

# VERİ SIKIŞTIRMA

## Görüntü Sıkıştırma Bölüm 7

Prof.Dr. Banu DİRİ

## Görüntü sıkıştırma

- **Kayıpsız**

Kayıpsız sıkıştırma genellikle pikseller arası korelasyonun düşük olduğu şekil gibi görüntülerde,

- **Kayıplı**

Kayıplı sıkıştırma ise fotoğraf görüntüleri gibi karmaşık ve yüksek korelasyona sahip görüntülerde kullanılır.

**Kayıpsız görüntü sıkıştırma**, olasılık veya sözlük tabanlı bir yöntem ile yapılabileceği gibi, ikisini bir arada kullanan bir yöntem ile de yapılabilir.

- PNG görüntü sıkıştırma yöntemi, LZ77 ve Huffman sıkıştırma algoritmalarını bir arada kullanan, DEFLATE sıkıştırma yöntemini kullanır.
- JPEG-LS, JBIG gibi özellikle kayıpsız görüntü sıkıştırmak için tasarlanmış yöntemler de vardır.

## ***RLE (Run Length Encoding)***

Her tür veri için kullanılır olsa da, aynı sembolün ardışık olarak çok defa tekrar etmesi durumunda iyi bir sıkıştırma oranı verir. Genellikle görüntü sıkıştırmada kullanılır.

BMP, PCX ve TIFF görüntü dosya formatları, RLE ile sıkıştırma yapar.

RLE'nin temel mantığı, bir değerin ardışık olarak çok sayıda tekrar etmesi durumunda, o değeri bir defa kodlayıp, ardından tekrar sayısını kodlamaktır.

**Örneğin** 1-bit renk derinliğine sahip bir görüntünün (2 renge sahip: sadece siyah ve beyaz) bir satırındaki 640 pikselin renkleri de sırayla; 200 adet beyaz, 240 adet siyah ve sonra 200 adet beyaz şeklinde ise, o satır RLE ile (200, 240, 200) şeklinde kodlanır.

Burada decoder ilk pikselin ne renk olduğunu bilmelidir. İlk pikselin rengi kodlamanın başında belirtilir veya ilk piksel her zaman beyaz kabul edilir

Fax makinelerinde kullanılan CCITT T4 (Group 3) standardı, RLE ve Huffman kodlamasını bir arada kullanır.

- İlk piksel her zaman beyaz kabul edilir. Tekrar sayıları çok fazla farklı sayıda olasılığa sahip olduğu için, 64'ten büyük olan tekrar sayıları önce 64'e bölünür, sonra bölüm ve kalan farklı Huffman Tabloları kullanılarak kodlanır.
- Bu kodlama *Modified Huffman (MH)* veya *Group 3 One-Dimensional (G31D)* olarak bilinir.

CCITT T4 standardı, IBM tarafından 1979 yılında geliştirilen MMR (Modified Modified READ) sıkıştırmasını da kullanabilir. *Group 3 Two-Dimensional (G32D)* olarak bilinir.

- Bu yöntemde bir önceki satır dikkate alınarak daha yüksek sıkıştırma oranlarına ulaşılır. MMR'nin sıkıştırma performansı, daha sonra geliştirilen JBIG standardının gerisinde kalmıştır.

## Zip ve Gzip

- Her bloğun Huffman ağacı ayrıdır ve her bloğun başında sıkıştırılarak saklanır
- LZ77'nin bir türevi ile statik Huffman kodlamanın birleşimidir
- 32KB'lık kayan bir sözlük ve 258byte'lık kaynak kod alanı kullanır. String sözlükte yoksa, olduğu gibi kodlanır
- Giriş verisi, kodlayıcı tarafından bloklara ayrılır. Blok uzunlukları keyfidir

## PKZip (Phil Katz) UnZip, Arc, Lite

- PK programları birkaç sıkıştırma algoritmasını içerisinde barındırır
- PKArc → LZW ve Huffman Coding
- Dinamik LZW, RLE
- LZ77 ve Shannon Fano

