \neq 0 \end{cases} \text{ Şeklinde olur.} \quad (2.4)$$

En sık kullanılan iki boyutlu AKD uygulamalarıdır.  $M \times N$  den oluşan bir matrisinin AKD dönüşüm formülü ise şöyledir.

$$\beta_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{mn} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{matrix} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{matrix} \quad (2.5)$$

$$\alpha_p = \begin{cases} 1/\sqrt{M} & p=0 \\ \sqrt{2/M} & 1 \leq p \leq M-1 \end{cases} \quad \alpha_q = \begin{cases} 1/\sqrt{N} & q=0 \\ \sqrt{2/N} & 1 \leq q \leq N-1 \end{cases}$$

$\beta_{pq}$  Değerleri AKD'nin katsayıları olarak isimlendirilir.  $8 \times 8$  bir matris için 64 temel fonksiyonlar şekilde gösterildiği gibidir.



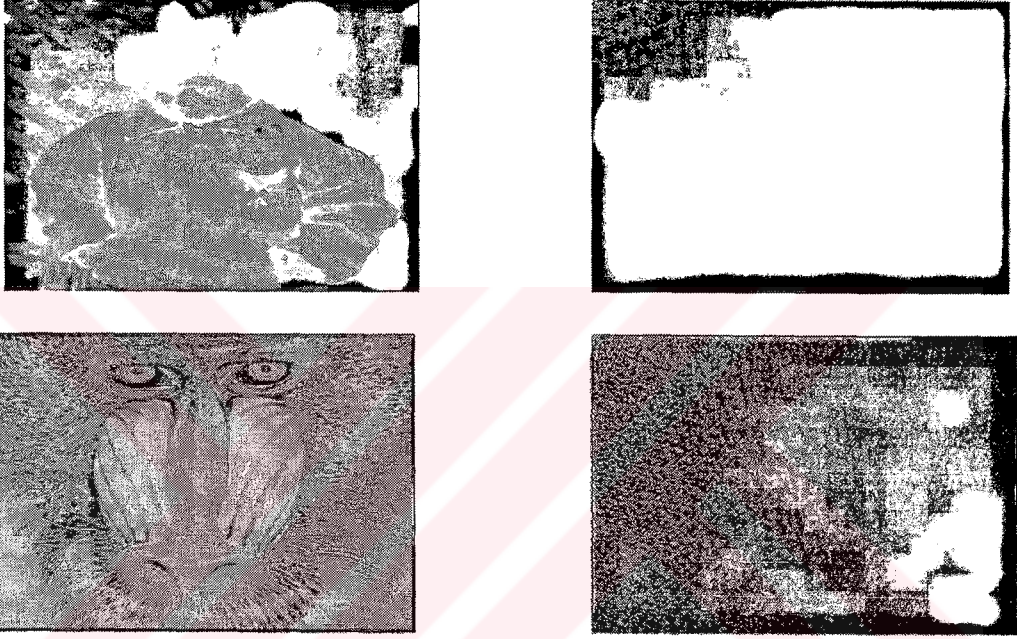
Şekil 2.5 AKD katsayılarının gösterimi

Yatay frekanslar soldan sağa doğru ve dikey frekanslar ise üstten altta doğru artarlar. Bundan dolayı yüksek frekanslar daha büyük satır ve sütun değerleri ile temsil edilirler. Şekil 2.5 'de gri tonlar sıfır noktasını, beyaz pozitif genliği siyah ise negatif genliği göstermektedir.

AKD 'nin ters dönüşüm formülü ise şöyledir.

$$A_{mn} = \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} \alpha_p \alpha_q \beta_{pq} \cos \frac{\pi(2m+1)p}{2M} \cos \frac{\pi(2n+1)q}{2N}, \quad \begin{array}{l} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{array}$$

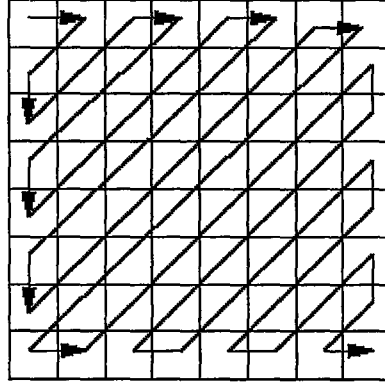
$$\alpha_p = \begin{cases} 1/\sqrt{M} & p=0 \\ \sqrt{2/M} & 1 \leq p \leq M-1 \end{cases} \quad \alpha_q = \begin{cases} 1/\sqrt{N} & q=0 \\ \sqrt{2/N} & 1 \leq q \leq N-1 \end{cases} \quad (2.6)$$



Şekil 2.6 İmge ve AKD dönüşüm örnekleri

Şekilde iki resmin AKD 'ye dönüştürülmüş şekilleri görülmektedir. Şekil-a da geniş bir alan üzerinde şiddet ve yoğunluk çok az değişmektedir bundan dolayı düşük frekanslara görüntülerine sahiptir. İkinci resmimizde ise yüksek ve alçak frekans geçişleri görülmekte bundan dolayı ise dönüşüm katsayıları alçak ve yüksek frekanslar arasında ayrılmıştır.

Dönüşüm katsayılarının bulunmasında sonra sistem şekilde gösterilen Zig-Zag taraması kullanılarak Run-Length kodlanır.

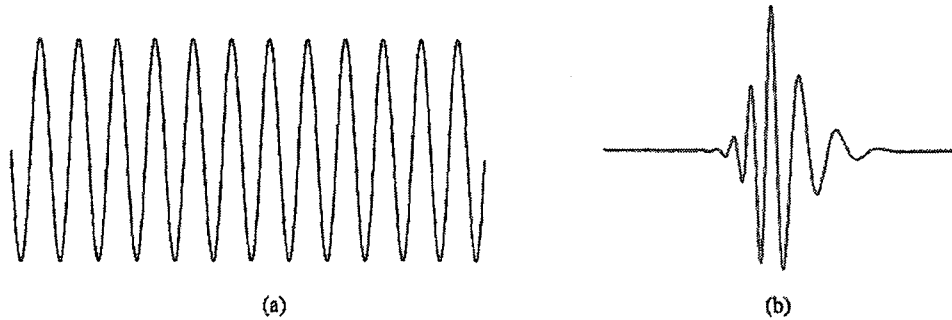


**Şekil 2.7** AKD katsayılarının Zig-Zag Taraması

Şekilden de görüldüğü gibi AKD katsayıları belirtilen doğrultuda kodlanarak en son hallerine dönüştürülürler.

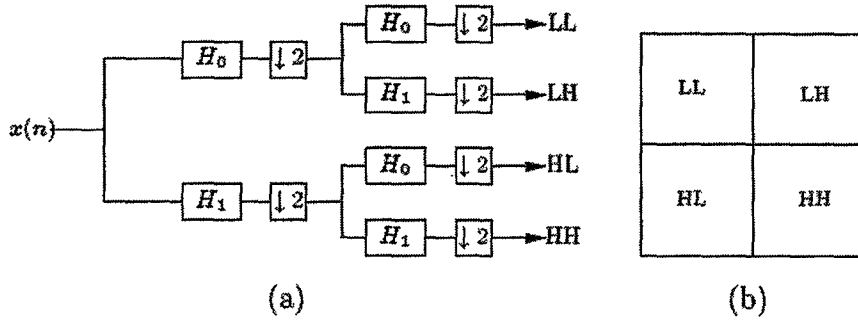
#### 2.2.4 Dalgacık dönüşümü

Bir sinyalin dönüşümü, sinyalin farklı bir şekilde gösterilmesidir. Bilginin içeriği değişmez. Wavelet(Dalgacık) Dönüşümü bir sinyalin Zaman-Frekans dönüşümünün gösterimidir. Dalgacık dönüşümü Kısa zamanlı fourier dönüşümü (KZFD) üzerine geliştirilmiş bir yöntemdir. KZFD tüm frekanslar için sabit çözünürlük değeri verirken. Dalgacık dönüşümü çoklu çözünürlük tekniğini kullanarak farklı frekansları farklı çözünürlüklerle incelemektedir (David S. Taubman 2002).



**Şekil 2.8** (a) Bir sinyalin gösterimi (b) Dalgacık gösterimi

Fourier dönüşümü ve KZFD sinyalleri analiz etmek için sinyal dalgalarını kullanırken Dalgacık dönüşümü ise onların enerjilerini kullanır. Dalgacık bir imgeyi  $H_0$  geçiren alçak geçiren filtre ve  $H_1$  yüksek geçiren filtreleri kullanarak yan bantlarına ayırır. İki boyutlu sinyal için filtre grubunun gösterimi Şekil 2.7 'de gösterildiği gibidir.



Şekil 2.9 Bir İşaretin Filtre Grubuna Uygulanması

Dalgacıklar, bir ana *Dalgacık* adı verilen temel dalgacığın ölçekleme ve ötelemelerden oluşurlar.

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}} \psi\left(\frac{t-b}{a}\right) \quad (2.7)$$

Buradaki  $a$  değeri ölçeklendirme katsayısı iken  $b$  ise kaydırma katsayısıdır. Bir boyutlu bir dalgacık dönüşüm formülü şöyledir;

$$W_f(a,b) = \int_{-\infty}^{+\infty} x(t) \psi_{a,b}(t) dt \quad (2.8)$$

Ters Dönüşüm Formülü ise;

$$x(t) = \int_0^{\infty} \int_{-\infty}^{\infty} W_f(a,b) \psi_{a,b}(t) db \frac{da}{a^2} \quad (2.9)$$

Burada,

$$C = \int_{-\infty}^{\infty} \frac{|\psi(w)|}{w} dw < \infty \text{ dir.} \quad (2.10)$$

Dalgacık dönüşümünün temelde Ayırık ve Sürekli dönüşüm olmak üzere iki türü bulunmaktadır.

Sürekli dalgacık dönüşümü (SDD) denklem 2,5’de gösterilen ifade ile gerçekleştirilir. Bu ifadede  $x(t)$  analiz edilen sinyal,  $\psi(t)$  ise ana dalgacık veya taban fonksiyondur. Dönüşümlerde kullanılan tüm dalgacık fonksiyonları ana dalgacık fonksiyonunun kaydırılması ve ölçeklendirilmesi ile üretilir.

$$X_{WT}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t) \psi^* \left( \frac{t-\tau}{s} \right) dt \quad (2.11)$$

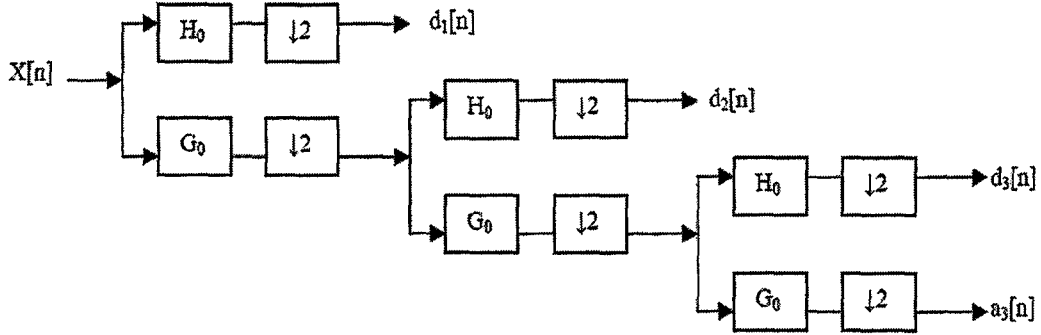
Dönüşüm parametresi  $\tau$  sinyal doğrultusunda kaydırılmış fonksiyonu ile ilişkilidir. Buda dönüşüm içerisindeki zaman bilgisine karşılık gelmektedir. Ölçülendirme parametresi  $s$  ise  $|1/frekans|$  ile gösterilir ve frekans bilgisine karşılık gelmektedir. Bu çoklu sinyali genişletebilir ya da sıkıştırabilir de. Büyük ölçekler (Alçak Frekanslar) sinyali genişletir ve sinyal içerisinde detaylı bilgi sağlarken küçük ölçekler sinyali sıkıştırır ve sinyal hakkında genel bilgi üretir.

Ayrık dalgacık dönüşümü (ADD) sürekli dönüşümün örneklenmiş halidir. Çözünürlüğe bağlı olarak gerekli olan zaman ve kaynak ihtiyacını azaltmaktadır. ADD, daha hızlı dalgacık dönüşüm hesaplaması üretebilmek için yan bant kodlamasına dayanmaktadır. Uygulaması kolay ve hesaplama zamanı nispeten kısadır.

Sürekli dalgacık dönüşümü’nde sinyaller dönüşüm ve boyutlandırma ile birbirine bağlı olan temel fonksiyonları kullanarak analiz yaparken. ADD’de ise zaman ölçekli sayısal bir sinyalin gösterimi sayısal filtreleme yolu ile gerçekleştirilir. Analiz edilecek olan sinyal farklı ölçülerde farklı kesim frekansları kullanan alçak ve yüksek geçiren filtrelerden geçirilir.

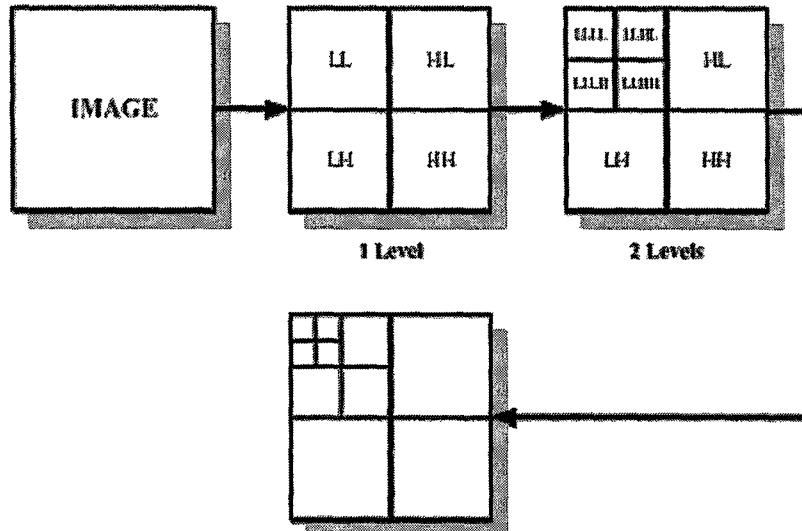
ADD şekil 2.10 ‘da gösterildiği gibi sinyalin ayrık zaman domeninde alçak ve yüksek geçiren filtreleme yolu ile hesaplanır. Buna Mallat Algoritması veya Mallat Ağaç Ayırıştırması denir. Bu algoritma sürekli zaman çok çözünürlüğünü ayrık zaman filtrelerine bağlanır. Şekil 2.8 ‘de işaret  $x[n]$  dizisiyle, alçak geçiren filtre  $G_0$

ile ve yüksek geçiren filtre ise  $H_0$  ile gösterir. Her bir katmanda yüksek geçiren filtre detay bilgisi  $d[n]$  'i üretirken alçak geçiren filtrede kaba yaklaşımlık  $a[n]$  'i üretir.



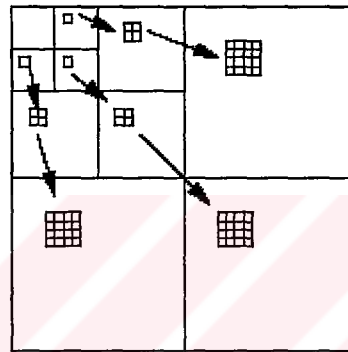
Şekil 2.10 Üç Katlı Dalgacık Bölünme Ağacı

Her bir arışım katmanında giriş frekansı yarım frekans bandı ile bölünür. Her bir kısımda frekans ikiye bölünür. Nyquist Kuralına göre orijinal sinyal  $\omega$  en yüksek frekansına sahip ise örnekleme için  $2\omega$  radyanlık frekans gerekmektedir. Böylece  $\omega/2$  radyanlık en yüksek frekansa sahip olan işaret  $\omega$  radyan frekansında örneklenebilir. Böylece örneklerin bölünmesi bilgi kaybına sebep olunmaz.



