

# Horn ve Schunck Optik Akış yöntemi ile hareket vektörlerinin gerçek zamanlı veya videolar üzerinden gerçekleştirilmesi.(Matlab)

**Dersin Adı:** Sayı İşaret İşleme Tas.&Uyg.

**Sınıf Öğitmeni:** Bilge Günsel Kalyoncu

**Hazırlayan:** Ahmet Haluk Açarççek **No:** 040100487

## 1 ÖZET

---

*<sup>1</sup>Berthold K.P. Horn ve Brian G. Schunck'ın 1981 yılında Artificial Intelligence Laboratuvarından yayınladıkları makaleye dayanarak MATLAB ve Python ortamlarında görüntülerin hareket vektörlerini bularak hareket vektörlerini hesaplayarak çeşitli farklı algoritmalarla ve algoritmada kullanılan parametrelerin farklı değerleriyle karşılaştırmaları.*

## 2 Giriş

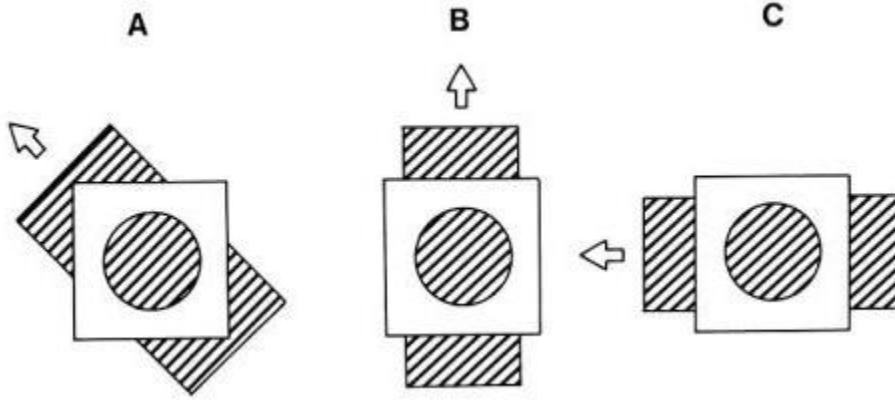
---

Matematiksel formülasyon kısmında karşılaşacağımız bazı terimlerde yabancılık çekilmemesi açısından aşağıda kısaca bir giriş yapmamız uygun olur.

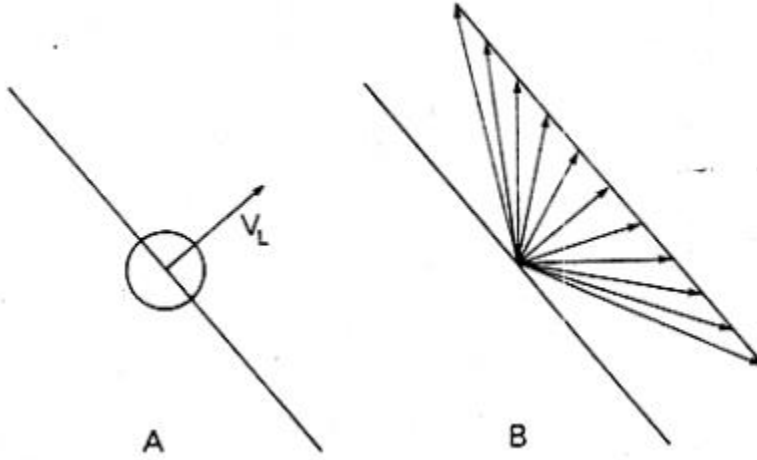
### 2.1 ARALIK (APERTURE) PROBLEMİ

Görüntü işlemedeki önemli problemlerden biri aralık(aperture) problemidir. Cisimlerin hareketini bulurken meydana gelen bu problem cismin global hareketini bulurken bizim durumumuzda her bir pnoktaya(kamerada sensöre) bakarak bulunamamasıdır. Bunun sebebi ise her bir nokta biraralıktan bakan sensörün algıladığı tek boyutlu bir değerdir. Bu tek boyutlu değere bakarak 2 boyutlu bir hareket çıkarmı yapabilmemiz için , yani aralık(aperture) problemini çözebilmemiz için çeşitli varsayımlar ortaya koymamız gerekmektedir.

