Bilgisayar Görmesi

Ders 2: GÖRÜNTÜLERİN GÖSTERİLMESİ

Dr. Öğr. Üyesi Serap ÇAKAR

Görsel Algı

Görüntüleri insan bakış açısı için işlerken, görüntülerin izleyen tarafından nasıl bilgiye dönüştüğünü göz önünde bulundurmak önemlidir.

Görsel algıyı anlamak algoritma geliştirme esnasında yardımcı olur. Görüntü verisi renksellik ve parlaklık gibi fiziksel nicelikleri temsil eder.

Renksellik dalga boyu ile tanımlanan ışığın renk kalitesidir.

Parlaklık ışık miktarıdır.

İzleyen için bu fiziksel nicelikler renk ve aydınlık olarak algılanabilir.

Renkli görüntü bilgisini nasıl algıladığımız üç algısal değişkene ayrılır: renk tonu (hue), renk yoğunluğu (saturation), aydınlık (lightness).

Dünyadaki renkleri kullandığımızda renk tonu ile karşılaşırız.

Renk tonu yeşil ve sarı gibi renkleri birbirinden ayırır.

Renk tonu, farklı dalga boyları ile meydana gelen, bir izleyici tarafından algılanan renk algısıdır.

Dalga boyunun 430 ve 480 nanometre arası mavi, 500 ve 550 nanometre arası yeşil, 570 ve 600 nanometre arası sarı, 610 nanometre ve yukarısı kırmızı olarak algılanır.

Siyah, gri ve beyaz renk olarak ele alınabilir fakat bunlar renk tonu değildir.

Renk yoğunluğu rengin beyaz ışık ile seyreltilmişlik derecesidir. Saf renk tonuna eklenmiş nötr renk miktarı arttıkça renk yoğunluğu azalır. Renk yoğunluğu genellikle saf renk olarak düşünülür. Yoğunluğu olmayan renkler rengi atmış ve solmuş gözükür, yoğunluklu renkler ise koyu ve parlaktır. Kırmızı yüksek derecede yoğun iken pembe yoğun değildir. Saf bir renk %100 yoğundur ve beyaz ışık içermez. Beyaz ışık ve saf renk karışımı %0 ve %100 yoğunluk arasında dağılım gösterir.

Aydınlık (ışıklılık) yansıyan nesnelerin yoğunluğunun algılanmasıdır. Beyazdan gri ve siyaha doğru geçen renklerin tamamı ile ilgilidir; grilik seviyesi olarak da düşünülebilir. Benzer bir terim olan parlaklık CRT gibi kendisi aydınlık olan nesnelerin algılanan yoğunluğudur. Parlaklık algılanan nicelik, aydınlık ise ölçülebilen niceliktir ve aralarındaki bağıntı logaritmik olarak değişir.

Zıtlık (contrast) görüntünün en koyu bölgesinden en açık bölgesine doğru olan derecelenmedir. I_{max} ve I_{min} bölgenin veya görüntünün maksimum ve minimum yoğunlukları olmak üzere, matematiksel ifadesi aşağıdaki gibidir.

$$Contrast = \frac{I_{max} - I_{min}}{I_{max} + I_{min}}$$

Yüksek zıtlıklı görüntüler geniş koyu ve açık bölgelere sahiptir. İyi zıtlıklı görüntüler bütün yoğunluklara sahiptir.

Bir görüntünün zıtlığı arttıkça, izleyici detayları daha iyi algılar. Bu sadece algılamadır, görüntüdeki detayların arttığını göstermez.

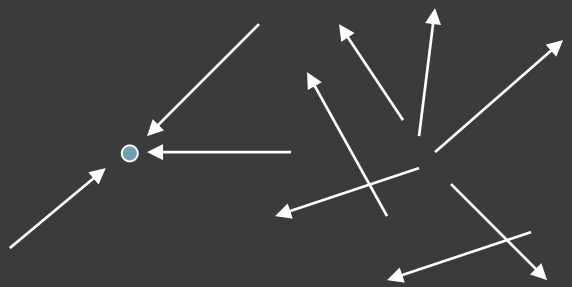
Işık nedir?

Elektromagnetic radyasyon (EMR) uzayda ışınlar boyunca hareket eder.

- R(λ) EMR dir, güç birimi ile ölçülür (watt)
  - λ dalgaboyudur

Işık alanı

- Işığı uzayın her noktasına ve her yöne yayılan radyasyon olarak tanımlayabiliriz.



$$R(X, Y, Z, \theta, \phi, \lambda, t)$$

Işık alanı

$$R(X, Y, Z, \theta, \phi, \lambda, t)$$

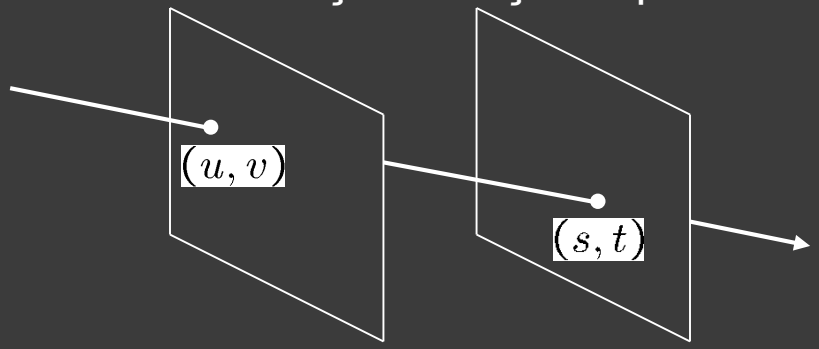
- **plenoptic fonksiyon** olarak bilinir.
- *R*'yi biliyorsanız, görüntünün herhangi bir bakış açısından nasıl görüneceğini kestirebilirsiniz.
- Bu, görünümün tam tanımını verir.

Işık alanı

$$R(u, v, s, t)$$

t zaman değildir (yukarıdaki t'den farklıdır)

- Parlaklığın bir ışın boyunca değişmediğini varsayalım.
  - Bu varsayım dünya ile ilgili neyi verir?
- İki plaka arasındaki kesişme ile ışınlar parametrelendirilebilir.



- Genellikle λ ve zaman parametreleri ihmal edilir.

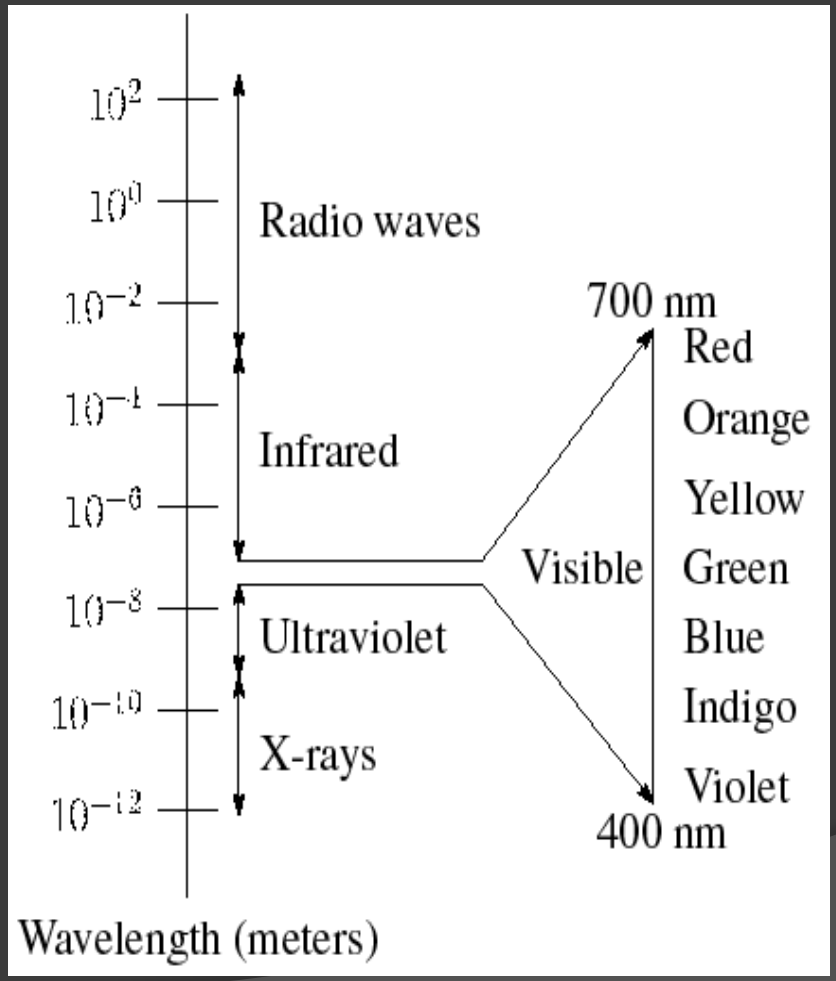
$$L(x,y,t,\lambda) = \varepsilon(x,y,t)M(\lambda) + r(x,y,t,\lambda)i(x,y,t,\lambda)$$

ε	Işık yayma 'emisitive'
L	Spektral Işınım 'radiance'
M	Objenin ışıması 'emittance'
R	Yansıma 'reflectance'
T	Zaman (time)
x,y	Koordinat (coordinate)
λ	Dalga boyu (Wavelenght)

Şekil: bir görüntünün matematiksel temsili

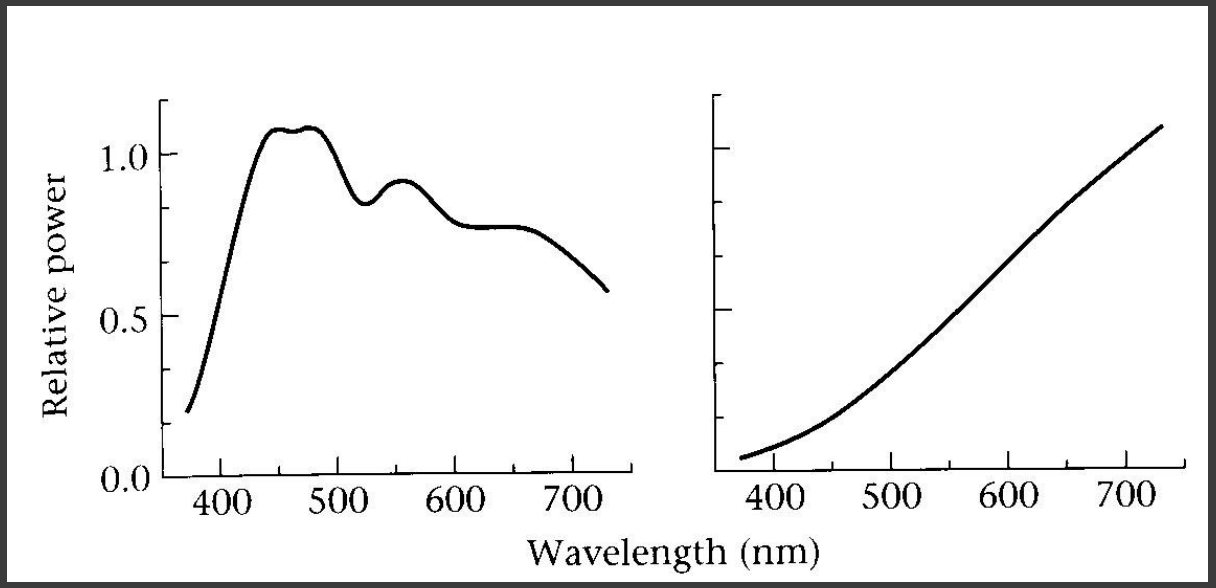
Görünür ışık spektrumu (tayf)

- Elektromagnetik radyasyonu belirli bir dalgaboyu aralığında görebiliriz.



Işık spektrumu

- Işığın görünümü onun güç spektrumuna bağlıdır.
- Bir dalgaboyunda ne kadar güç (veya enerji) vardır.

