

⇒ Çoklu ortam hesaplama ;

- Bilgisayar Ağları ve İşletim Sistemleri
- Bilgisayar Görme ve Örnekleme Tanıma
- Bilgisayar Grafiklerinde
- İmgenin, sesin, konuşma işleminde
- İnsan bilgisayar etkileşiminde kullanılır.

⇒ Çoklu ortam ve Analizde ;

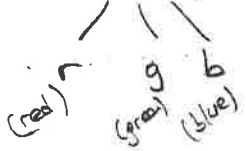
- Otomatik olarak veriden depoları çıkartma.
- Özet çıkarma
- Nesne takibi ve nesne tanıma yapılır.

Multimedia : Bilgilerimizin tutulma şeklidir, hepsi 1 ve 0'lerden oluşmaktadır.  
 Metin, video, resim

- Multimedia hesaplama sistemlerinde yapmak istenen ; örneğin youtube 'dan video izleme, whatsapp'tan görüntülü konuşma, gibi örneklerin her anında çoklu ortam vardır.
- Aynı anda birden fazla kısmın aynı videyu izlemesi, bunun veritabanında hazırlanmasında "çoklu ortam" vardır.
- Tomografi ve emarların üst üste çakıştırılması
- Farklı cihazlardan çekilen fotoğrafların birleştirilmesi } ÇOKLU ORTAM

Quantalama : En min. ile en max. değeri kaç parçaya bölmeyi sağlar. } Yani en koyudan → en açık renge geçişte kullanılır.

Renkli Fotoğraf



Video : Birden fazla fotoğrafın ard arda hızlıca sıralanması.

↓ saniyede → 24 çerçeve

24'ün üstüne çıkarsa gözbebeği ölmez.

fps = frame per second,,

- Kaliteye göre değişir. (24, 30, 60) (örneğin telefonlarda 30 olabilir)

\* Yavaş gösterim olacaksa, daha çok resim çekilir. Ağır çekimler düzenli ve zor, çok hızlı çalışması gerekir.



ipe  $\rightarrow$  BMP, JPG, GIF, TIF, PNG, PPM

Neden bu kadar fazla?

Çünkü birçok firma kendi fikirlerine göre söyleyip kısaltmışlardır. Her birinin birbirinden farklı özelliği vardır.

BMP  $\Rightarrow$  Kayıpsız Sıkıştırma (Büyük veriyi daha az yer kaplar)

JPG  $\Rightarrow$  Kayıplı, kayıpsız olabilir.

GIF  $\Rightarrow$  Hareketli

TIF  $\Rightarrow$  Tek katman, birden fazla katman

SES  $\rightarrow$  WAV, MP3

VIDEO  $\rightarrow$  AVI, MOV

Aslında WAV'in ses kalitesi MP3'ten daha iyidir.

Bir imgeyi, fotoğrafı wordpadde açarsak?

Header kısmı vardır  $\rightarrow$  Sıkıştırma Algoritması

Kaç bit derinlikte olduğunu söyler ve veri başlangıcı başlar.

Aynı şekilde videoyu veya sesi wordpadde açarsak benzer şeylerle karşılaşırız.

ASCII karakterlerine karşılık gelecek şekilde tam sayılar

Görüntü  $\rightarrow$  2 boyutlu görüntü

Ses  $\rightarrow$  Tek boyutlu görüntü

Verinin boyutları 2'nin kuvveti ise çok daha hızlı algoritmalar çalıştırabiliriz.

İmpede 32, 64, 128 - - - gibi boyutlar seçilir.

Seste 2'nin kuvveti olması da olur.

\* Bizim duyduğumuz max ses 15-16 KHz

Ses sıkıştırması  $\rightarrow$  44 KHz

İmge bir matris olarak düşünülebilir. Matrisi 2 boyutlu sunulabilir. Yani üçüncü boyutu bir renk veya yükseklik olarakta sunulabilirsin.

Gri renk tonu siyah ve tam beyaz arasında

Binary İmge  $\rightarrow$  Arapçası yok, grilik yok fotoğraf siyah veya beyazdan oluşur.

(0 ve 1)

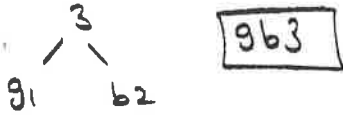


baba coddede pec' mesajını kodladım. {a,b,c,d,e,A,p,h}

Tekrarlama sırasına göre yazarsak?

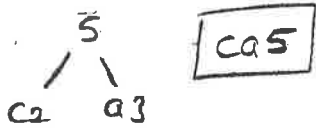
$p1, b2, c2, a3, d3, e3$

1.  $g1, b2 \quad 1+2=3$



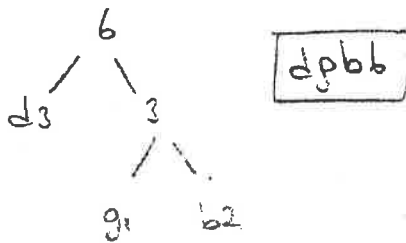
$c2, a3, d3, gb3, e3$

$c2, a3 \quad 2+3=5$



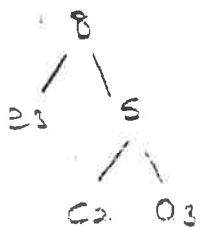
$d3, gb3, e3, ca5$

$d3, gb3 \quad 3+3=6$

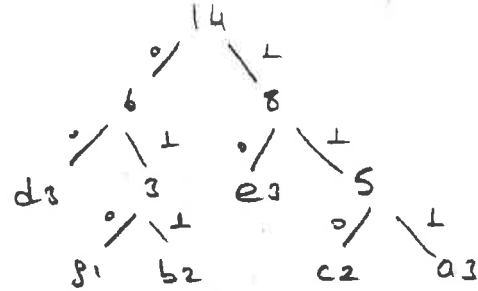


$e3, ca5, dpbb$

$e3, ca5 \quad 3+5=8$



⑤  $dpbb, eco8 \quad 6+8=14$



- Sapa pidi için 1, sola pidi için 0 yazılır.

d: 00

e: 10

p: 010

c: 110

b: 011

a: 111

\* Normalde şifrelessek  $14 \times 3 = 52$  bit olurdu.  
8 tane karakter olduğundan  $(2^8)$

Huffman ile;

tekrar x bit uzunluğu  
sayısı

$d \rightarrow 3 \times 2 = 6$

$p \rightarrow 1 \times 3 = 3$

$b \rightarrow 2 \times 3 = 6$

$e \rightarrow 3 \times 2 = 6$

$c \rightarrow 2 \times 3 = 6$

$a \rightarrow 3 \times 3 = 9$

36 bit oldu.

# Huffman

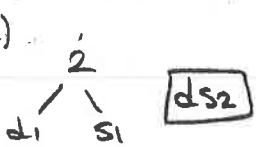
Kayıpsız veri sıkıştırması yapar. En büyük avantajı; sık kullanılan karakterleri daha az, nadir kullanılan karakterlerin ise daha fazla yer kaplamasını sağlamasıdır. Huffman kodlaması az sayıda sembol içeren ve olasılığı son derece çarpık olan durumlarda etkili değildir.