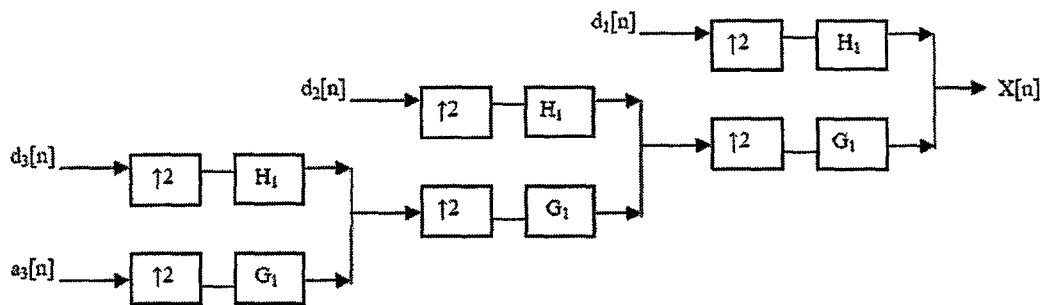
Şekil 2.11 İki Katmanlı Dalgacık Kodlama

Alt bant Kodlama tekniđi genel olarak iki kısma ayrılabilir. Bunlar gömülü yan bant ve çapraz bant teknikleridir. Gömülü yan bant tekniđinde her bir yan bant ayrı ayrı kodlanır. Çapraz bant tekniđinde ise farklı bantlar arasındaki ilişkiyi kullanan bir tekniktir. En popüler çapraz bant kodlama tekniđi sıfır ağaç (zero tree) tekniđi olarak adlandırılır. Sıfır ağaç tekniđi, aynı uzamsal konumlarda daha düşük frekanslı katsayıları sıfır ise aynı uzamsal konum için, yüksek geçiren filtre katsayılarınının 0 olması olasıdır gözlemine dayanmaktadır. Bir Sıfır Ağaç Yapısı Şekil 2.10'da görüldüğü gibidir.



Şekil 2.12 Çapraz Bant Kodlama için Sıfır Ağaç yapısı

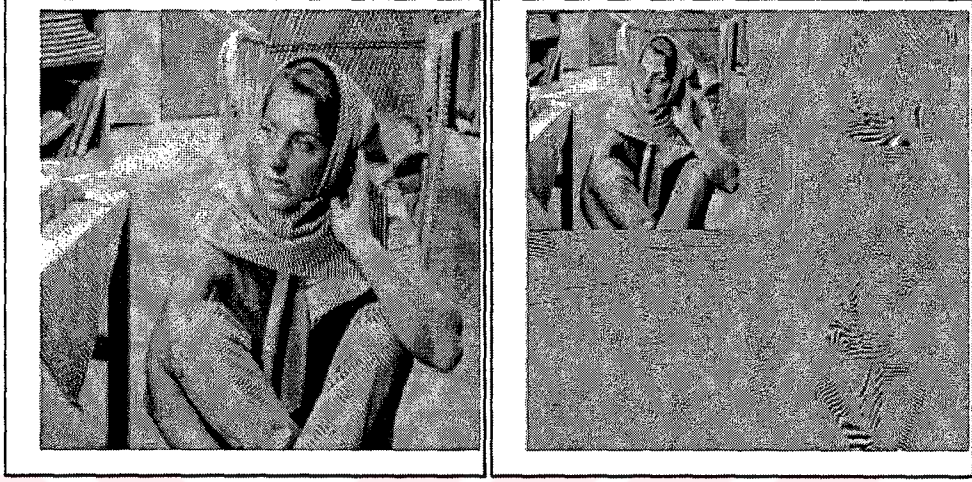
ADD yapılmış bir görüntüyü tekrar elde etmek için tüm katsayılar sondan başa doğru birbirlerine eklenerek bulunur. Bu işlem en baş katmana gelinceye kadar devam eder.



Şekil 2.13 Üç Katlı Dalgacık'tan Tekrar Elde Etme Ağacı



Temel olarak tekrar elde etme ayrıştırmanın tam tersi şeklindedir. Kaba ve detay katsayıları alçak ve yüksek geçiren filtrelerden geçirilerek birbirlerine eklenir. Bu işlemler orijinal görüntü elde edilinceye kadar devam eder.



**Şekil 2.14** 1.Aşama Ayrık Dalgacık Dönüşümü Görüntü Sonuçları

Şekil 2.11’de görülen resim orijinal resim olup ikinci resimde ise 1. aşama ADD ‘de uygulanmış sonuçları görülmektedir. 1.Aşamada resim 2 parça halinde gelirken 2. Aşama dönüşümde LL, LH, HL, HH olmak üzere 4 parça halinde gelir.

### **2.2.5 MPEG sıkıştırma**

MPEG , “Moving Picture Expert Group” adı verilen bir gruba dayanmaktadır. Bu grup ISO/IEC olup 1988 yılında ses ve görüntü formatları üzerine çalışmaya başlamıştır. Her bir sıkıştırma standardı, özel uygulamalar ve bit oranları ile tasarlanmıştır. MPEG sıkıştırma ölçülü şekilde bir bit oran sıkıştırması gerçekleştirmektedir. MPEG video sıkıştırma algoritmaları genel olarak iki temel tekniğe dayanmaktadır. Bunlar tek düzelilik bilginin azaltılması amaçlı blok temelli hareket tahmini tekniği ve uzamsal tek düzeliliklerin azaltılması için ayrık kosinüs dönüşümü (AKD) tekniğidir. Günümüzde kullanılmakta veya tasarım aşamasında olan beş çeşit MPEG standardı bulunmaktadır. Bunlar;

#### MPEG-1

Standart hareketli görüntü ve ses için sıkıştırma formatıdır. 1.5 Mbp'e ile kayıt imkanı sunar. CDROM video uygulamaları ve internet üzerindeki MPEG uygulamaları yaygın olarak kullanılan yerlerdir. 3. Seviyedeki MPEG-1 standardı ise şuan sık olarak kullanılan ses formatı ise MP3 'dür. Video CD'leri için Standard sıkıştırma tekniğidir ( Shanawaz A. Basith 1996).

#### MPEG-2

Sayısal televizyon ve DVD 'de bir standart olan bu teknik. 1.5 ile 15 MBps ile kayıt gerçekleştirmektedir. MPEG-1 temelli olmasına karşın sayısal yayınlarda kullanılmak üzere geliştirilmiştir. MPEG-1 'e göre en önemli özelliği çok daha fazla sıkıştırma oranına sahip olmasıdır ( Iain E. G. Richardson 2003).

#### MPEG-4

Multimedya ve Web için sıkıştırma standardıdır. MPEG-4, Sanal gerçeklik modelleme diline benzer şekilde nesne tabanlı bir sıkıştırma tekniğine sahiptir. Bir sahnedeki bireysel objeler ayrı ayrı takip edilirken, MPEG-4 'ü oluşturmak için birlikte sıkıştırılırlar. Bu ise iyi bir sıkıştırma sağlar. Bu işlem aynı zamanda film üreticilerinin veya video editörlerinin görüntüler üzerindeki objeleri kontrol etmelerini sağlar ( Iain E. G. Richardson 2003).

#### MPEG-7

Diğer adıyla "Çoklu ortam içerik tanım ara yüzü", Bu standart şuan gelişim aşamasındadır. Bu standart tamamlandıktan sonra özelleştirme, filtreleme ve işletme konularında bilgi içerecek olan birçoklu ortam sistemi üretecektir (Moving Picture Experts Group 2005).

#### MPEG-21

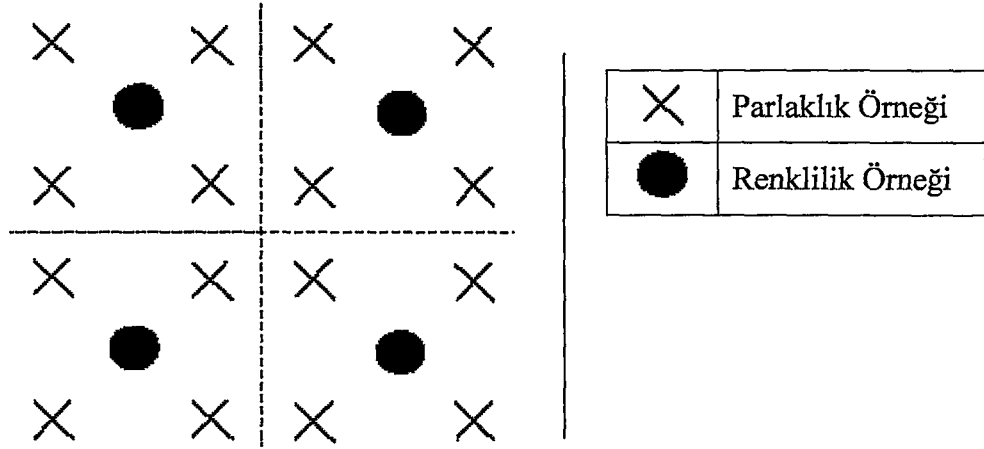
Bu standart üzerine yeni çalışılmaya başlanmış olup, Çoklu ortam Sistemi olarak adlandırılmaktadır. Bu sistem ile altyapının kurulması için gerekli olan elemanları ve birbirleri ile olan ilişkilerini tanımlamayı amaçlamaktadır (Moving Picture Experts Group 2005).

### 2.2.5.1 MPEG-1 video formatı

MPEG gurubu tarafından geliştirilen ilk sıkıştırma formatıdır. MPEG-1 sıkıştırmanın temeli, bir video çerçevesindeki uzamsal artıklığın ve video çerçeveleri arasındaki zamansal artıklığın çıkartılmasına dayanmaktadır. JPEG sıkıştırma formatında uzamsal artıklıkların çıkartılması kullanılmaktadır. AKD temelli sıkıştırma ise bu uzamsal artıklığı çıkarmak için kullanılır. Hareket dengeleme yöntemi ise video çerçeveleri arasındaki zamansal artıklığı yok etmede kullanılan diğer bir yöntemdir. Bir video dizisi belirli bir süre içerisinde ani değişme içermez. Buradan yola çıkarak hareket dengeleme yönteminin amacı bir video çerçevesini ona yakın çerçeveleri kullanarak kodlamaktır (Shanawaz A. Basith 1996).

Bir video dizisi birbirini izleyen video çerçevelerinden oluşur. Bu çerçevelerin her biri aslında bir imgedir. Bir video oynatıcısının yaptığı da bu imge leri sırayla göstermektir. Bu oran genellikle saniyede 30 çerçevedir. Çerçeveler 24 bit RGB renk formatında( Kırmızı Yeşil Mavi Her bir renk için 8 bit) sayısallaştırılmıştır. MPEG-1 formatı 1.5Mbps bit oranına sahip ve 352x288 İmge Boyutunda 24-30 Çerçeve/Saniye olacak şekilde kullanılmaktadır.

MPEG-1 algoritması imgeler üzerinde YUV (Y Cr Cb) renk uzayında işlem yapmaktadır. Bundan dolayı RGB formatında gelen bir imge dizisi önce YUV'a dönüştürülür. YUV formatında imgeler aynı şekilde 24 bit ile gösterilir. (8 bit Y parlaklık bilgisi, U ve Y için 8'er bit Renk Bilgisi). YUV formatının kullanılmasının sebebi RGB'ye göre daha verimli olarak kullanılabilir olmasından dolayıdır. Aynı zamanda istenilen 1.5Mbps 'lık bit oranına ulaşabilmek için video işareti MPEG-1 standart giriş formatına (SIF – Standart Input Format) dönüştürülür. MPEG-1 formatında 4:2:0 renk alt örnekleme formatı kullanılır.

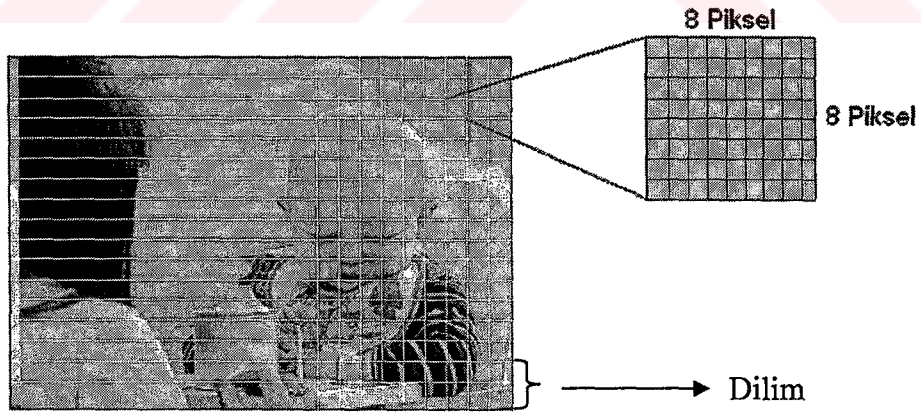


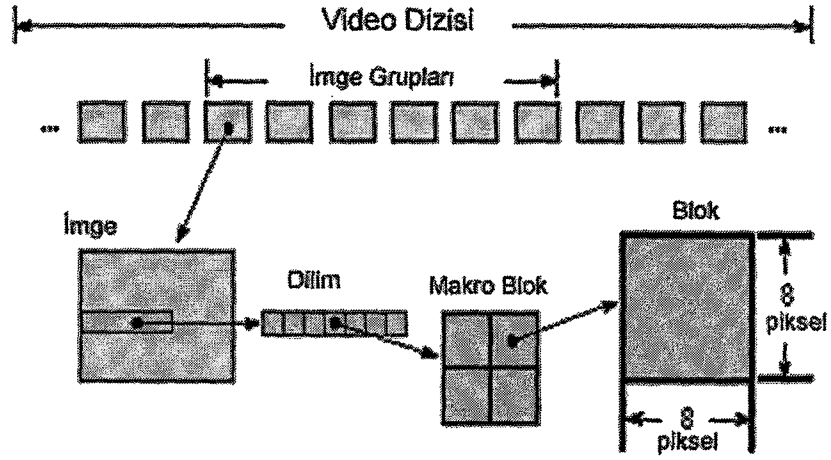
Şekil 2.15 MPEG-1 için 4:2:0 renk alt örnekleme Formatı

4:2:0 örnekleme formatının kullanılmasının sebebi insan gözü YUV formatındaki bir görüntü de en çok Y'yi algılar o yüzden Y tam çözünürlükte kodlanırken Cr ve Cb'ye olan hassasiyet azdır. Bundan dolayı daha düşük oranda kodlaya biliriz.

#### 2.2.5.1.1 Çerçeve kodlama

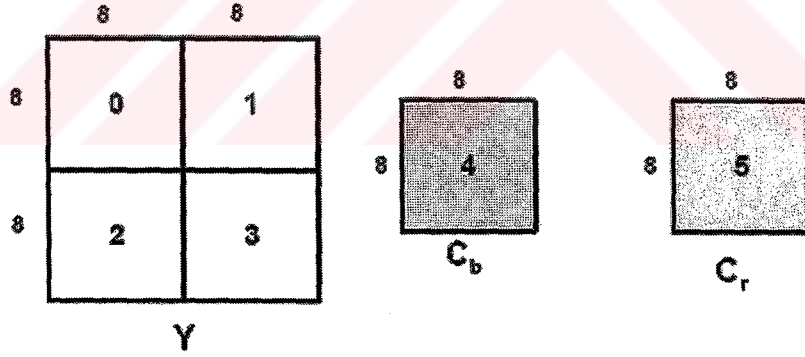
Bir MPEG görüntü dizisinde çerçeveler Şekil 2.12 'de olduğu gibi 8x8 lik bloklardan oluşur.