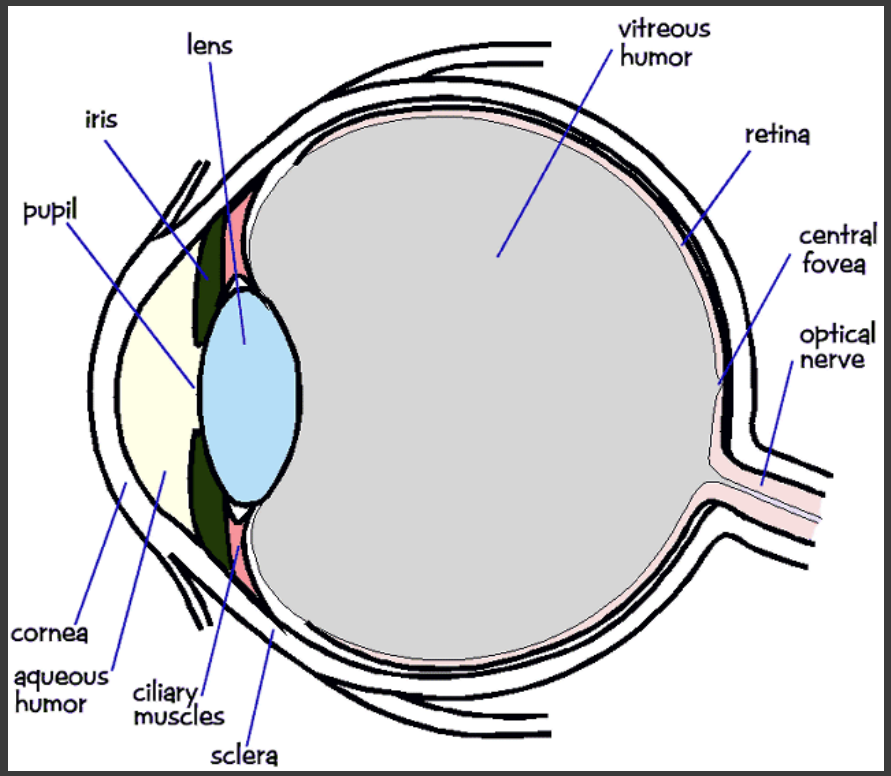


Gün ışığı Ampul ışığı

Görsel sistemimiz ışık spektrumunu renklere dönüştürür

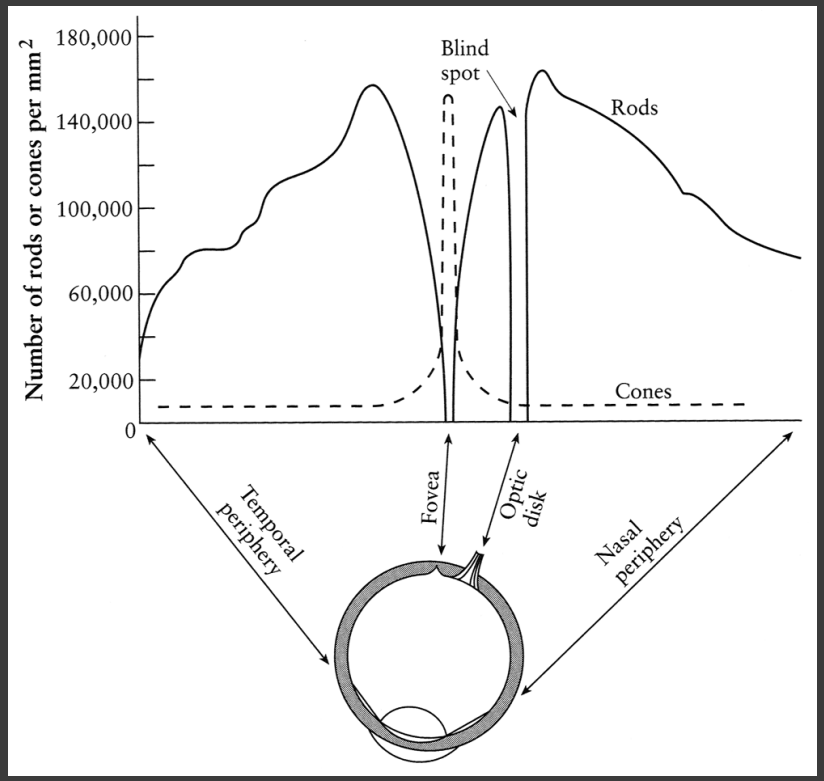
- Bu karmaşık bir dönüşümdür

İnsan görme sistemi



- Renk algısı
  - Işık fotosensetif hücrelere sahip olan retinaya gelir
    - Çubuklar (Rod) ve koniler (cone)
  - Bu hücreler tayfları birkaç ayrık değere dönüştürür

Çubukların ve konilerin yoğunlukları

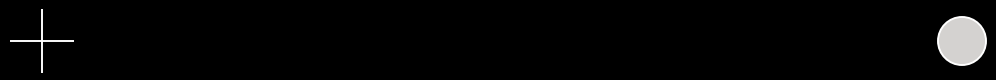


- Çubuklar ve koniler retinada tek biçimli bir şekilde dağılmamıştır.
  - Çubuklar gece görüşünü sağlar ve yoğunluktan sorumludur, koniler gündüz görüşünü sağlar ve renklerden sorumludur
  - **Fovea** – Konilerin en yoğun bulunduğu (çubukların olmadığı) görsel eksenin merkezindeki küçük bir bölgedir (1 veya 2°).
  - Çevredeki daha az görsel keskinlik için çoğu çubuk aynı nerona bağlanmıştır.

- İnsan görme sistemi optik sinirle birbirine bağlı iki ana bileşene sahiptir: Göz ve beyin. Yapısı hakkında en çok bilgiye sahip olduğumuz şey görüntü alma sensörü gözdür. Bilgisayar görüş sistemi insan beynine benzer bir bilgi işleme ünitesi olarak düşünülebilir. Bu ikisi alış sensöründen (göz) işleme ünitesine (beyin) yolculuk eden görsel bilgiyi taşıyan sinirler demeti ile birbirine bağlanmıştır.

- İnsan görsel sistemi şu şekilde çalışır: 1) Işık enerjisi retinadaki sensörlere lensler tarafından odaklanır 2) Bu sensörler bu ışık enerjisine optik sinirden beyine elektriksel sinyaller yollayan elektro-mekanik bir reaksiyonla cevap verirler 3)Beyin bu sinyalleri bizim görüntü olarak algıladığımız sinirsel bir modele çevirir.

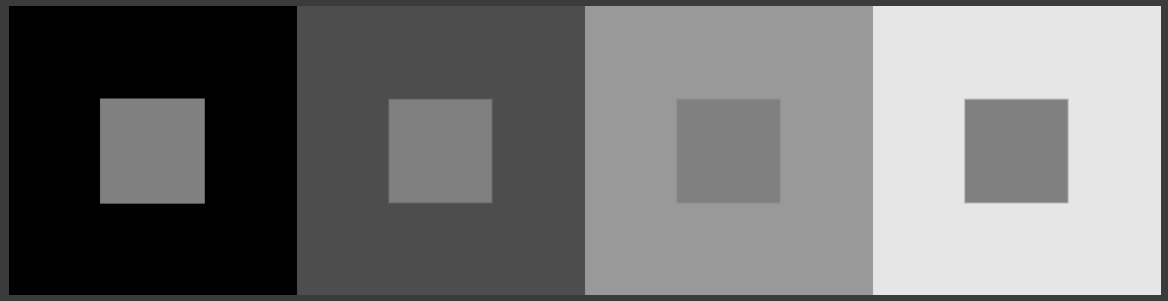
Görsel keskinliğin gösterilmesi



Sol gözünüzü kapatarak soldaki artı işaretine bakınız. Sağdaki daire kaybolacaktır. (Glassner, 1.8).

Parlaklık zıtlık ve tutarlılık

- Parlaklığın görünümü etrafındaki bölgeye bağlıdır.
  - **Parlaklık zıtlığı**: zıt bir renk bölgesi etrafındaki bölgeye bağlı olarak daha açık veya daha koyu gözükebilir:



  - **Parlaklık tutarlılığı**: bir yüzey değişik ışıklar altında bile aynı gözükebilir.

Işığa verilen tepki lineer değildir

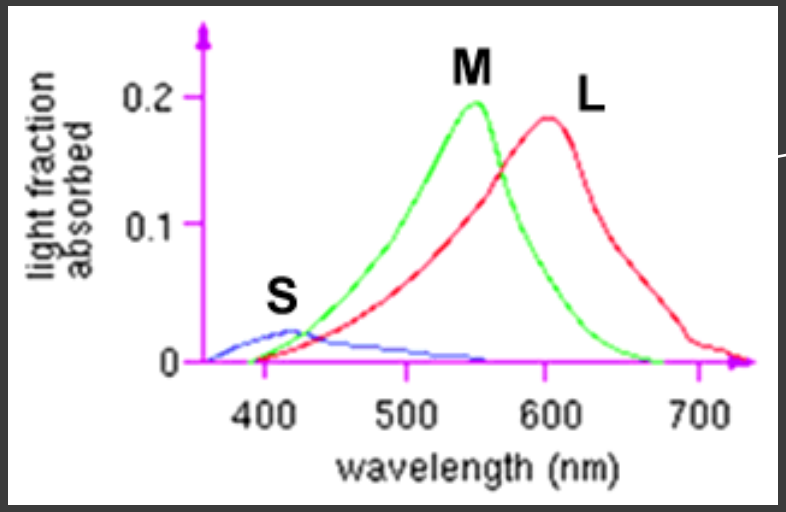
- Görsel sistemimiz büyük dinamik bir bölgeye sahiptir.
  - Aydınlık ve karanlık şeyleri aynı anda fark edebiliriz.
  - Bunu başarabilmek için bir mekanizma ışık yoğunluğunu logaritmik ölçekte algılayabilmemizdir.
    - Üstel bir yoğunluk rampası lineer bir rampa olarak görülebilir
  - Diğer bir mekanizma adaptasyondur.
    - Çubuklar ve koniler düşük ışıkta daha hassas, yüksek ışıkta daha az hassastır.

Görsel Dinamik Bölge

Background	Luminance (candelas per square meter)
Horizon sky	
Moonless overcast night	0.00003
Moonless clear night	0.0003
Moonlit overcast night	0.003
Moonlit clear night	0.03
Deep twilight	0.3
Twilight	3
Very dark day	30
Overcast day	300
Clear day	3,000
Day with sunlit clouds	30,000
Daylight fog	
Dull	300–1,000
Typical	1,000–3,000
Bright	3,000–16,000
Ground	
Overcast day	30–100
Sunny day	300
Snow in full sunlight	16,000

FIGURE 1.13
Luminance of everyday backgrounds. *Source:* Data from Rea, ed., *Lighting Handbook 1984 Reference and Application*, fig. 3-44, p. 3-24.

