

BSM409 Görüntü İşleme

Bölüm 10 Görüntü Sıkıştırma

Dr. Öğr. Üyesi Caner ÖZCAN

Yaşam kısa ve bilgi sonsuzdur... Kısaltma gerekli bir kötülüktür ve kısaltıcının işi, özünde kötü olmasına rağmen, bir işin en iyisini yapmaktır ki bu hala hiç yoktan iyidir. ~Aldous Huxley

İçerik

8. Görüntü Sıkıştırma

- ▶ Temeller
- ▶ Temel Bazı Sıkıştırma Yöntemleri (Huffman Kodlama)
- ▶ *Sayısal Görüntü Damgalama (Watermarking)*

İlgili Veri Artıklığı

- ▶ Aynı bilginin iki farklı gösterimindeki bit sayıları b ve b' ile gösterilirse, b bitleri ile tanımlanan ifadenin ilgili veri artıklığı R şu şekilde yazılabilir.

$$R = 1 - 1/C$$

- ▶ C sıkıştırma oranı olarak ifade edilir ve şu şekilde tanımlanır:

$$C = b/b'$$

Örneğin $C = 10$ ise daha büyük gösterim, daha küçük gösterimdeki verinin her 1 biti için 10 veri bitine sahiptir. Buna uygun daha büyük gösterimdeki ilgili veri artıklığı, verinin %90'ının artık olduğunu gösteren 0.9 ($R=0.9$) olacaktır.

Neden Sıkıştırılmaya İhtiyaç Duyulur?

► Veri Depolama

► Veri İletimi

Sıkıştırılmayı Nasıl Tanımlarız?

► **Kodlama Artıklığı**

Birçok 2-D yeğinlik dizisi, yeğinlikleri göstermek için ihtiyaç duyulandan daha fazla bit içermektedir.

► **Uzamsal ve Zamansal Artıklık**

Birçok 2-D yeğinlik dizisi uzamsal ve video dizilerinde de zamansal olarak bilgi tekrarlanır.

► **İlişkisiz Bilgi**

Birçok 2-D yeğinlik dizisi insan görme sistemi tarafından göz ardı edilen görüntü bilgisini içerir. Bu bilgi kullanılmayacak olması dolayısıyla artıktır.

