

F.Ü. Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümü

Bilgisayar Sistemleri Laboratuvarı

DENEY NO : 6

DENEY ADI : GÖRÜNTÜ İŞLEME UYGULAMALARI

1. GİRİŞ

Elde edilen görüntünün bilgisayara aktarılıp üzerinde herhangi bir işlem yapılması ve ardından görüntüleyici çıkışa iletilmesine sayısal görüntü işleme adı verilir. Görüntü işleme teknikleri ile sayısal görüntüler iyileştirilerek nesne tanıma, hedef tanıma gibi işlemler gerçekleştirilebilmektedir. Görüntü işleme, amaca göre çeşitli işlemlerden oluşmaktadır. Görüntü işleme; tasarım, imalat, güvenlik, tıp, elektronik, makine ve jeodezi gibi alanlarda çok geniş bir uygulama alanı bulmuştur. Aşağıda görüntü işlemenin kullanıldığı alanlardan bazıları, uygulamalarıyla beraber verilmiştir.

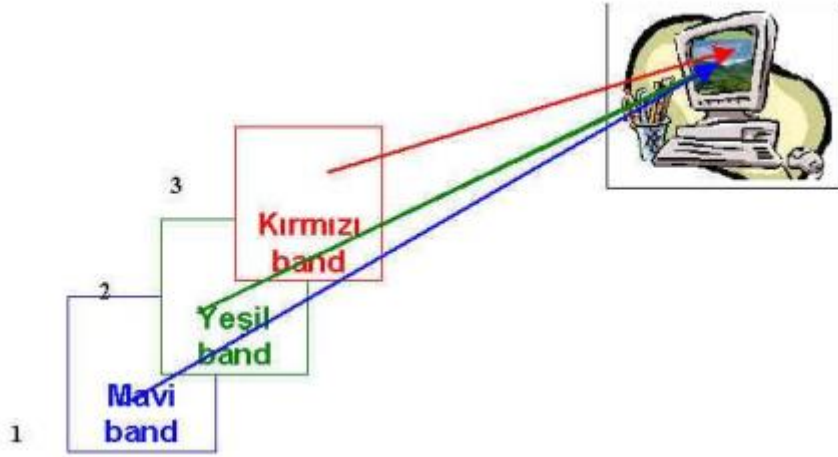
- Askeri (hedef tanıma, izleme)
- Tıp (damar analizi, bilgisayarlı tomografi, ultrason)
- Güvenlik (nesne takibi, hareket algılama, yüz tanıma)
- Trafik (trafik kontrol, plaka tanıma, trafik ışığı (işareti) tanıma)
- Endüstri (nesne sayma, kalite kontrol, robotik uygulamalar)
- Tarımsal uygulamalar (ekin verimliliği tespiti)
- Astronomi (uydu görüntüleri ile hava tahmini)
- Jeodezi ve Fotogrametri (uzaktan algılama)
- Perakende (insan sayma, davranış analizi, mağaza izleme)
- Çevre güvenliği (çevre kirliliği tespiti)

Sayısal resim haline getirilmiş olan gerçek yaşam görüntüsünün, girdi resim olarak işlenerek, resmin özelliklerinin ve görüntüsünün değiştirilmesi sonucunda yeni bir resmin oluşturulması işlemdir. Birçok amaç için kullanılmaktadır.

- Görüntü İyileştirme
- Görüntü Sıkıştırma
- Biyometrik Tanıma
- Otomatik yüz araç obje tanıma ve takip etme

2. İKİLİ GÖRÜNTÜ (BINARY IMAGE)

Bir resmin sayısallaştırılmasının açıklanması amacı ile öncelikle Siyah-Beyaz resim göz önünde bulundurulmuştur. Siyah-Beyaz resim sadece iki gri değerden oluşan bir resimdir. Böylesi bir görüntüde her bir piksel ya siyah ya da beyaz olarak oluşur. Burada sembolik olarak beyaz pikseller 1, siyah pikseller 0 değeri ile gösterilecektir. Renkli görüntüler bilgisayar ekranlarında 24 bit lik veri olarak görüntülenir. Görüntüleme R(Kırmızı), G(Yeşil), B(Mavi) kodlanmış aynı objeye ait üç adet gri düzeyli görüntünün üst üste ekrana iletilmesi ile oluşur. Elektro-manyetik spektrumda 0,4-0,5 mm dalga boyu mavi renge; 0,5-0,6 mm dalga boyu yeşil renge; 0,6-0,7 mm dalga boyu kırmızı renge karşılık gelir. Bu dalga boylarında elde edilmiş üç gri düzeyli görüntü bilgisayar ekranında sırası ile kırmızı-yeşil-mavi kombinasyonunda üst üste düşürülecek olursa renkli görüntü elde edilmiş olur.