# JPEG-LS (Lossless JPEG) Joint Photographic Experts Group

- ISO/IEC tarafından geliştirilmiştir (International Organisation for Standardisation /International Electrotechnical Commission)
- JPEG-LS, Görüntüleri kayıpsız veya az kayıplı sıkıştırabilen bir algoritmadır (1998)
- Standartı "Hewlett-Packard Laboratories" tarafından geliştirilen LOCO-I (LOW COMplexity LOSSless COMpression for Images) algoritması oluşturmaktadır
- Hem renkli hem de gri-tonlamalı görüntülerin sıkıştırılmasında kullanılır
- Gri tonlamalı görüntülerde ne yapıyorsa, renkli görüntülerde de kırmızı, yeşil ve mavi renk paletleri için aynı mantığı kullanır
- Kodlayıcı görüntünün sol üst pikselinden başlayarak sağ alt pikseline kadar satır tarama düzeninde işlem yapar
- Bir pikselin değeri tahmin edilirken kodlayıcı ve çözücü de o pikselin üstündeki ve solundaki piksellerin değerleri mevcuttur
- JPEG-LS algoritması bu değerlerden sadece 4 tanesini kullanır; soldaki (W), üst soldaki (NW), üstteki (N) ve üst sağdaki (NE)

$$G = \begin{cases} \min(W, N) & \max(N, W) \leq NW \\ \max(W, N) & \min(N, W) < NW \\ W + N - NW & \text{diğer durumlarda} \end{cases}$$

Tahmin iki aşamada gerçekleşir. İlk aşamada her piksel değeri için aşağıdaki gibi tahmin yapılır:

İlk iki madde dikey ve yatay kenarları yakalar

- Örneğin, eğer  $N > W$  ise ve  $N \leq NW$  ise, bu yatay kenarı işaret eder ve  $W$  tahmin olarak kullanılır
- Son madde ise yatay kenarları yakalar

Elde edilen  $G$  tahmininden sonra ikinci bir geçiş ile, yapılan bu tahminin çevresindeki piksel çiftlerinin değerlerinin farklarına göre düzeltme yapılır

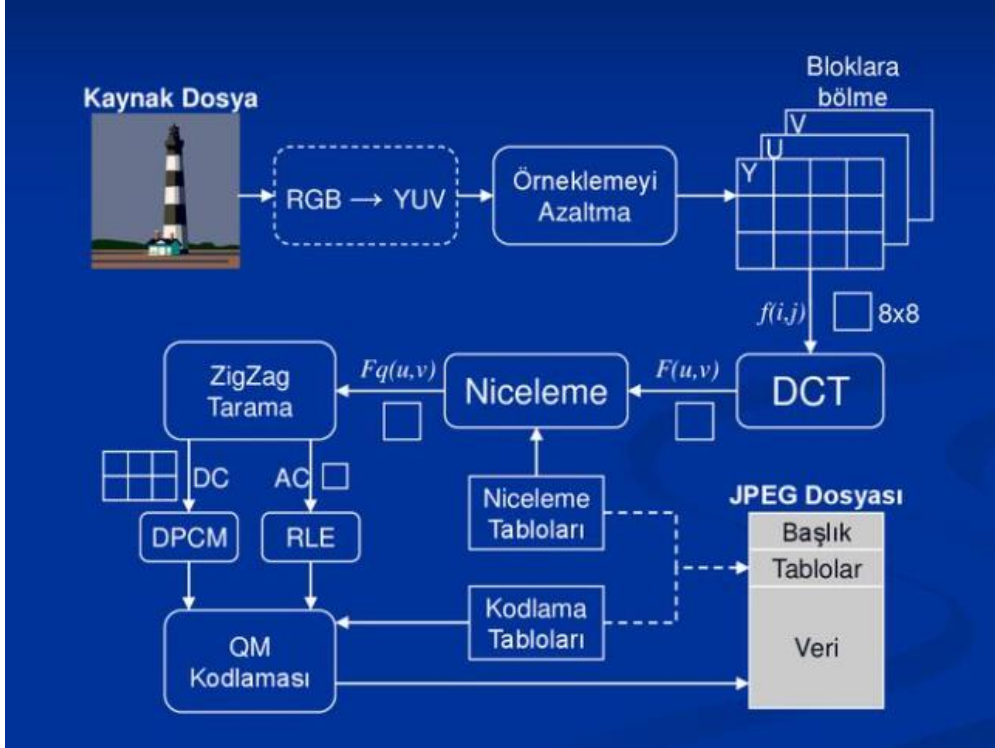
- Üç piksel çifti kullanılır;  $(NW, W)$ ,  $(NW, N)$  ve  $(N, NE)$
- İki bitişik pikselin değerlerinin farkına göre, her biri 9 grubun birine sınıflandırılır. Bu da toplamda  $9^3=729$  farklı durumu ortaya çıkarır. Her farklı durum, tahmini düzeltmek için kullanılan kendine ait düzeltme değerini ve o durum için daha önceki tahminlerin kaliteleri ile ilgili bilgiyi saklar. Algoritma ikinci geçişten sonra sonuç tahmini üretir ve gerçek değer ile arasındaki farkı kodlar.