ör

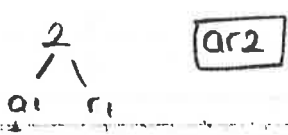
"codes-are-cool" mesajını sıkıştıralım? {c, o, d, e, s, -, a, r, l}

c<sub>2</sub>, o<sub>3</sub>, d<sub>1</sub>, e<sub>2</sub>, s<sub>1</sub>, -<sub>2</sub>, a<sub>1</sub>, r<sub>1</sub>, l<sub>1</sub>

→ d<sub>1</sub>, s<sub>1</sub>, a<sub>1</sub>, r<sub>1</sub>, l<sub>1</sub>, e<sub>2</sub>, -<sub>2</sub>, c<sub>2</sub>, o<sub>3</sub>



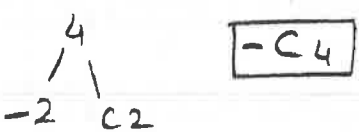
2) a<sub>1</sub>, r<sub>1</sub>, l<sub>1</sub>, e<sub>2</sub>, -<sub>2</sub>, c<sub>2</sub>, ds<sub>2</sub>, o<sub>3</sub>



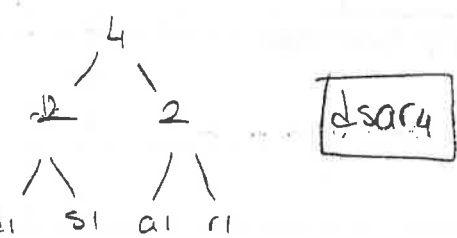
3) l<sub>1</sub>, e<sub>2</sub>, -<sub>2</sub>, c<sub>2</sub>, ds<sub>2</sub>, ar<sub>2</sub>, o<sub>3</sub>



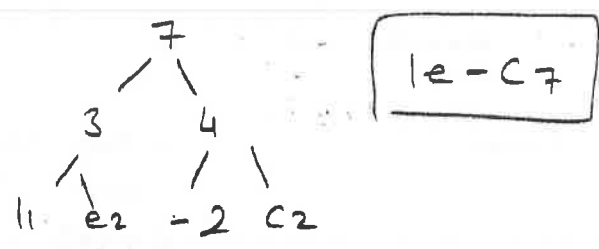
4) -<sub>2</sub>, c<sub>2</sub>, ds<sub>2</sub>, ar<sub>2</sub>, le<sub>3</sub>, o<sub>3</sub>



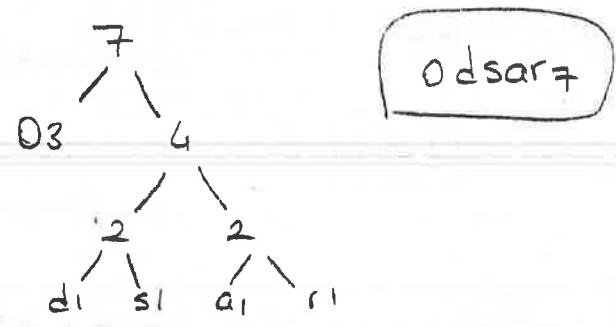
5) ds<sub>2</sub>, ar<sub>2</sub>, le<sub>3</sub>, -c<sub>4</sub>, o<sub>3</sub>



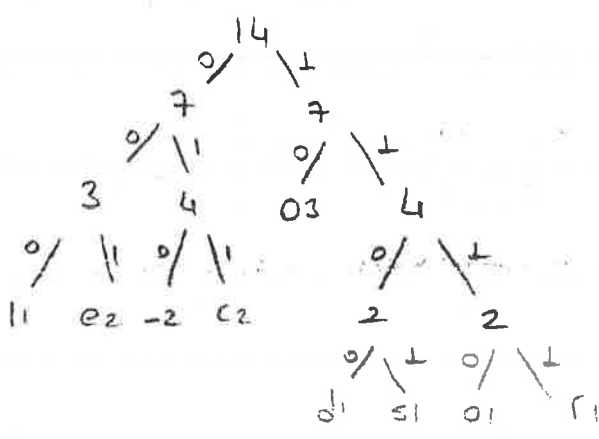
6) le<sub>3</sub>, -c<sub>4</sub>, dsar<sub>4</sub>, o<sub>3</sub>



7) o<sub>3</sub>, dsar<sub>4</sub>, le-c<sub>7</sub>

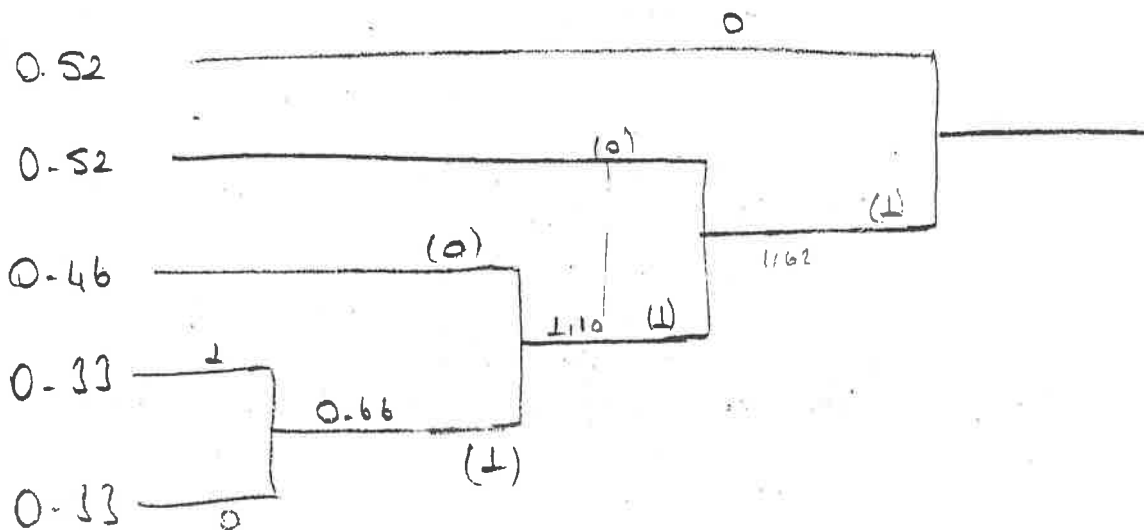


8) le-c<sub>7</sub>, odsar<sub>7</sub>



11015200

Sukistirdi,,



NO1 Eşitlik durumunda yukarıdaki 1 asapıddkını '0' aldım. Diğer durumda büyük olanı '1', küçük olanı '0' aldım.

(a) = 0	Bit Uzunluğu = 1
(c) = 10	" " = 2
(b) = 110	" " = 3
(d) = 1111	" " = 4
(f) = 1110	" " = 4

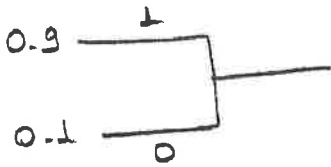
Bit Uzunluğu x Olasılık

$$(1 \times 0.52) + (2 \times 0.32) + (3 \times 0.46) + (4 \times 0.33) + (4 \times 0.33) = 5.58$$

Ortalama kod sayısı

ÖR

$P(a) = 0.9$   $P(b) = 0.1$   $\{a, b\}$  entropi değerleri verilmiştir.