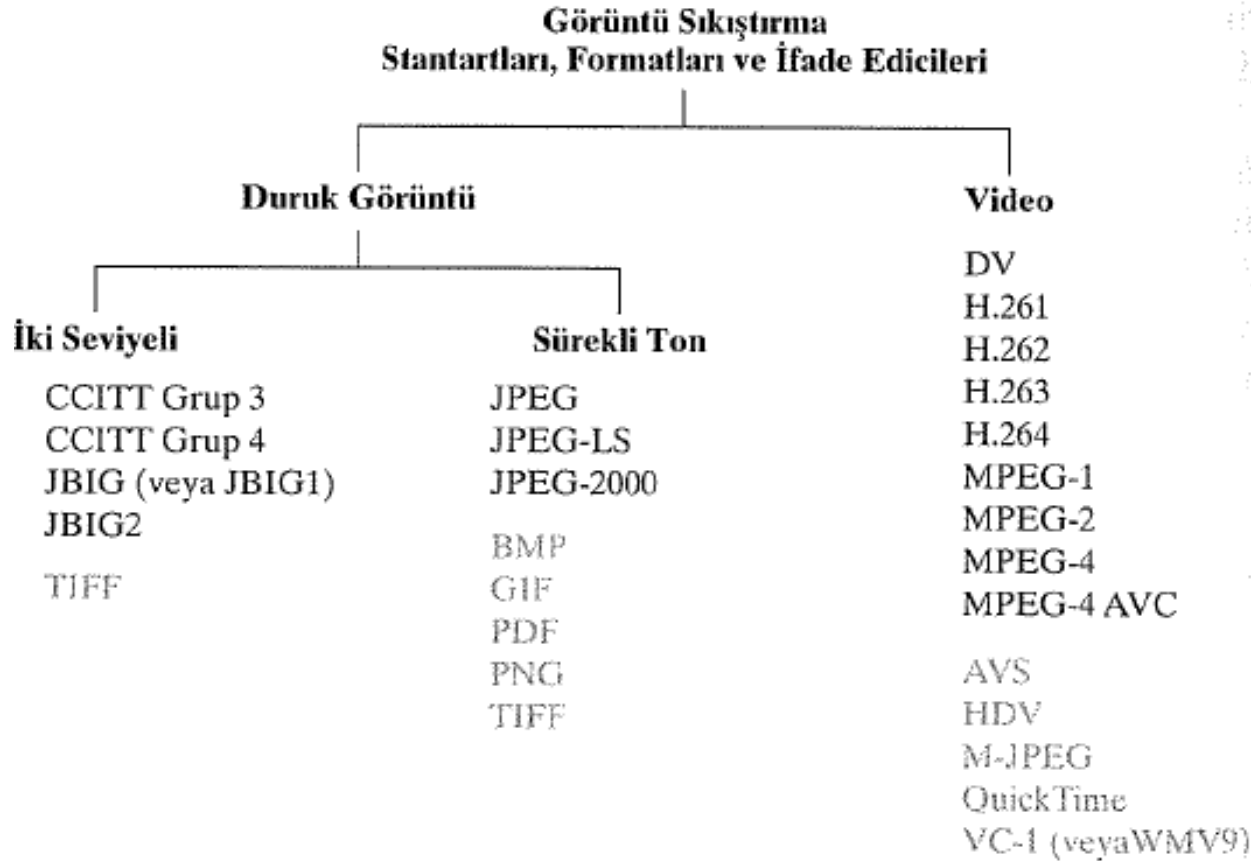
Artıklık Örnekleri



a b c

ŞEKİL 8.1 Bilgisayarla üretilmiş $256 \times 256 \times 8$ bit görüntüler ile (a) kodlama artıklığı, (b) uzamsal artıklık, (c) ilişkisiz bilgi. (Herbiri bir temel artıklığı göstermek için tasarlanmış, fakat diğerlerini de gösterebilirler.)

Görüntü Sıkıştırma Standartları



ŞEKİL 8.6

Bazı popüler görüntü sıkıştırma standartları, dosya formatları, ve ifade edicileri. Uluslararası onaylı satırlar siyah ve diğerleri de açık gri olarak gösterilmişlerdir.