# JPEG

- DCT (Ayrık Kosinüs Dönüşümü) tabanlı bir standarttır
- Sıkıştırmadaki kayıp oranı (sıkıştırma miktarı) kalite faktörünün (Q faktörü) değiştirilmesi ile ayarlanabilir
- 24-bit renk derinliğine sahip fotoğraflarda çok iyi sonuçlar verir (sayısal fotoğrafçılıkta kullanılır)
- JPEG, birçok görüntü sıkıştırma yönteminin bir araya getirilmesi ile oluşturulmuş bir standarttır
- Sıkıştırma ve açma süreleri tek algoritmaya sahip olan kayıpsız sıkıştırma yöntemlerine göre daha uzundur



# JPEG Mimarisi



- Renk bilgisi RGB türünde ifade edilmişse, YUV'a çevirme (isteğe bağlı)
- Örneklemeyi azaltma (down-sampling)
- Görüntüyü 8×8 büyüklüğünde bloklara bölme
- DCT kullanarak, piksel bilgisini zaman etki alanından frekans etki alanına çevirme
- DCT dönüşümü sonuç değerlerine nicelendirme
- Sonuç katsayıları köşegen tarama düzeninde RLE uygulama ve ardından QM kodlamasını kullanma

# Renk Dönüştürme

- RGB'den YUV'a dönüştürme JPEG sıkıştırması için gerekli değildir, ama bu dönüşüm sayesinde daha yüksek sıkıştırma oranlarına ulaşılır. R, G ve B bileşenleri arasındaki korelasyonu azalttığı için sıkıştırmanın etkinliği artar.
- YUV renk formatı renkleri, parlaklık (luminance) ve renklilik (chrominance) olarak ifade eder
- Y parlaklığı, U ve V değerlerinin kombinasyonları da renkliliği ifade eder
- RGB→YUV dönüşümü bir lineer dönüşümdür

$$Y = 0.299R + 0.587G + 0.114B$$

$$U = -0.169R - 0.331G + 0.500B$$

$$V = 0.500R - 0.149G - 0.081B$$

## Örnelemeyi Azaltma

- İnsan gözü parlaklığa daha çok, renkliliğe daha az hassastır. U ve V değerlerini bir miktar azaltmak, görüntüyü insan gözü için çok fazla farklılaştırmaz
- Bu sebeple 2×2 ebatlarında 4 piksellik bir blok için Y, U ve V değerlerinden 4'er adet (4:4:4) kullanmak yerine 4:2:2 veya 4:1:1 kullanabilir. Bu işleme down-sampling denir

## DCT (Discrete Cosine Transform)

- Görüntü 8×8 bloklara bölünür ve her blok diğerinden bağımsız olarak ayrık kosinüs dönüşümüne (DCT - Discrete Cosine Transform) tabi tutulur
- Sonuçta ortaya yine 64 katsayı çıkacaktır. Bunlardan ilkinde DC (ilk piksel - D(0,0)) ve diğerlerine de AC adı verilir.

- İnsan gözü yüksek frekanstaki değerleri algılayamaz
- Resimde sıkıştırma yapılacak ise yüksek frekanslar çıkarılır
- Resim 8x8 lik bloklara ayrılır (blok içerisinde renk değerleri 0-255 arasındadır)
- DCT (-128, 127) arasında çalıştığından bütün renk değerlerinden 128 çıkarılır (T matrisi)
- Her bloğa ayrı ayrı DCT uygulanır
- Düşük frekanslı değerler matrisin sol üst köşesinde, yüksek frekanslı değerler ise sağ alt köşede yer alır