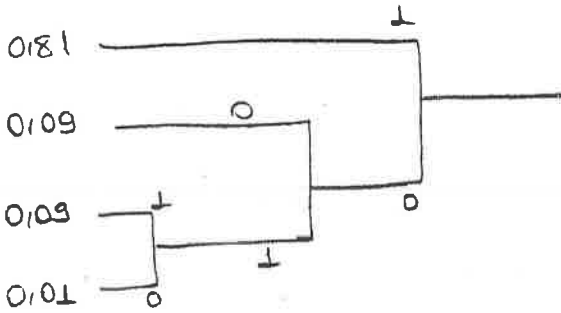
(a) = 1  
(b) = 0

$$(1 \times 0.9) + (1 \times 0.1) = 1$$

Ortalama kod sayısı

Örneğin kaynak sembolleri  $\{aa, ab, ba, bb\}$  olsun.

$$\begin{aligned} P(aa) &= P(a) * P(a) = (0.9) \times (0.9) = 0.81 \\ P(ab) &= P(a) * P(b) = (0.9) \times (0.1) = 0.09 \\ P(ba) &= P(b) * P(a) = (0.1) \times (0.9) = 0.09 \\ P(bb) &= P(b) * P(b) = (0.1) \times (0.1) = 0.01 \end{aligned}$$



$P(aa) = 1$   
 $P(ab) = 00$   
 $P(ba) = 011$   
 $P(bb) = 010$

$$1 \times (0.81) + 2 \times (0.09) + 3 \times (0.09) + 3 \times (0.01)$$

$$= 1.29/2 \rightarrow (ab, ba, aa, bb) \text{ olduğu için 2'ye bölceğiz.}$$

$$= 0.645$$

\* Eşitlik durumunda yukarıya 1 asayıp 0

Büyükçe "1" küçükçe "0"

Bit Uzunluğu x Olasılık



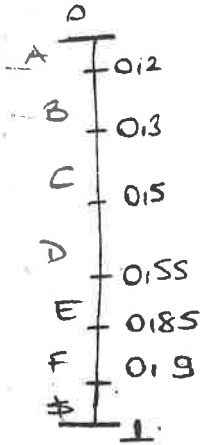
Semboller = { A, B, C, D, E, F, \$ }

Giris Dizisi = CAE\$

- Semboldü sırasına göre diz. Aralıkları olasılıklarını toplayarak belirle.

$$\Delta = 1 - 0 = 1$$

- A) = 0,12  
B) = 0,1  
C) = 0,2  
D) = 0,05  
E) = 0,3  
F) = 0,05  
(\$ ) = 0,1



$$\begin{aligned} 0 + \Delta \cdot P(A) &= 0 + 1 \cdot 0,12 = \underline{0,12} \\ 0,12 + \Delta \cdot P(B) &= 0,12 + 1 \cdot 0,1 = \underline{0,3} \\ 0,3 + \Delta \cdot P(C) &= 0,3 + 1 \cdot 0,2 = \underline{0,5} \\ 0,5 + \Delta \cdot P(D) &= 0,5 + 1 \cdot 0,05 = \underline{0,55} \\ 0,55 + \Delta \cdot P(E) &= 0,55 + 1 \cdot 0,3 = \underline{0,85} \\ 0,85 + \Delta \cdot P(F) &= 0,85 + 1 \cdot 0,05 = \underline{0,9} \\ 0,9 + \Delta \cdot P(\$) &= 0,9 + 1 \cdot 0,1 = \underline{1} \end{aligned}$$

Kimülatif olasılık toplamı,

- için

$$\Delta C = 0,5 - 0,3 = 0,2$$

$$\begin{aligned} 0,3 + \Delta \cdot P(A) &= 0,3 + 0,2 \cdot 0,12 = 0,34 \\ 0,34 + \Delta \cdot P(B) &= 0,34 + 0,2 \cdot 0,1 = 0,36 \\ 0,36 + \Delta \cdot P(C) &= 0,36 + 0,2 \cdot 0,2 = 0,4 \\ 0,4 + \Delta \cdot P(D) &= 0,4 + 0,2 \cdot 0,05 = 0,41 \\ 0,41 + \Delta \cdot P(E) &= 0,41 + 0,2 \cdot 0,3 = 0,47 \\ 0,47 + \Delta \cdot P(F) &= 0,47 + 0,2 \cdot 0,05 = 0,48 \\ 0,48 + \Delta \cdot P(\$) &= 0,48 + 0,2 \cdot 0,1 = 0,5 \end{aligned}$$

icin (Yine bir öncekine göre)

$$E = 0,334 - 0,322 = 0,012$$

$$\begin{aligned} 0,5 + \Delta \cdot P(A) &= 0,5 + 0,012 \cdot 0,12 = 0,5244 \\ 0,5244 + \Delta \cdot P(B) &= 0,5244 + 0,012 \cdot 0,1 = 0,5256 \\ 0,5256 + \Delta \cdot P(C) &= 0,5256 + 0,012 \cdot 0,2 = 0,528 \\ 0,528 + \Delta \cdot P(D) &= 0,528 + 0,012 \cdot 0,05 = 0,5286 \\ 0,5286 + \Delta \cdot P(E) &= 0,5286 + 0,012 \cdot 0,3 = 0,5322 \\ 0,5322 + \Delta \cdot P(F) &= 0,5322 + 0,012 \cdot 0,05 = 0,5328 \\ 0,5328 + \Delta \cdot P(\$) &= 0,5328 + 0,012 \cdot 0,1 = 0,534 \end{aligned}$$

Aralıklık herhangî bir sayı kod alınarak olur.

$$\begin{aligned} 0,333 \times 2 &= 0,666 \\ 0,666 \times 2 &= 1,332 \\ 0,332 \times 2 &= 0,664 \\ 0,664 \times 2 &= 0,128 \\ 0,128 \times 2 &= 0,256 \\ 0,256 \times 2 &= 0,512 \\ 0,512 \times 2 &= 1,024 \\ 0,24 \times 2 &= 0,48 \end{aligned}$$

01000010 //

A için (Bir önceki işleme göre hesapla)

$$\Delta A = 0,34 - 0,3 = 0,04$$

$$\begin{aligned} 0,3 + \Delta \cdot P(A) &= 0,3 + 0,04 \cdot 0,12 = 0,308 \\ 0,308 + \Delta \cdot P(B) &= 0,308 + 0,04 \cdot 0,1 = 0,312 \\ 0,312 + \Delta \cdot P(C) &= 0,312 + 0,04 \cdot 0,2 = 0,32 \\ 0,32 + \Delta \cdot P(D) &= 0,32 + 0,04 \cdot 0,05 = 0,322 \\ 0,322 + \Delta \cdot P(E) &= 0,322 + 0,04 \cdot 0,3 = 0,334 \\ 0,334 + \Delta \cdot P(F) &= 0,334 + 0,04 \cdot 0,05 = 0,336 \\ 0,336 + \Delta \cdot P(\$) &= 0,336 + 0,04 \cdot 0,1 = 0,34 \end{aligned}$$

\$ için (Yine bir öncekine göre)

$$\Delta \$ = 0,334 - 0,3328 = 0,0012$$

$$\begin{aligned} 0,3328 + \Delta \cdot P(A) &= 0,3328 + 0,0012 \cdot 0,12 = 0,33304 \\ 0,33304 + \Delta \cdot P(B) &= 0,33304 + 0,0012 \cdot 0,1 = 0,33316 \\ 0,33316 + \Delta \cdot P(C) &= 0,33316 + 0,0012 \cdot 0,2 = 0,3334 \\ 0,3334 + \Delta \cdot P(D) &= 0,3334 + 0,0012 \cdot 0,05 = 0,3336 \\ 0,3336 + \Delta \cdot P(E) &= 0,3336 + 0,0012 \cdot 0,3 = 0,33382 \\ 0,33382 + \Delta \cdot P(F) &= 0,33382 + 0,0012 \cdot 0,05 = 0,33388 \\ 0,33388 + \Delta \cdot P(\$) &= 0,33388 + 0,0012 \cdot 0,1 = 0,334 \end{aligned}$$