İsim	Organizasyon	Tanım
<i>İki-seviye Duruk Görüntüler</i>		
CCITT Grup 3	ITU-T	Telefon hatları üzerinden iki seviyeli dokümanları göndermek amacıyla bir faks (FAX) yöntemi olarak tasarlanmıştır.
CCITT Grup 4	ITU-T.	Sadece 2-D katar-uzunluk kodlamasını destekleyen CCITT Grup 3 standardının basitleştirilmiş ve kullanışlı hale getirilmiş versiyonu.
JBIG veya JBIG1	ISO/IEC/ ITU-T	İki-seviye görüntülerin Kademeli, kayıpsız sıkıştırılması için bir <i>Birleşik İki-seviyeli Görüntü Uzmanları Grubu (Bi-level Image Expert Groups)</i> standardı. 6 bit/piksel'e kadar sürekli-ton görüntüler bir bit-düzlemi esasında kodlanabilirler [8.2.7]. İçerik duyarlı aritmetik kodlama [8.2.3] kullanılır ve görüntünün başlangıç düşük çözünürlüklü bir versiyonu ilave sıkıştırılmış veri ile kademeli olarak zenginleştirilir (iyileştirilir).
JBIG2	ISO/IEC/ ITU-T	Masaüstü, İnternet ve FAX uygulamaları için iki-seviye görüntüler için ara vermeden (kesintisiz) JBIG1'e devam etme. Kullanılan sıkıştırma yöntemi metin ve yarımton bölgeleri için içeriğe dayalı sözlük tabanlı yöntemler [8.2.6], ve diğer görüntü içeriği için Huffman [8.2.1] veya aritmetik kodlamadır [8.2.3].
<i>Sürekli-Ton Duruk Görüntüler</i>		
JPEG	ISO/IEC/ ITU-T	Görüntü veya fotoğraf kalitesi için bir <i>Birleşik Fotoğrafılgili Uzmanlar Grubu (Joint Photographic Experts Group)</i> . Bu standardın <i>kayıp esaslı kodlama sistemi</i> (en yaygın kullanılan şekli) 8×8 görüntü öbeklerine dayalı nicemlenmiş ayrık kosinüs dönüşümü (DCT) [8.2.8], Huffman [8.2.1], ve katar-uzunluk [8.2.5] kodlaması kullanır. Bu İnternet üzerinde görüntüleri sıkıştırma için en çok kullanılan yöntemlerden birisidir.
JPEG-LS	ISO/IEC/ ITU-T	Uyarlanır öngörüne [8.2.9], içerik modellemeye [8.2.3], ve Golomb kodlamaya [8.2.2] dayalı sürekli ton görüntüler için kayıpsıza-yakın kayıpsız bir kodlamadır.
JPEG-2000	ISO/IEC/ ITU-T	Fotoğraf görüntü kalitesinin artırılmış sıkıştırması için JPEG'e ara vermeden devam etme. Aritmetik kodlama [8.2.3] ve nicemlenmiş dalgacık dönüşümleri [8.2.10] kullanılır. Sıkıştırma kayıpsız veya kayıph olabilir.

TABLO 8.3
Uluslararası onaylı görüntü sıkıştırma standartları. Parantez içerisindeki sayılar bu bölümdeki kısımlara atıf yapmaktadır.

TABLO 8.3
(Devam ediyor)

İsim	Organizasyon	Tanım
<i>Video</i>		
DV	IEC	<i>Sayısal Video</i> . Ev ve yarı profesyonel video oluşturma uygulamalarına ve Elektronik haber toplama ve video kameraları gibi cihazlara ayrılmış bir video standardı. W Çerçeveler, JPEG'e benzer şekilde bir DCT-tabanlı yaklaşım [8.2.8] kullanılarak tamamlanmamış kurgu için bağımsız olarak sıkıştırılırlar.
H.261	ITU-T	ISDN (<i>integrated services digital network-toplu servis sayısal ağ</i>) hatları için iki-yollu bir videokonferans standardı. Bu standart sırasıyla CIF (<i>Common Intermediate Format-Ortak ara format</i>), ve QCIF (<i>Quarter CIF-dört parçalı CIF</i>) olarak adlandırılan girişimsiz-taramalı (non-interlaced) 352×288 ve 176×144 çözünürlükteki görüntüler destekler. Zamansal artıklığı azaltmak amacıyla çerçeveden çerçeveye öngörücü farkı [8.2.9] ile JPEG'e benzer DCT-tabanlı bir sıkıştırma yaklaşımı [8.2.8] kullanılır. Çerçeveler arasındaki hareketi dengeleme için blok-tabanlı bir tekaik kullanılır.
H.262	ITU-T	Aşağıda verilen MPEG-2'ye bakınız.
H.263	ITU-T	İlave çözünürlüklerle tipik kullanılan telefon modemleri (örneğin 28.8 Kb/s) için tasarlanmış H.261'in geliştirilmiş versiyonu: SQCIF (<i>Sub-Quarter CIF</i> 128×96), 4CIF (704×576), ve 16CIF (1408×512).
H.264	ITU-T	Video konferans, kesintisiz Internet akışı, ve televizyon yayını için H.261 - H.263'ün genişletilmiş hali. O çerçevelerdeki öngörücü farklarını [8.2.9], değişken blok boyutlu tamsayı dönüşümleri (DCT'den başka), ve içerik uyarlanabilir aritmetik kodlamayı destekler [8.2.3].
MPEG-1	ISO/IEC	Girişimsiz-taramalı video ile 1.5 Mb/s'ye kadar CD-ROM uygulamaları için bir <i>Hareketli Resimler Uzman Grubu (Motion Pictures Expert Group)</i> standardıdır. H.261'e benzerdir, fakat çerçeve öngörümleri önceki çerçeveye, sonraki çerçeveye veya herikisinin bir aradeğerlemesine dayalı olabilir. Hemen hemen bütün bilgisayar ve DVD oynatıcılar tarafından desteklenir.
MPEG-2	ISO/IEC	15 Mb/s transfer hızına sahip DVD'ler için tasarlanmış MPEG-1'in bir uzantısı. Girişimli taramalı video ve HDTV'yi destekler. Günümüzdeki en başarılı video standardıdır.
MPEG-4	ISO/IEC	Çerçeveler içerisinde değişken blok büyüklükleri ve öngörücü fark almasını [8.2.9] destekleyen MPEG-2'nin bir uzantısı.
MPEG-4 AVC	ISO/IEC	MPEG-4 Kısım 10 Gelişmiş Video Kodlama (Advanced Video Coding - AVC). H.264 yukarıdaki için eşdeğerdir.