$D = M^T T M$  (DCT katsayılarından oluşan Matris)

$$M_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & i = 0 \\ \sqrt{\frac{2}{N}} \cos \left[ \frac{(2j+1)i\pi}{2N} \right] & i > 0 \end{cases}$$

- D matrisine önceden hazırlanmış Quantization matrisi uygulanır

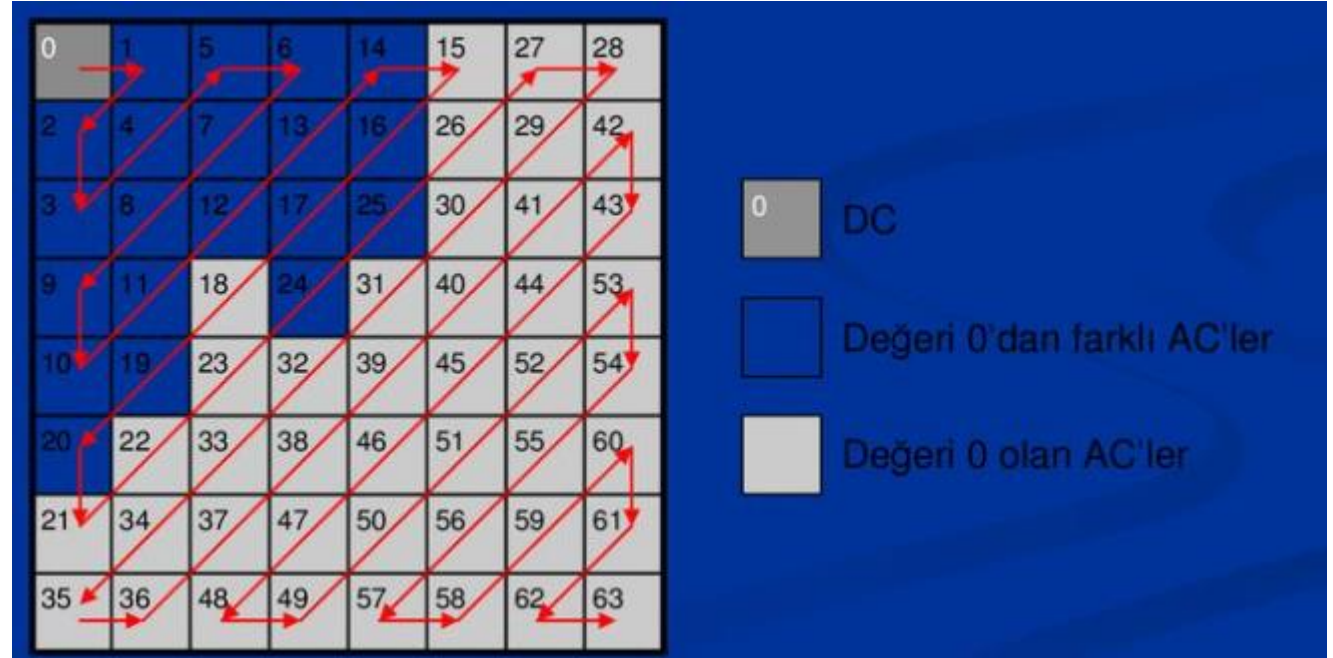
$$C_{i,j} = \text{round} \left[ \frac{D_{i,j}}{Q_{i,j}} \right]$$

8	6	5	8	12	20	26	31
6	6	7	10	13	29	30	28
7	7	8	12	20	29	35	28
7	9	11	15	26	44	40	31
9	11	19	28	34	55	52	39
12	18	28	32	41	52	57	46
25	32	39	44	52	61	60	51
36	46	48	49	56	50	52	50

75 kalite faktörü için parlaklık tablosu

- ❖ DCT dönüşümü sonrasında ortaya çıkan katsayılar genellikle ondalıklıdır, niceleme yöntemi ile tamsayıya dönüştürülür
- ❖ Nicemeleme yönteminde her katsayı 1 ile 255 arasında bir tamsayıya bölünür ve daha sonra en yakın tamsayıya yuvarlanır (bu esnada kayıp meydana gelir)
- ❖ Katsayıların hangi değere bölüneceği niceleme tabloları (8×8 büyüklüğünde bir nicemeleme matrisi) sayesinde belirlenir
- ❖ Parlaklık ve renklilik için farklı tablolar kullanılır.
- ❖ Görüntüdeki kayıp miktarını belirlemek için bir kalite çarpanı (quality factor) kullanılır
- ❖ Kullanılan kalite çarpanına göre tablolar değişiklik gösterir