Örn

Bir ham 30 frame (saniyede 30 cerçeve) video 640 x 480 çöz. ve 24 bit renk derinliği için

$$30 \times 640 \times 480 \times 3 = 27.6480 \text{ mbyte gerektirir.}$$

Çari Seviyeye İmge Sıkıştırma

İmge  $\Rightarrow$  1 2 4 10 128 205 210 201 195 120 50 30 40

Fark  $\Rightarrow$  1 1 2 6 118 77 5 -9 -6 -75 -70 -20 10  
1 2 4 10 128 205 210 201 195 120 50 30 40

RUN LENGTH CODING

0	1	2	3	4	5	6	7	8	9	10	11
0	0	0	0	0	1	1	1	1	1	1	1
1	1	0	0	0	0	1	1	1	0	0	0

(5081403130)

L277

abc oac da abc da bbbab

abc sonra  $\uparrow$  surda tekrar etmiş

(Amaç en çok tekrar eden stringi yakalamak ve en uzun string olacak!)

n. adım  $\rightarrow$  abc oac da abc da bbbab

$\leftarrow 8, 3, c(d) \rightarrow$

8'inci adımda  
sonra bir daha abc  
tekrar etmiş

abc uzunluğu  
3

ikinci abc'de  
c'den sonra d geldiği için

n+1. adım  $\rightarrow$  abc oac da abc da bbbab

$\leftarrow 8, 2, c(b) \rightarrow$

NEC / LZW

3 imge tipini destekler.

→ Gerçek renk, → Gri Seviye, → Palet Tabanlı (8 bit)

Alfa kanalını destekler.

Gecici kodlama ve kod çözümü destekler.

ALF ⇒ LZW kayıpsız sıkıştırmasına dayalıdır.

Sadece 8 bit (256) renkli imgeyi destekler.

Her imge kendi renk tablosuna sahip olabilir.

Gecirgen katmanı sahip ve basit animasyon fonk. destekler.

Gecirli kodlama ve kod çözümü destekler.

PEGA ⇒ 1992'de önerilen uluslararası bir standarttır.

Her renkli hemde gri imgeler için çalışır. 24 bit renkli imgeyi hedefler.

Ardışık kodlamayı destekler.

Kayıpsız Kodlama (Fark Kodlama)

ÖRNEK

1 3 4 5 3 2 1 0 3 4 → soruda verilen.

1 2 1 1 -2 -1 -1 -1 3 1 → Fark kodlaması FFS

1 3 4 5 3 2 1 0 3 4 → Toplama yaparak sıralamasını oluşturduk.

ÖRNEK

Şurada fark kodlamasını bulduğumuz değerlerin entropisini bulalım.

$$P(1) = -0,14 \log_2^{0,14} = 0,52$$

$$P(2) = -0,1 \log_2^{0,1} = 0,33$$

$$P(-2) = -0,1 \log_2^{0,1} = 0,33$$

$$P(-1) = -0,3 \log_2^{0,3} = 0,52$$

$$P(3) = -0,1 \log_2^{0,1} = 0,33$$

$$0,52 + 0,33 + 0,33 + 0,52 + 0,33 = 2,03 //$$

## Kayıplı Kodlama (Lark Kodlama)

$$\text{Eğer } x > 0 \text{ ise } Q(x) = 1$$

$$\text{Eğer } x = 0 \text{ ise } Q(x) = 0$$

$$\text{Eğer } x < 0 \text{ ise } Q(x) = -1$$

1 3 4 5 3 2 1 0 3 4

0

1 1 1 1 -1 -1 -1 -1 1 1 } Fark kodlama (Kayıplı)

0

1 2 3 4 3 2 1 0 1 2 } Kayıp oldu

Ör

Fark kodlamasını bulduğumuz değerlerin entropisini alalım.

1	1	1	1	-1	-1	-1	-1	1	1
---	---	---	---	----	----	----	----	---	---

$$P(1) = -0,6 \log_2^{0,6} = 0,44$$

$$P(-1) = -0,4 \log_2^{0,4} = 0,52$$

$$0,44 + 0,52 = 0,96$$

## Resim Uzantıları

TIFF, GIF, JPEG, PNG, BMP, PPM, PNM

## Ses Uzantıları

WAV, MP3, MPEG

## Video Uzantıları

AVI, MOV

## Binary Image Compression

### Run Length Coding

1	1	0	0	0	0	0	0	0	0	1	1
1	1	1	1	0	0	0	0	0	0	0	0

2b8w2b4b8w

Code:

0282048

NPEG4 → Kayıpsız ses kodlama standardıdır.

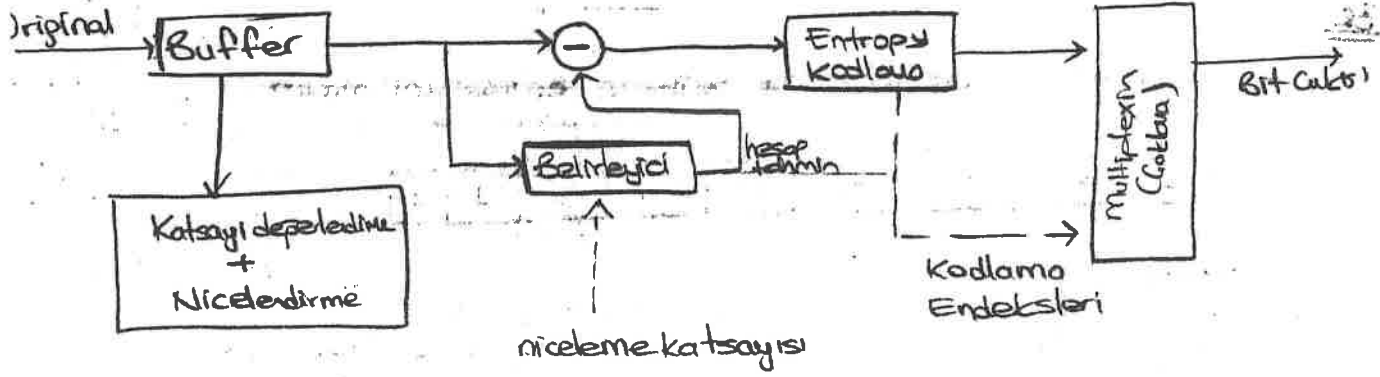
NPEG (moving Pictures Express Group)

Sesli ve görsel verilerin görüntü sıkıştırmasını tanımlar.

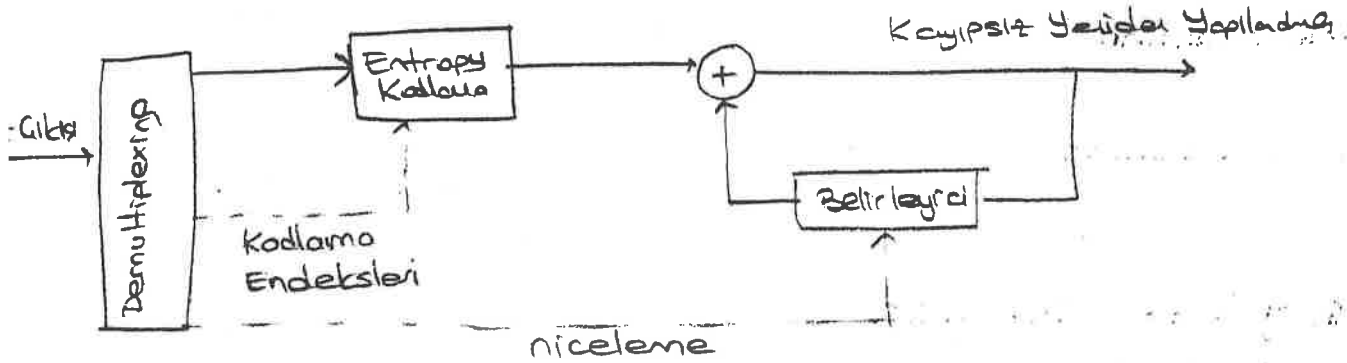
Web dosyaları / CD, TV, Ses kullanılır.

