



## GÖRÜNTÜ İŞLEME YARDIMCI NOTLARI -GÜZ 2021-2022

### VİDEO ve CANLI GÖRÜNTÜ İŞLEMLERİ

Bir video ardışık resim kare (çerçeve, frame)'lerinden oluşur. Bu karelerin saniyede belli bir miktarda oynatılmasıyla elde edilir. Örneğin 24 fps (frame per second), 30 fps, 60 fps gibi.

MATLAB, kısa video klipler ve sınırlı sayıda video formatı kullanarak temel video işleme için gerekli işlevselliği sağlar. Kısa bir süre önce, yerleşik MATLAB işlevleri tarafından desteklenen tek video tipi \*.avi idi. Son dönemlerde \*.mp4 \*.wmv tipindeki video dosyaları da işlenebilmektedir.

MATLAB'ın matrisleri işleme yeteneği, video dizilerinin kısa olması koşuluyla (birkaç dakikadan fazla olmayan video) sırasıyla monokrom ve renkli videoyu temsil etmek için 3D veya 4D veri yapılarını oluşturmayı ve değiştirmeyi kolaylaştırır.

Ayrıca, bir çerçevenin tek tek işlenmesi gerektiğinde, frame2im() işlevi kullanılarak bir görüntüye dönüştürülebilir ve bu daha sonra Görüntü İşleme Araç Kutusu'nda (IPT) bulunan işlevlerden herhangi biri kullanılarak işlenebilir.

Son olarak, başka bir MathWorks ürünü olan Simulink'in, MATLAB ve en yakın araç kutuları, özellikle IPT ve Görüntü Toplama Araç Kutusu (IAT) ile entegre edilebilen bir Video ve Görüntü İşleme Blok Seti içermektedir.

MATLAB'da video işleme aşağıdaki adımları içerir:

- Videoyu okumak
- Videoyu görüntüleme
- Videonun işlenmesi
- Videoyu yazmak

### Matlab R2020 için...

Adım 1: Videoyu okumak

```
>> vid = VideoReader('filename.avi')
```

MATLAB, video işleme için web kameralarını desteklerken, Image Acquisition Toolbox™ birçok endüstriyel ve bilimsel kameradan canlı görüntü alınmasını sağlar.

Adım 2: Videoyu görüntüleme

```
load cellsequence  
imshow(cellsequence,10)  
  
imshow('rhinos.avi')
```



### Adım 3: Video İşleme

Bir video, ardışık tekil resimlerin oluşturduğu bir dizidir. İşlemek için tekil resimler üzerinde çalışmak gerekir.

```
>> vid = VideoReader('filename.avi')  
>> img=readframe(vid)  
>> kenar=edge(img, 'roberts');
```

### Adım 4: Video Yazma

Bir video içindeki çerçeveleri işledikten sonra tekrar bir dosyaya yazabiliriz.

```
>> vid_w = VideoWriter('newfile.avi');  
  
>> open(vid_w)
```

Matlab R2016 Video İşlemleri:

Bazı fonksiyonlar R2020 ile benzerlik gösterir. Ancak aşağıdaki şekilde ifade edilebilir:

#### **Video Okuma:**

```
filename = 'daria_walk.avi';  
mov = VideoReader(filename);  
bir_Kare=read(mov,1);
```

ya da

```
mov = VideoReader('daria_walk.avi');  
bir_Kare=read(mov,1);
```

#### **Video görüntüleme**

```
figure(1);
```

```
imshow(bir_Kare);
```



### Video İşleme:

```
mov = VideoReader('daria_walk.avi');  
bir_Kare=read(mov,1);  
gri=rgb2gray(bir_Kare);  
kenar=edge(bir_Kare,'prewitt');
```

### Video Yazma:

```
writerObj=VideoWriter('Ornek.avi');  
open(writerObj);  
. . .  
writeVideo(writerObj,kenar);  
close(writerObj);
```

### Video dosyasının ardışık okunması:

#### a.) Kenar bulma

```
clc;clear;close all;  
filename='rhinos.avi';  
  
mov=VideoReader(filename);  
  
for k=1:mov.NumberOfFrames  
    r=read(mov,k);  
    gri=rgb2gray(r);  
    kenar=edge(gri, 'canny');  
    figure(1);  
    imshow(kenar);  
end
```

#### b.) Yeşil renkli nesnelere bulunması

```
clc;clear;close all;  
filename='rhinos.avi';  
mov=VideoReader(filename);  
for k=1:mov.NumberOfFrames  
    x=read(mov,k);  
    green=x(:, :, 2);  
    r=rgb2gray(x);
```



```
fark=imsubtract (green,r);  
bw=fark>10;
```

```
bwUint8=im2uint8 (bw);  
greenOrj=bitand (bwUint8,x);
```

```
figure (1);  
subplot (121), imshow (x);  
subplot (122), imshow (bw);  
subplot (122), imshow (green);
```

end

Seçilen alanları kendi orijinal  
renginde göstermek için kullanıldı.