TABLO 8.4
Tablo 8.3'de yer almayan popüler görüntü sıkıştırma standartları, dosya formatları, ve ifade edicileri.

İsim	Organizasyon	Tanım
<i>Süreklil-Ton Durak Görüntüler</i>		
BMP	Microsoft	<i>Windows Bitmap</i> . Temelde basit sıkıştırılmamış için kullanılan bir dosya formatı.
GIF	CompuServe	<i>Grafik Değiştirme Formatı (Graphic Interchange Format)</i> 1 ile 8 bit arası görüntüler için kayıpsız LZW kodlamayı [8.2.4] kullanan bir dosya formatı. Sıklıkla WWW (World Wide Web) için küçük animasyonlar ve kısa düşük çözünürlüklü filmler yapmada kullanılır.
PDF	Adobe Systems	<i>Taınabilir Doküman Formatı (Portable Document Format)</i> . 2-D dokümanları bir aygıt ve çözünürlükten bağımsız şekilde göstermek için bir format. Bu format JPEG, JPEG 2000, CCITT, ve diğer sıkıştırılmış görüntüler için bir ifade edicileri olarak çalışabilir. Bazı PDF sürümleri ISO standartlarına sahip olmuşlardır.
PNG	World Wide Web Consortium (W3C)	<i>Taınabilir ağ Grafikleri (Portable Network Graphics)</i> . Herbir pikselin değeri ile önceki piksellere dayalı öngörü yapılan değeri arasındaki farkı kodlayarak saydam bir şekilde (48 bit/piksel'e kadar) tam renkli görüntüleri kayıpsız sıkıştıran bir dosya formatı [8.2.9].
TIFF	Aldus	<i>Etiketli Görüntü Dosya Formatı (Tagged Image File Format)</i> . JPEG, JPEG-LS, JPEG-2000, JBIG2, ve diğerleri gibi değişik görüntü sıkıştırma standartlarını destekleyen esnek bir dosya formatı.
<i>Video</i>		
AVS	MII	H.264'e benzerdir fakat üstel Golomb kodlamasını kullanır [8.2.2]. Çin'de geliştirilmiştir.
HDV	Şirket konsorsiyumu	<i>Yüksek Çözünürlüklü Video (High Definition Video)</i> . Öngörücü fark alma ile zamansal artıklığı kaldıran, sıkıştırma olarak MPEG-2'yi kullanan HD televizyon için DV'nin bir uzantısı [8.2.9].
M-JPEG	Farklı şirketler	Hareketli JPEG (Motion JPEG). JPEG kullanarak herbir çerçevenin bağımsız olarak sıkıştırıldığı bir sıkıştırma formatı.
Quick-Time	Apple Computer	DV, H.261, H.262, H.264, MPEG-1, MPEG-2, MPEG-4, ve diğer video sıkıştırma formatlarını destekleyen bir ortam ifade edicisi.
VC-1 WMV9	SMPTE Microsoft	Internet üzerinde en çok kullanılan video formatı. HD ve <i>Mavi ışın (Blue-ray)</i> yüksek çözünürlüklü DVD'ler için uyarlanmıştır. Çerçevelerde öngörücü yapılmayan içerik bağımlı değişken uzunluklu kod tabloları [8.2.11] ve değişken blok boyutlu bir tamsayı