- Görüntü ve ses gibi ortam verilerini birlikte kodlar.
- Hataya karşı dirençlidir.
- Geliştirilmiş kodlama verimi mevcuttur.

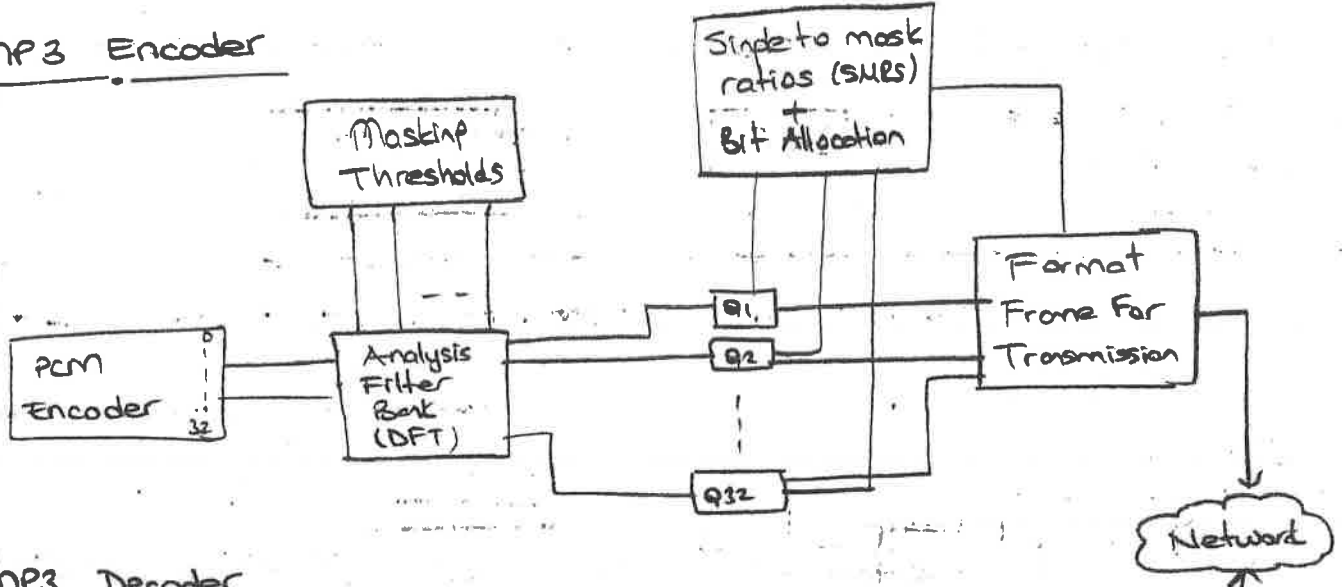
NPEG4 ALS Encoder



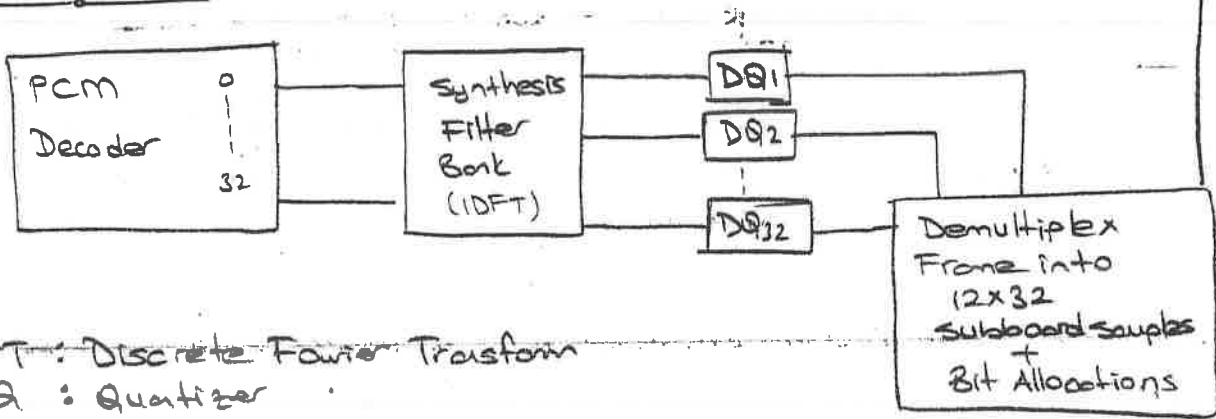
NPEG4 ALS Decoder



## MP3 Encoder



## MP3 Decoder



DFT : Discrete Fourier Transform

Q : Quantizer

DQ : Dequantizer

IDFT : Inverse Discrete Fourier Transform

DFT => (Fourier Dönüşümü)

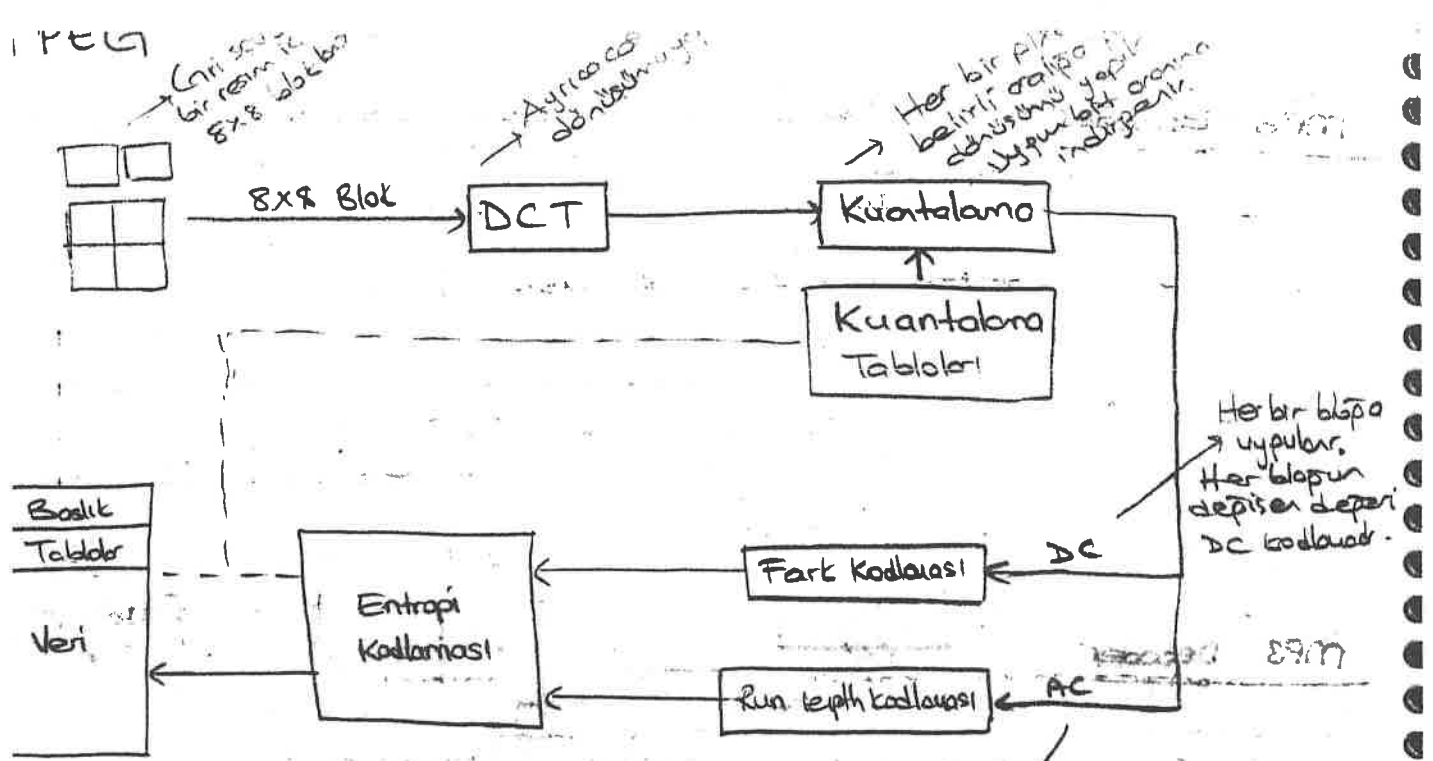
- Herhangi bir sinyal sonsuz sayıda sinüs dalgalarının birleşiminden oluşur. Fourier dönüşümü kullanarak frekans bileşenlerini hesaplayabiliriz. Herhangi bir sinyal için analizimizi genişletebiliriz.

