

Инициализация. Открытие изображение и получение его RGB матрицы.

```
RGBImage= imread('C:\Users\Baric\Desktop\MatlabCanny\Sample2.jpg');  
imshow(RGBImage)
```



Warning: Image is too big to fit on screen; displaying at 67%

Начальный этап алгоритма Canny. Преобразование в оттенки серого Цветого Изображения

Постоянные коэффициенты для перевода из RGB в YUV

```
MYUV=[0.299,0.587,0.144];
```

Получим матрицы описывающие компоненты R,G и B

```
rRGBM=RGBImage(:,:,1);%RGBMatrix([1 : size(RGBMatrix,1)],[1 : ceil(size(RGBMatrix,1)/3)]);% B M  
gRGBM=RGBImage(:,:,2);%([1 : size(RGBMatrix,1)],[ceil(size(RGBMatrix,1)/3) : ceil(2*size(RGBMatrix,1)/3)]);  
bRGBM=RGBImage(:,:,3);%RGBMatrix([1 : size(RGBMatrix,1)],[ceil(2*size(RGBMatrix,1)/3) : size(RGBMatrix,1)]);
```

Считаем Y компоненту

```
YComp=rRGBM*MYUV(1)+gRGBM*MYUV(2)+bRGBM*MYUV(3);  
imshow(YComp)
```



Warning: Image is too big to fit on screen; displaying at 67%

```
%imshow
```

Первый этап алгоритма Canny. Сглаживание

Применение фильтра Гаусса (из-за сложности реализации используем готовую функцию matlab)

Используются для подавления шума изображения

```
IBlured=imgaussfilt(YComp);  
imshow(IBlured)
```



Warning: Image is too big to fit on screen; displaying at 67%

```
%imshow
```

Второй этап алгоритма Canny. Поиск градиентов

Ядра фильтров

```
Gx=[-1 0 1;-2 0 2;-1 0 1]
```

```
Gx = 3x3
    -1     0     1
    -2     0     2
    -1     0     1
```

```
Gy=[-1 -2 -1;0 0 0;1 2 1]
```

```
Gy = 3x3
    -1    -2    -1
     0     0     0
     1     2     1
```

```
A=double(Iblured);
rows=size(Iblured,1);
columns=size(Iblured,2);
mag=zeros(rows,columns);
Vmap=zeros(rows,columns);
for i=2:columns-1
```

```

for j=2:rows-1
    templena=A(j-1:j+1,i-1 :i+1);

    SGx=sum(sum(Gx.*templena)) ; % Произвести умножение матриц и затем сложить всек элемент
    SGy=sum(sum(Gy.*templena)) ;

    mag(j,i)=sqrt(double((SGx).^2+(SGy).^2));

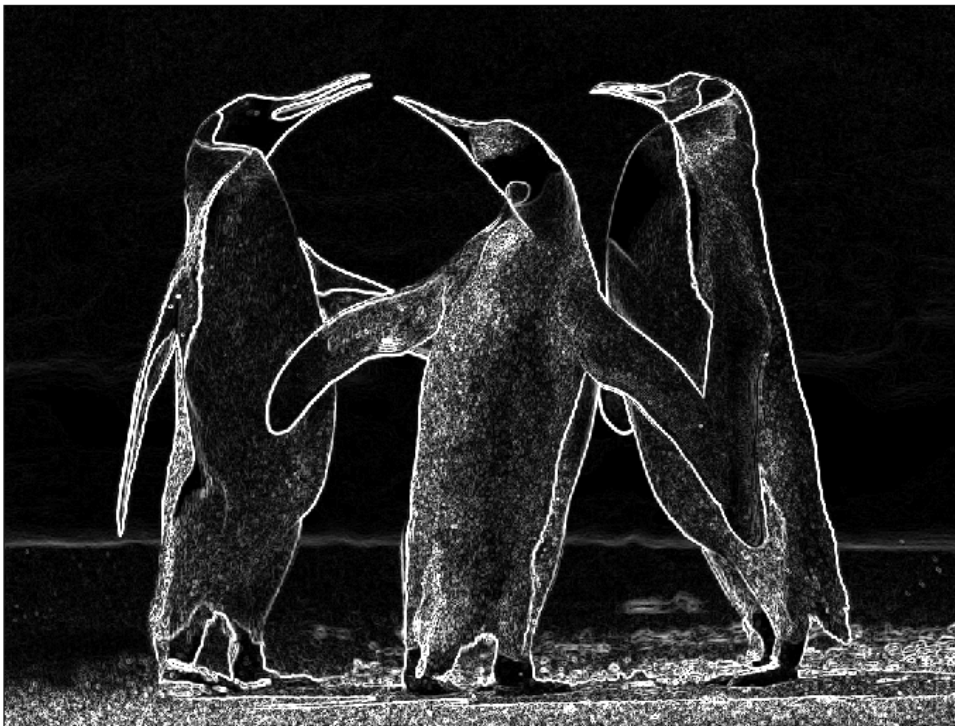
    tau=ceil(pi*atan2(double(SGx),double(SGy))/4)*pi/4-pi/2;
    Vmap(j,i)=tau;

    if 1==0%(mag(j,i) ~= 0)
        tau=ceil(pi*atan2(double(SGx),double(SGy))/4)*pi/4-pi/2;
        Vmap(j,i)=tau;
    end

end

end
%imshow(uint8(Vmap))
imshow(uint8(mag))