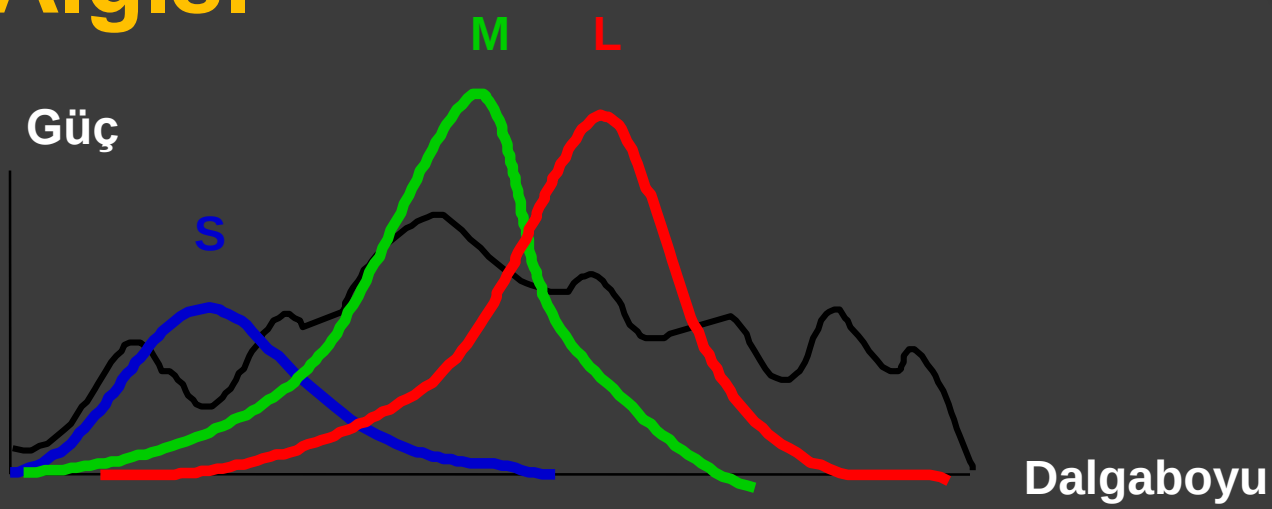
Renk algısı



L cevap eğrisi

- Üç tip koni vardır
  - Her biri spektrumun farklı bir bölgesinde hassastır.
  - Fakat bölgeler çakışır
    - Kısa (S) maviyi gösterir
    - Orta (M) yeşili gösterir
    - Uzun (L) kırmızıyı gösterir
  - Farklı algılama: Göz yeşile kırmızıdan daha hassastır.
  - Renk körlüğü—bir tip koninin eksikliği veya yetersizliği sonucu oluşur.

Renk Algısı



- Çubuklar ve koniler spektrumda filtreler gibi davranır.
  - Bir filtrenin çıkışını elde etmek için yanıt eğrisini spektrum ile çarpmak gerekir ve bütün dalga boyları birleştirilir.
  - Her bir koni tek bir rakam ile sonuçlanır.
  - S: Bütün spektrumu 3 rakamla nasıl temsil edebiliriz?

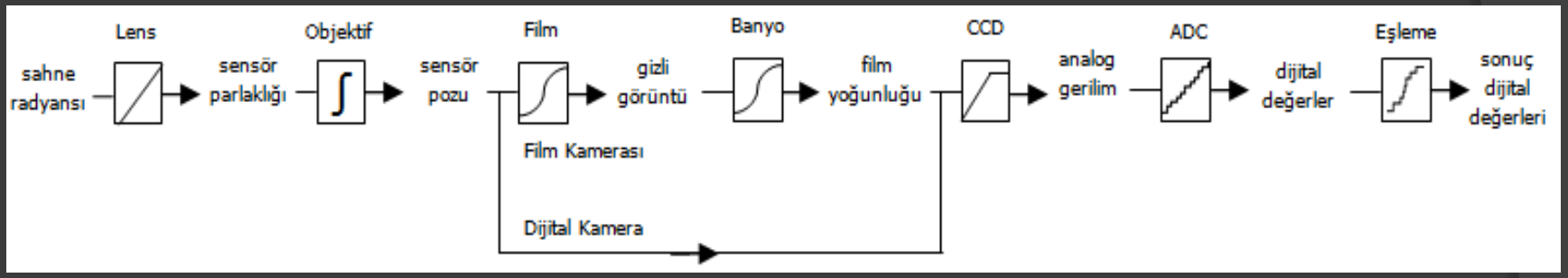
  - C: Yapamayız! Bilgilerin çoğu kaybolur.
    - Sonuç olarak, iki farklı spectruma ayrılamaz görülebilir.
      - » Böyle spectrumlar **metamer** olarak bilinir.
      - » http://www.cs.brown.edu/exploratory/research/applets/repository/spectrum/metamers.html

Algılama Özet

- Işımadan algılanan renge geçiş yapmak oldukça karmaşıktır!
  - Verinin çoğunu atarız
  - Logaritma uygularız
  - Parlaklık göz bebeğinin büyüklüğünü etkiler
  - Parlaklık zıtlık ve tutarlılığı etkiler
  - Ve görüntüler oluşur

Kamera yanıt fonksiyonu

- Işınımdan piksele f geçişi nasıl olur?
  - Bu da karmaşıktır, fakat daha anlaşılırdır
  - Bu f geçişi film veya kamera cevap fonksiyonu olarak bilinir



Işınım değerlerinden piksel değerlerini nasıl elde ederiz?

Bu neden önemlidir?

- Maddenin özelliklerini kestirmek istiyorsak bu faydalıdır
- Gölgelemeden şekle geçiş ışınım gerektirir
- Yüksek dinamikli görüntüleri oluşturmayı mümkün kılar

Yanıt fonksiyonu nelere bağlıdır?

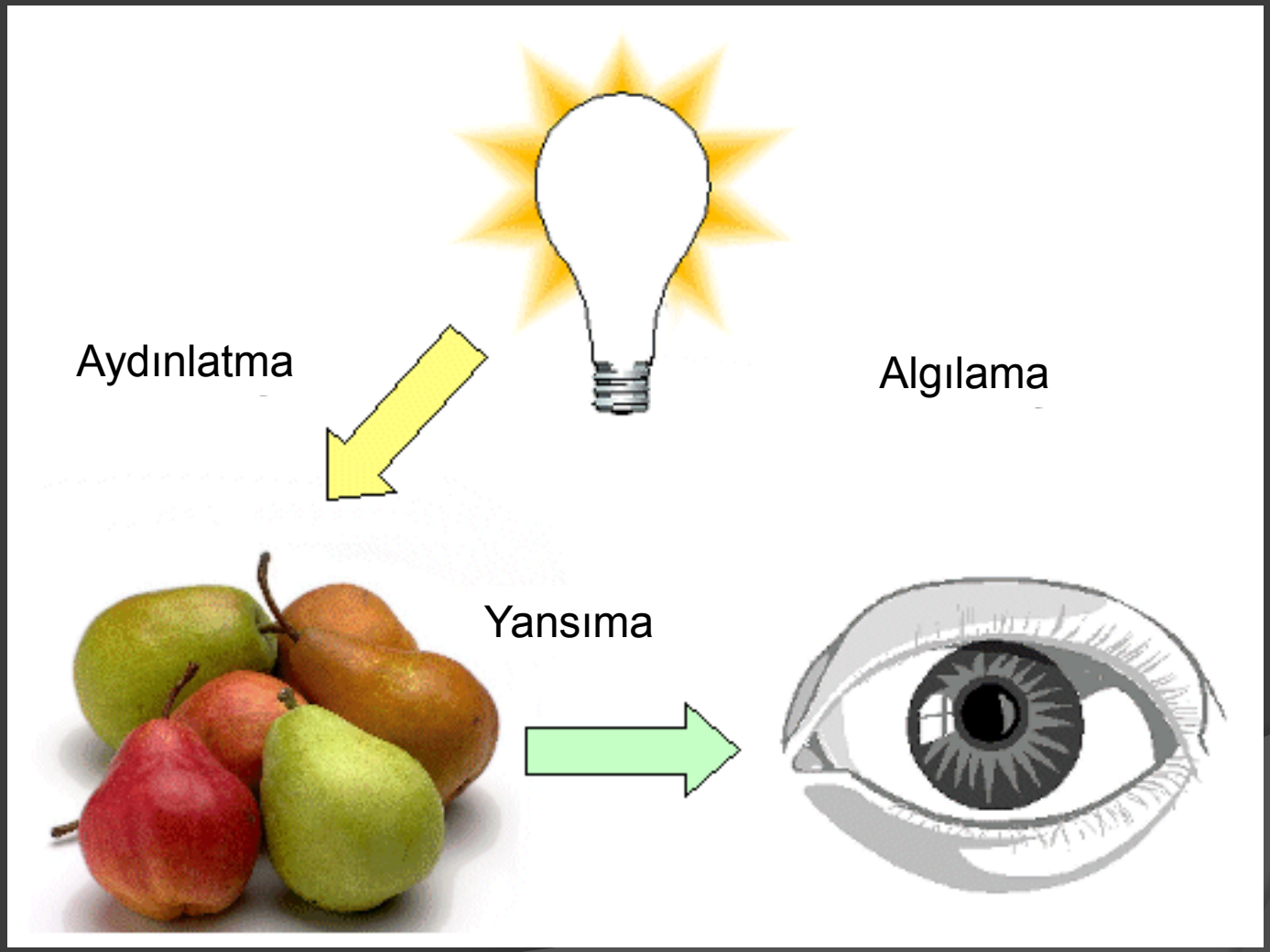
f(objektif kapağı hızı, diyafram açıklığı, film hammaddesi, dijitalleştirici..)

Yüksek dinamikli görüntüler

Figure 6: *Sixteen photographs of a church taken at 1-stop increments from 30 sec to* $\frac{1}{1000}$ *sec. The sun is directly behind the rightmost stained glass window, making it especially bright. The blue borders seen in some of the image margins are induced by the image registration process.*

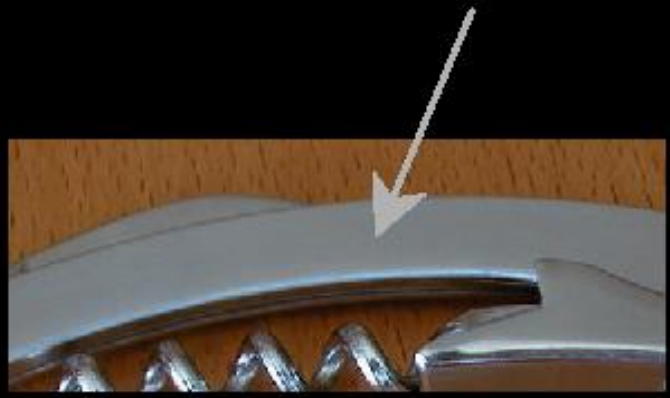
- Teknikler
  - Debevec: http://www.debevec.org/Research/HDR/
  - Columbia: http://www.cs.columbia.edu/CAVE/tomoo/RRHomePage/rrgallery.html

Işığın yansıtılması

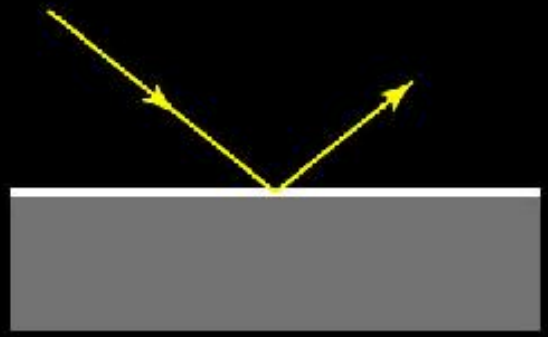


Materials

iletken

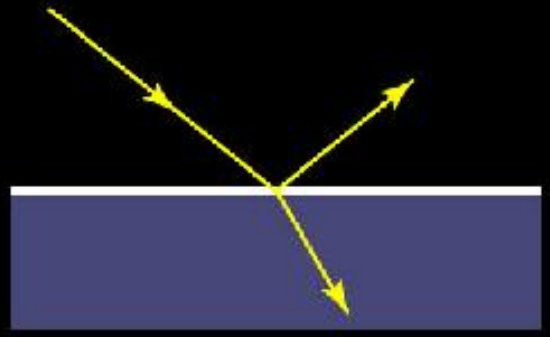


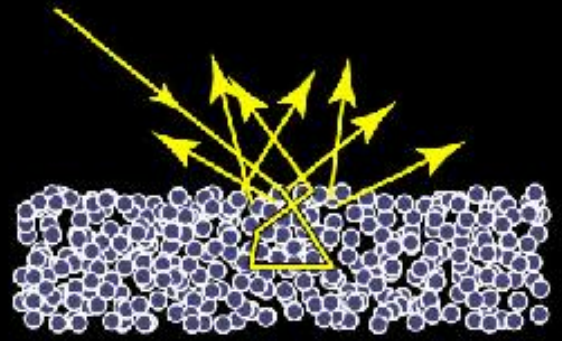
iletken +
mikrogeometri



yalıtkan

yalıtkan +
mikrogeometri





Işığın etkileşimi ve yansıtılması

- Bir ışık bir nesneye vurduğunda ne olur?
  - Işığın bir kısmı emilir
    - Enerjinin diğer biçimlerine dönüştürülür (örn. ısı)
  - Bir kısmı nesnenin içerisinden iletilir
    - Bükülür ve yansıtılır
  - Bir kısmı yansıtılır
    - Daha önce de görüldüğü gibi, bir kerede birçok yöne iletilebilir

Renklerin Temsil Edilmesi

Bir renk modeli (veya renk uzayı) renklerin temsil edilme yoludur ve birbirleri arasındaki ilişkidir.