Bazı Temel Sıkıştırma Yöntemleri: Huffman Kodlaması

Orjinal kaynak		Kaynak indirgeme			
Sembol	Olasılık	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6 0.4
a_6	0.3	0.3	0.3	0.3	
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1			
a_3	0.06	0.1			
a_5	0.04				

ŞEKİL 8.7
Huffman kaynak indirgemesi.

Bazı Temel Sıkıştırma Yöntemleri: Huffman Kodlaması

Orjinal kaynak			Kaynak indirgeme					
Sembol	Olasılık	Kod	1	2	3	4		
a_2	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00
a_1	0.1	011	0.1	011	0.2	010	0.3	01
a_4	0.1	0100	0.1	0100	0.1	011	0.4	1
a_3	0.06	01010	0.1	0101				
a_5	0.04	01011						

ŞEKİL 8.8
Huffman kod atama süreci.

Kodun ortalama uzunluğu:

$$L_{avg} = 0.4 * 1 + 0.3 * 2 + 0.1 * 3 + 0.1 * 4 + 0.06 * 5 + 0.04 * 5$$

$$= 2.2 \text{ bits/pixel}$$

Huffman Örnek

- ▶ 6 sembolden oluşan bir mesajınız olduğunu varsayalım.
- ▶ Her sembolün 8 bit genişliği vardır.
- ▶ Mesajın uzunluğu 100 semboldür. $(8 * 100) = 800$ bit.
- ▶ Aşağıdaki tabloda sembollerin sıklığı verilmektedir.

Symbol (Sembol)	frequency	Probability (Olasılık)
a ₁	8	0.08
a ₂	7	0.07
a ₃	19	0.19
a ₄	45	0.45
a ₅	3	0.03
a ₆	18	0.18

Huffman Örnek

Sembol	Code	Probability Olasılık	1	2	3	4
a ₄	1	0.45	0.45	0.45	0.45	+ 0.55
a ₃	01	0.19	0.19	0.19	+ 0.36	0.45
a ₆	000	0.18	0.18	0.18	0.19	
a ₁	0011	0.08	+ 0.1	+ 0.18		
a ₂	00100	0.07	0.08			
a ₅	00101	0.03				

Huffman Örnek

İlk durumda 0,55 olasılığının 0 olarak ifade edildiğine ve 0,45 olasılığının 1 olduğuna dikkat edin.

Symbol Sembol	Code	Probabilty Olasılık	1	2	3	4
a ₄	1	0.45	0.45	0.45	0.45	0.55 (0)
a ₃	01	0.19	0.19	0.19	0.36 (00)	0.45 (1)
a ₆	000	0.18	0.18	0.18 (000)	0.19 (01)	
a ₁	0011	0.08	0.1 (0010)	0.18 (001)		
a ₂	00100	0.07 (00100)	0.08 (0011)			
a ₅	00101	0.03 (00101)				