- $C$  matrisinin sağ alt köşesi 0 ile doludur. Çünkü quantization işlemi yüksek frekansları yok eder
- $C$  matrisine zigzag coding uygulanarak, arkasından değişken uzunluklu kodlama kullanılarak sıkıştırılır
- Ters dönüşüm için  $R_{ij} = Q_{ij} * C_{ij}$
- $DCT^{-1}$  için  $N = \text{round}(M^T \times R \times M) + 128$



# JPEG2000

- Kayıpsız JPEG2000 ' Joint Photographic Experts Group tarafından geliştirilmiştir
- Wavelet teknolojisini temel alan en iyi kodlama algoritmasıdır
- JPEG standardının kısıtlamalarını gidermek ve düşük bit oranlarında yüksek kalitede görüntüler elde etmek amacıyla tasarlanmıştır
- Wavelet teknolojisinin avantajı sayesinde, DCT'de olduğu gibi sadece kayıplı sıkıştırma değil, kayıpsız sıkıştırma da yapabilmektedir
- JPEG2000, istemci/sunucu görüntü uygulamaları için ve sınırlı kaynağa sahip olan kablosuz cihazlar için ilave fonksiyonel özelliklere sahiptir

# GIF (Graphics Interchange Format)

- CompuServe tarafından geliştirilmiştir (1987)
- LZW tabanlı çalışan bir algoritmadır
- En fazla 8-bit renk derinliğine ( $2^8 = 256$  renk) sahip görüntülerde çalışır
- Renkli fotoğraf görüntülerinin sıkıştırılmasında yetersiz kalır
- Birkaç rengin çoğunlukta olduğu grafiksel gösterim, basit şekiller, vb. gibi görüntülerin kayıpsız olarak sıkıştırılmasında kullanılır
- 87a ve 89a diye iki versiyonu vardır. 89a, tek bir GIF dosya içerisinde birden çok GIF formatlı resim yerleştirilmesine ve anime edilmesine olanak tanır
- Katmanlı görüntü saklama özelliğine sahiptir
- Kullanım oranı düşüktür (patent koruması altında)
- Donanımdan bağımsız çalışır



## TIFF (Tagged-Image File Format)

- Aldus Corporation tarafından geliştirilmiştir, şu an telif hakkı Adobe Systems şirketine aittir
- 1988'de yayınlanan 5.0 sürümü ile RGB bitmap verilerini, LZW algoritması kullanarak sıkıştırır
- 1992'de yayınlanan 6.0 sürümü ile JPEG sıkıştırması da eklenmiştir
- TIFF-LZW'nin de patent sorunlarından dolayı, TIFF altında ZIP sıkıştırması geliştirilmiştir
- TIFF 24-bit renk derinliğini desteklemesi sayesinde fotoğrafların kayıpsız olarak sıkıştırılmasında kullanılır
- Çok renkli resim dosyalarında zaman, yerden daha önemli ise hızından dolayı tercih edilir
- JPEG göre daha az küçülme sağlamasına rağmen hızı ile açığı kapatır
- TIFF formatının GIF'ten temel farkı; hem 256 sıralı-renk görüntülerle, hem de 24-bit RGB (8-bit Red + 8-bit Green + 8-bit Blue) ile çalışabilmesidir. 24-bit renk derinliğini desteklemesi sayesinde fotoğrafların kayıpsız olarak sıkıştırılmasında kullanılır.



## PNG (Portable Network Graphics)

- GIF ve altyapısındaki LZW algoritmalarının, patent anlaşmaları ile koruma altında olmaları, ücretsiz olarak kullanılabilecek bir kayıpsız görüntü sıkıştırma standardı ihtiyacını gündeme getirmiştir. Thomas Boutell, Scott Elliott, Mark Adler, Tom Lane ve birçok başka girişimci tarafından geliştirilmiştir (1996)
- 48-bit gerçek renk, 16-bit gri tonlama desteği ve yüksek oranda sıkıştırma becerisi sayesinde GIF ve TIFF standartlarından daha iyidir
- Altyapısında ZLIB ve DEFLATE veri sıkıştırma teknikleri yer almaktadır