Şekil 1. RGB Görüntü

3. NİTELİKLENDİRME

Görüntünün piksel değerlerinin belirli aralıklarda olması, meydana gelen görüntünün niteliğini değiştirir. Örneğin 0 beyazı ve n1 de siyahı temsil ederse ve bu değerler arası gri tonlarını ifade eder. Burada $n = 2^b$ olmak üzere, b değeri görüntünün 1 pikselini ifade etmek için gereken bit sayısıdır. Örneğin $b=8$ ise 256 adet gri tonu bulunmaktadır. $b=1$ ise resim sadece 0 ve 1 'lerden oluşur ve buna İkili resim(Binary Image) denir.

4. MATLAB İLE GÖRÜNTÜ İŞLEME

Söz konusu uygulamaları geliştirmek için kullanılan, Matlab'ın görüntü işleme komutlarının ve image processing tools'un bazılarını örneklerle inceleyeceğiz.

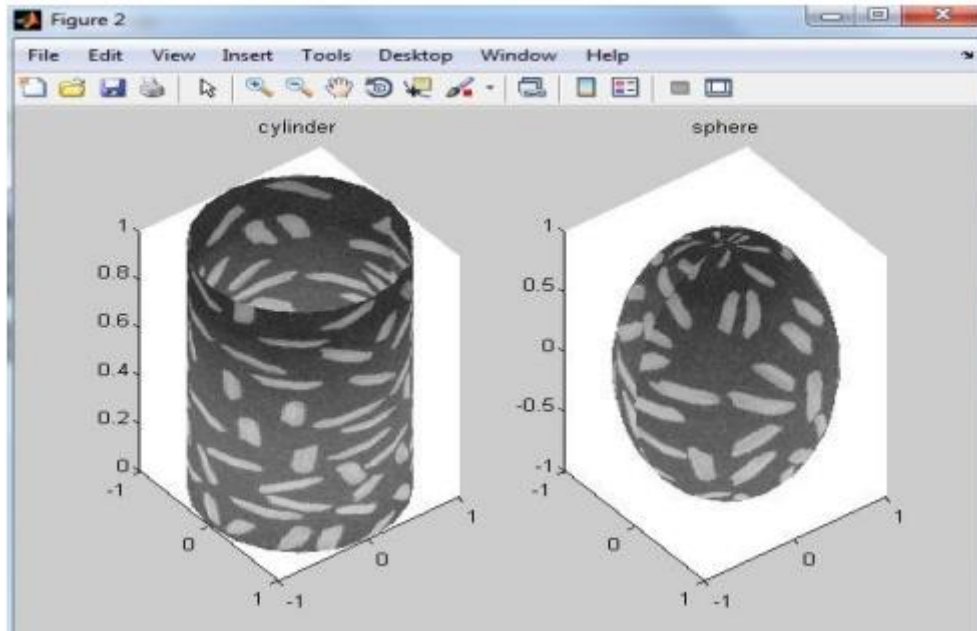
imread: imread ile üzerinde çalışılmak istenilen fotoğraf bir değişkene atanarak matlab workspace için tanımlanır ve ardından imshow ile pencerede gösterilebilir. Workspace'te imgenin boyutu, çözünürlüğü gibi bilgileri görebilirsiniz.

imtool: imtool ile bazı işlemlerin yapılabileceği bir pencere açılır. Örnek kodları yazarken workspacedeki verileri temizlemek için, clear all ve clc komutlarını en başa yazmayı unutmamamız gerekir.

```
I = imread('ornek.jpg')
imshow(I)
imtool(I)
```

warp: görüntüyü (içinde yazı olan görüntü daha iyi olur) belli yüzeylerde gösterir.

```
resim = imread('rice.png');
[x1,y1,z1] = cylinder;
[x2,y2,z2] = sphere;
imshow(resim); title('Orijinal resim');
figure;
subplot(1,2,1);
warp(x1,y1,z1,resim);
title('cylinder');
subplot(1,2,2);
warp(x2,y2,z2,resim);
title('sphere');
```



Şekil 2. Elde Edilen Görüntü

4.1. Tür Dönüşüm Kodları

rgb2gray: renkli görüntüyü gri seviyeye çevirir.