```



Warning: Image is too big to fit on screen; displaying at 67%

```
%imsave
```

Третий этап алгоритма Сanny. Подавление не-максимумов


```

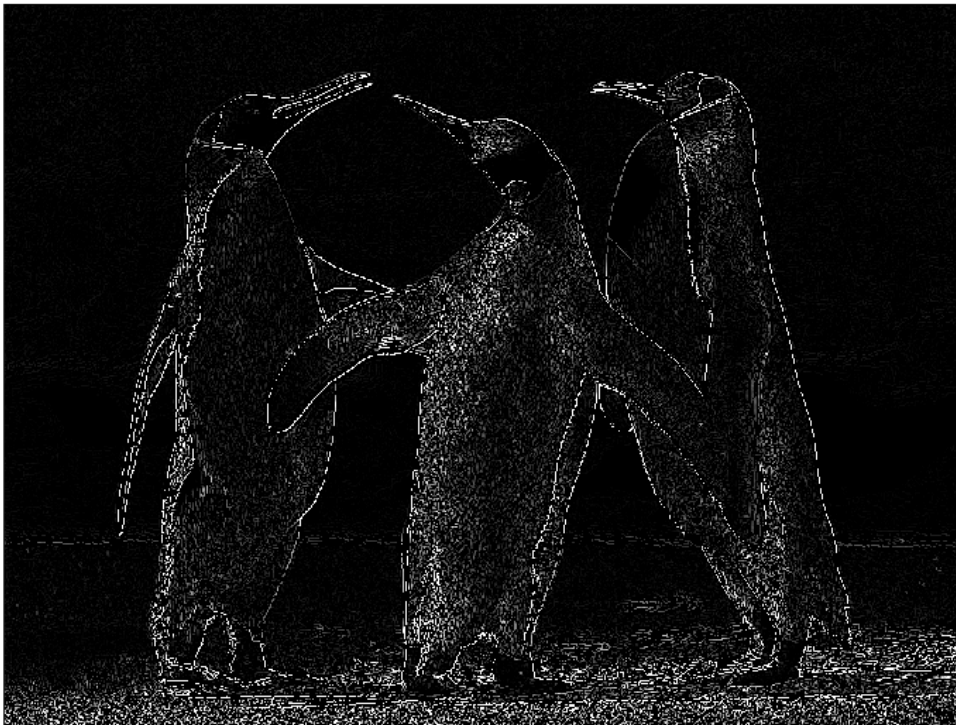
RMatrix=mag;
for i=1:columns-1
    for j=1:rows-1
        di=sign(cos(Vmap(j,i)));
        dj=-sign(sin(Vmap(j,i)));

        if(j+dj>0&& j+dj<rows&&i+di>0&&i+di<columns)
            if(mag(j+dj,i+di)<=mag(j,i))
                RMatrix(j+dj,i+di)=0;
            end
        end

        if(j-dj>0&& j-dj<rows&&i-di>0&&i-di<columns)
            if(mag(j-dj,i-di)<=mag(j,i))
                RMatrix(j-dj,i-di)=0;
            end
        end

    end
end
imshow(uint8(RMatrix))