DCT => (Ayrık Cosinus Dönüşümü)

- Sadece cosinüs bileşenler bulunur. Yani bir sinyali ifade etmek için sadece cos bileşenleri kullanırız. Bu dönüşümde negatif değerler bulunmaz. Elimizde reel bir bilgi varsa ayrık cos dönüşümü uygularız.

## Farkı

- DFT hem sinüs hem de cosinüs bileşenlerinden oluşur. Ayrık cosinüs ise sadece cosinüs bileşenlerinden oluşur.
- Fourier Dönüşümü negatif değerler içerebilirken ayrık cosinüs sadece pozitif değerler içerir. Negatif değer içermez.



- \* DCT (Ayrık cosinüs tabanlı)
- \* 24 bit RGB görüntüler ile çalışır.
- \* Renk bilgisi RGB ise YUV'a çevirme
- \* Örnekleme hızı azaltma

DC'den sonra yapılır. Değerler DC değerleri AC değerleridir.

Adımlar:

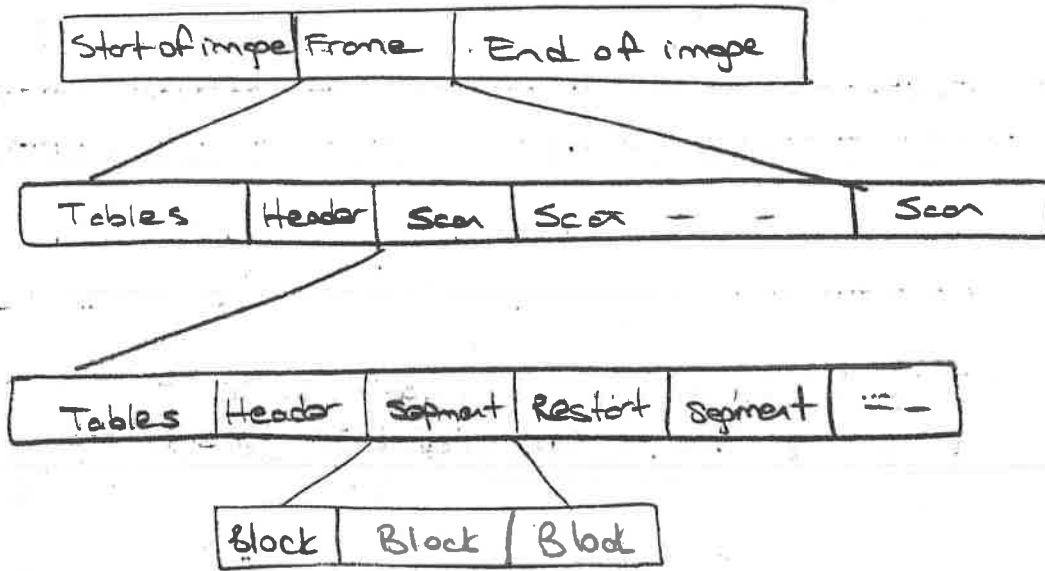
- Öncelikle resim 8x8 bloklara bölünür.
- Ayrıca cosinüs dönüşümü alınır (DCT)
  - \* Amaç olarak frekansa yüksek frekansa sıralamak
  - \* Veri daha az bit ile ifade edilecek hale gelir
- Daha sonra kuantalama işlemi ile uygun bit oranına indirgenir.
- Sonuç zikzak tarama düzeyinde RLE uygulanır ve ardından Huffman kullanılarak kayıpsız sıkıştırılır.

DC : Farklı detayları farklı katsayılarla göstermek için DC katsayısı yüksek seviyede kuantalanmalıdır. Aynı tıyı daha iyi göstermek için.

AC : AC katsayılarında kuantalama işlemi arttıkça görüntü kaba ve büyük görünür.

JPEG 24 bit renk derinliği sunar.  
- GIF ise 256 renk derinliği sunar.

### JPEG CERCİVE YAPISI



### Gri Tonlamalı Resim

Genellikle her pixel için 256 seviyelerini kullanır.

### Binary Resim

Sadece 1 ve 0 değerlerini alır. Görüntü analizi ve nesne algılamada uygulamalarda oldukça önemlidir.

### Digital Video

Digital video 3D fonksiyonlarının sayısal fonksiyonlarıdır. PAL'da YUV kullanılır. PAL; bir analog video standarttır ve digital videolar için kullanılır.

Y parlaklık bileşeni:

$$Y = 0.299R + 0.587G + 0.114B \quad \left. \vphantom{Y = 0.299R + 0.587G + 0.114B} \right\} \begin{array}{l} \text{U ve V renk} \\ \text{bileşeni} \end{array}$$

$$\left\{ \begin{array}{l} U = B - Y \\ V = R - Y \end{array} \right.$$



## Dijital Ses

Duyduğumuz ses 44-KHz'dir. 16 bittir. 44 KHz örnekleme frekansıdır. Müzik, konuşmadan daha yüksek bir frekansa sahiptir. 8 KHz örnekleme telefon konuşmaları için yeterince yoldur. 16 bit demek 16 bit örnek anlamındadır. Dijital ses, bir kanaldan daha fazlasına sahip olabilir.

## OpenCV

OpenCV görüntü işleme ve bilgisayarlı grafik için kullanılan bir C kütüphanesidir.

- OpenCV'nin windows, linux, mac versiyonları vardır.
- OpenCV matris işlemleri ve görüntü işleme için basit veri yapılarına sahiptir.
- Görüntüleme için bazı fonksiyonları ve resim yakalama gibi işlemler barındırır.
- Optimize ve hızlıdır. iyi bir kütüphanedir.
- Dokümantasyonu kötü ve hiç koruması yoktur.

## Zip2op Kodlama

- AC kodlamayı yok edip sadece tek bir satır haline getirip DC kodlama yapmak için kullanılır. Yani görüntüyü tek bayuta indirgeyerek yazar.
- Yüksek frekanslı bileşenler en uc noktadır. Zip2op taraması ile düşük frekansa yakın yüksek frekansa doğru bu frekansları sıralarız. Zip2op taramadan sonra yüksek frekanslı bileşenler atılır bazı ayrıntılar bulanık görünür.
- Örneğin kumlar desenlerini birbirinden ayırt etmek için ayırık cos dönüşümü alınıp, zip2op taraması ile tek bayuta indirip entropisine bakılır.