**Şekil 2.16** Bir Görüntü Dizisinin Piksel Gösterimi ve Makro Blok Temelli Gösterimi

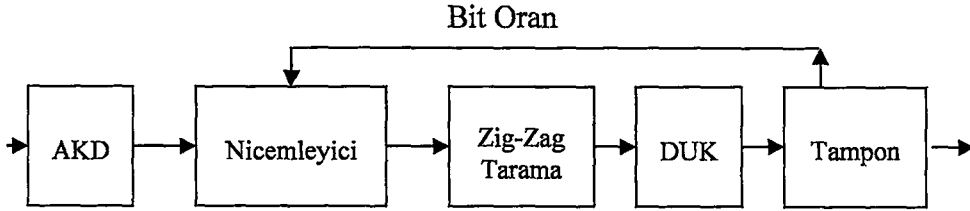
Resimler dilimlerden oluşur ve birinci kodlama ünitesidir. Her bir çerçeve 16x16'lık Makro Bloklardan meydana gelir. Her bir makro blok dört adet parlaklık bloğu ve iki renk bloğundan oluşur.



**Şekil 2.17** 4:2:0 Örnekleme Blok Formatı

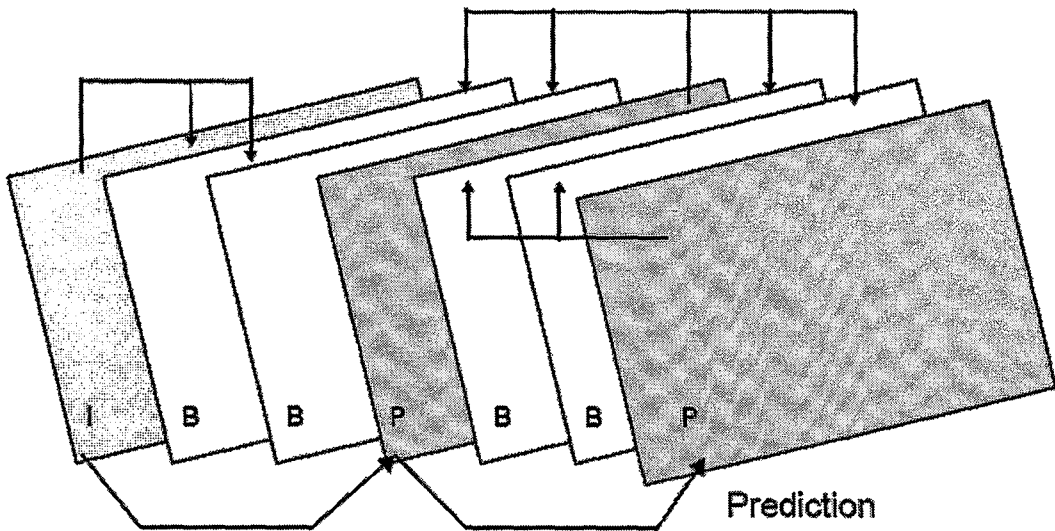
Makro bloklar hareket dengeli sıkıştırma için, bloklar ise AKD sıkıştırması için kullanılır. Çerçeveler 4 çeşitte kodlanır. Bunlar I çerçevesi (Intra) , ileri tahmin çerçevesi (P) , iki yönlü tahmin çerçevesi (B) ve D resimleri sadece her bloğa ilişkin ortalama bileşeni içerir ve imge dizisi içinde çok düşük bir oranında tarama yapmak için kullanılır.

I Çerçevesi, tek bir imge olarak geçmiş ve ileri çerçevelerden bağımsız olarak kodlanır. Kodlanması JPEG benzer bir yapıda dır. Bir tanesi hariç diğer 8x8 blokların hepsi bağımsız olarak kodlanır. I çerçevelerinin kodlanması AKD'ye uyarlanması ile başlar. Elde edilen ayrık kosinüs katsayıları önem derecelerine göre nicemleyici ile ağırlıklandırılır. Son olarak zig - zag tarama sonucu elde edilen veriler değişken uzunluklu kodlama (DUK) kullanılarak kodlanır.



**Şekil 2.18** I Çerçeve Kodlanması

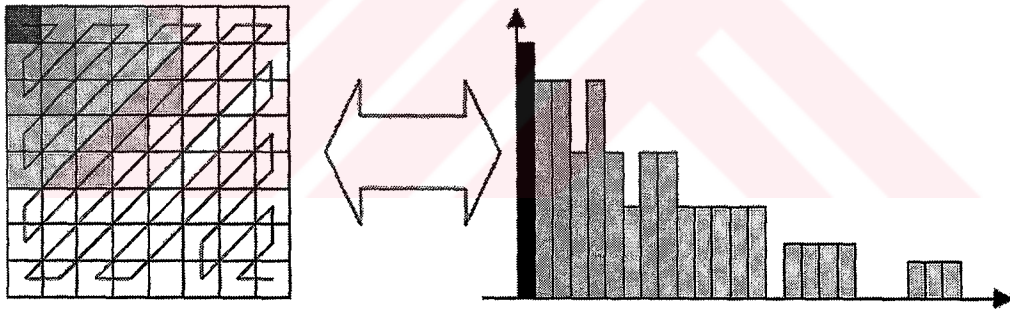
P Çerçevesi, P çerçeveleri, bir önceki I veya P resmini referans alarak, hareket dengelenmiş ileri tahmin kodlamasına olanak sağlar. Bir önceki çerçeve referans alınarak 16x16 blok halinde kodlanır. Bir P makro bloğu 16x16 lık geçmiş referans çerçevesi ve hata terimi ile kodlanır. Referans çerçevedeki 16x16 lık bloğu belirlemek için hareket vektörü eklenir. Hareket vektörünün (0,0) olması 16x16 lık referans bloğunun aynı yerde olması demektir.



**Şekil 2.19** Ara Çerçeveler Arasında Geçişler

B Çerçevesi, B çerçevesi, geçmiş veya gelecek referans çerçevesi ile veya her ikisi de kullanılarak kodlanır. Yüksek sıkıştırma oranına sahip olmasına karşın bunun için daha fazla çerçevenin depolanmasına ihtiyaç duyar. Buda bir miktar gecikmeye sebep olur. Kodlanan blok ile referans blok arasındaki hareketi elde etmek için blok uyumlaması (Blok Matching) kullanılır.

Zig-Zag Tarama, Nicemlenmiş ortalama katsayılar arasındaki artıklık Ayrık darbe kod modülasyonu ile azaltılır ve elde edilen işaret değişken uzunluklu kodlama ile kodlanır. Nicelenmiş AC katsayılar zig-zag tarama kullanılarak Şekil 2.16 'da gösterildiği gibi [Run,Level] şeklinde çiftli forma dönüştürülür ve Değişken uzunluklu kodlama (DVK) ile kodlanır. Burada "Level" sıfır olmayan katsayıların değeri , "Run" ise o anki katsayıdan önceki sıfır olan katsayıların sayısıdır. Kodlamada Huffman benzeri bir kod tablosu bulunmaktadır. Kod haritasında dizilim alçak frekanstan yüksek frekansa doğrudur Zig-Zag tarama ile 2 boyutlu bu frekans katsayılarını alçak tan yüksek frekansa doğru bir boyutlu hale getirir.

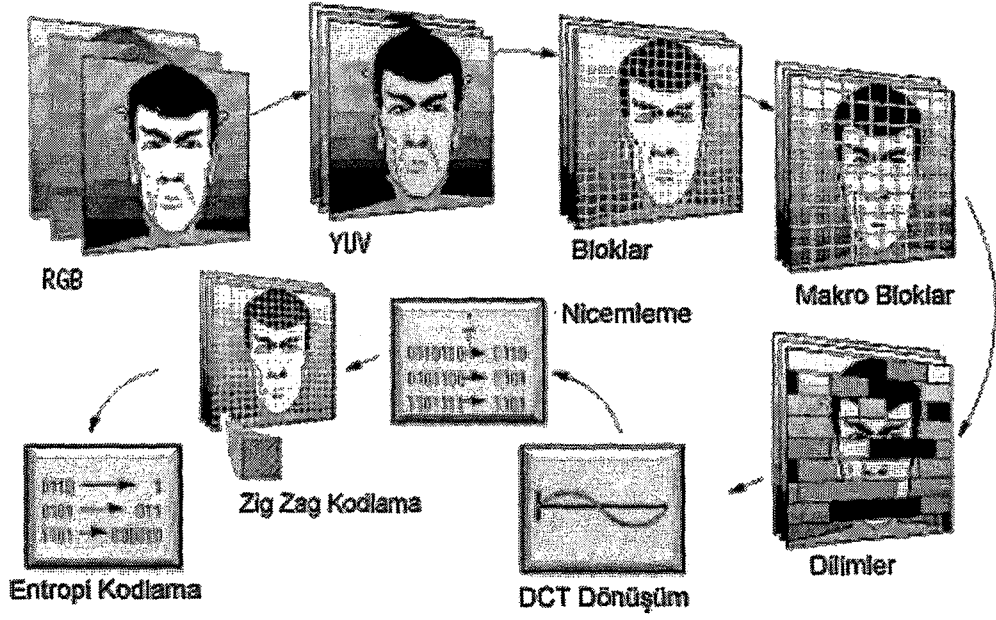


Şekil 2.20 Zig-Zag Tarama Yapısı

#### 2.2.5.1.2 MPEG-1 sıkıştırma yapısı:

Bir MPEG dönüşüm algoritması şu aşamalardan oluşur;

- 1.Ayrık Kosinüs Dönüşümü (AKD)
- 2.Nicemleme (Quantization)
- 3.Run-Length Kodlama



Şekil 2.21 MPEG Dönüşüm Blokları

Hem imge blokları hem de hata kestirim blokları uzamsal bir fazlalığa sahiptir. MPEG algoritması AKD'yi kullanarak 8x8 'lik piksel bloklarını veya 8x8 'lik hata terim bloklarını uzamsal domenin den frekans domenine dönüştürür.

Bu AKD ve nicemleme birçok frekans katsayısının özellikle yüksek frekans katsayılarının 0 olmasının sağlar. Bu katsayılardan maksimum verim elde edilebilmesi için zig - zag şeklinde yerleşmişlerdir. Böylece uzun sıfır işlemleri elde edilebilir. Katsayılar Çalışma-Genlik çifti serilerine dönüştürülür. Her bir çift sıfır olan ve olmayan katsayıları göstermektedir. Bu kodlar daha sonra değişken uzunluklu kod (DUK) ile kodlanarak daha kısa kod dizimleri elde edilir.

#### 2.2.5.2 MPEG-2 sıkıştırma yapısı

MPEG-1 'in 1.5 Mbps ile sıkıştırma kalitesinin birçok uygulamada yetersiz olduğu görülmüş ve yapılan çalışmalar sonucunda 6 Mbps ile sıkıştırılmış bir görüntü dizisinin çok iyi kaliteli sonuçlar verdiği görülmüştür. MPEG-1 ile uyumlu ve farklı uygulamalar için 2-20 Mbps aralığında bit oranları kodlama imkanı sağlayan MPEG-2 standardı geliştirilmiştir. MPEG-1 'e göre farkları ve özellikleri şöyledir;



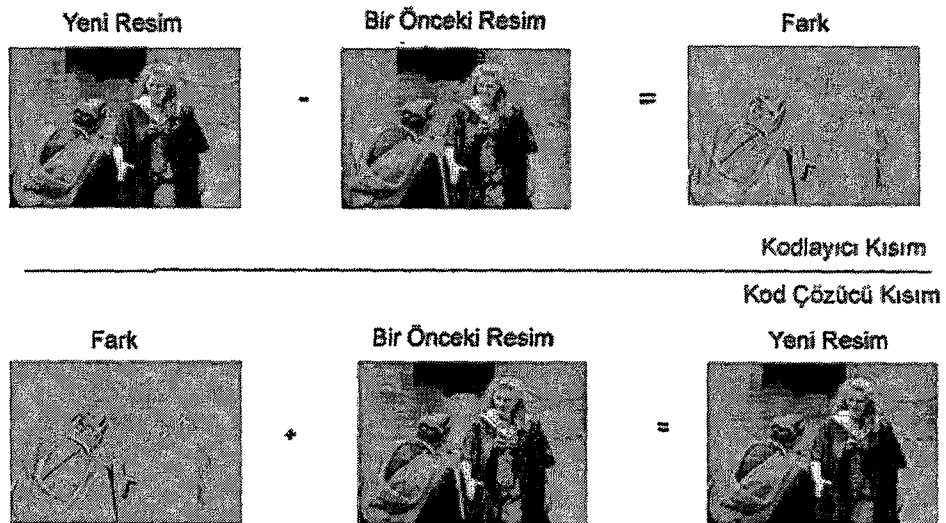
- Ölçeklenebilir bit dizilerinin kullanımına olanak sağlar
- Geliştirilmiş niceme ve kodlama seçenekleri içerir.
- Ters ayırık kosinüs dönüşümü uyumsuzluk kontrolü
- 2 ile 20 Mbps arasında bit oranında kodlama

Şeklinde özetlenebilir. MPEG-2 4 çeşit sıkıştırma algoritması içermektedir.

- Ön işlem
- Hareket Tahmini
- Hareket Dengeleme
- Niceme

Hareket Dengeleme, Hareket dengelemeli tahmin video dizisi içerisindeki zamansal artıklığı atmakta kullanılır. Bir görüntü dizisinde çerçeveler birbirlerine çok benzerdir. Bir referans çerçevesi kullanılarak ondan sonraki çerçevelerin tahmini büyük oranda bilgi miktarının azaltılmasını sağlar. P resimlerinde 16x16'lık makro blokları kendisinden önce kodlanmış I bloklarından tahmin edilebilir. Bir P resmi içerisindeki makro bloğun I resmi içerisinde aranması ve ona en yakın olanının bulunması tahmin hatası olarak karşımıza çıkar. Bu hata kodu AKD domenin de kodlanabilir( Shanawaz A. Basith 1996).

#### Sadece Hareket Eden Nesnelere Kodlanır



Şekil 2.22 MPEG-2 Kodlama Özelliği

Şekil 2.18 'de görüldüğü gibi bir resim kodlanmasında bir önceki resim ile arasındaki fark alınır ve kodlanır. Alıcı tarafa ulaşan kod bir önceki resme eklenerek olması gereken resim elde edilmiş olur. Bu yöntem ile büyük oranlarda sıkıştırma oluşturmak mümkündür.

### **2.2.5.3 MPEG-4 video sıkıştırma**

MPEG 1 ve 2 standartları VCD, DVD ve Sayısal televizyon Yayınları için yeterli olmasına karşın çoklu ortam uygulamalarında gerekli esnekliğe sahip değildir. Yapılan çalışmalar sonucunda 1999 yılında ortaya çıkmıştır. MPEG-4 üç alanda başarılı sonuçlar vermektedir. Bunlar;

- Sayısal Televizyon yayınlarında
- İnteraktif grafik uygulamalarında
- İnteraktif çoklu ortam uygulamalarında

Diğer MPEG standartları sayısal ve analog video ile aynı özellik ve işlevselliklere sahiptirler. Eğer içerik sayısal olarak sunulabiliyorsa, kullanıcı belirli bir içeriği izleme, erişme ve değiştirme gibi birçok işlevsellik kolayca sağlanabilir. Bu standardı ise MPEG-4 oluşturmaktadır.