Farklı görüntü işleme sistemleri farklı sebeplerden farklı renk modellerini kullanır.

Renkli resim yayınlama endüstrisi CMY renk modelini kullanır. Renkli CRT monitörleri ve çoğu bilgisayar grafik sistemleri RGB renk modelini kullanır.

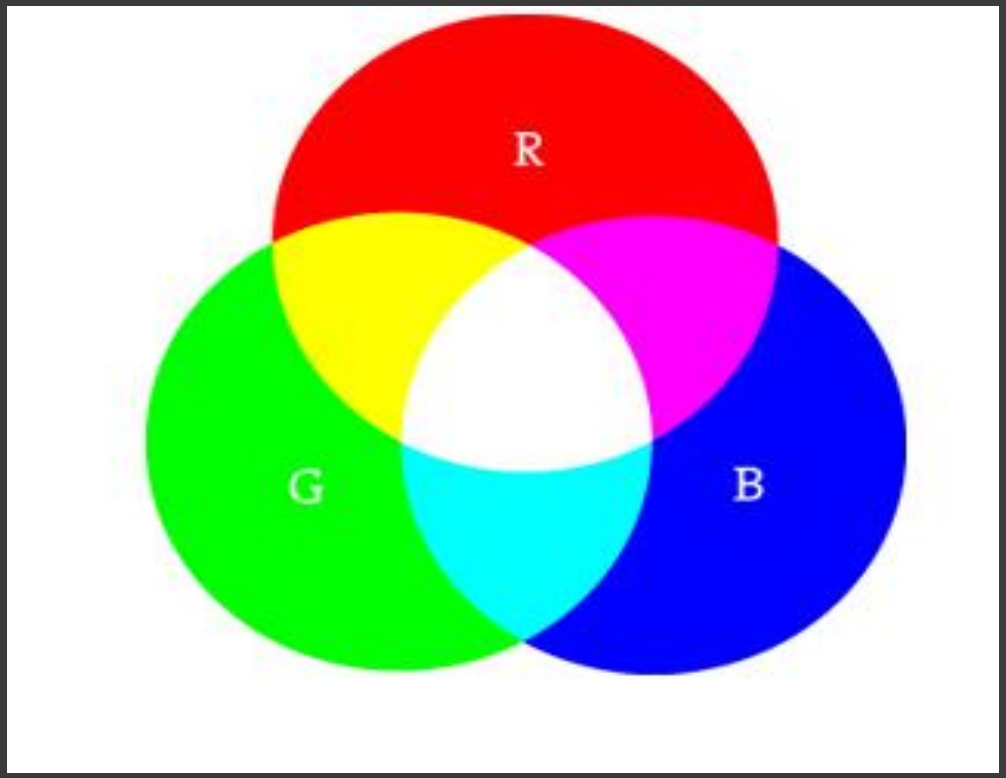
Renk tonu, renk yoğunluğu ve parlaklık ile çalışmak zorunda olan sistemler HSI renk modelini kullanır.

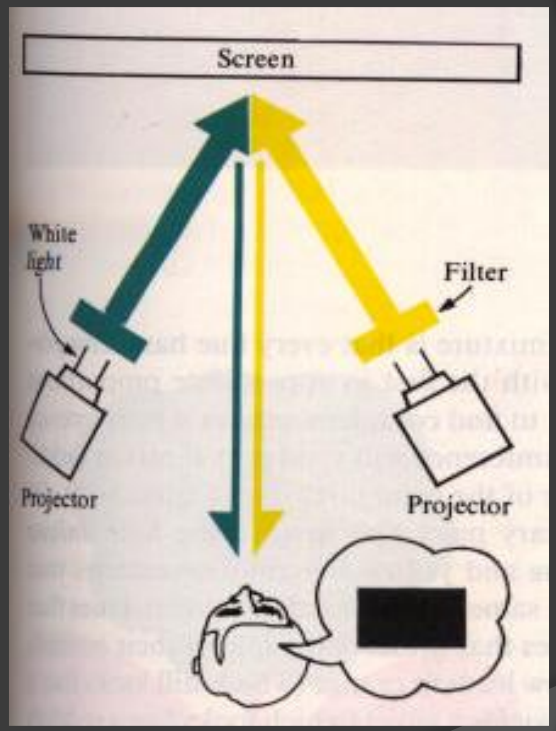
İnsanların renkleri algılaması üç tip koninin yanıt fonksiyonu ile oluşur. Bunun için, renk sistemleri üç tane sayıya dayanır. Bu sayılar tristimulus değerleri olarak adlandırılır.

Tristimulus değerlerine dayanan pek çok renk uzayı vardır. YIQ renk uzayı geniş bantlı televizyonlarda kullanılır. XYZ uzayı fiziksel birincillerle alakalı değildir fakat renk standardı olarak kullanılır. Basit bir matris çarpımı kullanılarak XYZ'den diğer renk uzaylarına dönüşüm oldukça kolaydır. Diğer renk modelleri Lab, YUV, ve UVW'yi içerir.

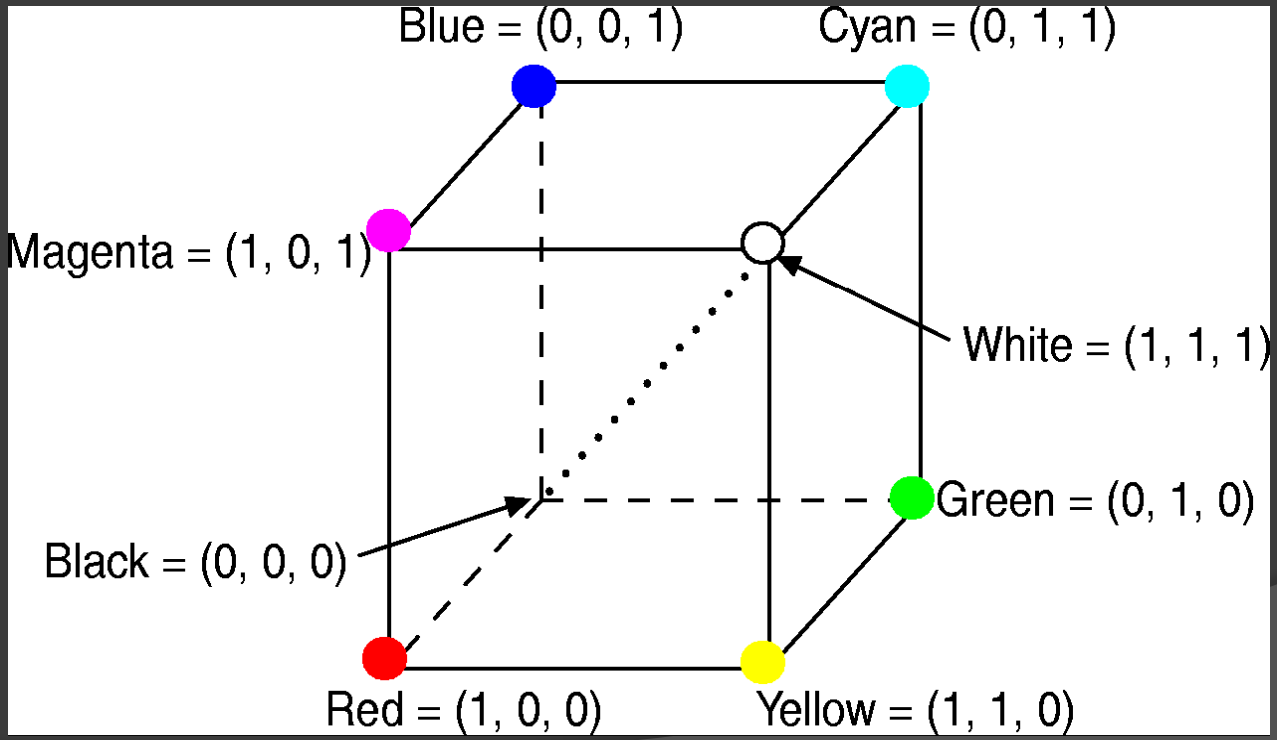
Renk uzayı tartışmalarının hepsi bütün renklerin normalleştirildiğini varsayar (0 ve 1.0 arası değerler). Bu, renklerin maksimum değere bölünmesi ile kolayca elde edilebilir. Örneğin, 8-bitlik renk sistemi 255'e bölme ile normalleştirilir.

RGB renk uzayı üç ana renkten oluşur: kırmızı, yeşil ve mavi. Diğer renkleri oluşturmak için bu renklerin tayfsal bileşenleri ard arda kombine edilir.





RGB modeli her bir eksende köşelerde kırmızı, yeşil ve mavi olan 3-boyutlu küple temsil edilir. Siyah orijinde bulunur. Beyaz küpün sonunda çaprazda bulunur. Gri seviye siyahtan beyaza çizgiyi takip eder. Her bir renk kanalı 8-bitlik olan 24-bitlik bir renk grafik sisteminde, kırmızı (255,0,0). Renk küpünde, bu (1,0,0)'dir.



RGB modeli bilgisayar grafikleri sistemlerinin tasarımını basitleştirir fakat bütün uygulamalar için ideal değildir. Kırmızı, yeşil ve mavi renk bileşenleri yüksek oranda ilişkilidir. Bu, bazı görüntü işleme algoritmalarını gerçekleştirmeyi zorlaştırır. Histogram denkleştirme gibi çoğu işleme teknikleri, görüntünün sadece yoğunluk bileşeni üzerinde çalışır. Bu işlemleri HSI renk modelinde yapmak daha kolaydır.