- 
- <sup>1</sup> B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*, vol 17, pp 185–203, 1981. [Manuscript](#) available on MIT server.



Sadece yuvarlak aralıktan bakacak olursak, yukarıdaki dikdörtgen levhanın A,B veya C hareketlerinden hangisini yaptığını kestirmemiz olanaksızdır.



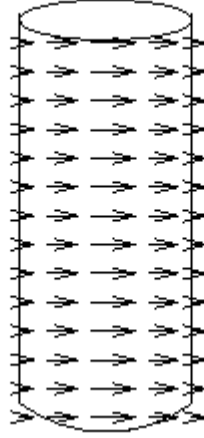
Yukarıdaki yuvarlak aralığımızı tek nokta halinde bize 1 boyutlu bir sonuç vermektedir. Bu sonuca göre  $V_L$  vektörünün yönünü kestirmemiz aynı şekilde olanaksızdır.

## 2.2 OPTİK AKIŞ(OPTICAL FLOW)

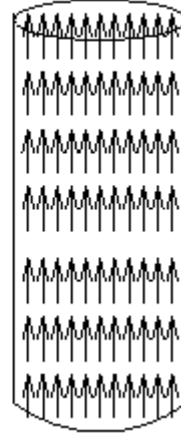
Hareket algılama, sanayide, akademik araştırmalarda, tıpta hatta son kullanıcıya yönelik ürünlerde bile sıklıkla kullanılan bir görüntü işleme dalıdır. Bu yüzden de farklı ihtiyaçlara yönelik bu konuda çeşitli yöntemler ve teknikler geliştirilmiştir. Bu tekniklerden biri olan optik akış ise kolay gerçekleştirilebilir olması ve basitçe anlaşılabilmesiyle video sıkıştırma, objelerin hareket vektörlerinin bulunması vb. gibi uygulamalarda sıklıkla kullanılan bir yöntem olmuştur. Bu makalede ilgilenilecek olan hareket vektörü bulunması konusunda ise optik akışa göre aslında gerçek hareketin gözlemciye göre yansımaları denebilir. Bu yüzden optik akış her durumda gerçek hareketi bize veremese de çoğu uygulamada yakın ve faydalı sonuçlar üretir.



Optik akışın gerçek hareketle örtüşmediği durumlardan biri



Gerçek Hareket



Optik akış vektörleri

Üstteki durum haricinde örüntülerin hareket etmemesine rağmen üzerilerinden yansıyan ışık miktarının değişmesi veya örüntüleri hareket etmesine rağmen parlaklık düzeylerinin değişmemesi gibi durumlarda da optik akış gerçek hareketle örtüşmez. İleride de detaylı görebileceğimiz gibi optik akış gözlemciye göre görüntünün; gri düzey görüntü ise parlaklık ,renkli ise renk değerlerinin görüntü süresince değişmediği ve hareket eden cisimlerin kısa zamanda kısa hareketler yaptığı varsayımları optik akışın temellerini oluşturur. Fakat sadece bu varsayımlara dayanarak optik akış hesaplaması yapılamaz(bknz. Denklem 1). Daha doğrusu optik akışın bulunması için ortaya çıkan yöntemler bu noktadan sonra ayrılmaya başlar. Bu makale boyunca gri düzey görüntüler üzerinde çalışılacaktır.

## 2.3 HORN VE SCHUNCK YÖNTEMİ (HORN AND SCHUNCK METHOD)

Matematiksel formülasyon(3.1) bölümünden de görebileceğimiz gibi optik akışın bulunması için çeşitli varsayımlar yapılmalıdır. Bu yüzden iki ana grupta toplayabileceğimiz çeşitli optical flow hesaplama teknikleri ortaya çıkmıştır. Horn and Schunck yöntemi global ve lokal olarak bölünebilecek bu tekniklerden global yöntemlerde uzun süredir kullanılan bir yöntemdir. Ayrıca bu yöntemle yoğun optik akışın ,her piksele ait hareket vektörlerinin ayrı ayrı hesaplanması, bulunabilmesi sayesinde tercih edilir bir yöntem olmuştur. Horn ve Schunck'ın makalesinde de görülebildiği gibi optik akışın bulunması için ekstra bir koşul daha gereklidir. Bu koşul görüntüdeki örüntülerin sahip olduğu optik akış vektörlerinin görüntü boyunca ufak varyasyonlarla değiştiğini yani görüntü boyunca yumuşak bir akış olduğunu varsayar. Bu varsayımın gerçek görüntüler göz önüne alınarak ortaya atıldığı aşikardır. Görüntüdeki cismin her noktasında hareket boyunca teoride aynı hız vektörünün oluşması gerektiği açıktır. Fakat gerek gürültü gerekse görüntü kayıtlarındaki kuvantalama işlemleri yüzünden cisim boyunca hız vektörlerinde ufak değişimler meydana gelebilir. Biz ilerki aşamalarda bu değişimi minimum yapacak optik akış vektörlerini bulmaya çalışacağız(bknz denklem 2). Horn and Schunck'ın global bir yöntem olması dolayısıyla

cismin köşelerinde oluşan çeşitli sorunlar, ki varsayımı düşünerek böyle bir sorunun çıkacağını zaten anlayabiliriz, ve gürültü sebebiyle ortaya çıkan hatalar lokal yöntemlere göre daha belirgindir. Avantajına bakarsak ise lokal yöntemlerin aksine görüntü boyunca sadece yatay, dikey ve görüntüler arası zamansal türevler alınarak hesaplandığından daha hızlı bir yöntemdir.

### 3 MATEMATİKSEL FORMÜLASYON

---

#### 3.1 OPTİK AKIŞ DENKLEMİNİN BULUNMASI

Optik akış denklemi taylor serisi açılımı kullanarak veya diferansiyeldeki zincir kuralı kullanılarak bulunabilir. Kısaca zincir kuralına göre denklemi elde edecek olursak:

Görüntünün x,y koordinatlarında ve t anında parlaklık değeri  $E(x,y,t)$  ile gösterirsek bu koordinatlardaki görüntünün hareket etmesiyle parlaklık değeri değişmez. Yani:

$$\frac{dE}{dt} = 0 ,$$

Zincir kuralı kullanılarak,

$$\frac{\partial E}{\partial x} * \frac{dx}{dt} + \frac{\partial E}{\partial y} * \frac{dy}{dt} + \frac{\partial E}{\partial t} = 0 ,$$

x yönündeki optik akış hızı  $u = \frac{dx}{dt}$  ve y yönündeki optik akış hızı  $v = \frac{dy}{dt}$  olarak gösterilirse,

optik akış denklemini(denklem 1) :  $E_x * u + E_y * v + E_t = 0$  olarak elde ederiz.

Görüldüğü gibi en az iki adet görüntü ile optik akış denklemindeki u ile v haricindeki değerleri hesaplayabiliriz.

#### 3.2 HORN VE SCHUNCK YÖNTEMİNİN FORMÜLASYONU

Yumuşaklık şartıyla optik akış vektörlerinin değişiminin minimum olduğunu açıklanmıştı (bknz. 2.2). Bunu bir diğer şekilde ifade edecek olursak optik akış vektörlerinin yön türevlerinin karelerinin toplamını minimize ederek optik akış denkleminin çözümüne ulaşabilir. Yani :

$$\text{denklem2:} \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 + \left( \frac{\partial v}{\partial x} \right)^2 + \left( \frac{\partial v}{\partial y} \right)^2$$

değerini minimize(idealde sıfırlayacak) edecek u ve v değerleri sayesinde optik akış denklemini çözmemiz mümkündür. Fakat açıklandığı gibi kuvantalama ve gürültü yüzünden ne Horn ve Schunck'ın yumuşaklık

şartı ne de optik akış denklemi sıfıra eşit olamayacaktır. Bu yüzden de problem bu aşamadan sonra bir minimizasyon problemine dönüşür.

### 3.3 OPTİK AKIŞ DENKLEMİNİN ÇÖZÜMÜ

Giriş bölümünde de açıklandığı gibi gerek optik akış denklemi, gerek ise yumuşaklık varsayımımız sonucu elde ettiğimiz denklemlerin sıfıra mümkün olduğunca yakın(gürültünün de çok yüksek seviyelerde olmadığı varsayılarak) olan u ve v için çözümleri, bize gerçeğe yakın optik akış vektörlerini vermektedir.

Bu yüzden de 2 denklemi de ayrı ayrı minimize edersek:

Parlaklık sabitliği varsayımımız yüzünden denklem.1 in minimizasyonu sonucu ortaya çıkan hata fonksiyonu:

$$e_c = \iint (E_x u + E_y v + E_t)^2 dx dy,$$

Optik akış yumuşaklığı varsayımımız yüzünden denklem.2 nin minimizasyonu sonucu ortaya çıkan hata fonksiyonu:

$$e_s = \iint ((u_x^2 + u_y^2) + (v_x^2 + v_y^2)) dx dy,$$

Olarak elde edilir. Hatayı minimum yapacak “u” ve “v” değerleri bulurken 1. Denklemimizin gürültüden 2. Hata fonksiyonuna göre daha kolay etkilenmesi yüzünden  $e_s$  hata fonksiyonumuz bir katsayıyla kontrol edip yüksek gürültü beklediğimiz durumlarda bu fonksiyonumuzun ağırlığını arttırarak gürültüye daha dayanıklı bir model elde etmiş oluruz. Bu katsayıya  $\alpha$  dersek u ve v için minimum çözümüne ulaşmamız gereken denklem şu hali alır:

$$e_s + \alpha \cdot e_c$$

Bu denklemi minimize edecek çözümü euler lagrange denklemi<sup>2</sup> ile adım adım “u” ve “v” için çözecek olursak :

---

<sup>2</sup> [Weisstein, Eric W. "Euler-Lagrange Differential Equation." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Euler-LagrangeDifferentialEquation.html](http://mathworld.wolfram.com/Euler-LagrangeDifferentialEquation.html)

$$f_u = 2I_x(I_x u + I_y v + I_t)$$

$$f_v = 2I_y(I_x u + I_y v + I_t)$$

$$f_{u_x} = 2\alpha u_x$$

$$f_{u_y} = 2\alpha u_y$$

$$f_{v_x} = 2\alpha v_x$$

$$f_{v_y} = 2\alpha v_y$$

$$\frac{df_{u_x}}{dx} = 2\alpha u_{xx} , \quad \frac{df_{u_y}}{dy} = 2\alpha u_{yy}$$

$$\frac{df_{v_x}}{dx} = 2\alpha v_{xx} , \quad \frac{df_{v_y}}{dy} = 2\alpha v_{yy}$$

için euler lagrange denkemimizi aşağıdaki gibi elde ederiz:

$$f_u - \frac{df_{u_x}}{dx} - \frac{df_{u_y}}{dy} = 0 \quad \text{ve} \quad f_v - \frac{df_{v_x}}{dx} - \frac{df_{v_y}}{dy} = 0$$

$u_{xx} + u_{yy}$  "u"nun ;  $v_{xx} + v_{yy}$  ise "v" nin laplasyenlerine eşit olacağından laplasyeni

iteratif olarak laplasyen yaklaşıklığını kullanarak çözebiliriz. Yani :

$$\nabla^2 u \approx \bar{u} - u \quad \text{ve} \quad \nabla^2 v \approx \bar{v} - v \quad \text{için euler lagrange denklemlerini}$$

iterative çözüm için düzenlersek "u" ve "v" için aşağıdaki iterative çözümü elde etmiş oluruz:

$$u^{k+1} = \bar{u}^k - \frac{-E_x[E_x \bar{u} + E_y \bar{v} + E_t]}{(a + E_x^2 + E_y^2)}$$

$$v^{k+1} = \bar{v}^k - \frac{-E_y[E_x \bar{u} + E_y \bar{v} + E_t]}{(a + E_x^2 + E_y^2)}$$

## 4 ÇÖZÜMÜN MATLAB ORTAMINDA GERÇEKLENMESİ

Algoritmamızın çekirdeğini oluşturan kısmı açıklamak gerekirse.

### 4.1 TÜREVLERİN HESAPLANMASI

İlk aşamada herhangi iki resim arasındaki  $Ex, Ey, Et$  değerlerinin bulunabilmesi için “turev.m” fonksiyonu iletilen iki resmin ayırık zamanda gerekli türevlerini daha verimli sonuçlar ürettiğinden<sup>3</sup>  $Ex, Ey$  değerlerini sobel operatörünü kullanarak hesaplamaktadır.

“turev.m” fonksiyonu:

```
function [Dx, Dy, Dt] = turev(im1,im2)
    Dx = conv2(im1,0.25* [-1 0 1; -2 0 2;-1 0 1], 'same');
    Dy = conv2(im1, 0.25*[1 2 1;0 0 0; -1 -2 -1], 'same');
    Dt = conv2(im1, 0.25*[-1 1], 'same') + conv2(im2, -0.25*[-1 1], 'same');
end
```

### 4.2 İTERASYON İŞLEMİ

İki resmin arasındaki optic akışı bulmak için 3. Bölümde elde ettiğimiz algoritmamızı matlab ile hesaplırsak :

$\bar{u}^k$  Ve  $\bar{v}^k$  değerlerini sırasıyla uAvg ve vAvg olarak gösterip bu ortalama değerlerini 3 e 3 lük ağırlıklı ortalama kernelini kullanırsak :

Kernel_1 :	1/12	1/6	1/12
	1/6	-1	1/6
	1/12	1/6	1/12

ve başlangıç için optic akışları 0 kabul ederek aşağıdaki çözümü elde ederiz:

```
for i=1:ite
    uAvg=conv2(uvid(:,:,k-1),kernel_1,'same');
    vAvg=conv2(vvid(:,:,k-1),kernel_1,'same');
    uvid(:,:,k)= uAvg - ( Dx.*(( Dx.*uAvg ) + ( Dy.*vAvg ) + Dt ) )./( alpha^2 + Dx.*Dx + Dy.*Dy);
    vvid(:,:,k)= vAvg - ( Dy.*(( Dx.*uAvg ) + ( Dy.*vAvg ) + Dt ) )./( alpha^2 + Dx.*Dx + Dy.*Dy);
end
```

<sup>3</sup> Scharr, Hanno, 2000, Dissertation (in German), [Optimal Operators in Digital Image Processing](#) .

## 4.3 GAUSS FİLTRESİ

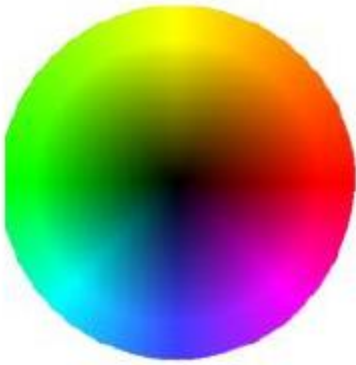
Horn and schunck yöntemi gürültüden daha rahatlıkla etkilenen bir yöntem olduğundan yumuşaklık faktörü(alpha) dışında ekstra bir önlem alacak olursak. Gürültüyü giderme amacıyla gauss filtresi uygulayabiliriz. Ayrıca programın yüksek çözünürlüklü videolarda daha rahat çalışabilmesi amacıyla görüntülerin çözünürlüğü düşürülür.

```
H = fspecial('gaussian',3);

for k=1:NumFrames
    resized=imresize(rgb2gray(read(Object,k)),360/oldvidWidth);
    vid(:,:,k)=imfilter(im2double(resized),H,'replicate');
end
```

## 4.4 OPTİK AKIŞI RENK İLE KODLAMAK

Yöntemle her piksel için bulduğumuz optik akış vektörünü ifade etmemiz için sonucu quiver() fonksiyonuyla oklar halinde gösterebiliriz. Fakat ok ile gösterim hareketli görüntülerde hareketlerin seçilmesi zorlaştırmaktadır. Bunun yerine optik akışın yönünü renk ile şiddetini de rengin tonuyla belli edebiliriz.



## 5 TEST SONUÇLARI

---

### 5.1 VIPTRAFFIC.AVI

Programımızın farklı etmenler kullanarak inceleyelim.





### 5.1.1 Farklı alpha katsayıları için sonucun değişimi

Sabit bir iterasyon sayısı(5) kullanarak sadece alpha değerinin değişimine bakalım



Alpha = 0



alpha = 1



alpha =100

### 5.1.2 Farklı iterasyon sayıları için sonucun değişimi

Alpha değerini sabit tutarsak.