Huffman Örnek

- ▶ Orijinal mesaj uzunluğu $b = 8 \text{ bits} * 100 = 800 \text{ bits}$
- ▶ Huffman kodlu mesaj uzunluğu

$$b' = \text{length}(a_1) * \text{freq}(a_1) + \text{length}(a_2) * \text{freq}(a_2) + \text{length}(a_3) * \text{freq}(a_3) + \\ \text{length}(a_4) * \text{freq}(a_4) + \text{length}(a_5) * \text{freq}(a_5) + \text{length}(a_6) * \text{freq}(a_6)$$
$$b' = 4 * 8 + 5 * 7 + 2 * 19 + 1 * 45 + 5 * 3 + 3 * 18 = 219$$

$$\text{compression ratio } c = \frac{b}{b'}$$

$$c = \frac{800}{219} = 3.65$$

$$\text{Redundancy } R = 1 - \frac{1}{c} = 0.72$$

Sabit uzunluk kodlamasını kullanırsanız, mesaj uzunluğu $3 \text{ bits} * 100 = 300 \text{ bits}$ olmakta.

Bu, değişken uzunluklu kodun daha iyi olduğu anlamına gelir.

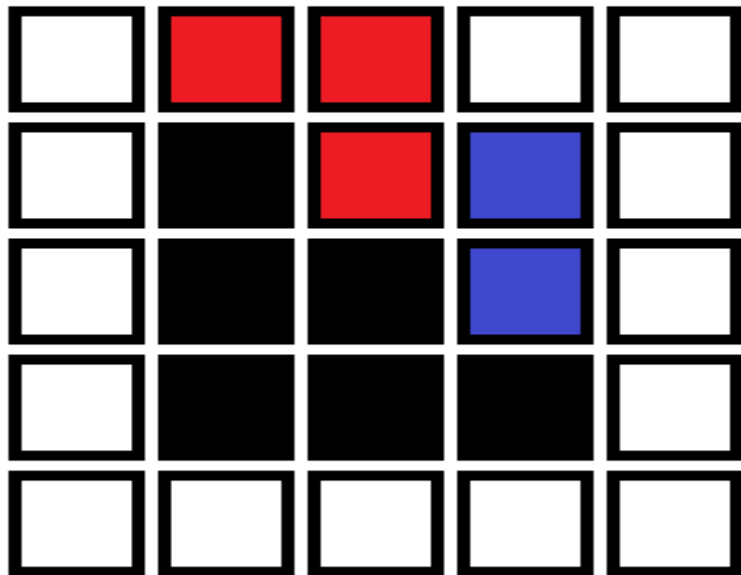
Symbol Sembol	code	Length In bits	freq
a_4	1	1	45
a_3	01	2	19
a_6	000	3	18
a_1	0011	4	8
a_2	00100	5	7
a_5	00101	5	3

Huffman Kodlama - JPEG Görüntü Sıkıştırma Tabanı

In 1952 David Huffman, a graduate student at the famous Massachusetts Institute of Technology developed an elegant algorithm for lossless compression as part of his schoolwork. The algorithm is now known as Huffman coding.

Huffman coding can be used to compress all sorts of data. It is an entropy-based algorithm that relies on an analysis of the frequency of symbols in an array.

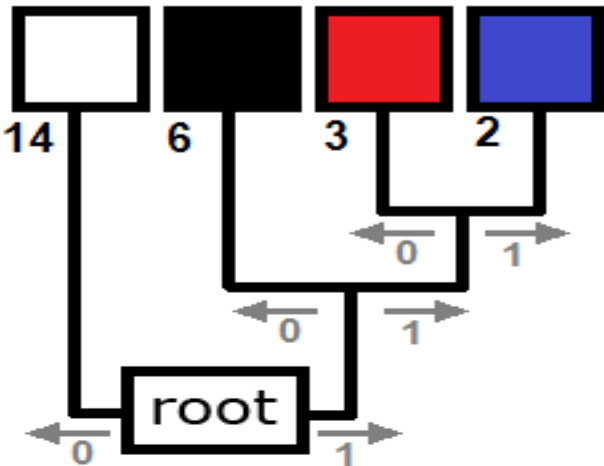
Huffman coding can be demonstrated most vividly by compressing a raster image. Suppose we have a 5x5 raster image with 8-bit color, i.e. 256 different colors. The uncompressed image will take $5 \times 5 \times 8 = 200$ bits of storage.