Çoğu zaman RGB görüntüleri gri seviyeli görüntülere dönüştürmek gerekir. Bir görüntüyü RGB'den gri seviyeye dönüştürmek için aşağıdaki eşitlik kullanılır:

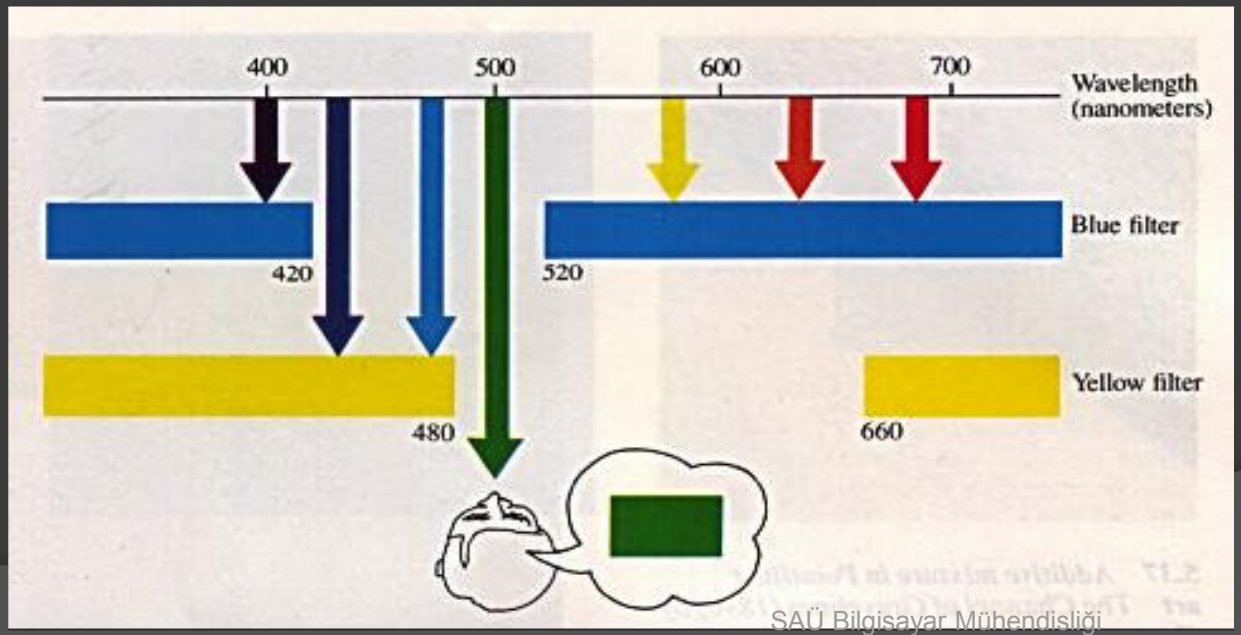
Gri seviyeli yoğunluk = 0.299R + 0.587G + 0.114B

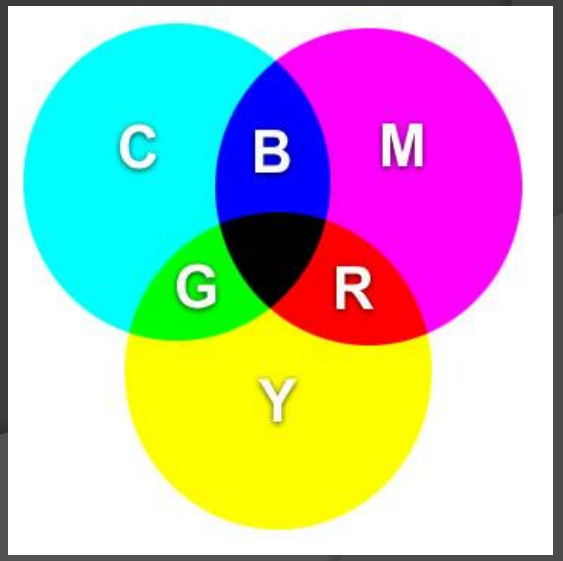
Bu eşitlik parlaklık için NTSC standardından gelir.
Diğer bir dönüştürme şekli aşağıdaki gibidir.

Gri seviyeli yoğunluk = 0.333R + 0.333G + 0.333B

CMY/CMYK

CMY renk uzayı cam göbeği (cyan), mor pembe (magenta) ve sarı (yellow) renklerinden oluşur. Cam göbeği, mor pembe ve sarı sırası ile kırmızı, yeşil ve mavinin tamamlayıcısı olduğu için RGB renk uzayının tamamlayıcısıdır. Cam göbeği, mor pembe ve sarı çıkarılan renkler olarak bilinir. Bu renkler istenilen rengi elde etmek için beyaz ışıktan çıkarılır. Cam göbeği kırmızıyı emer, mor pembe yeşili emer ve sarı maviyi emer. Sarıyı ve cam göbeğini arttırarak veya mor pembeyi azaltarak yeşili arttırabiliriz.





RGB ve CMY birbirinin tamamlayıcısı olduğu için, iki renk uzayı arasındaki dönüşüm kolaydır. RGB'den CMY'ye geçmek için, bileşenler beyazdan çıkarılır.

$C = 1.0 - R$

$M = 1.0 - G$

$Y = 1.0 - B$

CMY'den RGB'ye:

$R = 1.0 - C$

$G = 1.0 - M$

$B = 1.0 - Y$

Diğer bir renk modeli CMYK olarak adlandırılır. Siyak (K) yazdırma işleminde eklenir çünkü bu, diğer üç rengin kombinasyonundan daha saf bir siyahtır. Saf siyah daha büyük bir zıtlık sağlar.

CMY’den CMYK’ya dönüşüm yapmak için:

K = min(C, M, Y)
C = C - K
M = M - K
Y = Y – K

CMYK’dan CMY’ye dönüşüm için siyah bileşen C, M, ve Y bileşenlerine eklenir.

HSI

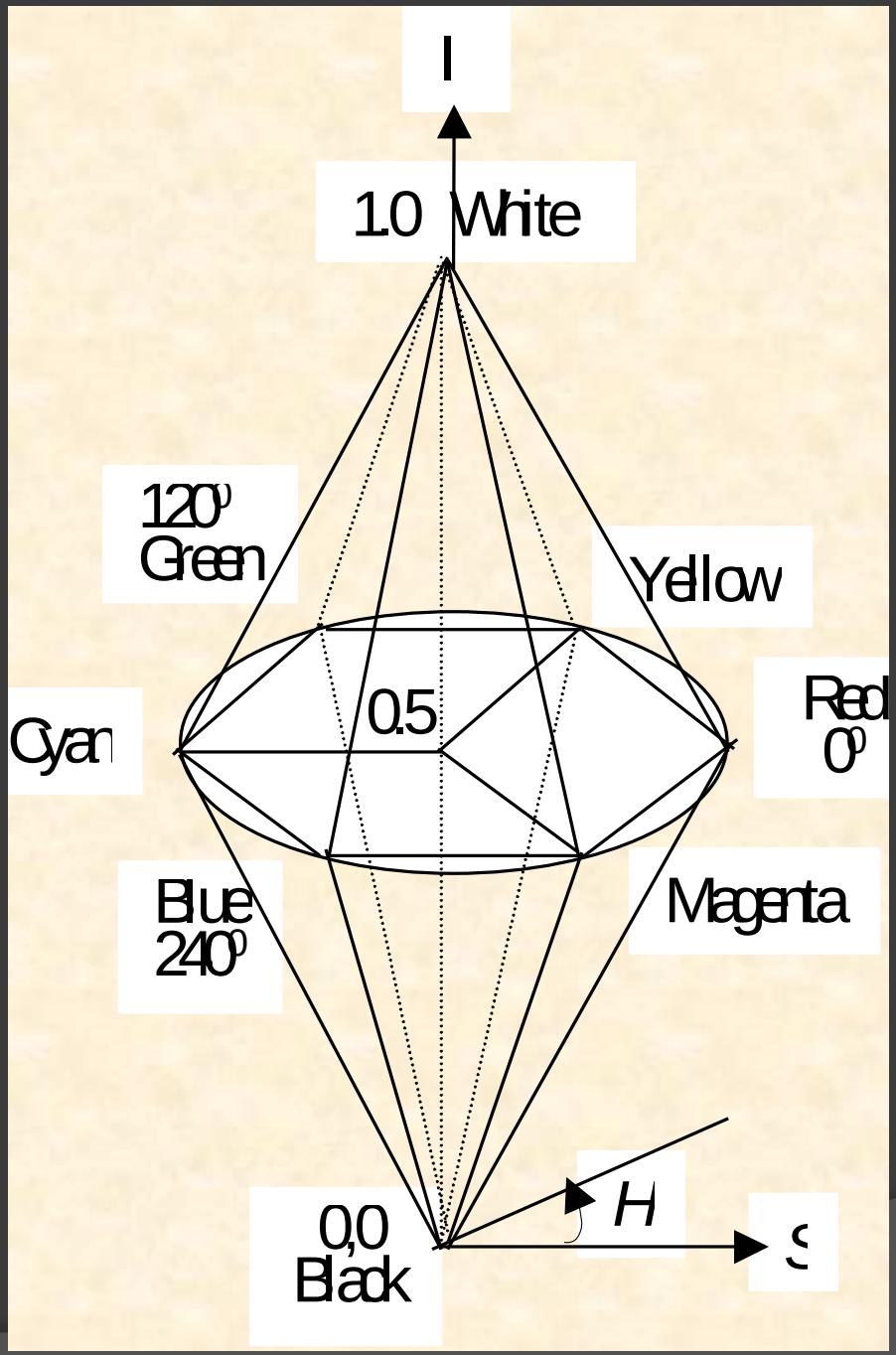
Renk tonu, parlaklık ve renk yoğunluğu rengi tanımlamak için kullanılan üç özellik olduğundan, HSI gibi ilgili bir renk modeli olması mantıklı gözükür. HSI renk uzayı kullanılırken bir rengi oluşturmak için mavi ve yeşilin hangi oranlarda olduğunu bilmemize gerek yoktur. İstediğiniz rengi elde etmek için renk tonunu ayarlamanız yeterlidir. Kırmızının derinliğini pembeye dönüştürmek için parlaklığı ayarlamanız gerekir. Rengi daha koyu ve açık yapmak için yoğunluğu değiştirmeniz gerekir.

Çoğu uygulama HSI renk modelini kullanır. Makine görmesi farklı nesnelerin rengini tanımada HSI renk uzayını kullanır. Histogram işlemleri, yoğunluk dönüşümleri ve konvolusyon gibi görüntü işleme uygulamaları sadece görüntünün yoğunluğu üzerinde işlem yapar. Bu işlemleri HSI renk sisteminde gerçekleştirmek daha kolaydır.

HSI silindirik koordinatlarla modellenmiştir. Renk tonu (*H*) 0^o 'den 360^o'ye değişen açılardan 0 açısı ile tanımlanır. Parlaklık (*S*) 0'dan 1'e değişen açılarla tanımlanır. Yoğunluk (*I*) 0 siyah ve 1 beyaz olmak üzere *z* ekseni boyunca değişir.

S = 0 olduğunda, renk 1 yoğunluk seviyesindedir. *S* = 1 olduğunda, renk koninin tepesindeki sınırdadır. Parlaklık daha yüksek olduğunda renk beyaz/gri/siyah 'dan daha uzaktır (yoğunluğa bağlı olarak değişir).

Renk tonunu ayarlamak rengi 0^o 'de olan kırmızıdan, 120^o'de olan yeşile ve 240^o 'de olan maviye ve tekrar 360^o 'de olan kırmızıya değiştiririz. *I* = 0 olduğunda, renk siyahtır ve böylece *H* tanımsızdır. *S* = 0 olduğunda , renk gri seviyelidir. *H* bu durumda yine tanımsızdır. *I*'nın değiştirilmesi ile, bir renk açık veya koyu yapılabilir. *S* = 1 iken *I*'nın değiştirilmesi ile, rengin gölgeleri oluşturulur.


I
1.0 White
120°
Green
Yellow
Cyan
0.5
Red
0°
Blue
240°
Magenta
0,0
Black
H
S

Aşağıdaki formüller RGB uzayından HSI uzayına nasıl dönüşüm yapılacağını gösterir:

$$I = \frac{1}{3}(R+G+B)$$

$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)]$$

$$H = \cos^{-1}\left[\frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right]$$

B, G'den büyük ise, $H = 360^0 - H$ olur.

To convert from HSI to RGB, the process depends on which color sector H lies in. For the RG sector ($0^0 \le H \le 120^0$):

$$b = \frac{1}{3}(1 - S)$$

$$r = \frac{1}{3}\left[1 + \frac{S\cos(H)}{\cos(60^0 - H)}\right]$$

$$g = 1 - (r + b)$$

For the *GB* sector ($120^0 \le H \le 240^0$):

$$H = H - 120^0$$

$$g = \frac{1}{3}\left[1 + \frac{S\cos(H)}{\cos(60^0 - H}\right]$$

$$r = \frac{1}{3}(1 - S)$$

$$b = 1 - (r + b)$$

For the *BR* sector ($240^0 \le H \le 360^0$):

$$H = H - 240^0$$

$$g = \frac{1}{3}\left[1 + \frac{S\cos(H)}{\cos(60^0 - H)}\right]$$

$$r = \frac{1}{3}(1 - S)$$

$$b = 1 - (r + b)$$

The values *r*, *g*, and b are normalized values of *R*, *G*, and *B*. *To* convert them to *R*, *G*, and *B* values use:

$$R=3Ir,\ G=3Ig,\ 100B=3Ib.$$

Remember that these equations expect all angles to be in degrees. To use the trigonometric functions in C, angles must be converted to radians.

