Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

# 4C8 Imaging Processing Lab2 Assignment

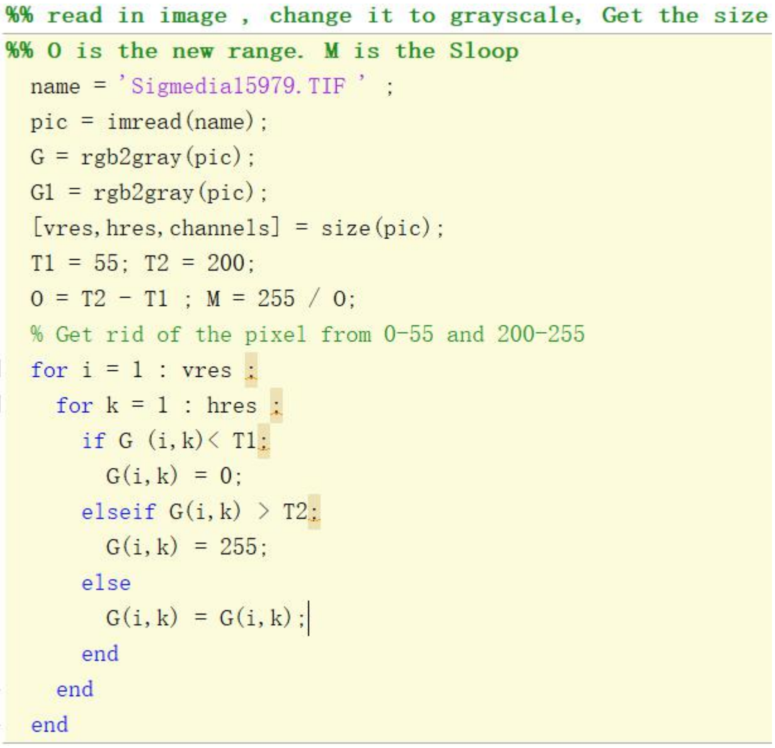
## February 9, 2020

| Contributor | Student Number |
|---|---|
| Shida Sheng | 19303554 |

Q1. Write an .m script to do a linear contrast stretch on the jfk image from lab 1 so that the image values in the range 55 to 200 are stretched to occupy the whole 8 bit grayscale range. The code will have to ensure that any intensities outside of the range 0:255 are appropriately clipped. Display the output image in a separate figure window when you are done. In your report, include your script along with the image you generate. Explain clearly how the code implements all of the listed requirements.

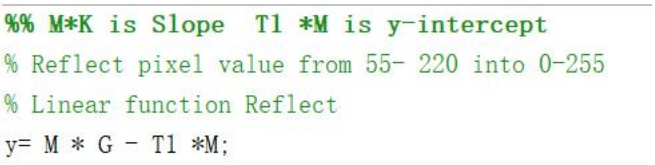
Answer:

The Question 1 is to do a Contrast enhancement. It is a Conversion in mathematics. The Conversion is a linear function, So the First thing is to find out the sloop of the new linear function. First I use a easy for-loop to get rid of the useless range. It will be set two threshold (T1 and T2) to control the new range.

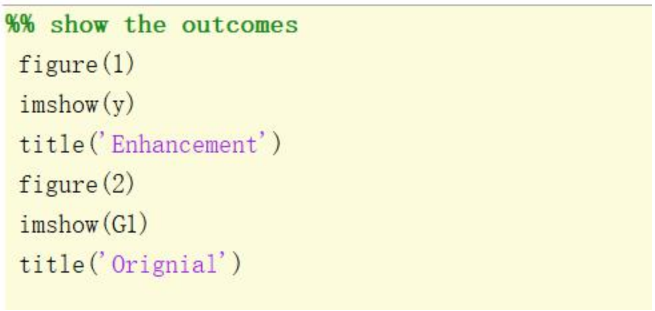


```
%% read in image , change it to grayscale, Get the size
%% O is the new range.  M is the Sloop
  name = 'Sigmedia15979.TIF ' ;
  pic = imread(name);
  G = rgb2gray(pic);
  G1 = rgb2gray(pic);
  [vres,hres,channels] = size(pic);
  T1 = 55; T2 = 200;
  O = T2 - T1 ; M = 255 / O;
  % Get rid of the pixel from 0-55 and 200-255
  for i = 1 : vres ;
    for k = 1 : hres ;
      if G (i,k)< T1;
        G(i,k) = 0;
      elseif G(i,k) > T2;
        G(i,k) = 255;
      else
        G(i,k) = G(i,k);
      end
    end
  end
```

After getting the Pixel inside the new range, Make a  Linear Function Conversion



```
%% M*K is Slope  T1 *M is y-intercept
% Reflect pixel value from 55- 220 into 0-255
% Linear function Reflect
y= M * G - T1 *M;
```



```
%% show the outcomes
 figure(1)
 imshow(y)
 title('Enhancement')
 figure(2)
 imshow(G1)
 title('Orignial')
```

The Result of the Question 1 :