## RLE Kodlama

- Birbirini takip eden aynı karakterlerin hepsini tutmak yerine sadece karakteri ve kaç defa tekrar ettiğini tutma mantığına dayanır.
- Hızlıdır, kolay uygulanabilir.
- BMP, TIFF dosyaları tarafından desteklenir.
- Ancak, yüksek sıkıştırma oranlarına sahip değildir.

Örnek 1 { a a a a a a a a a a b b b b b b b b c c c c c c d d d d d d d d } bu diziyi RLE kodlamayla sıkıştırınız.

- Karakterin tekrar etme sıklığı ve karakteri yazıyoruz.

{ 1 1 a, 8 b, 6 c, 8 d }

Örnek 2 { 2 2 2 3 2 4 2 4 2 4 2 4 2 5 2 6 2 6 2 6 2 6 2 5 2 4 } bu diziyi RLE kodlamaya uygulayınız.

- Eğer 3'ten fazla tekrar etmişse özel bir karakter olan 'FF' konulur.

{ 2 2 2 3 FF 2 4 0 5 2 5 FF 2 6 0 4 2 5 2 4 }

Özel karakter      Tekrar eden karakter      Tekrar etme sıklığı

Örnek 3 300 x 400 büyüklüğünde renkli, 8 bit derinliğinde bir resmi kaç KB ile ifade ederiz?

$$300 \times 400 = 120000$$

renkli (rgb) olduğu için  $120000 \times 3 = 360000$

8 bit derinlik dediği için  $360000 \times 8 \text{ bit}$

$$1 \text{ byte} = 8 \text{ bit}$$

$$\frac{360.000 \times 8}{8} = 360.000 \text{ byte}$$

$$1 \text{ KB} = 1024 \text{ Byte}$$

$$\frac{360000}{1024} = 351.56 \text{ KB,,}$$

5 saniyelik, bir video'da saniyede 30 frame geçiyorsa  
360000 byte'ı bulmuskuk.

1 kare: 360000

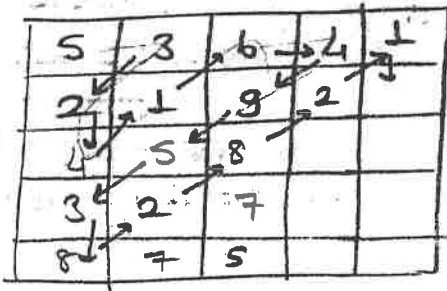
1 sn'de 1080000

5 sn'de x

$$x = 30 \times 360000 \\ = 10800000 \text{ byte}$$

$$x = 54000000 \text{ byte}$$

ŞİZİNG

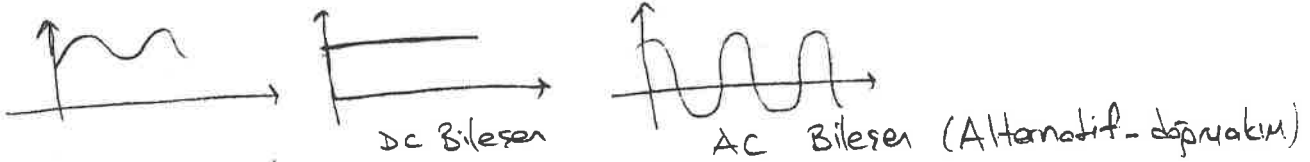
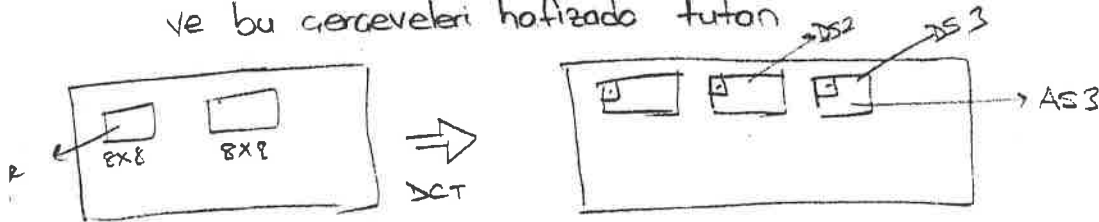


5 3 2 4 1 6 4 9 5 3 8 2 8 2 1

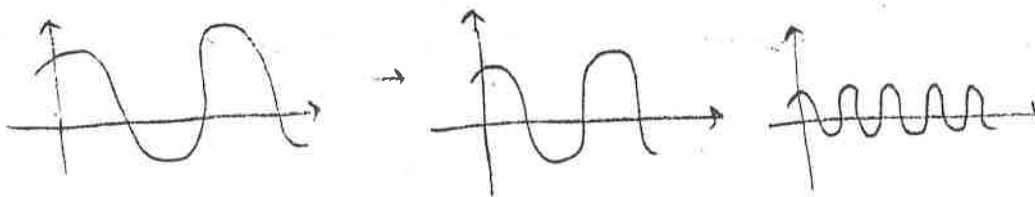
## Multimedia Systems

Frame  $\Rightarrow$  Bir imge karesi.

Video  $\Rightarrow$  Beğenimiz 24 Gerçeveden fazla algalayamaz.  
Küçük farklarla ard arda bu gerçevelerin gelmesi  
ve bu gerçeveleri hafızada tutan



2'ye parçalayız.

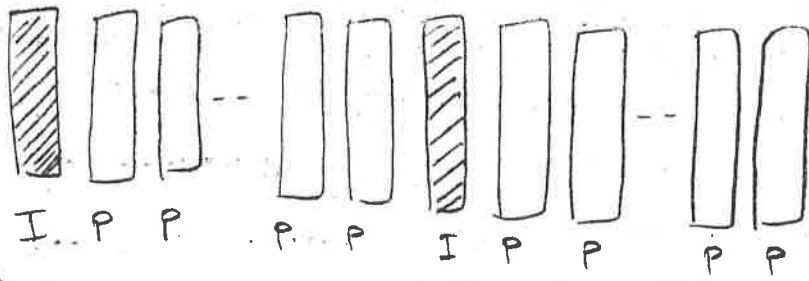


Şekil farklı  
değerler gittikçe  
genlikleri azalır.

- DS'ler fark kodlaması

AS'ler Huffman kodlamasına tabi tutulur.

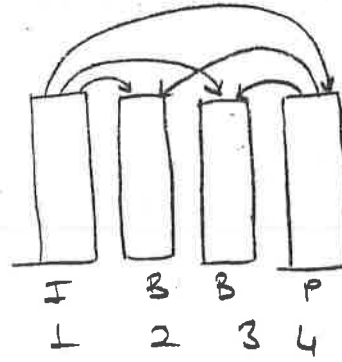
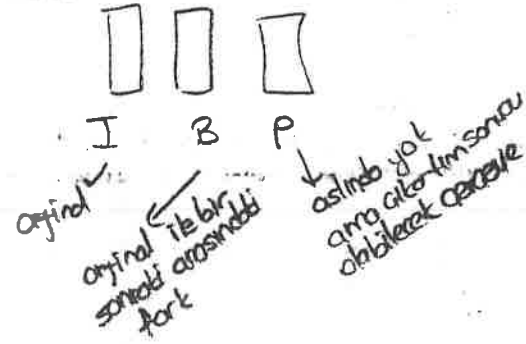
## GOP



GOP

GOP

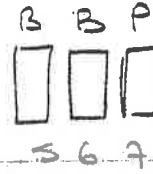
Bir süre sonra hatadan dolayı beyin alır, bu yüzden belli bir tetrardan sonra orijinal çerçeveye tekrar döndürülür.