İterasyon sayısı: 1



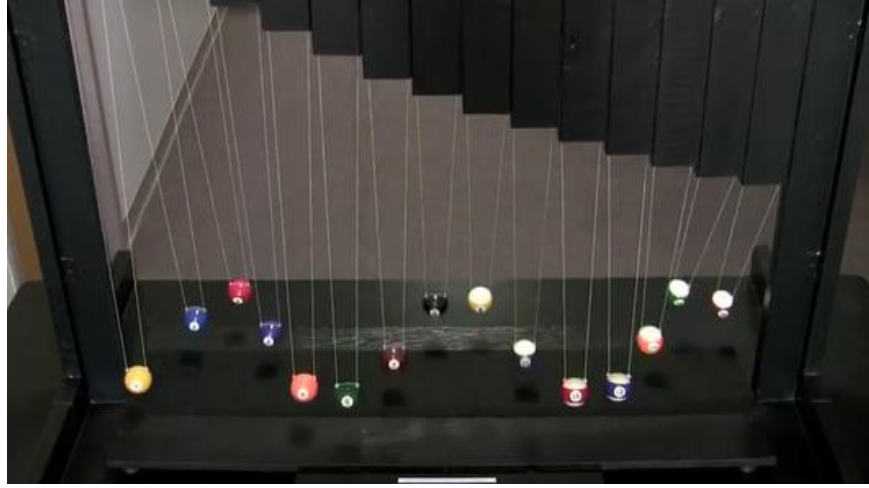
5



100

### 5.1.3 Gürültüye duyarlılık

Horn and schunck algoritmamızı gürültülere karşı daha dayanıklı kılmak için işlemlerden önce görüntüye gauss filtresi uygulamıştık. Bu uygulamanın etkisine bakacak olursak.