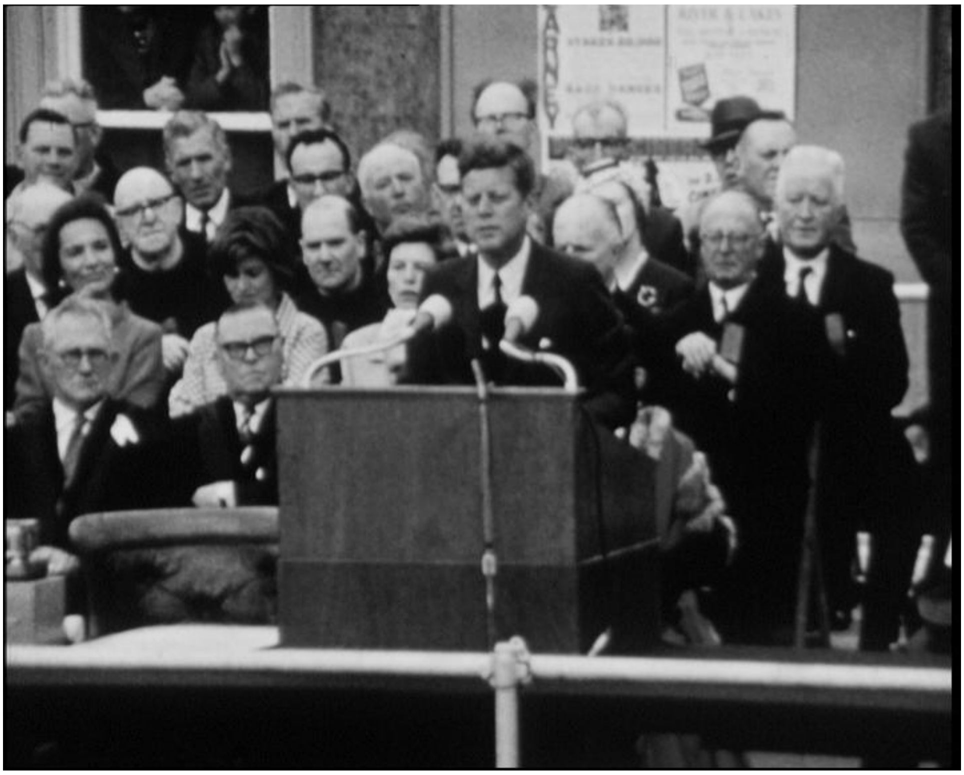


Figure.1   Original Picture

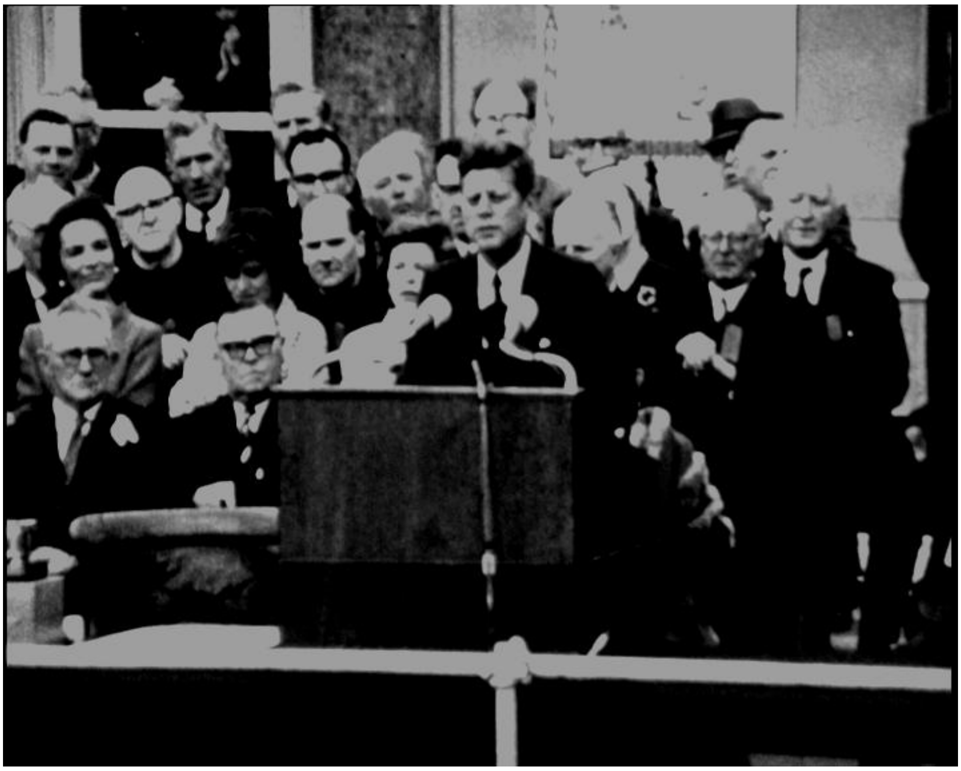


Figure.2 Enhancement Picture

The Enhancement is based on the range conversion in of the linear function. It make the bright area become brighter,and make the dark area become darker. The enhancement makes image have more specific features in the strength area, and reduce the interference from the useless information.