YC_bC_r

YC_bC_r renk bilgisinden parlaklığı ayıran diğer bir renk uzayıdır. Parlaklık Y ile, mavilik ve kırmızılık C_bC_r ile kodlanır. *RGB'den* YC_bC_r*'ye dönüşüm oldukça kolaydır.*

$$Y = 0.29900R + 0.58700G + 0.11400B$$
$$C_b = -0.16874R - 0.33126G + 0.50000B$$
$$C_r = 0.50000R - 0.41869G - 0.08131B$$

ve tekrar *RGB'ye dönüşüm*

$$R = 1.00000Y + 1.40200C_r$$
$$G = 1.00000Y - 0.34414C_b - 0.71414C_r,$$
$$B = 1.00000Y + 1.77200C_b$$

YC_bC_r dönüşümleri için birçok yöntem vardır. Bu, CCIR (International Radi Consultive Committee) 601-1 önerisidir ve JPEG sıkıştırmasında kullanılan tipik bir modeldir.

Görüntü Yakalama, Gösterme ve Saklama

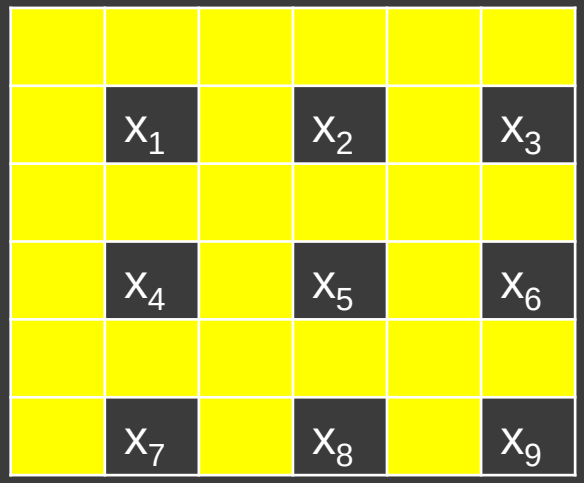
Görüntüler bilgisayarlarda 2-boyutlu sayı dizisi şeklinde depolanır. Bu sayılar, renk, grilik seviyesi yoğunluğu, parlaklık, krominans gibi farklı bilgilerle alakalı olabilir.

Bir görüntüyü bilgisayarda işlemeden önce, görüntüyü dijital forma dönüştürmemiz gerekir. Bu işlem dijitalleştirici gerektirir. En yaygın kullanılan dijitalleştiriciler tarayıcılar (scanners) ve dijital kameralardır. Dijitalleştirici iki fonksiyona sahiptir; örnekleme ve kuantalama. **Örnekleme** bir veriyi aralıklı veri noktaları ile temsil eder. Bu veri noktaları bilgisayarda saklanması gerektiği için, ikili (binary) biçime dönüştürülmesi gerekir. **Kuantalama** her bir değeri ikili bir değere atar.

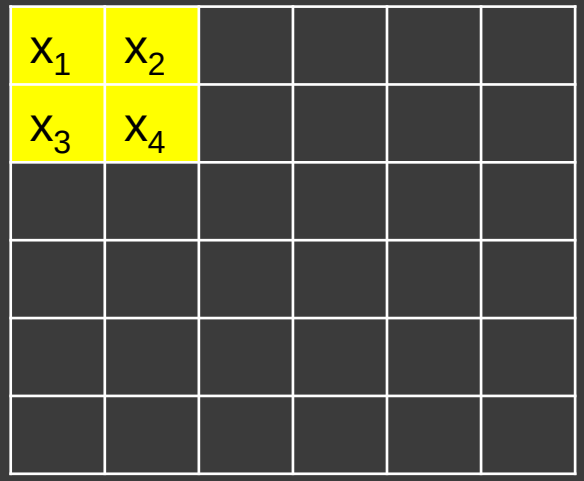
Aşağıdaki şekil bir görüntünün çözünürlüğünün azaltılmasının etkisini gösterir. Her bir kare onun etrafındaki alanının ortalama parlaklık değerini temsil eder.

Değişik örnekleme miktarları: (a) 512x512, (b) 128x128, (c) 64x64, (d) 32x32.

Görüntüyü iki kat küçültme için iki yöntem

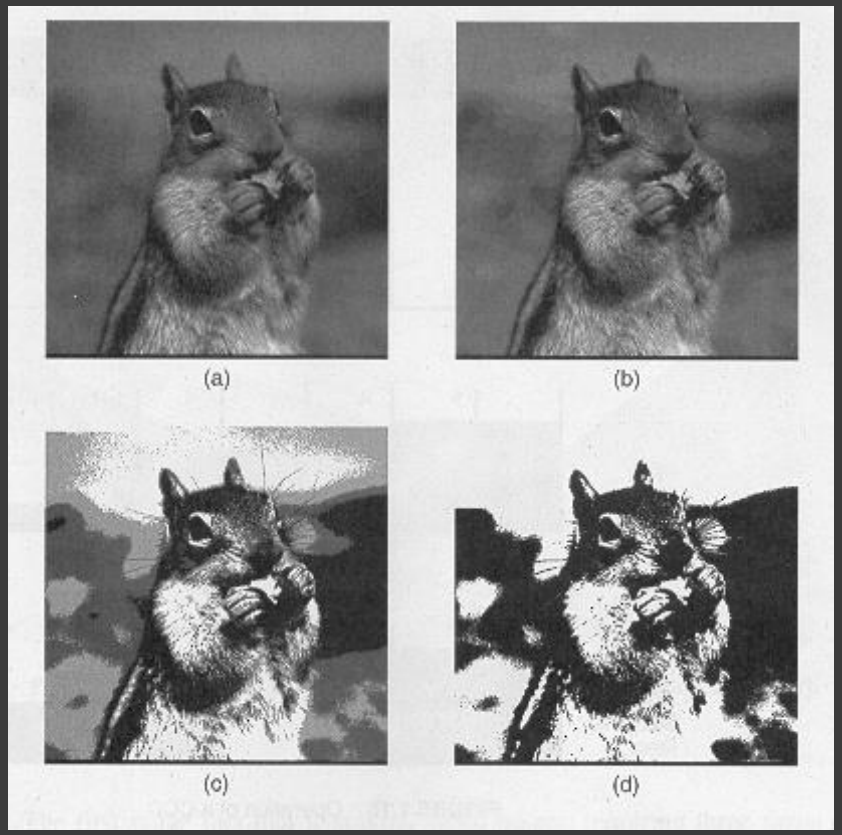


x_1	x_2	x_3
x_4	x_5	x_6
x_7	x_8	x_9



$$y=(x_1+x_2+x_3+x_4)/4$$

Aşağıdaki şekil bir görüntünün kuantalamasında kullanılan bitlerin sayısının azaltılmasının etkisini gösterir. 4-bitlik ve daha az olan görüntülerde yanlış biçimleme veya tonlandırma denilen etki oluşur.



Değişik kuantalama seviyeleri: (a) 6 bit; (b) 4 bit; (c) 2 bit; (d) 1 bit.

8 bitlik bir görüntüde 0'dan 255'e kadar 256 tane grilik seviyeleri vardır ($_2{}^8$).

6 bitlik bir görüntüde 64 tane grilik seviyesi vardır.

0-3 ⟶ 0

4-7 ⟶ 4

8-11 ⟶ 8

. . .

. . .

4 bitlik bir görüntüde 16 tane grilik seviyesi vardır.

0-15 ⟶ 0

16-31 ⟶ 16

32-47 ⟶ 32

. . .

. . .

Bir resim sürekli bir görüntünün dijitalleştirilmiş hali ile temsil edilir. Resim örneklendikçe dijitalleştirici ışığı parlaklığı temsil eden bir işarete dönüştürür. Bir dönüştürücü (transducer) bu dönüşümü yapar. Bir analogdan dijitale dönüştürücü (AID) dijital olarak saklanabilen veriyi üretmek için bu işareti kuantalar. Bu veri yoğunluğunu temsil eder. Böylece, siyah 0 ile ve beyaz mümkün olan maksimum değer ile temsil edilir.

Term	Definition
binary image	Image containing only black and white pixels. In MATLAB, a binary image is represented as a logical array of 0's and 1's (which usually represent black and white, respectively). This documentation uses the variable name `BW` to represent a binary image in the workspace.
image type	Defined relationship between array values and pixel colors. The toolbox supports binary, indexed, intensity, and RGB image types.
indexed image	Image whose pixel values are direct indices into an RGB colormap. In MATLAB, an indexed image is represented by an array of class `uint8`, `uint16`, or `double`. The colormap is always an m-by-3 array of class `double`. This documentation uses the variable name `X` to represent an indexed image in the workspace, and `map` to represent the colormap.
intensity image	Image consisting of intensity (grayscale) values. In MATLAB, intensity images are represented by an array of class `uint8`, `uint16`, or `double`. While intensity images are not stored with colormaps, MATLAB uses a system colormap to display them. This documentation uses the variable name `I` to represent an intensity image in the workspace.

RGB image	Image in which each pixel is specified by three values -- one each for the red, green, and blue components of the pixel's color. In MATLAB, an RGB image is represented by an m-by-n-by-3 array of class `uint8`, `uint16`, or `double`. This documentation uses the variable name `RGB` to represent an RGB image in the workspace. This type of image is also known as a true-color image.
storage class	The numeric storage class used to store an image in MATLAB. The storage classes used in MATLAB are uint8, uint16, and double. The reference documentation for some functions includes a section called "Class Support" that specifies which image classes the function can operate on. When this section is absent, the function can operate on all supported storage classes.

İkili (Binary) Görüntüler

İkili bir görüntüde, her bir piksel sadece iki ayrı değerden biri ile gösterilir.

Temel olarak, bu iki değer açık (on) ve kapalı (off) ile alakalıdır. İkili bir görüntü 0'lar (off pixel) ve 1'ler (on pixel) olmak üzere iki boyutlu matrisler halinde saklanır.