### c.) Yeşil renkli nesnelere bulunması 2. yol

```
clc;clear;close all;%yeşil bulma  
mov=VideoReader ('rhinos.avi');%okuma  
for k=1:100  
    x=read (mov,k);%okuma  
    gri=rgb2gray (x);  
    yesil=x (:,:,2);%yeşil bant ayrıldı  
    %renk=yesil-gri;  
    renk=imsubtract (yesil,gri);%yesil renk bulundu  
    BW=im2bw (renk,0.02);BW=im2uint8 (BW);  
    %BW=renk>10;  
    figure (1);  
    subplot (121);imshow (x);title ('orjinal');  
    yeni=bitand (BW,x (:,:,1));  
    yeni2=bitand (BW,x (:,:,2));  
    yeni3=bitand (BW,x (:,:,3));  
    x (:,:,1)=yeni;  
    x (:,:,2)=yeni2;  
    x (:,:,3)=yeni3;  
    subplot (122);imshow (x);title ('yeşil');  
end
```

Seçilen alanları kendi orijinal  
renginde göstermek için kullanıldı.

### d.) Hareketli nesne bulma

```
clc;clear;close all;  
filename='video1.mp4';  
  
mov=VideoReader (filename);  
artis=5;  
  
rEski=zeros (mov.Height,mov.Width);  
rEski=uint8 (rEski);  
  
for k=1:artis:mov.NumberOfFrames
```



```
x=read(mov,k);  
r=rgb2gray(x);  
fark=imsubtract(rEski,r);  
fark=fark>30;  
fark=im2uint8(fark);  
figure(1);  
subplot(121), imshow(x);  
subplot(122), imshow(fark);  
fprintf('Frame No:%d Fark Değeri:%d\n',k,sum(fark(:)));  
rEski=r;  
end
```

### Video Yazma

#### a.) Hareketli nesne çıkışının yazılması

```
clc;clear;close all;  
filename='video1.mp4';  
mov=VideoReader(filename);  
artis=5;  
rEski=read(mov,1);%ilk çerçeve okundu
```

```
writerObj=VideoWriter('Ornek.avi');  
writerObj.FrameRate=3;  
open(writerObj);
```

```
for k=2:artis:mov.NumberOfFrames  
x=read(mov,k);  
r=rgb2gray(x);  
fark=imsubtract(rEski,r);  
fark=fark>30;  
fark=im2uint8(fark);  
figure(1);  
subplot(121), imshow(x);  
subplot(122), imshow(fark);  
fprintf('Frame No:%d Fark Değeri:%d\n',k,sum(fark(:)));  
rEski=r;
```

```
writeVideo(writerObj,fark);
```

```
end
```

```
close(writerObj);
```

Video yazmak için kullanılan kodlar.

#### b.) Kenar bulma çıkışının yazılması

```
clc;clear;close all;  
filename='rhinos.avi';
```



```
mov=VideoReader(filename);

writerObj=VideoWriter('Ornek.avi');
writerObj.FrameRate=3;
open(writerObj);

for k=1:mov.NumberOfFrames
    r=read(mov,k);
    gri=rgb2gray(r);
    kenar=edge(gri, 'canny');
    figure(1);
    imshow(kenar);
    writeVideo(writerObj,kenar);
end
close(writerObj);
```

## ARDUINO ile DIŞARIYA SİNYAL GÖNDERME

Arduino geliştirme kartını kullanarak yaptığımız görüntü işleme programlarından dış dünyaya ses ya da ışık gibi bir takım sinyaller gönderebiliriz. Aşağıdaki programda video1.mp4 videosunda ekrana hareketli bir nesne girdiği anda alarm üreten bir mekanizma gerçekleştirilmiştir.

```
clc;clear;close all;
%Arduino kartı matlab'a tanıtılıyor..
a=arduino('COM3','Uno');
filename = 'video1.mp4';
mov = VideoReader(filename);
artis=3;

%Bir önceki çerçeve başlangıç için boş olarak açılıyor.
rEski=read(mov,1);

%okuma işlemi for döngüsünde başlıyor.
for k=2:artis:mov.NumberOfFrames;
r=read(mov,k);
r=rgb2gray(r);
fark=imsubtract(rEski,r);
fark=fark>30;
%aşğıdaki kod ile her fark çerçevdeki fark piksellerin
toplamı alınıyor.
%En yüksek fark iki çerçeve arasındaki en yoğun değişimi
gösterir.
fprintf('Frame No=%d Fark Pikseller=%d\n',k,sum(fark(:)));
figure(1);
imshow(fark);
if sum(fark(:))>500
    writeDigitalPin(a,'D7',1)
```



```
else  
    writeDigitalPin(a, 'D7', 0)  
end  
rEski=r;  
end
```

## CANLI VİDEO OKUMA İŞLEMLERİ

Bilgisayarın USB portuna takılacak bir webcam ile gerçek zamanlı okuma yapabilir ve elde ettiğimiz görüntüleri işleyebiliriz. Ardından işlenmiş görüntüyü kaydedebiliriz.