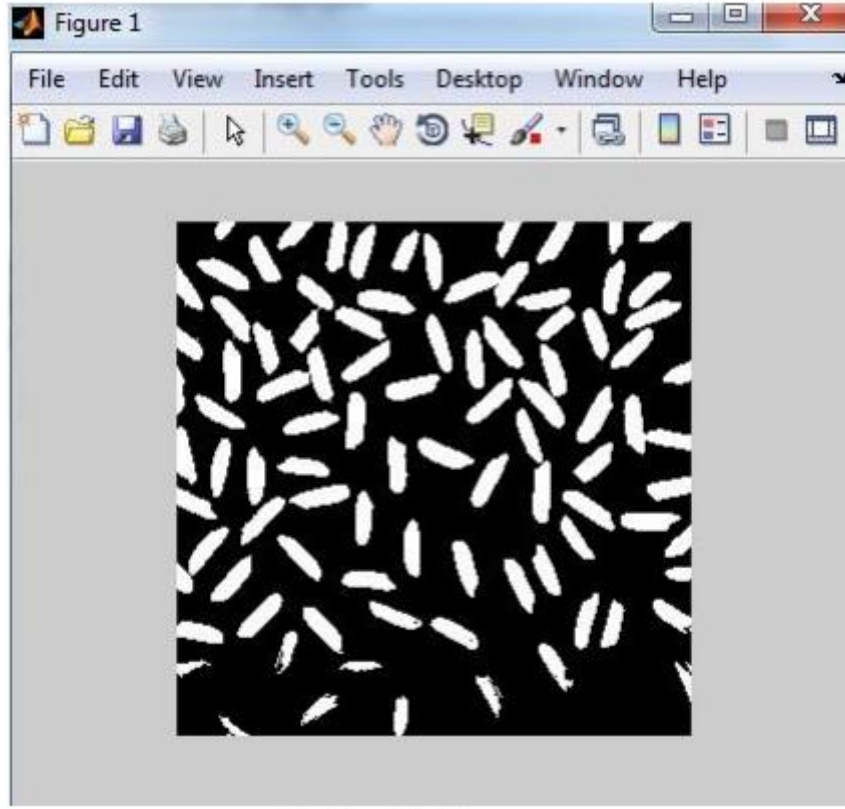
graythresh: graythreshold komutu görüntüdeki parlaklık eşiğini otomatik olarak belirler ve sonuç olarak 0 – 1 arasında bir sayı (level) oluşturur. Görüntüdeki parlaklık sınırları ile yapacağımız işlemlerde graythresh’ tan elde ettiğimiz sayıyı kullanırız. Görüntü üzerinde belirli işlemleri yapabilmemiz için öncelikle gri seviyede çalışmamız gerekir. Bunun için **I=rgb2gray(I)** komutu kullanılır.

im2bw: renkli görüntüyü ikili görüntüye çevirir. Kullanımı aşağıdaki gibidir.

```

I = imread('image.png');
level = graythresh(I);
bw = im2bw(I,level);
bw = bwareaopen(bw, 50);
figure; imshow(bw);

```



Şekil 3. Elde Edilen Görüntü

im2uint8: uint8 renk hassasiyetine dönüştürür. Genelde görüntüler uint8 türündedir. Her piksel işaretli 8 bit (0 – 255) arası değere sahiptir.

Diğer dönüşüm komutları; demosaic, gray2ind, grayslice, im2int16, label2rgb, im2double, im2uint16, mat2gray...