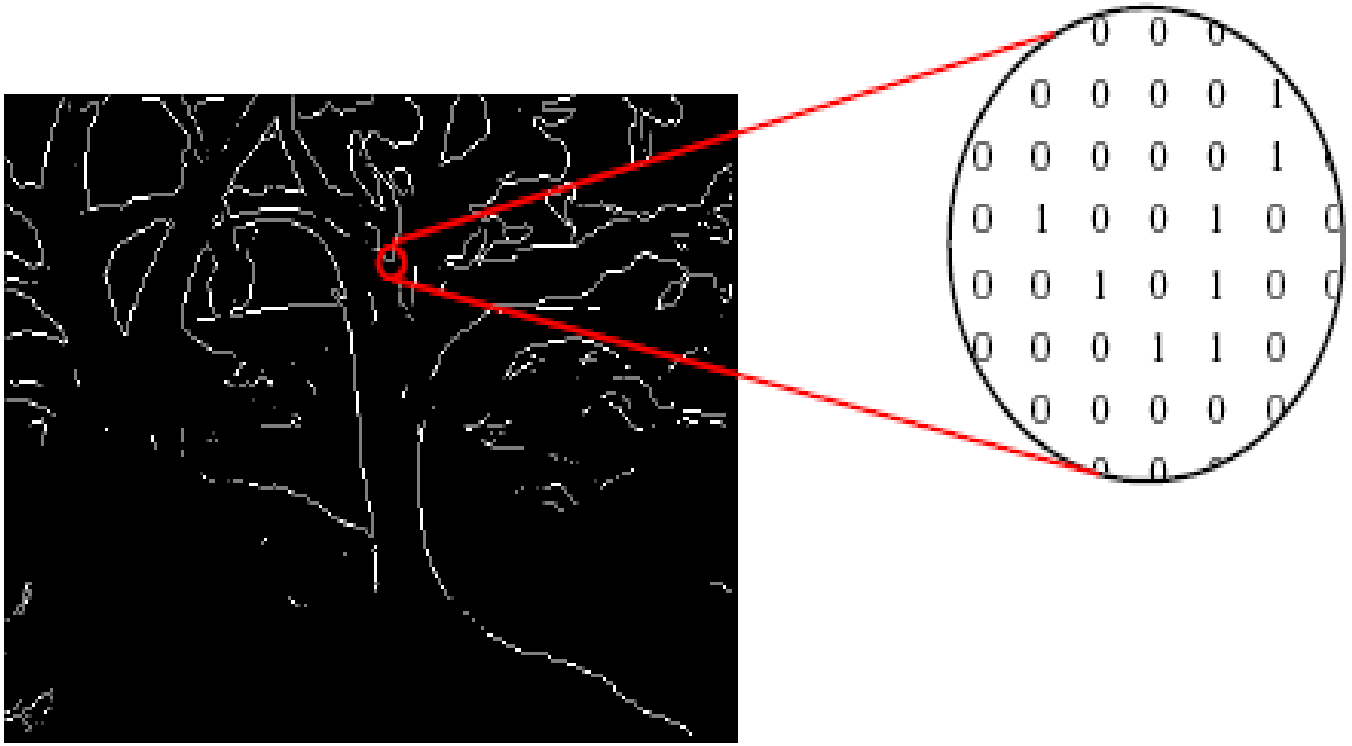
İkili bir görüntü keskin (intensity) görüntülerin özel bir biçimi olarak ele alınabilir. Diğer yorumlar da mümkündür, bununla birlikte ikili görüntüyü sadece iki renkli indeksli görüntü olarak da düşünebilirsiniz.

İkili bir görüntü double veya uint8 dizisi halinde saklanabilir. Bununla birlikte, bir uint8 dizisi tercih edilir, çünkü hafızada daha az yer kaplar. Görüntü İşleme Toolbox'ında, herhangi bir fonksiyon ikili görüntü olarak geri dönebilir. Toolbox [0,1] aralığında mantıksal bayrak kullanır (Eğer mantıksal bayrak kapalıysa, Toolbox veri aralığının [0,255] olduğunu varsayar).

İkili görüntüler sık olarak genel şekil veya taslak gibi bilgi gerektiren bilgisayar görüş uygulamalarında kullanılır. Bir nesneyi sürüklemek için robot sürükleyiciyi pozisyona getirmek için, üretilmiş bir mamulü hatalara karşı denetlemek için, kopya (FAX) görüntüleri veya optik karakter tanımlamaları (OCR) gibi uygulamalar örnek olarak verilebilir.

İkili görüntüler genellikle eşik değeri üzerindeki her değer için pikselin beyaza (‘1’) ve altındakilerin siyaha (‘0’) döndüğü eşik işlemleriyle ilgili gri-skala görüntülerinden oluşturulur.

This figure shows an example of a binary image.



Şekil ikili bir görüntü örneğini gösterir.

RGB Görüntüleri

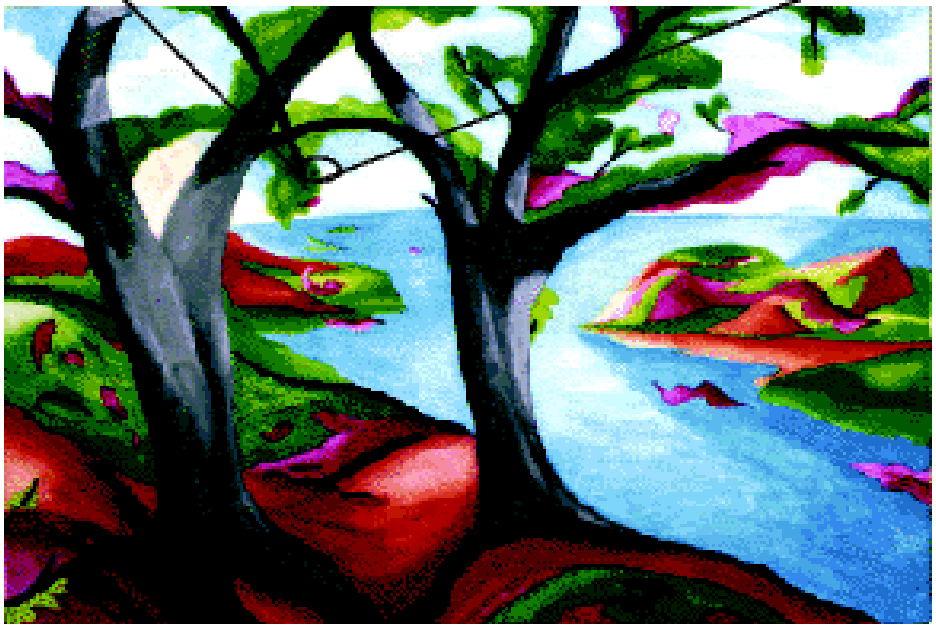
Bir RGB görüntüsü, (sometimes referred to as a true-color image) MATLAB'de her piksel için kırmızı, yeşil ve mavi renk bileşenlerinin tanımlandığı m-by-n-by-3'lik veri dizisi şeklinde saklanır.

RGB görüntüleri palet kullanmaz. Her pikselin rengi pikselin yerindeki renk düzleminde saklanan kırmızı, yeşil ve mavi yoğunluklarının kombinasyonu ile belirlenir. Grafik dosya sistemi RGB görüntülerini kırmızı, yeşil ve mavi bileşenlerin her biri 8-bitlik olmak üzere, 24-bitlik görüntüler halinde saklar. Bu, 16 milyon tane rengin üretilmesi ile sonuçlanır. Gerçek hayattaki görüntülerin netliği true-color görüntülü ledlerin kullanılması ile sağlanabilir.

Bir RGB dizisi double, uint8, veya uint16 sınıfından olabilir. Double bir RGB dizisinde, her renk bileşeni 0 ve 1 arasında bir değer alır. Renk bileşenleri (0,0,0) olan bir piksel siyah rengi gösterirken (1,1,1) olan bir piksel beyaz rengi gösterir.

Her piksel için üç renk bileşeni üç boyutlu veri dizisi şeklinde saklanır. Örneğin (10,5) pikselinin kırmızı, yeşil ve mavi renk bileşenleri sırası ile RGB(10,5,1), RGB(10,5,2), and RGB(10,5,3) şeklindedir. Aşağıdaki görüntü bir double RGB görüntüsünü gösterir.

	0.2235	0.1294	**Blue**	0.4196		
3804	0.2902	**0.0627**	0.2902	0.2902	0.48	
0.5804	0.0627	0.0627	0.0627	0.2235	0.2588	
.5176	0.1922	0.0627	**Green**	0.1922	0.2588	0.2588
0.5176	0.1294	**0.1608**	0.1294	0.1294	0.2588	0.2588
0.5176	0.1608	0.0627	0.1608	0.1922	0.2588	0.2588
.5490	0.2235	0.5490	**Red**	0.7412	0.7765	0.7765
5490	0.3882	**0.5176**	0.5804	0.5804	0.7765	0.7765
490	0.2588	0.2902	0.2588	0.2235	0.4824	0.2235
	0.2235	0.1608	0.2588	0.2588	0.1608	0.2588
	2588	0.1608	0.2588	0.2588	0.2588	



İndeksli Görüntüler

İndeksli bir görüntü veri matrisi X'i ve renk-tablosu matrisini içerir. Veri matrisi uint8, uint16, veya double olabilir. Renk-tablosu matrisi [0,1] aralığında double ve floating-point değerlere sahip olan m-by-3'lük bir dizidir. Her bir satır kırmızı, yeşil ve mavi bileşenlerini gösterir. Her bir pikselin rengi X'in ilgili değerinin kullanılması ile belirlenir. 1 değeri ilk satırı, 2 değeri ikinci satırı gösterir, ve devam eder.

Bir renk-tablosu genellikle indeksli görüntüde saklanır ve imread fonksiyonu kullanıldığında otomatik olarak görüntü ile birlikte yüklenir. Bununla beraber, default veya isteğe bağlı olarak seçilen bir renk-tablosu kullanılabilir. Aşağıdaki şekil indeksli bir görüntünün yapısını gösterir. Görüntüdeki pikseller tamsayılarla temsil edilir.