Aşağıdaki programda gerçek zamanlı olarak kameradan alınan görüntülerin kenarları çıkarılarak yeni bir isimle kaydedilmektedir.

```
%% Setup  
clear;clc;close all;  
% 1. webcam'a bağlanılıyor.  
cam = webcam(1);  
%%tek görüntü alma  
% kameradan sadece bir görüntü alıyoruz.  
img = snapshot(cam);  
imshow(img)  
  
%% Video  
set(gcf,'Position',[200 10 600 600]);%figür penceresinin yeri ayarlandı.  
vidWriter = VideoWriter('Kenar_Canny.avi');  
open(vidWriter);  
% 500 çerçeve alalım  
for frames = 1:100  
% tek bir çerçeve alıyoruz  
a = snapshot(cam);  
a=rgb2gray(a);  
kenar=edge(a, 'canny');  
figure(1);  
subplot(1,2,1); imshow(a);  
subplot(1,2,2); imshow(kenar);  
writeVideo(vidWriter,a);  
end  
close(vidWriter);  
clear cam
```



## Arduino ile gerçek zamanlı görüntü işleme

Kırmızı renk gördüğünde led yakan bir uygulama:

```
%% Setup
clear;clc;close all;
% 1. webcam'a bağlanılıyor.
cam = webcam(1);
ard=arduino('COM3','Uno');
% Video
set(gcf,'Position',[200 10 600 600]);%figür penceresinin yeri
ayarlandı.
vidWriter = VideoWriter('sinifRenk.avi');%Canlı çekilen
videoyu kaydediyoruz.
open(vidWriter);
% 100 çerçeve alalım
for frames = 1:100
% her döngüde tek bir çerçeve alıyoruz
a=snapshot(cam);
red=a(:,:,1);
a=rgb2gray(a);
kirmizi=imsubtract(red,a);
BW=kirmizi>50;

if sum(BW(:))>1000
    writeDigitalPin(ard,'D7',1);
else
    writeDigitalPin(ard,'D7',0);
end

figure(1);
subplot(1,3,1); imshow(a);
subplot(1,3,2); imshow(BW);
subplot(1,3,3); imshow(kirmizi);
%siyah-beyaz'a dönüştürülen kırmızı renkli nesnelere video
olarak kaydediliyor.
writeVideo(vidWriter,im2uint8(BW));
end
writeDigitalPin(ard,'D7',0);
close(vidWriter);
clear cam
```

Canlı görüntü Okuma

Video işleme

Arduino kontrol

Canlı Video  
görüntüleme

Video Yazma





## Canlı Video İşlemleri (Çift kamera kullanımı)

Aşağıdaki programda hem usb kamera hem de dizüstü bilgisayarın entegre kamerası aynı anda kullanılmıştır.

```
%% Setup
% Çift kamera kenar bulma
clear;clc;close all;
% 1. ve 2. webcam'a bağlanılıyor.
cam1 = webcam(1);
cam2 = webcam(2);
% Video
set(gcf,'Position',[200 10 700 700]);%figür penceresinin yeri
ayarlandı.
vidWriter = VideoWriter('CiftKamera.avi');%Canlı çekilen
videoyu kaydediyoruz.
vidWriter.FrameRate=3;
open(vidWriter);
% 100 çerçeve alalım
for frames = 1:100
% her döngüde tek bir çerçeve alıyoruz
a=snapshot(cam1);
b=snapshot(cam2);
kenar1=edge(rgb2gray(a),'canny');%çıktısı logical
kenar2=edge(rgb2gray(b),'canny');%çıktısı logical
son=[rgb2gray(a) rgb2gray(b);im2uint8(kenar1)
im2uint8(kenar2)];%Dört görüntü birleştirildi
figure(1);
subplot(1,3,1); imshow(a);
subplot(1,3,2);imshow(b);
subplot(1,3,3);imshow(son);
%siyah-beyaz'a dönüştürülen kırmızı renkli nesnelere video
olarak kaydediliyor.
writeVideo(vidWriter,son);
end
close(vidWriter);
clear cam
```



AKÜ TEKNOLOJİ FAKÜLTESİ  
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ