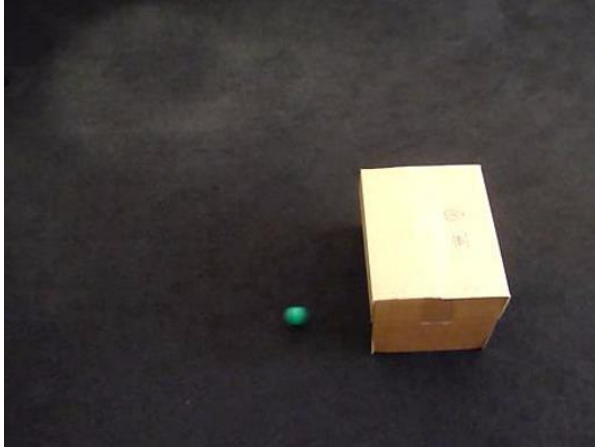
5x5 gauss filtresi uygulanmış sonuç



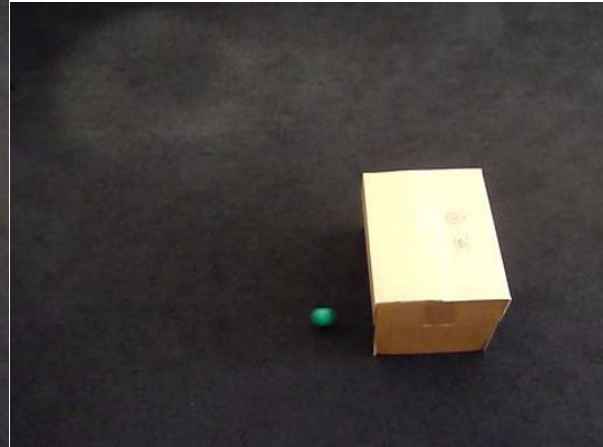
gauss filtresi uygulanmamış sonuç

## 5.2 FARKLI GÖRÜNTÜLER İÇİN SONUÇLAR

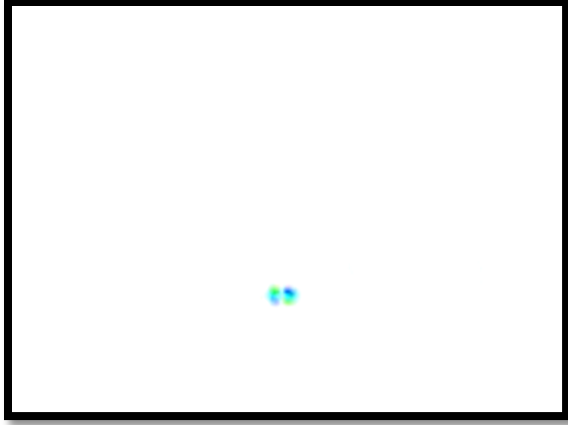
### 5.2.1 Singleball.avi



19. kare



20. kare



Optik akış (hsv renk koduyla) gösterimi

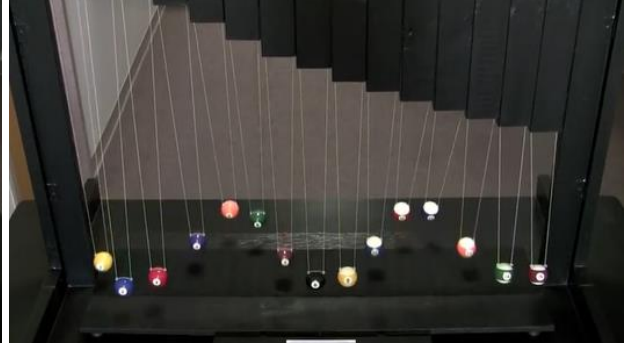


optik akışa göre yapılmış arka plan ayrıştırma

### 5.2.2 Pend.mp4

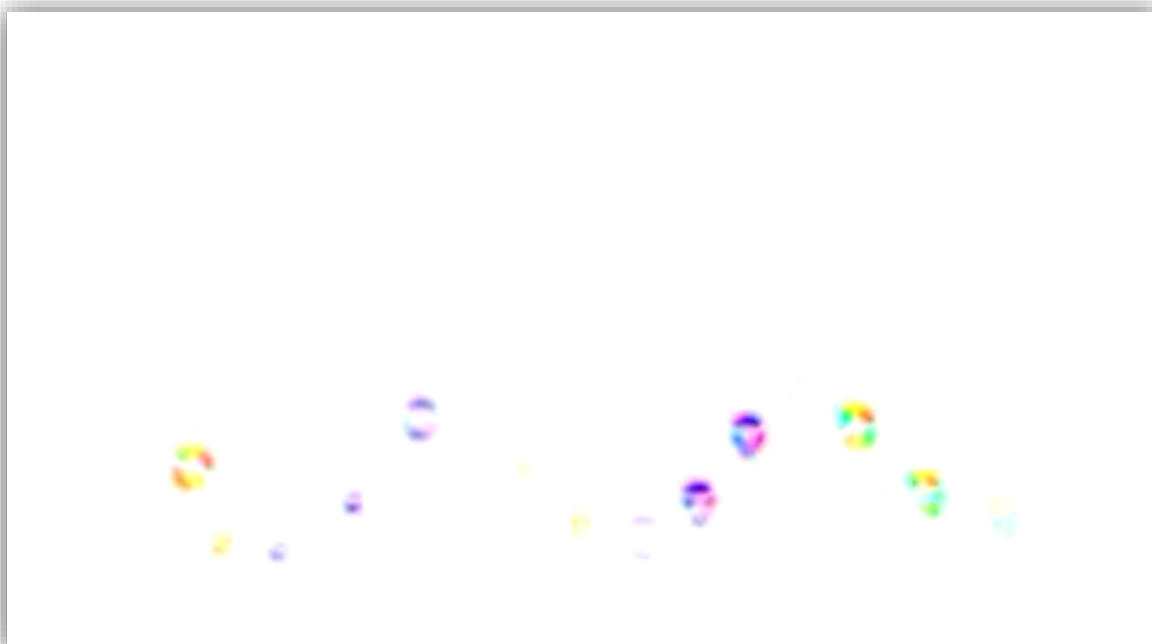


36. kare



37. kare





5.2.3 Viptrain.avi



## 6 REFERANSLAR

---

- B.K.P. Horn and B.G. Schunck, "Determining optical flow." *Artificial Intelligence*, vol 17, pp 185–203, 1981. [Manuscript](#) available on MIT server.
- [Weisstein, Eric W.](#) "Euler-Lagrange Differential Equation." From *MathWorld*--A Wolfram Web Resource. <http://mathworld.wolfram.com/Euler-LagrangeDifferentialEquation.html>
- Scharr, Hanno, 2000, Dissertation, [Optimal Operators in Digital Image Processing](#) .