Her 8 çerçevenin gösterilme seti

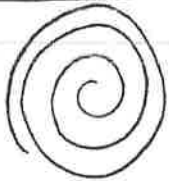
Encoding  
= 1 4 2 3 7 6 5

Gösterimi  
= 1 2 3 4 5 6 7



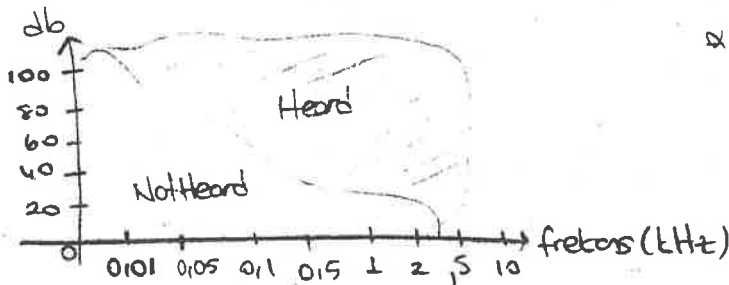
devam eder

## MUSIC CODING



### GENELİK HASSASİYETİ (AMPLITUDE SENSITİVE)

\* max işitebileceğimiz 22 kHz'dir.

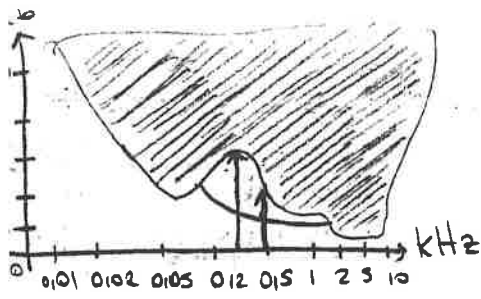


\*  $db = 20 \cdot \log V$   
Ses perlipinin logaritması  
alın 20 ile çarparsak  
db'yi buluruz.

Telefondaki konuşma frekans aralığı  
(500 - 2500 Hz) arasındadır.

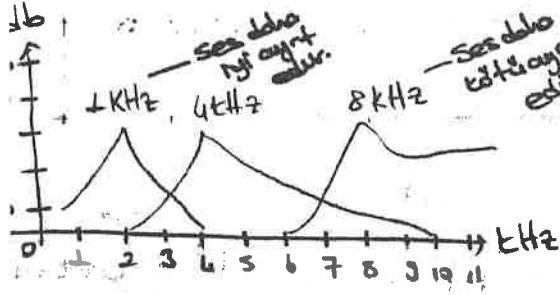
\* 4-8 kHz arası bilgilerimiz varsa konuşmazdakinin  
hislerini, duyuplarını hissedebiliriz.

\* Kulakın hassasiyeti frekansa bağlıdır.



- Genliği yüksek olan altındaki maskeler
- Yüksek deni alabiliriz, alıyamadığımızı bilgisi saklamaya gerek yoktur.

\* Duyulabilirliği yüksek seslerin varlığı nedeniyle bozuluyor.



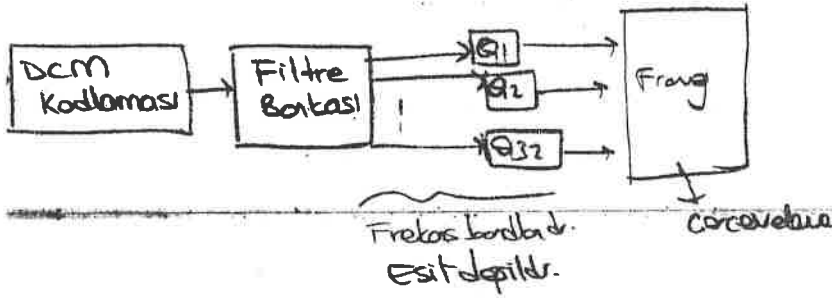
\* 1 - 4 - 8 kHz'e göre duyulabilir frekans aralıklarıdır.

\* En yüksek seviyede deni diğer, altındaki maskelerdir.

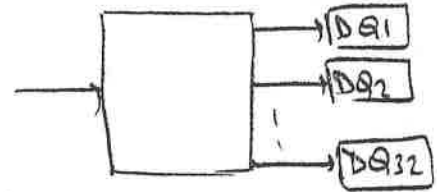
\* Genliği en yüksek deni diğer, diğerlerini tutmasında olur.

- Düşük frekanslı ses daha iyi duyulur.

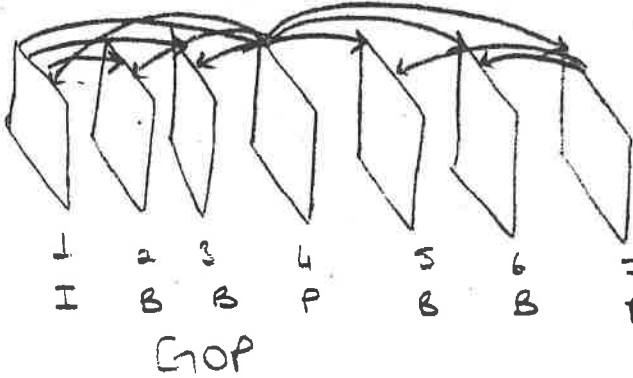
### Kodlama Kısım



### Decoder Kısım



### MULTIMEDIA SYSTEMS



### Encoding Order

1 4 2 3 7 5 6

### Decoding Order

1 4 2 3 7 5 6

### Display Order

1 2 3 4 5 6 7

I Frame : Hareketsiz görüntü olarak kodlanır.  
• GOP başlığı için veya sahne değişimliklerinde kullanılır.

P Frame : Önceki karelere dayanan öngörü tabanlı kodlama

B Frame : Hem önceki hem sonraki çerçevelere öngörüleri daha fazla esneklik.

- Daha iyi tahmin

- Gelecekteki çerçeveler için referans olarak kullanılmak.

Vizelerde ve Finalde Çıkan Konular

\* 5 dk, 44 KHz, stereo, 16 bit imge kaç byte?

$$\frac{5 \times 60 \times 44.000 \times 2 \times 16}{8} = 52.800.000 \text{ byte}$$

\* 5 dk, 16 KHz, mono, 8 bit

$$\frac{5 \times 60 \times 16.000 \times 1 \times 8}{8} = 4.800.000 \text{ byte}$$

\* 512 x 512 siyah-beyaz

$$\frac{512 \times 512}{8} = 32768 \text{ byte}$$

\* 512 x 512 8 bit gri

$$\frac{512 \times 512 \times 8}{8} = 262.144 \text{ byte}$$

\* 512 x 512 16 bit gri

$$\frac{512 \times 512 \times 16}{8} = 524.288 \text{ byte}$$

\* 512 x 512 8 bit renkli

$$\frac{512 \times 512 \times 8 \times 3}{8} = 786.432 \text{ byte}$$

\* 512 x 512 60 dk 8 bit renkli 30 fps

$$\frac{512 \times 512 \times 60 \times 60 \times 8 \times 3 \times 30}{8} = 8.643.4636 \times 10^3$$

### Vizelerde Çıkan Konular (2018)

- Huffman Kodlama
- Entropi Soruları
- Yukarıdaki soru
- Fork Kodlaması
- Run Length Coding

### Finalde Çıkan Konular (2018)

- Yukarıdaki soru örneği
- Frekans / Genlik oranları
- Frekans maskelene
- MP3 Diagrafları
- Genlik Hassasiyeti
- ZipZap Kodlama
- DCT nedir?
- Entropi
- Fork Kodlaması

Bu konulardan soru geldi.