#### **2.2.5.3.1 MPEG-4 'ün özellikleri**

MPEG-4 'ün çeşitli fonksiyonel araçları destekleyen bir yapıya sahiptir. En önemli özellikleri üç gruba ayrılır;

- 1.Nesneye yönelik erişim
- 2.Düşük Bit Oranı
- 3.Birlikte çalışabilirlik (Diğer uygulamalarla uyum)

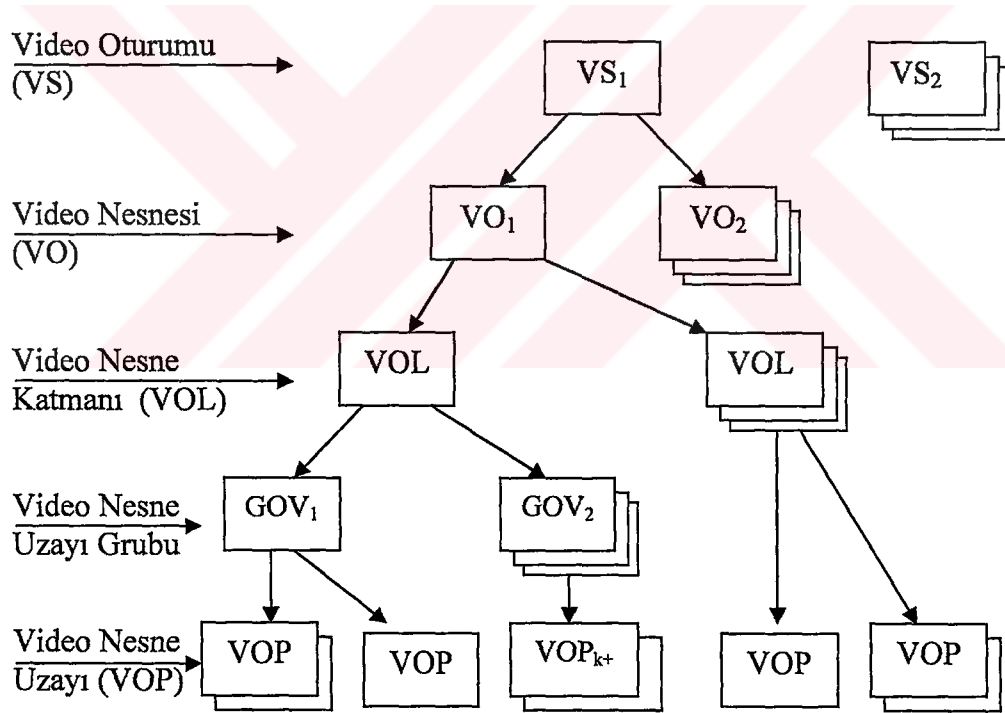
MPEG-4 formatında diğer formatlarda olduğu gibi en önemli özelliklerden birisidir. Arttırılmış kodlama verimliliği ve birçok veri dizisinin aynı anda kodlanması MPEG-4 standardının kullanan uygulamaların uygunluğunu arttıracaktır.

Nesneye yönelik erişim; video nesnelere video çerçeveleri yerine video nesnesi olarak kodlanması ve sunulması nesneye yönelik çalışmalara olanak sağlar. Bu ise

MPEG-4 'ün en önemli özelliklerinden biridir. Nesnelerin etkin sunumunu esas alan nesne düzeltme, bit dizisi düzeltme ve nesne temelli ölçekleme yeni içerik etkileşimlerine imkân verir ( Iain E. G. Richardson 2003).

MPEG-4 hem sürekli tarama hem de geçmeli tarama kullanıma olanak sağlar. Desteklenen alt örnekleme formatı 4:2:0 'dır. Bu alt örnekleme formatında Cr ve Cb örnekleri hem yatay hem düşey doğrultuda ışıklılık bileşeninin yarısı kadardır.

MPEG-4 'ün öne sürdüğü temel görüş nesne tabanlı sunumun temelini sağlayan sesli görsel nesnelere. Bir MPEG-4 görsel ortamı bir veya daha fazla video nesnesinden oluşabilir. Görsel bilgi içeren bir MPEG-4 bit dizisinin görsel ortamının gösterimi Şekil 2.19 da belirtilmiştir.



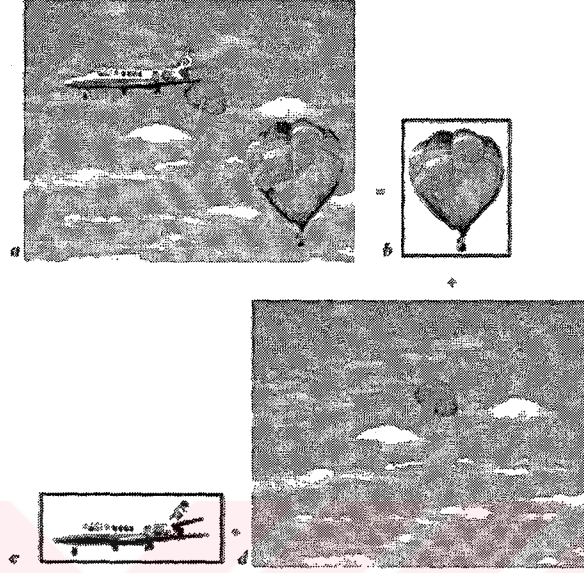
Şekil 2.23 MPEG-4 Yapısı

- Görsel Nesne görüntü dizini (Visual Object Sequence –VS): 2-Boyutlu veya 3-Boyutlu gerçek veya yapay nesnelere ve onların geliştirilmiş katmanlarını içeren bütün bir MPEG-4 manzarasıdır.

- Video Nesnesi (Video Object- VO) : Bir video nesnesi ortamdaki belirli bir 2-boyutlu gerçek önüne gelir. En basit biçimde bu dikdörtgensel bir çerçeve veya bir nesneye veya ortamın arka planına karşılık gelen gelişigüzel şekillendirilmiş bir nesne olabilir.
- Video Nesne katmanı (Video object Layer –VOL) : Her bir video nesnesi uygulamaya bağlı olarak, ölçeklenmiş (çok katmanlı) veya ölçeklenmemiş (tek katmanlı) şekilde video nesne katmanında sunulmak üzere kodlanabilir. Video nesne katmanı ölçeklenebilir kodlamayı desteklemeye imkân verir.
- Video Nesnesi Uzayı Grubu (GOV) : Video nesne uzayı grupları, video nesne uzaylarını gruplandırır. Video nesne uzayı grupları vasıtasıyla, bit dizisinde video nesne uzaylarının birbirinden bağımsız olarak kodlandığı noktalar bulunabilir. Bu özellik ile bit dizisi içerisinde rasgele erişim noktalarına ulaşmaya olanak sağlar. Video nesne uzayı grupları isteğe bağlıdır.
- Video Nesne uzayı (Video object Plane-Vop) : Bir video nesne uzayı, bir video nesnesinin zamanda örneklemevidir. Video nesne uzayları birbirlerinden bağımsız olarak veya bir birlerine bağımlı şekilde hareket dengeleme (compensation) kullanılarak kodlanabilirler. Geleneksel bir video çerçevesi dikdörtgensel köşeleri olan bir video nesne ile gösterilebilir. Bir video nesne uzayı birkaç farklı şekilde kullanılabilir. En genel kullanımda video nesne uzayı, bir video nesnesinin herhangi bir zaman anında kodlanmış video bilgisini içerir. Bu durumda video nesne uzayı, içinde hareket para metreleri, biçim bilgisi ve doku verisini de bulundurur ve bunlar makro bloklar kullanılarak kodlanır.

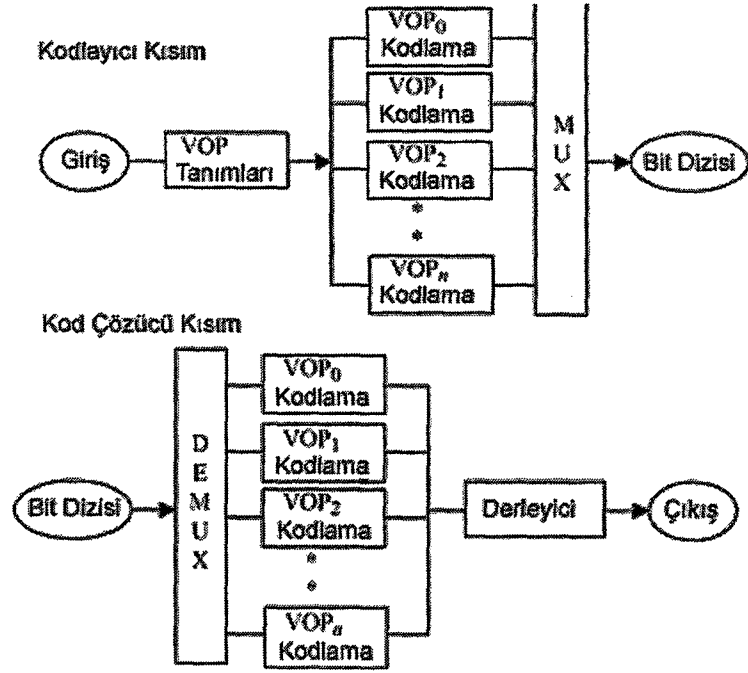
Video Nesne Uzayı, Nesne temelli bir kodlamada video çerçeveleri video nesne uzayı katmanları olarak tanımlanır. Her bir video nesne uzayı ilgili nesne video çerçevesi kodlanır. Şekil 2.20 'de bir video çerçevesi üç adet VOP 'dan meydana gelmektedir. Her bir obje kendi video nesne uzayında tanımlanır(VOP<sub>1</sub>,VOP<sub>2</sub> vb). Nesnelerin dışında kalan kısım ise arka plandır ve VOP<sub>0</sub> ile tanımlanır. Kodlama uygulamalarında arka plan bir defa kodlanır. Diğer nesne uzayları ise zaman içerisinde kodlanır. Kod çözücü tarafında ise arka plan her bir nesneye eklenir ve görüntü oluşur. Bir video dizisinde nesnelere bilgi dizisinde küçük bir kısım oluşturmaktadır. Bundan dolayı bu algoritmayla yüksek orandan sıkıştırma meydana gelir.

Video dizisinde nesnelere Şekil 2.24 de olduğu gibi uçak, balon veya şemsiye gibi olabilir. Bu şekiller Y,U ve V bileşenlerinden oluşur. Nesnelere arka planı maskelemeye kullanılır ve buda bize nesnelere tanımlamakta yardımcı olur.



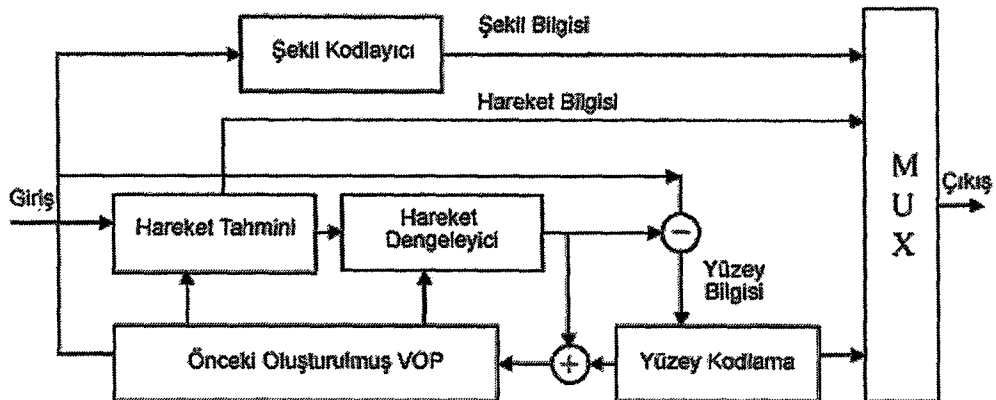
Şekil 2.24 MPEG-4 'de Video Dizisi Nesneleri

Nesnelerin Kodlanması, Her bir nesneye ait VOP kodlanır ve bit dizisi içerisine eklenir. Kodlayıcı VOP bilgilerini ve kod çözme bilgisini alıcıya gönderir. Kod çözme bilgisi VOP'ların nerde ne zaman kullanılacağını belirler. Kullanıcılara bu bit dizisi içerisinde ilgilendikleri nesnelere takip etme imkânı verilir.



Şekil 2.25 Nesne Tabanlı Kodlama

Şekil 2.25 'de Nesne tabanlı kodlamanın şekli gösterilmektedir. Nesneler uzayları belirlendikten sonra her bir VOP kodlanır ve bit dizisine dönüştürülür. Bu bit dizileri bir çoklayıcıdan geçirilerek tek bir bit dizisine dönüştürülür. Alıcı tarafta ise alınan bit dizisi nesnelere dönüştürülür ve derleyici yardımıyla görüntülenirler.

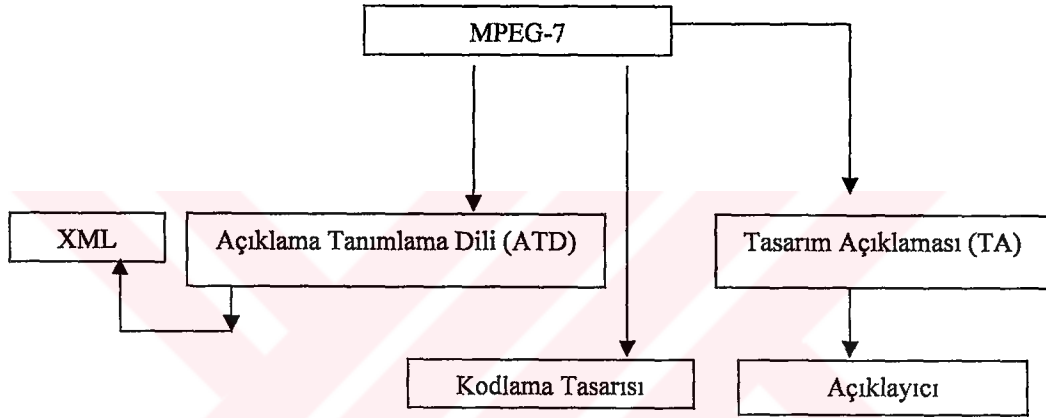


Şekil 2.26 VOP Kodlama Yapısı

Şekil 2.2 bir VOP kodlayıcı yapısı gözükmemektedir. Bu kodlayıcı iki kısımdan oluşmaktadır. Birincisi şekil kodlayıcısı ve geleneksel hareket ve yüzey kodlayıcısıdır.

### 2.2.5.3 MPEG-7 video sıkıştırma algoritması

MPEG-7 çoklu ortam içerik tanımlayıcı bir ara yüzdür. MPEG-7, çeşitli çoklu ortam bilgisini Standard kodlayıcılar ile diğer veri tabanları ile hatta analog görsel-ses bilgisi ile kodlanabilir.



Şekil 2-27 MPEG-7 Yapısı

MPEG-7 tarafından açıklama bilgisi içeriğinde kodlanan form bilgisi, erişimde bulunduğumuz birimlerin açıklamaları (Telif Hakkı, Parası vb.) ve diğer nesnelere ile olan ilişkilerini içerir.

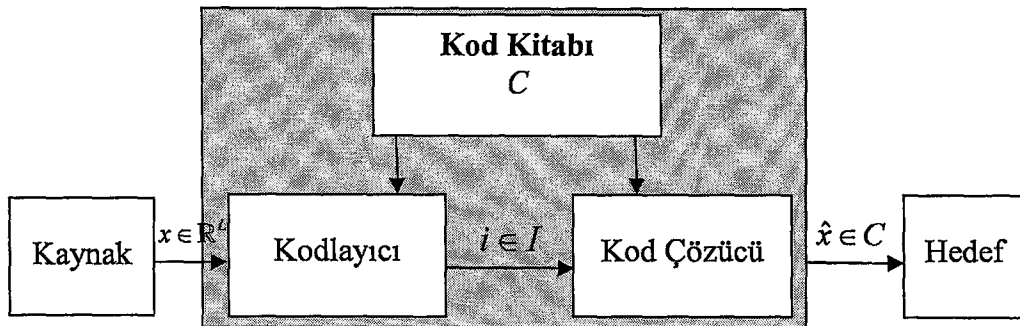
Uygulama alanı çok geniştir eğitimden sayısal kütüphanelere kadar birçok alanda kullanılabilir. Bir görüntü dizisi içerisindeki tüm nesnelere tanımlamak ve bunlar hakkında açıklamalar üretilmesinde kullanılan bir algoritmadır. Bu açıklamalar ile bir görüntü dizisi içerisinde arama gerçekleştirilebilir aradığımız nesnelere kolayca ulaşabiliriz. ATD dili olarak XML (Extensible Markup Language) kullanılır.