First, we count up how many times each color occurs in the image. Then we sort the colors in order of decreasing frequency. We end up with a row that looks like this:



Now we put the colors together by building a tree such that the colors farthest from the root are the least frequent. The colors are joined in pairs, with a node forming the connection. A node can connect either to another node or to a color. In our example, the tree might look like this:

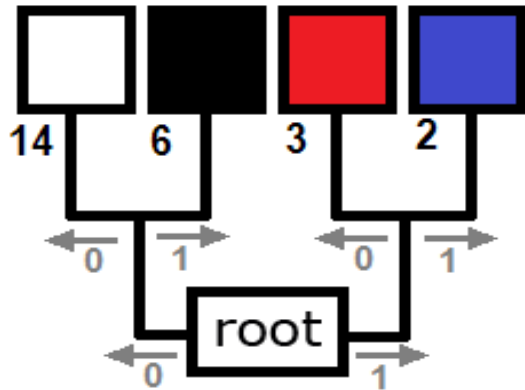


Our result is known as a Huffman tree. It can be used for encoding and decoding. Each color is encoded as follows. We create codes by moving from the root of the tree to each color. If we turn right at a node, we write a 1, and if we turn left – 0. This process yields a Huffman code table in which each symbol is assigned a bit code such that the most frequently occurring symbol has the shortest code, while the least common symbol is given the longest code.

color	freq.	bit code
	14	0
	6	10
	3	110
	2	111

Huffman Kodlama - JPEG Görüntü Sıkıştırma Tabanı

The Huffman tree and code table we created are not the only ones possible. An alternative Huffman tree that looks like this could be created for our image:



The corresponding code table would then be:

color	freq.	bit code
	14	00
	6	01
	3	10
	2	11

Using the variant is preferable in our example. This is because it provides better compression for our specific image.

Huffman Kodlama - JPEG Görüntü Sıkıştırma Tabanı

Using the variant is preferable in our example. This is because it provides better compression for our specific image.

Because each color has a unique bit code that is not a prefix of any other, the colors can be replaced by their bit codes in the image file. The most frequently occurring color, white, will be represented with just a single bit rather than 8 bits. Black will take two bits. Red and blue will take three. After these replacements are made, the 200-bit image will be compressed to $14 \times 1 + 6 \times 2 + 3 \times 3 + 2 \times 3 = 41$ bits, which is about 5 bytes compared to 25 bytes in the original image.

Kaynaklar

- ▶ Sayısal Görüntü İşleme, Palme Yayıncılık, Üçüncü Baskıdan Çeviri (*Orj: R.C. Gonzalez and R.E. Woods: "Digital Image Processing", Prentice Hall, 3rd edition, 2008*).
- ▶ “Digital Image Processing Using Matlab”, Gonzalez & Richard E. Woods, Steven L. Eddins, Gatesmark Publishing, 2009
- ▶ Ders Notları, CS589-04 Digital Image Processing, F.(Qingzhong) Liu, <http://www.cs.nmt.edu/~ip>
- ▶ Ders Notları, BIL717-Image Processing, E.Erdem
- ▶ Ders Notları, EBM537-Görüntü İşleme, F.Karabiber
- ▶ Huffman Coding – Base of JPEG Image Compression, <https://www.print-driver.com/stories/huffman-coding-jpeg>
- ▶ <https://docs.opencv.org/>
- ▶ Bekir Aksoy, Python ile İmgeden Veriye Görüntü İşleme ve Uygulamaları, Nobel Akademik Yayıncılık