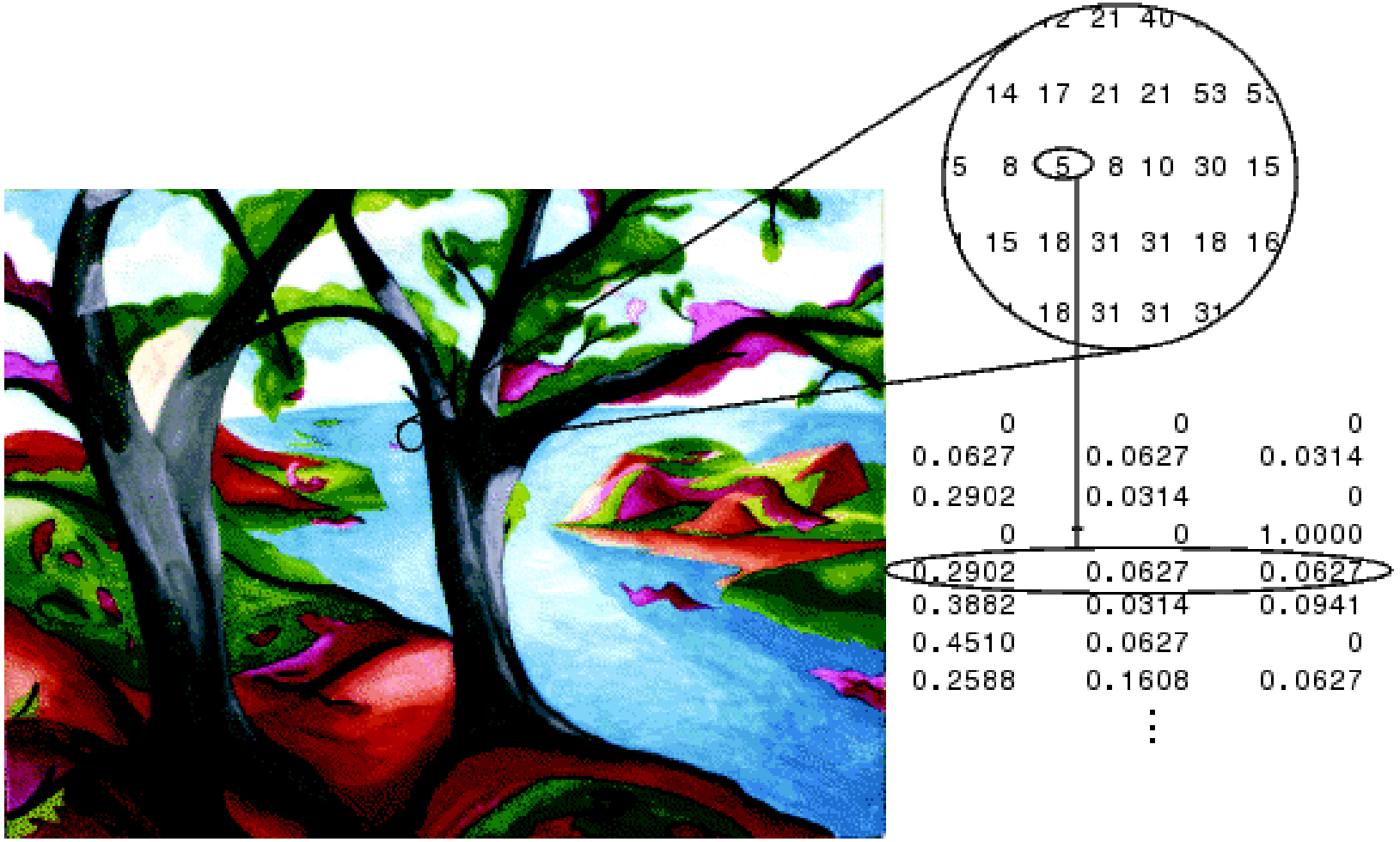


Image Courtesy of Susan Cohen

Relationship of Pixel Values to Colormap in Indexed Images

Gri Seviyeli Görüntüler

Gri-skala görüntüleri mono-krom veya tek renk olarak adlandırılan görüntülerdir. Bunlar renk bilgisi değil, sadece parlaklık bilgisi içerirler. Tipik bir görüntü 256 (0-255) değişik parlaklık (gri) seviyesine sahip olmamıza olanak sağlayan 8 bit/piksel veri içerir. Bu gösterim insan görme sisteminin gereksinimleri açısından yeterli parlaklık çözünürlüğünden fazlasını ve ayrıca gerektiği kadar gri seviye sağlayarak gürültü toleransı sağlar. Birçok değişik tipteki gürültü (sinyaldeki yanlış bilgi) gerçek sistemlerin doğasında bulunduğundan, bu gürültü toleransı gerçek dünya uygulamalarında yararlıdır. Ek olarak 8 bit temsili tipik olarak 8 bit veriye karşılık gelen ve sayısal bilgisayarlar dünyasında küçük bir ünite olan byte kelimesiyle tanımlanır. MATLAB gri seviyeli bir görüntüyü tek bir matris halinde saklar. Matrisin her bir elemanı bir görüntü pikselini gösterir. Matris double, uint8, veya uint16 sınıfından olabilir. Aşağıdaki görüntü bir double gri seviyeli görüntüyü göstermektedir.

	.2251	0.2563	0.2826	0.2826	0.4	
0.5342	0.2051	0.2157	0.2826	0.3822	0.4391	0.4391
0.5342	0.1789	0.1307	0.1789	0.2051	0.3256	0.2483
0.4308	0.2483	0.2624	0.3344	0.3344	0.2624	0.2549
	3344	0.2624	0.3344	0.3344	0.33	

Pixel Values in an Intensity Image Define Gray Levels

- Tıbbi görüntüleme ve astronomi gibi kesin uygulamalarda, 12 veya 16 bit/piksel gösterim kullanılmaktadır. Bu ekstra parlaklık seviyeleri sadece görüntünün küçük bir kısmı daha büyük yapıldığında işe yarar olmaktadır. Bu durumda bu ek parlaklık çözünürlüğü olmaksızın gözden kaçabilecek olan detayları görebiliriz. Tabii ki yararlı olabilmesi için, ayrıca yüksek seviyeli uzaysal çözünürlük (piksel sayısı) gerektirmektedir. Eğer bu parlaklık çözünürlüğü seviyelerinin ötesine geçersek ışık enerjisini görünebilir görüntü spektrumunun özel alt-bölümlerini oluşturan enerji bantlarına bölmüş oluruz.

Çok çerçeveli Görüntü Dizileri

Bazı uygulamalar için, magnetik rezonans görüntüleme (MRI) veya film görüntüleri gibi görüntü dizileri ile çalışmak zorunda kalabiliriz. Görüntü İşleme Toolbox'ı aynı dizide çoklu görüntü depolamayı sağlar. Her bir ayrık görüntü çerçeve (frame) olarak adlandırılır. Eğer bir dizi çoklu çerçeve barındırıyorsa, bunlar 4-boyutlu olarak ard arda bağlanır. Örneğin, 400x300 RGB görüntüye sahip olan beşli bir dizi, 400x300x3x5 şeklinde olacaktır. Benzer bir multiframe gri seviyeli veya indeksli görüntü, 400x300x1x5 şeklinde olacaktır. Bir multiframe dizisinde çoklu görüntü saklamak için cat komutu kullanılır. Örneğin, A1, A2, A3, A4 ve A5 görüntü gurubunuz varsa, bunları aşağıdaki şekilde tek bir dizide saklayabilirsiniz.

```
A = cat(4,A1,A2,A3,A4,A5)
```

Ayrıca multiframe görüntüden çerçeveleri çıkartabiliriz. Örneğin, MULTI adında çoklu görüntünüz varsa, aşağıdaki komut üçüncü çerçeveyi çıkarır.

```
FRM3 = MULTI(:,:,:,3)
```

Multiframe bir görüntü dizisinde, her bir görüntü aynı boyutta olmalıdır. Multiframe indeksli bir görüntüde, her bir görüntü aynı renk-tablosunu kullanmalıdır.

Çok çerçeveli modelin kısıtlılığı

Toolboxdaki birçok uygulama sadece ilk iki veya ilk üç boyutta işlem yapar. Bu fonksiyonlarla dört boyutlu dizileri de kullanabilirsiniz, fakat her bir çerçeveyi bireysel olarak işlemelisiniz. Örneğin, aşağıdaki çağırma dizideki yedinci çerçeveyi gösterir.

MULTI. imshow(MULTI(:,:,:,7))

Summary of Image Types and Numeric Classes

This table summarizes the way MATLAB interprets data matrix elements as pixel colors, depending on the image type and storage class.

Image Type	Storage Class	Interpretation
Binary	`logical`	Array of zeros (0) and ones (1)
Indexed	`double`	Array of integers in the range [1, *p*]. The associated colormap is a *p*-by-3 array of floating-point values in the range [0, 1].
	`uint8` or `uint16`	Array of integers in the range [0, *p*-1]. The associated colormap is a *p*-by-3 array of floating-point values in the range [0, 1].
Intensity	`double`	Array of floating-point values. The typical range of values is [0, 1]. The associated colormap, typically grayscale, is a *p*-by-3 array of floating-point values in the range [0, 1].
	`uint8` or `uint16`	Array of integers. The typical range of values is [0, 255] or [0, 65535]. The associated colormap, typically grayscale, is a *p*-by-3 array of floating-point values in the range [0, 1].
RGB (true-color)	`double`	m-by-n-by-3 array of floating-point values in the range [0, 1]
	`uint8` or `uint16`	m-by-n-by-3 array of integers in the range [0, 255] or [0, 65535]

Grafik Görüntülerini Okuma

imread fonksiyonu herhangi bir formattaki ve herhangi bir bitteki görüntüyü okur. Çoğu görüntü dosya formatı piksel değerlerini saklamak için 8 bit kullanır. Bunlar okunduğunda MATLAB bunları class uint8 olarak depolar.

```
f=imread('c:\bm.jpg');
```

Bu örnekte, imread dosya formatını dosya içeriğinden okur. MATLAB çoğu grafik dosya formatını destekler, bunlar; Microsoft Windows Bitmap (BMP), Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG) ve Tagged Image File Format (TIFF) formatlarıdır.

Bir Grafik Dosyasından Çoklu Görüntüleri Okuma

MATLAB, HDF ve TIFF gibi çoklu görüntü barındıran birçok grafik formatını destekler. Default olarak, imread fonksiyonu dosyadan sadece ilk görüntüyü alır. Dosyadan diğer görüntüleri almak için dosya formatı tarafından desteklenen syntax kullanılmalıdır. Örneğin, TIFF dosyaları kullanıldığında, almak istediğiniz dosyadaki görüntüyü tanımlayan bir indeks değeri kullanılabilir. Aşağıdaki örnek bir TIFF dosyasından 27 görüntü serisini okur ve görüntüleri dört-boyutlu bir dizide depolar. Dosyada kaç tane görüntü olduğunu tespit etmek için imfinfo komutu kullanılabilir.

```
mri = uint8(zeros(128,128,1,27)); % 4-D diziye yerleştirir

for frame=1:27
    [mri(:,:,:,frame),map] = imread('mri.tif',frame);
end
```

Grafik Görüntülerini Yazma

imwrite fonksiyonu desteklenen formatlardan birinde bir görüntüyü grafik dosyasına yazar. imwrite için çoğu temel syntax görüntü değişkenini ve dosya ismini alır. MATLAB dosyayı istenilen formata dönüştürüp istenilen yere yazabilir.

```
imwrite(f,'c:\b.bmp');
```

Dosya Formatını ve Parametresini Belirleme

imwrite bazı grafik formatları ile kullanılırken, ek parametreler belirlenebilir. Örneğin, PNG dosyaları ile, ek parametrelerle bit derinliğini belirleyebilirsiniz. Aşağıdaki örnek bir gri seviyeli *I* görüntüsünü 4-bitlik PNG dosyasına yazar.

```
imwrite(I,'clown.png','BitDepth',4);
```

Aşağıdaki örnek, sıkıştırma kalitesi parametresini belirlemek için bir ek parametre kullanarak A görüntüsünü JPEG dosyasına yazar.

```
imwrite(A, 'myfile.jpg', 'Quality', 100);
```

1-bitlik formatta ikili görüntüleri okuma ve yazma

Belirli dosya formatlarında, ikili bir görüntü 1-bitlik formatta saklanabilir. Eğer dosya formatı bunu destekliyorsa, MATLAB default olarak ikili görüntüyü 1-bitlik görüntü halinde yazabilir. İkili bir görüntüyü 1-bitlik bir formatta okuduğumuzda, MATLAB bunu workspace'de mantıksal bir dizi olarak saklar. Aşağıdaki örnek, ikili görüntüyü okur ve onu TIFF dosyası olarak yazar. Çünkü TIFF dosya formatı 1-bitlik görüntüleri destekler, dosya hafızaya 1-bitlik formatta yazılır.

```
BW = imread('text.png');

imwrite(BW,'test.tif');
```

test.tif dosyasının bit derinliğini doğrulamak için, **imfinfo** komutu çağırılır ve info.BitDepth kontrol edilir.

```
imfinfo('test.tif');

info.BitDepth
ans =

     1
```

imfinfo fonksiyonu toolbox tarafından desteklenen herhangi bir formatta olan grafik dosyaları ile ilgili bilgi edinmemizi sağlar. Elde edilen bilgi dosya tipine bağlıdır, fakat en azından aşağıdakileri içerir:

- Dosyanın ismi
- Dosya formatı
- Dosya formatının versiyon sayısı
- Dosya değiştirme tarihi
- Byte'lar halinde dosya boyutu
- Görüntü piksel genişliği
- Görüntü piksel yüksekliği
- Her pikseldeki bit sayısı
- Görüntü tipi: RGB (true-color), gri seviyeli (grayscale), veya indeksli

```
>> info=imfinfo('c:\b.bmp')

info =

              Filename: 'c:\b.bmp'
          FileModDate: '29-Jan-2009 22:13:25'
             FileSize: 15116598
               Format: 'bmp'
        FormatVersion: 'Version 3 (Microsoft Windows 3.x)'
                Width: 2592
               Height: 1944
             BitDepth: 24
            ColorType: 'truecolor'
      FormatSignature: 'BM'
   NumColormapEntries: 0
             Colormap: []
              RedMask: []
            GreenMask: []
             BlueMask: []
      ImageDataOffset: 54
     BitmapHeaderSize: 40
           NumPlanes: 1
      CompressionType: 'none'
           BitmapSize: 15116544
       HorzResolution: 0
       VertResolution: 0
        NumColorsUsed: 0
   NumImportantColors: 0

>>
```

Grafik Dosyalarını Dönüştürme

Bir görüntünün grafik formatını dönüştürmek için imread fonksiyonu kullanılarak görüntü okunur ve imwrite fonksiyonu ile istenilen uygun formatta saklanabilir.

Aşağıdaki örnek bir BMP görüntüsünü okumak için imread fonksiyonunu kullanır, daha sonra imwrite komutu ile dosyayı PNG formatında yazar.

```
bitmap = imread('mybitmap.bmp','bmp');

imwrite(bitmap,'mybitmap.png','png');
```