### BÖLÜM 3. VEKTÖR NİCEMLEME

Vektör Kuantalama (Nicemele) ilk olarak Shannon tarafından bulunmuştur. Bu çalışma ile teorik olarak veri sıkıştırmanın sınırları gösterilmek istenilmiştir. Shannon 'nın teoreminin temeli bir kaynak içerisinde ard arda gelen ayrı sembollerin vektörlerden oluşan bir paket hale dönüştürmektir. Böylece Kaynağı daha verimli olarak kullanılması amaçlanmıştır.

Vektör Nicemenin pratik çalışmaları 1970'lerin sonu 1980'lerin başı gibi başlamıştır. Bundan sonrasında ise pratik yapılar ve tasarım algoritmaları geliştirilmiştir (Linde 1980). Bu çalışmaların sonucunda Vektör niceme başarılı bir şekilde imge, imge dizileri ve ses kodlamalarında kullanılmaya başlanmıştır (Gersho 1982).

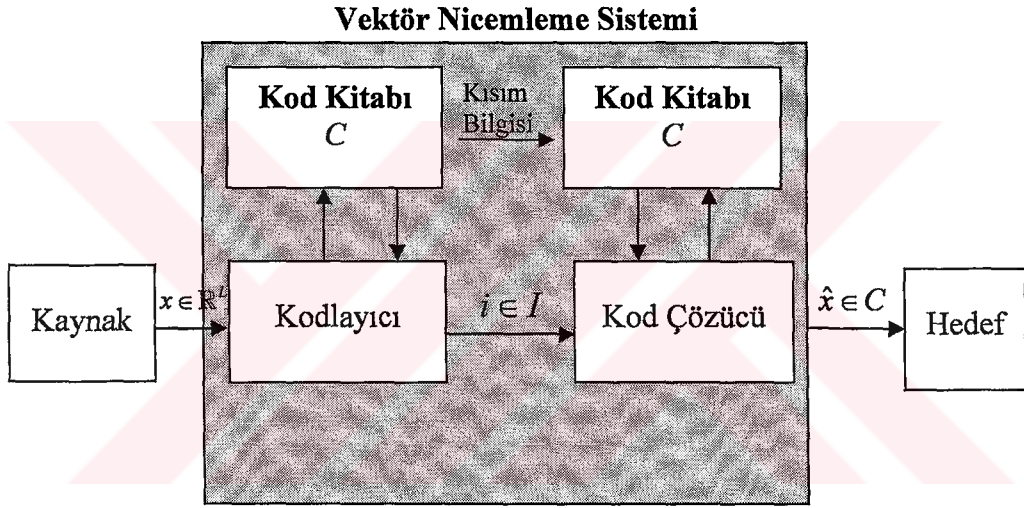
Şekil 3.1 basit bir Vektör Nicemenin yapısı gösterilmektedir. Kodlayıcı kaynaktan  $x \in \mathbb{R}^d$  vektörünü okur. Okuma işleminden sonra kodlayıcı Kod Kitabı ( $C$ ) olarak adlandırılan kod dizisinden  $x$  vektörüne karşılık gelen en iyi  $\hat{x} \in C$  vektörü bulur.  $\hat{x}$  vektörünün kod kitabı içerisindeki adresi  $i \in I$  kod çözücüye gönderilir. Gönderilen  $i$  indeks değeri ile kod çözücü  $\hat{x}$ 'i kod kitabından belirleyebilir. Bu durum yalnızca kodlayıcı ve kod çözücüleri aynı kod kitabını kullanmaları durumunda geçerlidir.



Şekil 3.1 Vektör Niceme Blok Diyagramı



Vektör nicemlemenin temel dezavantajı kaynağın istatistiksel davranışının çok iyi bir şekilde bilinmesi gereklidir. Bu sistem sabit olmayan kaynaklar için uygun değildir. Kodlama ve kod çözme işlemleri tanımlı olan kod kitabına bağlı olarak kodlama ve çözme işlemleri gerçekleştirir. Bundan dolayı kodlayıcı kod kitabının içeriğini bağımsız olarak değiştiremez veya kod çözücü eşzamanlı olarak kod kitabındaki değişikliklerden haberdar edilmelidir. Kod kitabındaki değişikliklerin bildirilebilmesi için özel bir haberleşme yapısına ihtiyaç vardır. Bu iletilen ek bilgiye kısım bilgisi adı verilir. Şekil 3.2 de Vektör niceleme ile kısım bilgi yapısı gösterilmektedir. Bu yapıya örnek vektör nicemleme, adaptif (uyarlamalı) vektör nicemleme (AVN) gösterilebilir.



**Şekil 3.2** Uyarlamalı Vektör Nicemleme Blok Diyagramı

Bu bölümde, Kısım 3.1 'de Vektör niceleme sisteminin geçerli açıklamalarını gösterilecektir. Kısım 3.2 ve 3.3 de ise genel vektör niceleme tasarım ve yapıları belirtilecek. Son kısımda ise uyarlamalı vektör niceleme konusu işlenecektir.

### 3.1 Temel Tanımlamalar ve Özellikler

Bir Vektör Niceleme sistemi,  $N$  vektörlü  $c_i \in \mathbb{R}^L$  'den ,  $I = \{0, \dots, N-1\}$  indeks dizisi ve  $N$  kod kelimesinin  $V$  dizisi ile tanımlanan değişken uzunluklu kod'dan (DUK) oluşan bir  $C = \{c_0, \dots, c_{N-1}\}$  kod kitabına dayanır.

Bu diziler arasındaki bağlantı  $\alpha$  vektör kodlayıcı ,  $\beta$  vektör kod çözücü ve bire bir indeks kodlayıcı  $\gamma$  ile tanımlanır.

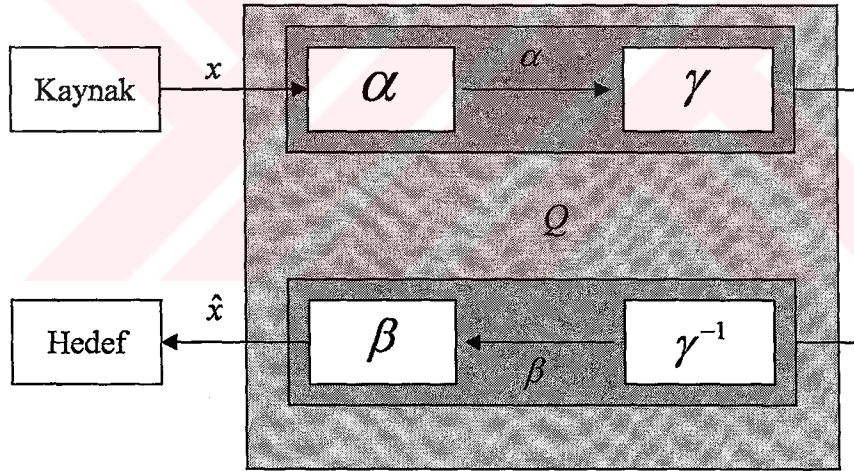
- $\alpha : \mathbb{R}^L \rightarrow I$  vektörü indekse eşler
- $\beta : I \rightarrow C$  bir kod kitabı içerisindeki indeksi ise vektöre eşleştirir.
- $\gamma : I \rightarrow V$  indeksi bir Değişken uzunlukta kod kelimesine eşler

$\alpha : \mathbb{R}^L \rightarrow I$  kodlayıcı  $\gamma \circ \alpha$  ile verilir ve  $\beta : I \rightarrow C$  ise  $\beta \circ \alpha$  ile tanımlıdır.

$Q = \mathbb{R}^L \rightarrow C$  Şeklinde gösterilen niceleme şöyle tanımlanabilir;

$$Q = \beta \circ \alpha = \beta \circ \gamma^{-1} \circ \gamma \circ \alpha \quad (3.1)$$

Denklem 3.1 'in gösterimi şekil 3.3 belirtilmiştir.



**Şekil 3.3** Vektör Nicemleme Yapısı

Kısım 3.2 'de C kod kitabının nasıl oluşturulduğu gösterilecektir. Fakat bu kısımda bir kod kitabının var olduğunu kabul ederek işlemlerimizi gerçekleştireceğiz.

$x = (x^{(0)}, \dots, x^{(L-1)})'$  ve  $Q(x) = \hat{x} = (\hat{x}^{(0)}, \dots, \hat{x}^{(L-1)})'$  arasındaki bozulmayı belirlemek için bir bozulma ölçüsü  $d(x, \hat{x})$  gereklidir. Genellikle  $d(.,.)$  şöyle tanımlanır.

$$d(x, \hat{x}) = \|x - \hat{x}\|_2^2 = \sum_{i=0}^{L-1} (x^{(i)} - \hat{x}^{(i)})^2 \quad (3.2)$$

Bozulma ölçüsünü de belirlendikten sonra, vektör kodlayıcı  $\alpha$  genellikle en yakın komşu (NN) kuralı ile gerçekleştirilir. Bu kurala göre Vektör kodlayıcı, Kod kitabı içerisindeki vektör indeksini en küçük bozulmaya sahip olanından seçer  $\alpha(x) = \arg \min_{i \in I} d(x, \beta(i))$ . Bu kuralın uygulanabilmesi için en yakın değer iki tane olması durumunda bunlardan bir tanesini seçecek bir kabul yapılır.

Vektör niceleyicisinin performansını değerlendirebilmek için,  $x$  vektörlerini rasgele değişken  $X$  ile tanımlandığını ve olasılık yoğunluk fonksiyonu ise  $f_X$  olduğu kabul edilsin. Ortalama bozulma  $D$  ve Ortalama oranı da  $R$  ile gösterilsin bu durumda;

$$D = E[d(X, Q(X))] = \int_{\mathbb{R}^L} d(x, Q(x)) f_X(x) dx \quad (3.3)$$

ve

$$R = E[|\gamma(\alpha(X))|] = \int_{\mathbb{R}^L} |\gamma(\alpha(X))| f_X(x) dx \quad (3.4)$$

Denklem 3.4 'de tanımlanan  $|\gamma(i)|$ ,  $\gamma(i)$ 'nin bit formatında VLC uzunluğudur.  $E$  ise beklenti operatörüdür. Vektör kodlayıcı  $\mathbb{R}^L$  uzayını  $N$  parçalı hücrelere böler  $R_i = \{x \in \mathbb{R}^L : \alpha(x) = i\}$ . Her bir bölüm hücrenin içinde bulunduğu bölmenin olasılığına  $p_i$  ve bölmenin bozulmasına  $d_i$  ise şu şekilde tanımlanır;

$$p_i = \text{prob}(X \in R_i) = \int_{R_i} f_X(x) dx \quad (3.5)$$

ve

$$d_i = E[d(X, Q(X)) : X \in R_i] \quad (3.6)$$

Bu terimleri kullanarak bölmeler için bozulma;

$$D_i = d_i \cdot p_i \quad (3.7)$$

Vektör niceleme teorik olarak genel sınıflı kaynakların kodlanmasında en iyi yöntemlerden biridir. Vektör nicemleme, keyfi bir kod kitabı boyutu ve vektör

boyutları kullanıldığı zaman durağan bir kodlamada ve rasgele işlemlerde en iyi sonucu vermektedir.

### 3.2. Kod Kitabı Tasarımı

Vektör niceleyicinin başarımı genel olarak kod kitabının kalitesine bağlıdır. Kaynak karakteristikleri kaynak ile istatikselsel olarak bire bir uyuşmuyorsa diğer vektör nicemeleme parametreleri  $\alpha, \beta$  ve  $\gamma$  ile bunun temsili imkânsızdır. Bundan dolayı kod kitabının tasarımı büyük önem taşımaktadır. Birçok tasarım algoritması mevcuttur (Lloyd 1957) bunlardan en ünlüsü Lloyd Algoritmasıdır (GLA). Bu algoritma Linde, Buzo ve Gray tarafında geliştirilmiştir. Diğer bir teknik ise kod kitabı vektörlerinin sayısını belirlemede kullanılan bölümlenme metodudur.

#### 3.2.1 Genelleştirilmiş lloyd algoritması (GLA)

GLA 'nın açıklana bilmesi için 2 duruma gerek vardır.

Durum 1 : (En yakın Komşu Durumu ) Oluşturulmuş bir kod kitabı için en uygun bölüm hücreleri  $R_i$   $R_i \subset \{x : d(x, \beta(j)), \forall j \in I\}$  'yi tamamlar.

Durum 2 : (Kütle merkez konumları) Verilen bir bölüm için  $\{R_i : i \in I\}$ , En iyi kod kitabı vektörleri  $cent(R) = E[X : X \in R]$  Beklenti operatörü E ile  $\beta(i) = cent(R_i)$  'yi tamamlamaktadır.

Birinci durumda vektör kodlayıcı  $\alpha$  sabit bir vektör kod çözücüsü  $\beta$  'nın olduğu kabul edilerek formülize edilmiştir. 2. durumda ise  $\beta$  kod çözücüsü sabit bir  $\alpha$  kodlayıcısının olduğu kabul edilerek kodlanmıştır. Bu kabuller şu iterasyon adımlarını önerir: Sabit bir  $C_n$  kod kitabı ile başlanır. Vektör kodlayıcı  $\alpha_{n+1}$  en yakın komşu kuralına bağlı olarak bölüm hücreleri  $R_i$  'leri belirler. Sonrasında yeni bir kod kitabı  $C_{n+1}$ ,  $\beta_{n+1}(i) = cent(R_i)$  'nin belirlenmesi ile hesaplanabilir. Bu iterasyon adımları Lloyd İterasyonu olarak adlandırılır.

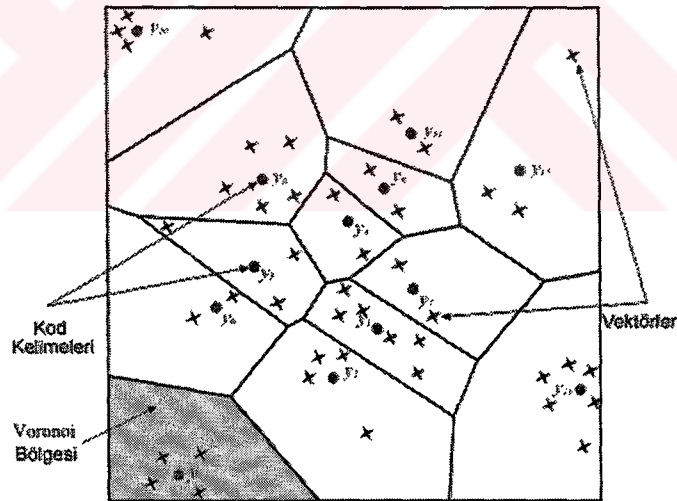
Yukarıda gösterilen iterasyon formu pdf  $f_X(x)$  'in bilindiği ve  $R_i$  bölüm hücrelerinin analitik olarak belirlendiğini kabul eder. Genelde böyle bir şey imkânsızdır. Olasılık

dağılımı eğitim vektör dizileri ile bulunur. Bundan dolayı Lloyd iterasyonun'da kütle merkezleri hesaplanması ve  $R_i$  'lerin hesaplanması ayrı hesaplamaya ile tayin edilir.

$$R_i = \{x \in B : d(x, \beta(i)) \leq d(x, \beta(j)), \forall j \in I\} \quad (3.8)$$

$$cent(R_i) = \frac{1}{|R_i|} \sum_{x \in R_i} x \quad (3.9)$$

Bölüm bozulması  $d_i$  için bölüm olasılığı  $p_i$  ve kısmi dozumla  $D_i$  'yi benzer bir yolla bulabiliriz. Bu da bize gösterilebilir ki ortalama bozulma  $D$  bir Lloyd iterasyonun dan sonra artmaz. Bundan dolayı Lloyd iterasyon'unun sonrasında gelen uygulamalar artmayan ortalama bozulması ile birlikte  $C_m$  kod kitapları dizisi oluşturmaktadır. Bu iterasyon algoritmasına Genellenmiş Lloyd algoritması (GLA) adı verilir. Eğer  $D_m$  ortalama bozulma değişimi yeteri kadar küçükse iterasyon durmaktadır. Bu bir eğitim dizisi'nden sonra her zaman gerçekleşmektedir. Buda bize göstermektedir ki bir minimum değerine yaklaşmaktadır.