```



Warning: Image is too big to fit on screen; displaying at 67%

```
%imsave
```

Четвертый этап алгоритма Canny. Двойная пороговая фильтрация

Коэффициенты:

```
low_pr=0.55
```

```
low_pr = 0.5500
```

```
high_pr=0.60% в процентах
```

```
high_pr = 0.6000
```

Непосредственно алгоритм

```
down=low_pr*255
```

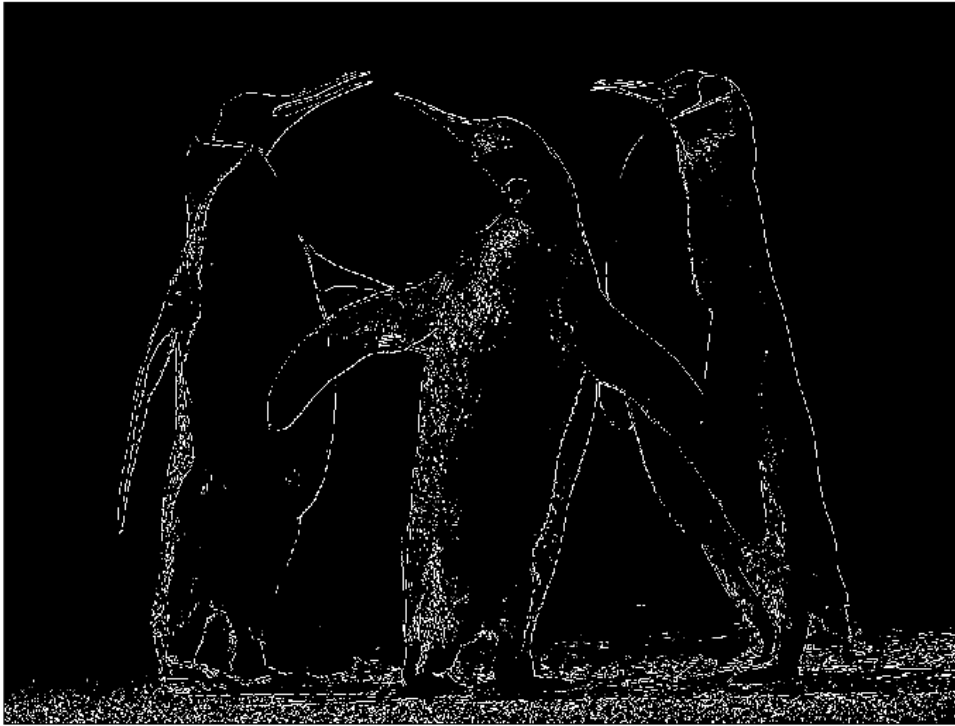
```
down = 140.2500
```

```
up=high_pr*255
```

```
up = 153
```

```
RM=RMatrix;
for i=1:columns-1
    for j=1:rows-1
        if(RM(j,i)>=up)
            RM(j,i)=255;
        else
            if(RM(j,i)<=down)
                RM(j,i)=0;
            else
                RM(j,i)=127;
            end
        end
    end
end

imshow(uint8(RM))
```



Warning: Image is too big to fit on screen; displaying at 67%

```
%imshow
```

Финальный этап алгоритма Canny. Двойная пороговая фильтрация

```
movdir=[-1 -1 -1 0 0 1 1 1; -1 0 1 -1 1 -1 0 1]
```

```
movdir = 2x8
    -1    -1    -1     0     0     1     1     1
    -1     0     1    -1     1    -1     0     1
```

```
High=255;
Low= 127;
CMat=RM;
rows=size(CMat,1);
columns=size(CMat,2);
```

Фильтрация

```
for i=1:columns-1
    for j=1:rows-1
        if(CMat(j,i) <= Low)
            CMat(j,i) = 0;
        elseif (CMat(j,i) >= High)
            CMat(j,i) = 255;
```

```

else
    if((CMat(j+1,i)>High)|| (CMat(j-1,i)>High)|| (CMat(j,i+1)>High)|| (CMat(j,i-1)>High))
        CMat(j,i) = 255;
    end
end
end
end
imshow(uint8(CMat))

```



Warning: Image is too big to fit on screen; displaying at 67%

```

imsave

```