



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 (greenOrj);
```

end

Seçilen alanları kendi orijinal renginde göstermek için kullanıldı.

c.) Hareketli nesne bulma

```
clc;clear;close all;  
filename='videol.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

```
clc;clear;close all;  
filename='videol.mp4';
```

```
mov=VideoReader (filename);  
artis=5;
```

```
rEski=zeros (mov.Height,mov.Width);  
rEski=uint8 (rEski);
```



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

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

Video yazmak için kullanılan kodlar.

ARDUINO ile DIŞARIYA SİNYAL GÖNDERME

Arduino geliştirme kartını kullanarak yaptığımız görüntü işleme programlarından dış dünyaya ses ya da ışık gibi bir takım sinyaller gönderebiliriz.

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

```
%Bir önceki çerçeve başlangıç için boş olarak açılıyor.  
rEski=zeros(mov.Height,mov.Width);  
rEski=uint8(rEski);
```

```
%okuma işlemi for döngüsünde başlıyor.
```

```
for k=1:artis:mov.NumberOfFrames;  
    r=read(mov,k);  
    r=rgb2gray(r);  
    fark=imsubtract(rEski,r);  
    fark=fark>30;
```

```
%şağıdaki kod ile her fark çerçevedeki fark piksellerin toplamı alınıyor.
```

```
%En yüksek fark iki çerçeve arasındaki en yoğun değişimi gösterir.
```



AKÜ TEKNOLOJİ FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ



```
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
```