**Şekil 3.4** Vektör Nicemeleme'de Vektör ve Kod Kelimeleri

Bu bozulma dizileri küçük bir değerde bir araya gelmektedirler. Bundan dolayı  $c_0$  seçimi çok önemlidir. En basit başlatma tekniği  $N$  adet kod kitabı vektörlerini test dizisi  $B$  'den seçmektir.

GLA ortalama bozulma D 'yi optimize eder ve bunu yaparken ortalama oranı R'yi dikkate almaz. Bundan başka indeks kodlayıcının sabit uzunlukta bir kodlayıcı olduğu kabul edilir. Bundan dolayı R iterasyon sırasında değişmez. Optimizasyon sırasında R önemli ise bu algoritma tipine Entropi Kısıtlı Vektör Niceleme (EKVN ) adı verilir.

Algoritma 1: Genelleştirilmiş Lloyd Algoritması

Verilenler :

Başlangıç Kod Kitabı  $C_0$  ,

Vektör Çözücü  $\beta_0$  ,

Eğitim Dizisi B,

Durma Eşik Değeri ,  $\varepsilon$

İlk Vektör niceleyicisinin  $D_0$  ortalama dağılımı

$m=0$

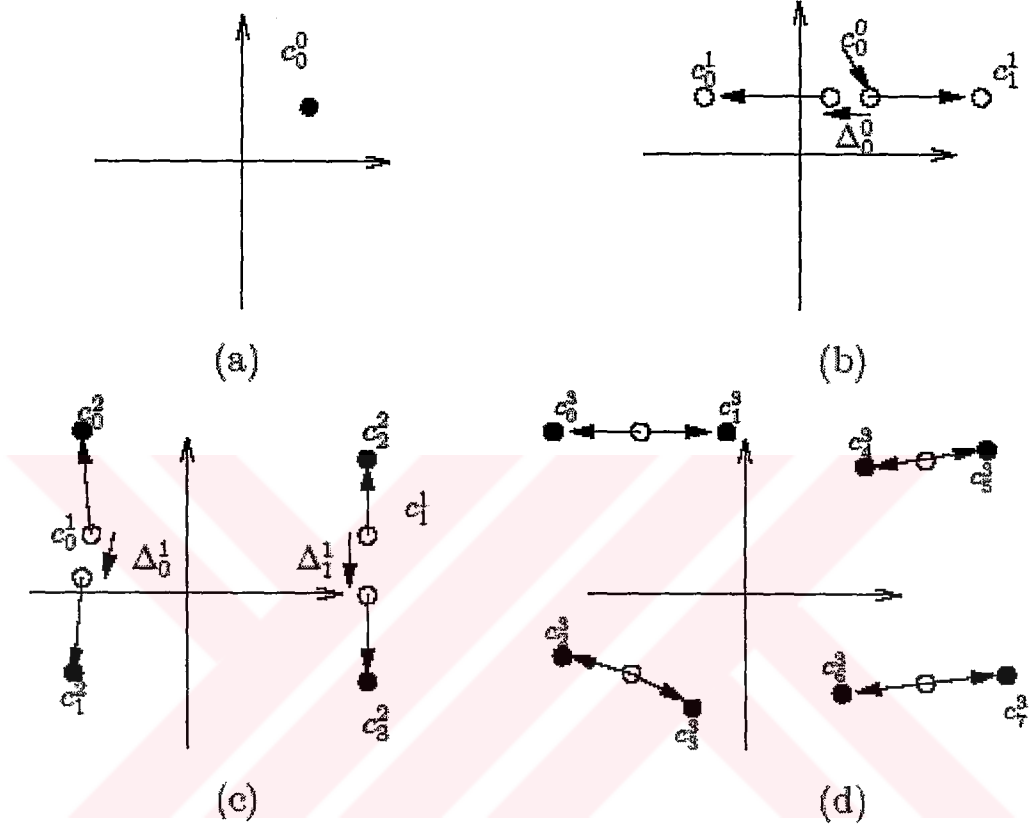
Döngü

$$\begin{aligned} \alpha_{m+1}(x) & \stackrel{def}{=} \arg \min_n d(x, \beta_m(n)) \\ R_n & \leftarrow \{x \in B : \alpha_{m+1}(x) = n\}, \forall n \in I \\ \beta_{m+1}(n) & \leftarrow \frac{1}{|R_n|} \sum_{x \in R_n} x, \forall n \in I \\ C_{m+1} & \leftarrow \{\beta_{m+1}(n) : n \in I\} \\ D_{m+1} & \leftarrow \frac{1}{|B|} \sum_{x \in B} d(x, \beta_{m+1}(\alpha_{m+1}(x))) \\ m & = m + 1 \\ \frac{D_m - D_{m-1}}{D_m} & < \varepsilon \text{ Oluncaya Kadar Dön} \end{aligned}$$

### 3.2.2 Kod kitabı bölümlenme

GLA 'nın dezavantajlarından bir tanesi de Kod vektör (N) sayısı sabit bir boyuttadır. Kod kitapların büyüye bilmesi için Linde tarafından bir teknik geliştirilmiştir. Bu teknikte ilk kod kitabı  $C^0 = \{c_0^0\}$   $c_0^0 = cent(B)$  ile gösterilen sadece bir vektör içerir. Bu vektör daha sonrasında daha ufak  $\Delta$  vektörü ile böler ve sonuçta  $\{c_0^0, c_0^0 + \Delta_0^0\}$

meydana gelir. GLA sonucu  $C^1$  meydana gelir. Yeni kod kitabı  $C^1 = \{c_0^1, c_1^1\}$  bölünür ve  $\{c_0^1, c_0^1 + \Delta_0^1, c_1^1, c_1^1 + \Delta_1^1\}$ 'e dönüşür. Bu böyle devam eder. Bu algoritmanın gösterimi Şekil 3.5 'de olduğu gibidir.



Şekil 3.5 Bölümlleme Algoritması

$$C^0 = \{c_0^0\}, C^1 = \{c_0^1, c_1^1\}, C^2 = \{c_0^2, c_1^2, c_2^2, c_3^2\}, C^3 = \{c_0^3, c_1^3, c_2^3, c_3^3, c_4^3, c_5^3, c_6^3, c_7^3\}$$

Algoritma sonucunda bu kod kitap dizisi oluşur.

Bu işlem adımları bir kod kitabı dizisi meydana getirir. Her bir adımda kod kitabı boyutu ikiye katlanmaktadır. Algoritma istenilen sayıda kod vektörüne ulaşıldığında veya ortalama bozulma  $D$  değeri yeteri kadar küçük olduğunda durur.

### 3.3 Yapısal Vektör Nicemleme

$\alpha$  fonksiyonu belirlenmiş vektörü  $x \in \mathbb{R}^L$  indeks dizisine  $i \in I$  atar. Tipik olarak NN kuralını işletir. Seçilen minimum dağılmaya sahip vektör için  $i$  indeksi seçilir.  $N$  boyutundaki ve  $L$  yüksek vektör boyutundaki büyük bir kod kitabı için bu belirleme çok daha fazla karışıktır. Bundan dolayı kod vektörleri taramak için belirli yapılar geliştirilmiştir.

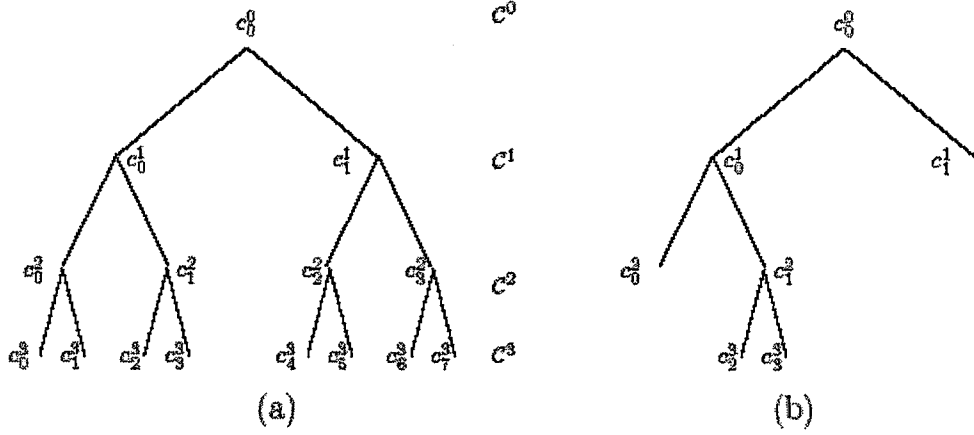
#### 3.3.1 Ağaç yapısında vektör nicemleme

Bölmeleme yöntemi bir çeşit ağaç yapısında vektör nicemlemdir.  $X$  vektörü için belirlenmiş  $\alpha(x)$  indeksi çeşitli katmanlarda hesaplanır. Her bir katmanda bazı kod kitabı vektörleri taramandan hariç tutulur. Sadece son katman hangi indeks'in ve kod kitabının seçildiğini gösterir (Manber 1989).

Bu şekil 3.5 gösterilmektedir.  $C=C^3$  olduğunu kabul edelim. Kod kitabı içerisinde  $x$ 'e en yakın komşuyu belirleye bilmek için  $C^1$  kod kitabının ilk seviyesinin içerisindeki vektörler olan  $d(x, c_0^1)$  ve  $d(x, c_1^1)$  arasındaki bozulmayı hesaplarız.  $d(x, c_0^1)$ 'in  $d(x, c_1^1)$ 'den küçük olduğunu kabul edelim. İkinci katmanda  $C^2$  kod kitabı içerisindeki vektörleri düşünelim. Bu vektörler birinci katmanda en düşük bozulmaya sahip  $c_0^1$  ile bölünmesi sonucunda oluşmuştur.  $(c_0^2, c_1^2)$ .  $d(x, c_0^2)$ 'in  $c_0^3$  ve  $c_1^3$ 'den küçük olduğunu kabul edelim. Bu durumda bölüm vektörü  $c_0^2$  dir ve  $c_0^3$  ve  $c_1^3$  bulunur. Buradan  $\alpha(x)$ ,  $\log_2 N$  katmanı içerisinde bulunabilir. Her bir katmanda ikili vektörlerin dağılımı karşılaştırılmalıdır. Bu yapı şekil 3.6 'da gösterilmiştir. Yukarıda belirtilen arama yapısı temelden tek bir tane kalıncaya kadar bir yol belirlemektedir. Dallardaki düğümler  $C = C^3$  kod kitabını göstermektedirler.  $c_n^1$ 'deki 1. seviyedeki tüm vektörler aradaki  $C^1$  kod kitabı içerisinde yer alır. Tasarım yapısında ara seviyedeki kod kitapları için bunlarda son kod kitabıdır ki, bölmeleme metodundan belirli farklarla ayrılır. Orijinal bölmeleme yönteminde  $C^k$  kod kitabı oluşturulduğu zaman,  $\alpha^k$  en yakın komşu kuralı ile belirlenir ve bu yapılırken ara kod kitapları hesaba katılmaz.



Ağaç yapısı içerisinde bulunan her bir uç sadece ikiye ayrılır ve indeks kodlayıcı  $\gamma$  AYVN( Tree Structured VQ) için temelden istenilen noktaya kadar binary olarak kodlar.



**Şekil 3.6 Ağaç Yapısında Vektör Nicemleme**

(a) Tam Ağaç Yapısı Kod Kitabı  $C = \{c_0^3, c_1^3, c_2^3, c_3^3, c_4^3, c_5^3, c_6^3, c_7^3\}$

(b) Budanmış Ağaç Yapısı VN Kod Kitabı  $C_t$

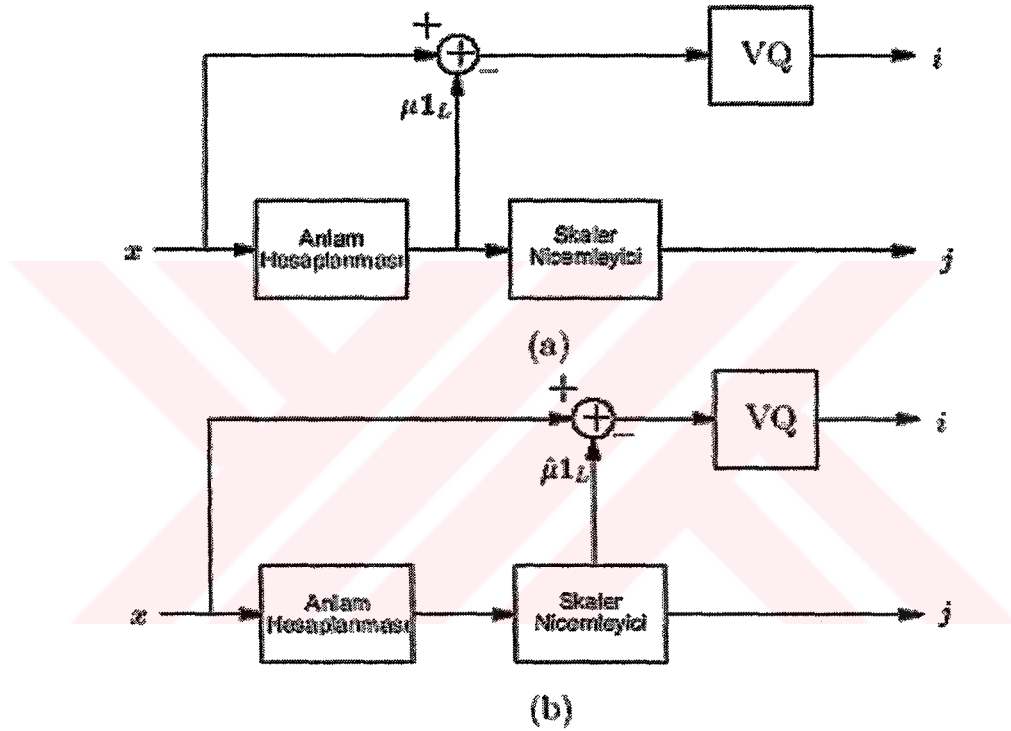
Eğer bazı dallar elenirse bu durumda temel den uç noktalara olan uzunluklar çeşitlilik gösterir. Bundan dolayı  $\gamma$  indeks kodlayıcısı elenmiş ağaç tarafından yönlendirilir. Artık sabit uzunlukta kod üretmemektedir. Buna budanmış ağaç yapısında vektör nicemleme (AYVN) adı verilir.

### 3.3.2 Ortalaması atılmış vektör nicemleme

Vektör elemanlarının istatiksik anlamları 0 'dan 1 'e doğru çeşitlilik gösteriyor ise bu durumda kod kitabı yapısı daha iyi organize edilebilir. Daha büyük kod kitabı kullanmak yerine, kod kitabı boyutu vektörlerin ortalamalarının ayrı olarak işlemesi yolu ile azaltılabilir. Böylece 0 ortalamalı vektörler nicelenmiş olur.

Tipik bir ortalaması atılmış vektör nicemleme yapısı (OAVN) şekil 3.7 'de gösterilmiştir.  $x$  vektörünün  $\mu$  ortalaması hesaplandıktan sonra  $x$ 'in ortalaması-

çıkartılmış fazlalık değeri ( $r = x - \mu 1_L$  Burada  $1_L$ ,  $L$  değişkeni ile  $(1, \dots, 1)^t$  vektörünü tanımlamaktadır.) vektör niceleyiciye gönderilir.  $\alpha$  vektör kodlayıcısı ile indeks  $i$ 'ye eşlenir. Skaler nicemleyici ortalaması  $\mu$  nicemleme için kullanır. Skaler nicemleyici kullanarak indeks  $j$  yi üretir. Kod çözme işlemi ise  $\hat{x} = Q_s(\mu)1_L + Q(r)$  ile gerçekleştirilir. Bu yapıda her iki nicemleyici bağımsız olarak çalışır (Simon Lucey 1998).



Şekil 3.7 Ortalaması Atılmış Vektör Nicemleme Yapısı

Diğer bir OAVN yapısı ise Şekli 3.7 (b) görüldüğü gibidir. Bir önceki yaklaşıma karşın artıklik nicemlenmiş ortalama ile hesaplanır. Bundan dolayı VN artık Skaler nicemleyici'den bağımsız olarak çalışamaz. Böyle bir yapıyı kullanmak bir öncekine göre daha düşük bir ortalama bozulma değerine ulaşmaktadır.

### 3.3.3 Çarpım kod vektör nicemleme (ÇKVN)

OAVN 'da vektör nicemleyici iki kısımdan oluşmaktadır. Skaler nicemleyici ve anlam-çıkartılmış artıklık vektör nicemleyicisidir. Bu yapı tarama karmaşıklığını önemli bir biçimde azaltmaktadır. Bu işlemler sırasında tüm ortalamaların ve ortalaması-azaltılmış artıklıkların taranmasına gerek yoktur. Bundan dolayı bu görev karmaşıklığı daha küçük bölümlere ayrılmıştır. Bu yapıya çarpım kod vektör nicemeleme adı verilir (Simon Lucey 1998).

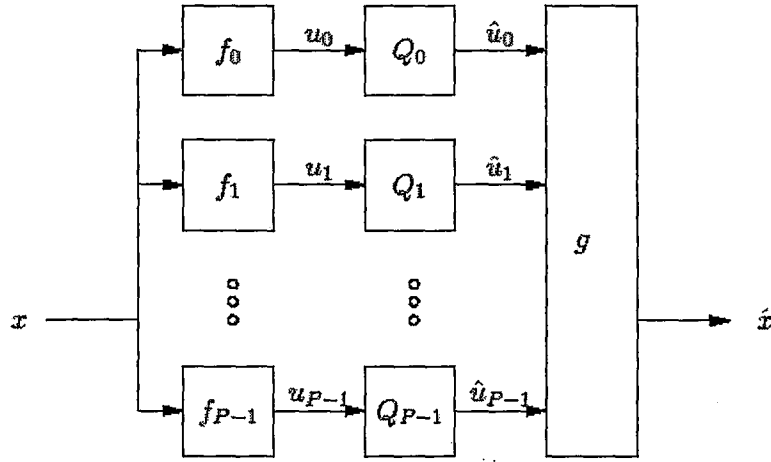
Bağımsız nicemleyiciler için ÇKVN şekil 3.8 'de gösterildiği gibidir (Gersho 1983) .  $x$  vektörü  $P$  vektör elemanlarına ayrılmıştır.  $f_i$  fonksiyonu ile  $u_i : 0 \leq i < P$  .Sonrasında elemanlar bağımsız nicemleyici  $Q_i$  ler ile uygulanmıştır. Nicemlenmiş eleman vektörü  $\hat{u}_i$  ,  $g$  fonksiyonu ile yeniden elde etme değeri  $\hat{x}$  üretmek için kullanılır.

En basit ÇKVN bölümlere ayrılmış alt vektörlerden oluşur. Bu duruma bağlı olarak  $P=2$  ve vektör  $x = (x^{(0)}, \dots, x^{(L-1)})^t$  için ;

$$f_0 : \mathbb{R}^L \rightarrow \mathbb{R}^K, (x^{(0)}, \dots, x^{(L-1)})^t \mapsto (x^{(0)}, \dots, x^{(K-1)})^t \quad (3.10)$$

ve

$$f_1 : \mathbb{R}^L \rightarrow \mathbb{R}^{L-K}, (x^{(0)}, \dots, x^{(L-1)})^t \mapsto (x^{(K)}, \dots, x^{(L-1)})^t \quad (3.11)$$



Şekil 3.8 Bağımsız nicemleyiciler için ÇKVN

### 3.4 Adaptif Vektör Nicemleyici (AVN)

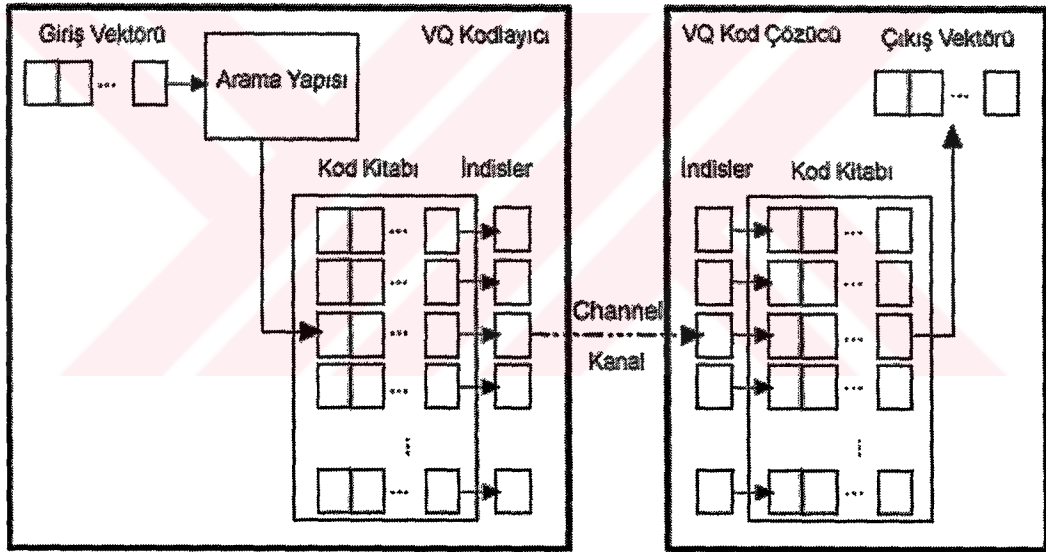
Daha önceki bölüm 3.1 'de bahsedildiği gibi vektör nicemleyici, durağan ve ergodik rasgele işlemler ile üretilmiş vektörlerin kodlamasını için kullanılır. Teorik olarak vektör nicemleme en iyisi ve kod kitabı ve vektör boyutları önemli ölçüde büyük, rasgele işlem olasılık yoğunluk fonksiyonu ise bilindiği kabul edilir. Fakat rasgele süreç durağan değil ise yani zaman içerisinde yavaş olarak değişiyor ise ve olasılık yoğunluk fonksiyonu bilinmiyor ise yapı yeni istatiklere uyum göstermesi gerekir. Bundan dolayı daha genel nicemleme sistemine ihtiyaç vardır (A. Gersho 1985).

Adaptasyon AVN sistemi tarafından sağlanır. Bu sistem içerisinde tüm parametreler zamanla değişen bir yapıdadır. Tipik vektör nicemleme sisteminden farklı olarak rasgele işlemlerde vektörleri kodlamaya gerek yoktur. Ayrıca değişen parametreler için bilgiyi kodlamamız gerekir. Bu bilgiye kısım bilgisi adı verilir.

Özellikle  $C_t$  'nin içeriği zaman içerisinde değişir ve bu değişim özel olarak dağılır. Yeni vektörler eklenir eskilerin ise çıkartılması gerekir. Etkili bir biçimde kısım bilgisinin kodlana bilmesi için, bir süper kod kitabının  $C^*$  bulunması gerekmektedir.  $C_t$  her zaman  $C^*$  'ın bir alt kümesidir.

## BÖLÜM 4. VEKTÖR NİCEMLEME İLE MPEG

Bir imge dizisinin vektör nicemleme ile kodlana bilmesi için geliştirdiğimiz algoritma temelinde Bölüm 3'de anlatılan vektör nicemleme algoritması bulunmaktadır. Bir imge dizisinin kodlanması çeşitli vektör ve kod kitabı boyutları için gerçekleştirilmiş ve sonuçları ekte sunulmuştur. Bu bölümde Vektör nicemlemenin temel yapısı, GLA eğitim algoritması ve imge dizisinin kodlanması yöntemleri açıklanmaktadır. Bir vektör nicemleme sistemin temel yapısı Şekil 4.1'de gösterilmektedir.



Şekil 4.1 Vektör Nicemleme Yapısı

Kodlayıcı kaynaktan  $x$  vektörünü okur. Okuma işleminden sonra kodlayıcı Kod Kitabı ( $C$ ) olarak adlandırılan kod dizisinden  $x$  vektörüne karşılık gelen en iyi  $\hat{x} \in C$  vektörü bulur.  $\hat{x}$  Vektörünün kod kitabı içerisindeki adresi  $i \in I$  kod çözücüye gönderilir. Gönderilen  $i$  indeks değeri ile kod çözücü  $\hat{x}$ 'i kod kitabından belirleyebilir. Bu durum yalnızca kodlayıcı ve kod çözücülerinin aynı kod kitabını kullanmaları durumunda geçerlidir. Kod kitabının oluşturulması ve eğitimi Llyod

algoritmasına bağılı olarak gerçekleştirilmiş olup aşağıdaki algoritma adımları izlenmiştir.

Verilenler :

Başlangıç Kod Kitabı  $C_0$  ,

Vektör Çözücü  $\beta_0$  ,

Eğitim Dizisi B,

Durma Eşik Değeri ,  $\varepsilon$

İlk Vektör niceleyicisinin  $D_0$  ortalama dağılımı

$m=0$

Döngü

$$\alpha_{m+1}(x) \stackrel{def}{=} \arg \min_n d(x, \beta_m(n))$$

$$R_n \leftarrow \{x \in B : \alpha_{m+1}(x) = n\}, \forall n \in I$$

$$\beta_{m+1}(n) \leftarrow \frac{1}{|R_n|} \sum_{x \in R_n} x, \forall n \in I$$

$$C_{m+1} \leftarrow \{\beta_{m+1}(n) : n \in I\}$$

$$D_{m+1} \leftarrow \frac{1}{|B|} \sum_{x \in B} d(x, \beta_{m+1}(\alpha_{m+1}(x)))$$

$$m = m + 1$$

$$\frac{D_m - D_{m-1}}{D_m} < \varepsilon \text{ Oluncaya Kadar Dön}$$



Bir sıkıştırılmış imgedeki bozulmanın ölçütü PSNR değeri olarak adlandırılır ve gösterimi şöyledir;

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2 \quad (4.1)$$

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left( \frac{MAX_I}{\sqrt{MSE}} \right) \quad (4.2)$$

Bu eğitim  $\varepsilon$  değerinin altına düşünceye kadar devam eder. Eğitimin tamamlanması ile birlikte oluşturulan kod kitabına bağılı olarak gelen imge çerçeveleri bloklara ayrılarak kodlanmaya başlar. Bu kodlama işleminde seçilen bloğa kod kitabı içerisindeki en yakın değer tespit edilir. Ve o bloğu temsil ettiği indeks bilgisi alıcıya ulaştırılır. O koda uygun bir değerde blok bulunmaması durumunda ise blok

kodlanarak kod kitabına eklenir ve yeni bir indeks değeri alır. Bu en yakın indeks değeri tespit edilirken Öklit Uzaklığı yönteminden faydalanılır. Bu yöntemle göre,

- Vektör Boyutu =  $k$  ( $= w \cdot h$ )
- Giriş Vektörü  $x = (x_1, x_2, \dots, x_k)$
- Kod Vektörü  $y_i = (y_{i1}, y_{i2}, \dots, y_{ik})$  ise

Bu durumda  $x$  ile  $y_i$  arasındaki Öklit uzaklığı,

$$d(x, y_i) = \|x - y_i\|^2 = \sum_{j=1}^k (x_j - y_{ij})^2 \quad (4.3)$$

Bunu bir örnek ile açıklamak gerekirse;

$cw_1$	(184, 192, 193, 197)
$cw_2$	(34, 50, 43, 47)
$cw_3$	(191, 198, 190, 188)
$cw_4$	(77, 83, 84, 65)
$cw_5$	(63, 70, 94, 98)
$cw_6$	(23, 29, 16, 19)
$cw_7$	(210, 213, 192, 230)
$cw_8$	(151, 153, 169, 171)

**Şekil 4.2** Kod Kitabı Örneği

Kodlanacak örnek vektör,  $v = (150, 145, 121, 130)$  olsun

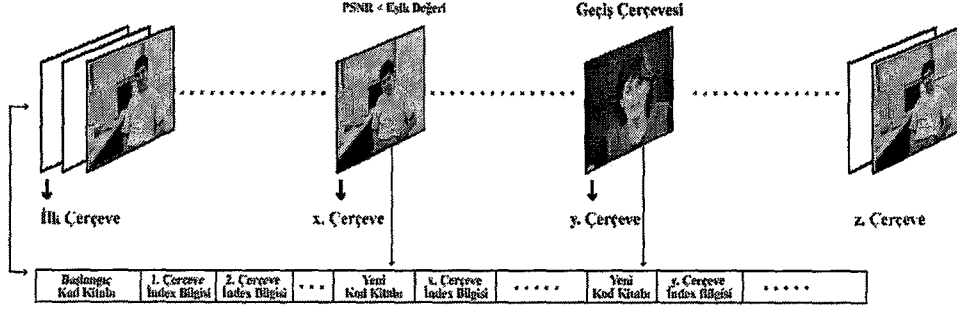
(1) Kod kitabı içerisindeki vektörler ile  $v$  vektörü arasındaki uzaklığı hesaplırsak

$$\begin{aligned} d(v, cw_1) &= 114.2 & d(v, cw_2) &= 188.3 & d(v, cw_3) &= 112.3 \\ d(v, cw_4) &= 124.6 & d(v, cw_5) &= 122.3 & d(v, cw_6) &= 235.1 \\ d(v, cw_7) &= 152.5 & d(v, cw_8) &= 63.2 \end{aligned}$$

(2) Buradan çıkan sonuç  $cw_8$  kod vektörü,  $v$  giriş vektörünü temsil eder.

Vektör nicemleme ile imge dizisinin kodlanmasının temeli Şekil 4.2'de gösterildiği gibidir. Bilindiği üzere bir imge dizisinde durağan görüntülere sahip olduğu gibi ani değişimlerin olduğu çerçevelerden de oluşmaktadır. Bu imgeler eşik çerçevesi olarak

adlandırılacaktır. İlk çerçevenin kodlanması ile birlikte ilk kod kitabı oluşturulur. Bu kod kitabı sonrasında gelen çerçeveler için de referans olmaktadır.



**Şekil 4-2** Vektör Nicemleme ile gerçekleştirilmiş bilgi dizisi

İlk çerçevenin kodlanmasından sonra ikinci ve diğer çerçeveler kodlamaya başlar. Her bir çerçevenin kodlama işleminin tamamlanmasından sonra ilk çerçeve referans alınarak PSNR değeri tespit edilir. Bulunan değer belirlediğimiz eşik değerinden düşük olması durumunda yeni bir kod kitabı oluşturulur. Bu uygulamamızda 26 dB PSNR değerini eşik değeri olarak kabul ettik. Bir imge dizisinin kodlanması esnasında görüntüde oluşan ani değişimlerin olduğu eşik çerçevelerinde ise yeni bir kod kitabı oluşturulur. Bu kodlama yapısı tüm çerçeveler tamamlanuncaya kadar devam eder. Sonuçta ilk kısımda kod kitabı sonrasında çerçevelerin indeks bilgileri, değişimin olduğu veya PSNR'ın eşik seviyesinin altına düştüğü durumlarda ise yeni kod kitabı bilgi dizisine eklenerek bu yapı tamamlanır.

Vektör nicemleme kullanılarak gerçekleştirilmiş olan bu yapıyı MPEG kodlama algoritmaları ile karşılaştırmak gerekirse; sistemimizin en büyük avantajı kod çözücü sistemimizin çok basit bir yapıya sahip olması ve hızlı kod çözme gerçekleştirebilmesidir. Buna karşın görüntü kalitesinde ve sıkıştırma oranında MPEG'den daha kötü olduğu durumlar bulunmaktadır. Bununla ilgili karşılaştırma Tablo 4-1 de gösterilmiştir. Sistemimiz yeni geliştirilmekte olan MPEG-7 algoritmasına benzer bir yapıya sahip olup bir imge dizisi içerisinde nesne tanımayla ve onlar üzerinde işlem yapmaya olanak sağlayan özellikler içermektedir.



Kod kitabı oluşturulurken seçilen vektör boyutu ve kod vektörü sayısı görüntünün kalitesini önemli ölçüde etkilemektedir. Vektör boyutunun küçük seçilmesi görüntü kalitesini arttırmakta, buna karşın sıkıştırma oranı küçük değerlerde kalmaktadır. Uygulamalarımızda 2x2 ve 4x4 vektör boyutları kullanılarak çeşitli imge dizilerine uygulanmış ve elde ettiğimiz sonuçlar Tablo 4.3 de gösterilmiştir.

**Tablo 4.1** Vektör Nicemleme ile MPEG Yapısının Karşılaştırılması

	VQ			MPEG
Kod Kitabı Boyutu	128	256	512	-
Sıkıştırma Oranı	*26,02	*16,32	*8,74	30
PSNR	29,8	31,1	33,8	25 - 26
Vektör Boyutu	4x4			

\*Labotatuar Frame 202 uygulaması esas alınmıştır.

**Tablo 4.2** Vektör Nicemleme ile MPEG Kodlama Yapılarının işlem adımlarının karşılaştırılması

İşlemler	
VQ	<ul style="list-style-type: none"> <li>• Kod Kitabının Oluşturulması ve Eğitim</li> <li>• İmge Çerçevesindeki Blokların Kod Kitabına bağlı olarak indeks değerlerinin atanması</li> <li>• Kod kitabı ve kodlanan imge çerçevelerinin bir dizi haline getirilmesi</li> </ul>
MPEG	<ul style="list-style-type: none"> <li>• Ters Ayrık Kosinüs Dönüşümü</li> <li>• Prediction / Blok Uyumlaması</li> <li>• Nicemleme</li> <li>• Zig Zag Kodlama</li> <li>• Entropi Kodlama</li> </ul>

**Tablo 4.3 Ölçüm Sonuçları**

İmge Dizisi	Frame	Matris Boyutu	Kod Kitabı	PSNR	Sıkış. Oranı
Laboratuar Frame	38	2x2	64	29.2	15,378
			128	30.5	12,85
			256	34.6	10,70
Laboratuar Frame	202	4x4	64	27.18	38,871
			128	29.7	26,020
			256	31.6	16,329
Miss America	21	2x2	64	38,6	15,402
		4x4	128	40.8	26,176
			256	42.6	16,376

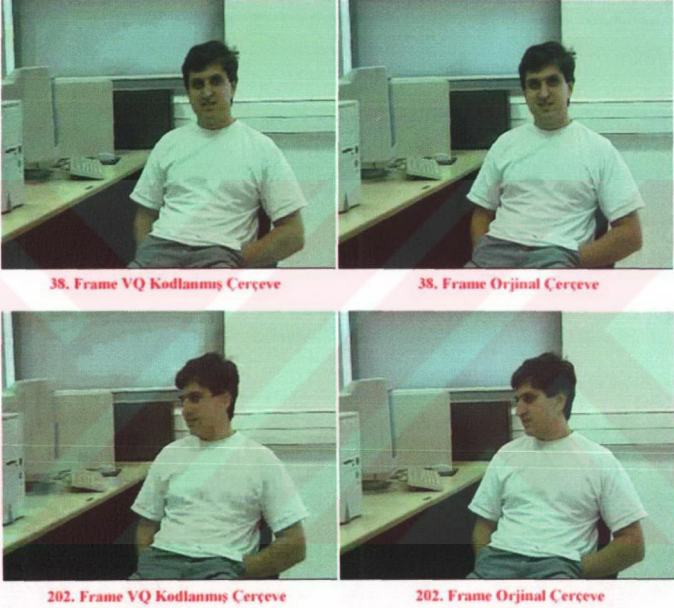
Tablo 4.2 de görüldüğü gibi bir vektör nicemleme yapısının MPEG karşısındaki basitliği görülmektedir. Kod çözücü sistemin görevi gelen indeks bilgisine bağlı olarak kod kitabından ilgili vektörleri kullanarak görüntüyü oluşturmaktır. MPEG çözücüsünde ise Ters Ayrık Kosinüs dönüşümünden zig zag kodlamaya kadar birçok adım bulunmaktadır. Bu da sistemimizin önemli özelliklerinden biridir.

İyi bir sıkıştırma ve görüntü kalitesi için 4x4 vektör boyutunda 512 kod kitabı ile kodlanma yapılmasının iyi sonuçlar verdiği görülmüştür.

Böyle bir sistem ileride birçok alanda kullanılabilir, bunların başında mobil teknolojiler gelmektedir. Basit bir çözme yapısına sahip olduğundan en basit işlemcilerde bile bu kod çözme işlemi kolaylıkla gerçekleştirilebilir. Bunun yanı sıra görüntünün gösterimi için gerekli olan süre çok kısa bir süre olduğundan internet teknolojilerinde de kolaylıkla kullanılabilir. Ayrıca obje tanıma da kolaylık sağlayacak bir yapı içermektedir, bu özelliğe sahip olmasından dolayı ileride birçok uygulamada bize kolaylık sağlayacaktır.

## SONUÇLAR VE ÖNERİLER

Laboratuar imge dizisi için yapılan deneme sonuçları aşağıda görüldüğü gibidir.



Şekil 5.1 Laboratuar 38. ve 202 çerçevelerinin orijinal ve kodlanmış görüntüleri

38 . Çerçevenin 2x2 vektör boyutunda 256 kod vektörü ile kodlanmış görüntüsü ilk kısımda görüldüğü gibidir. Bu kodlama sonucunda 10,5 kat sıkıştırma ve 34.6 dB lik bir PSNR değeri elde edilmiştir.

202. Çerçevenin 4x4 vektör boyutunda ve 128 kod vektörü ile kodlama neticesi ise alt kısımda görüldüğü gibidir. Sonuçta 26 kat sıkıştırma oranı ve 30.3 dB PSNR değeri elde edilmiştir.

Sistemimizin “Miss America” imge dizisi için ( 29. ve 65. Çerçevesel arasındaki) kodlama sonuçları şu şekildedir.

(Çerçeve 36)



Kodlanmış Çerçeve



**Şekil 5.2** Miss America 36. çerçevenin orijinal ve kodlanmış görüntüsü

Bu görüntü dizisi için şu sonuçlara ulaşılmıştır:

Ortalama PSNR değeri 38.6 dB

Vektör Boyutu : 2 x 2

Kod Vektör Sayısı : 64

Sıkıştırma Oranı : 15,4

(Çerçeve 57)



Kodlanmış Çerçeve



**Şekil 5.3** Miss America 57. çerçevenin orijinal ve kodlanmış görüntüsü

Bu görüntü dizisi için şu sonuçlara ulaşılmıştır:

Ortalama PSNR değeri 40.8 dB  
Vektör Boyutu : 4 x 4  
Kod Vektör Sayısı : 128  
Sıkıştırma Oranı : 26,176

Orijinal Çerçeve 43



Kodlanmış Çerçeve

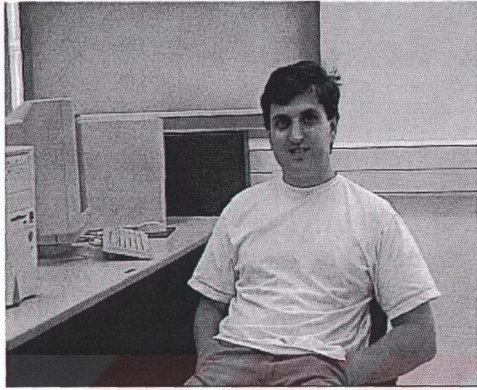


Şekil 5.4 Miss America 43. çerçevenin orijinal ve kodlanmış görüntüsü

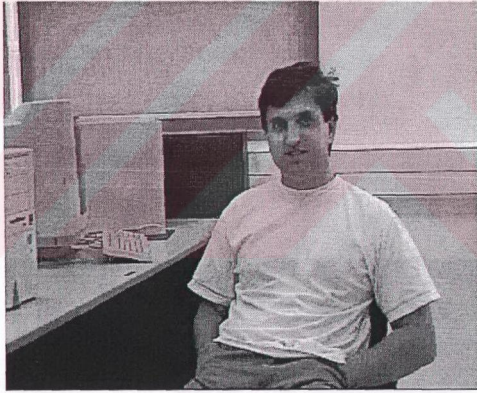
Bu görüntü çerçevesi için şu sonuçlara ulaşılmıştır:

Ortalama PSNR değeri 42.4 dB  
Vektör Boyutu : 4 x 4  
Kod Vektör Sayısı : 1024  
Sıkıştırma Oranı : 5,32

Orijinal Çerçeve 35



Kodlanmış Görüntü



Şekil 5.5 Laboratuar 35. çerçevenin orijinal ve kodlanmış görüntüsü

Bu görüntü çerçevesi için şu sonuçlara ulaşılmıştır:

Ortalama PSNR değeri 30.5 dB

Vektör Boyutu : 2 x 2

Kod Vektör Sayısı : 128

Sıkıştırma Oranı : 12,85

Sonuç olarak sistemimiz 4x4 ve daha büyük boyutlu vektörler için yüksek sıkıştırma oranına sahip olmasına karşın görüntü kalitesi oldukça düşmüştür. 10 saniyelik imge dizisinin kodlanması yaklaşık üç saat almış olup kod vektör sayısının artmasına bağlı olarak bu sürede artmıştır. Vektör nicemlemenin en önemli özelliklerinden biri olan basit bir çözücü sistemine sahip olmasından dolayı hızlı bir çözümleme gerçekleştirmektedir. Bundan dolayı böyle bir sistem

- Mobil Teknolojiler
- Web Teknolojileri
- İmge ve Video'nun Web Sayfalarında kullanılması
- Gerçek zamanlı video ve İmge Görüntüleme Sistemlerinde

Kullanılabilir yapıdadır. Şuan ki aşamada sistemimiz her bir imge için indeks bilgisini göndermektedir. Bundan sonraki çalışmalarımızda hareket tahmini yöntemi ile imgeler arasında sadece değişen bölgelerin kodlanarak indeks bilgisinin gönderilmesi ve böylelikle tekrarlanan indeks bilgileri çıkartılarak daha fazla sıkıştırma elde edilmesi amaçlanmaktadır. Ayrıca diğer araştırmacıların geliştirdikleri yeni kodlama ve eğitim algoritmaları ile daha hızlı ve kaliteli kod kitabı elde edilerek daha iyi görüntü ve sıkıştırma oranı elde edilmesi amaçlanmaktadır.



## KAYNAKLAR

1. Linde Y. , Buzo A. , and R. M. Gray. An algorithm for vector quantizer design. IEEE Trans. Commun., 28:84, 95, Jan. 1980.
2. Gersho A. and Ramamurthi B. Image coding using vector quantization. In ICASSP'82, volume 1, pages 428{431, May 1982.
3. Lloyd S.P. Least squares quantization in PCM. Unpublished Bell Laboratories Technical Note. Published in IEEE Trans. on Inf. Th. 1982, 1957.
4. Manber U. Introduction to algorithms. Addison-Wesley, 1989.
5. Gersho A. and Gray R. M.. Vector Quantization and Signal Compression. Kluwer Academic Publishers, 1992.
6. Gersho A. and B. Ramamurthi. Image coding using vector quantization. In ICASSP'82, volume 1, pages 428{431, May 1982.
7. A. Gersho and M. Yano. Adaptive vector quantization by progressive codevector replacement. In Proc. of ICASSP, pages 4.6.1{4.6.4, 1985.
8. Lloyd S.P. Least squares quantization in PCM. Unpublished Bell Laboratories Technical Note. Published in IEEE Trans. on Inf. Th. 1982, 1957
9. Gersho A. and Yano M. Adaptive vector quantization by progressive codevector replacement. In Proc. of ICASSP, pages 4.6.1{4.6.4, 1985.}
10. Gregory N. Video Compression Employing algorithm using temporal blocking structures. IEEE April 2001 Vol:37 Page 563
11. Khalil H. Predictive Multistage Vector Quantizer Design Using Asymptotic Closed Loop Optimization IEEE November 2001 Vol No: 10 Page 1765
12. Ricardo L. De Queiroz Senior Member IEEE , Very Fast JPEG Compression Using Hierarchical Vector Quantization IEEE May 2000 Vol : 7 Page 97
13. Yang H. Source Channel Prediction In Error resilient video coding IEEE 2003 Page :233
14. Kenneth R. , Open Loop Design of Predictive Vector Quantizers for Video Coding IEEE 1998 Page 953

15. Raouf H. Rate Distortion Based Video Coding With Adaptive Mean-Removed Vector Quantization IEEE Oct.1998 International Conference
16. Chin – Chan Chang , An Efficient and effective Method for VQ Codebook Design IEEE December 2003 Page 782
17. Chuan Y. , A Fast Linde , Buzo Gray Algorithm in Image Vector Quantization IEEE March 1998 Vol :45 Page 432
18. Lucey S. Low Rate Image Compression Using Vector Quantization Book, Bachelor Electric and Electronic Engineering , November 1998
19. Khayam Syed A. The Discrete Cosine Transform Theory and Application, Michigan State University Electric and Computer Engineering ,March 2003
20. Bodden E. Arithmetic Coding Revealed , RWTH Aachen University May 2004
21. Taubman David S., JPEG2000 Image Compression Fundamentals Book , 2002
22. Shanawaz A. Basith , Digital Video, MPEG and Associated Artifacts , Imperial College Departments of Computing & Electrical Engineering June 1996
23. Iain E. G. Richardson , H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia Book , October 2003
24. Moving Picture Experts Group (MPEG) Web Site [www.mpeg.org](http://www.mpeg.org)
25. William H. Press, NUMERICAL RECIPES IN C: THE ART OF SCIENTIFIC COMPUTING Book (ISBN 0-521-43108-5) 2002

## ÖZGEÇMİŐ

1978 yılında İzmit'te doğdu. İlk, orta ve lise öğrenimini Gölcük'te tamamladı. 1995 yılında girdiđi Kocaeli Üniversitesi Elektronik ve Haberleşme Mühendisliđi bölümünden 2000 yılında Elektronik ve Haberleşme Mühendisi olarak mezun oldu.

2001 yılından beri Kocaeli Üniversitesi Karamürsel Meslek Yüksek Okulunda Öğretim görevlisi olarak görev yapmaktadır.