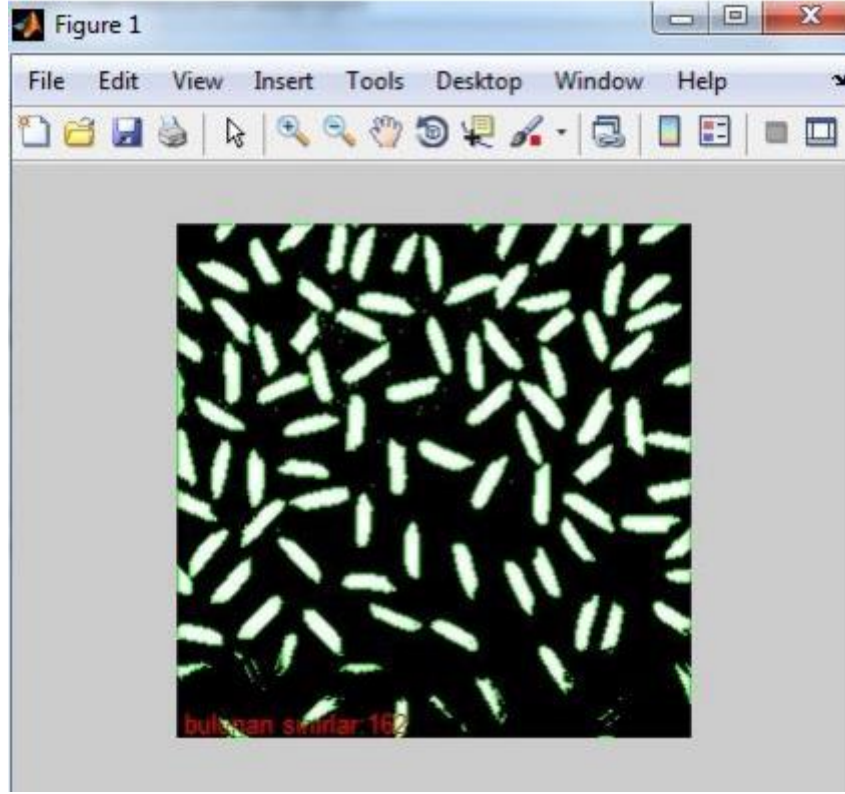
4.2. Görüntü Analizi Kodları

Bwboundaries: binary modda bölgelerin sınırlarını belirler. Bu örnekte, yeşil çizgiyle sınırlanmış alanların sayısını belirleyebiliriz. Bu örnekle daha sonra bahsedeceğimiz, morfolojik işlemler gerçekleştirilerek basit nesne sayma uygulaması yapılabilir.

```

I = imread('image.png');
BW = im2bw(I, graythresh(I));
B = bwboundaries(BW);
figure; imshow(BW);
text(10,10,strcat('\color{red}bulunan sınırlar:',num2str(length(B))))
hold on;
for k = 1:length(B)
    boundary = B{k};
    plot(boundary(:,2), boundary(:,1), 'g', 'LineWidth', 1)
end

```



Şekil 4. Elde Edilen Görüntü

Edge: özel filtreler yardımıyla gri seviyedeki görüntülerin sınırları belirlenir.

```
I = imread('image.png');  
imshow(I);  
BW1 = edge(I,'prewitt');  
BW2 = edge(I,'canny');  
figure; imshow(BW1);  
figure; imshow(BW2);
```

Diğer analiz komutları: hough, houghlines, corner...

4.3. Görüntü İyileştirme Komutları

imadjust: görüntü yoğunluğu değerini ve renk haritasını ayarlar.

histeq: histogramı eşitleyerek kontrastı artırır.

adapthisteq: CLAHE(kontrast sınırlı adaptif histogram eşitleme) algoritması kullanarak kontrastı artırır.

medfilt2: 2 boyutlu medyan filtreleme yapar.

wiener2: 2 boyutlu adaptif gürültü temizleme filtresidir.

5. DENEY

Bu uygulamada verilen benzer iki resim arasındaki farklar bulunup bu farklar hem sayıca hesaplanmakta hem de işaretlenmektedir.

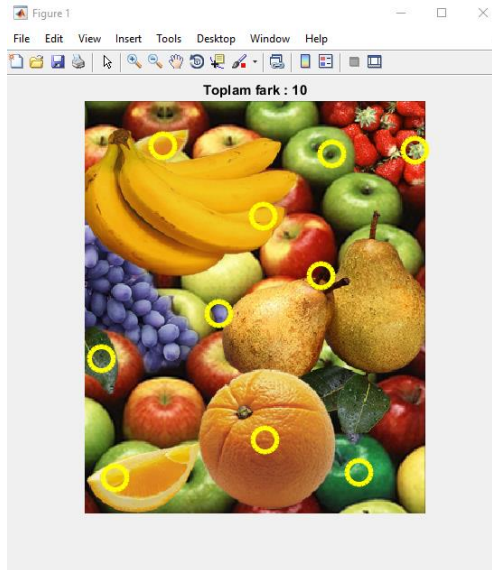


Şekil 5. Benzer GörSEL 1



Şekil 6. Benzer GörSEL 2

```
im1 = rgb2gray(imread('im1.png'));  
im2 = rgb2gray(imread('im2.png'));  
fark = imabsdiff(im1,im2);  
bw = bwareaopen(fark,55);  
bw = imfill(bw,'holes');  
SE = strel('square',1);  
bw2 = imerode(bw,SE);  
fark = regionprops(bw2, 'all');  
c = [fark.Centroid];  
imshow('im2.png');  
title(['Toplam fark : ',num2str(length(fark))]);  
hold on;  
x = c(1:2:end);  
y = c(2:2:end);  
plot(x,y,'yo','MarkerSize',20,'LineWidth',4);
```



Şekil 7. Elde Edilen Çıktı

Not: Deney esnasında deney föyündeki ve slayttaki tüm bilgi ve kod içeriğinden sorumlu olduğunuzu unutmayınız...

KAYNAKLAR

1. <https://www.mathworks.com/>
2. <https://www.elektrikport.com>
3. https://guraysonugur.aku.edu.tr/wp-content/uploads/sites/11/2019/03/GI-Ders4_1.pdf
4. <https://akademik.duzce.edu.tr/Content/Dokumanlar/pakizeerdogmus/Dosya/ab1930a7-d01f-44bf-9715-98261cbe69a4.pdf>