## Q2.The Gaussian Filter

Write a matlab function that implements a Gaussian low pass filter according to the above criteria. Include the code in your submission and explain what each line of your function does.

Answer: First,The Result Of My Gaussian low Pass Filter.

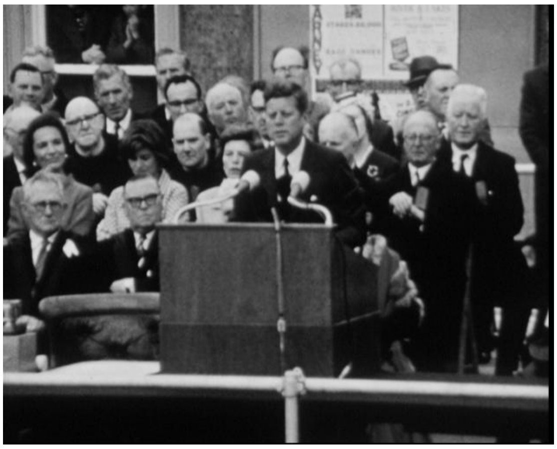


Figure.2 Original Picture

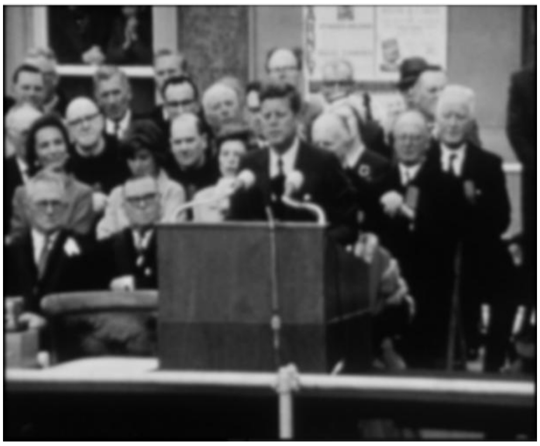


Figure.3 After Gaussian filter(by separable)

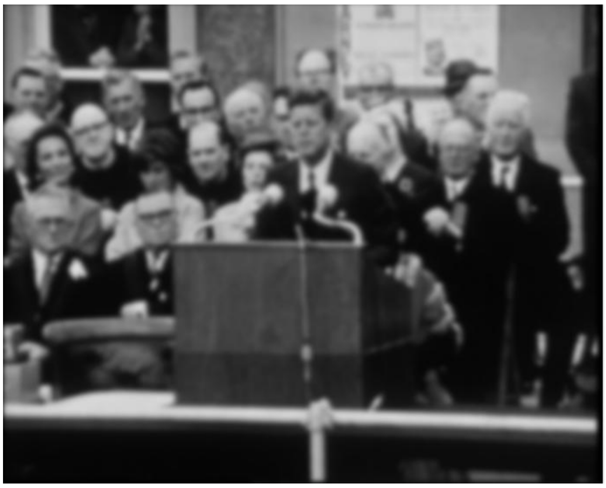
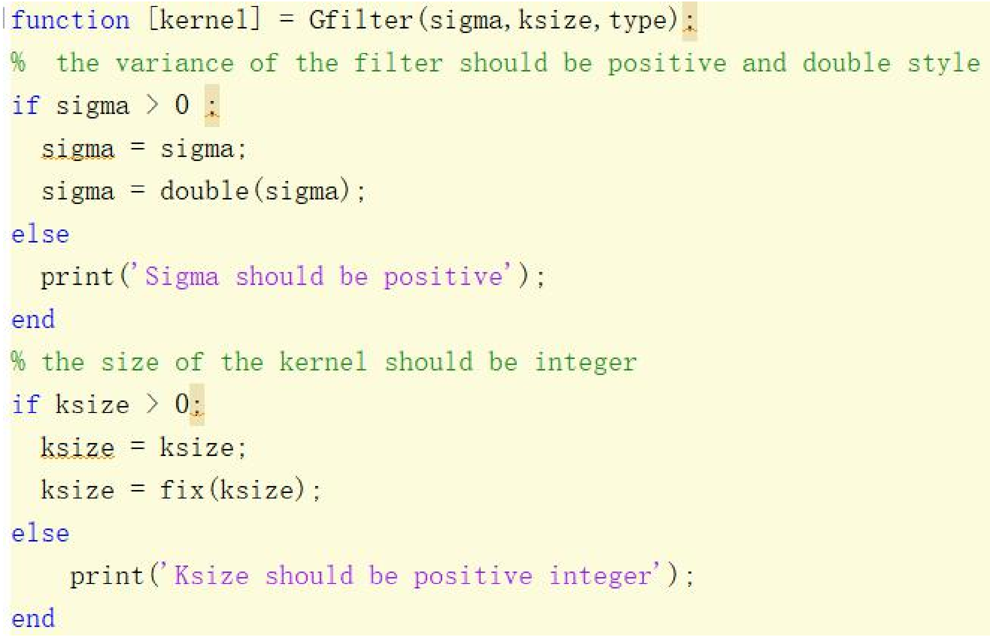


Figure.4 After Gaussian filter(by combined)

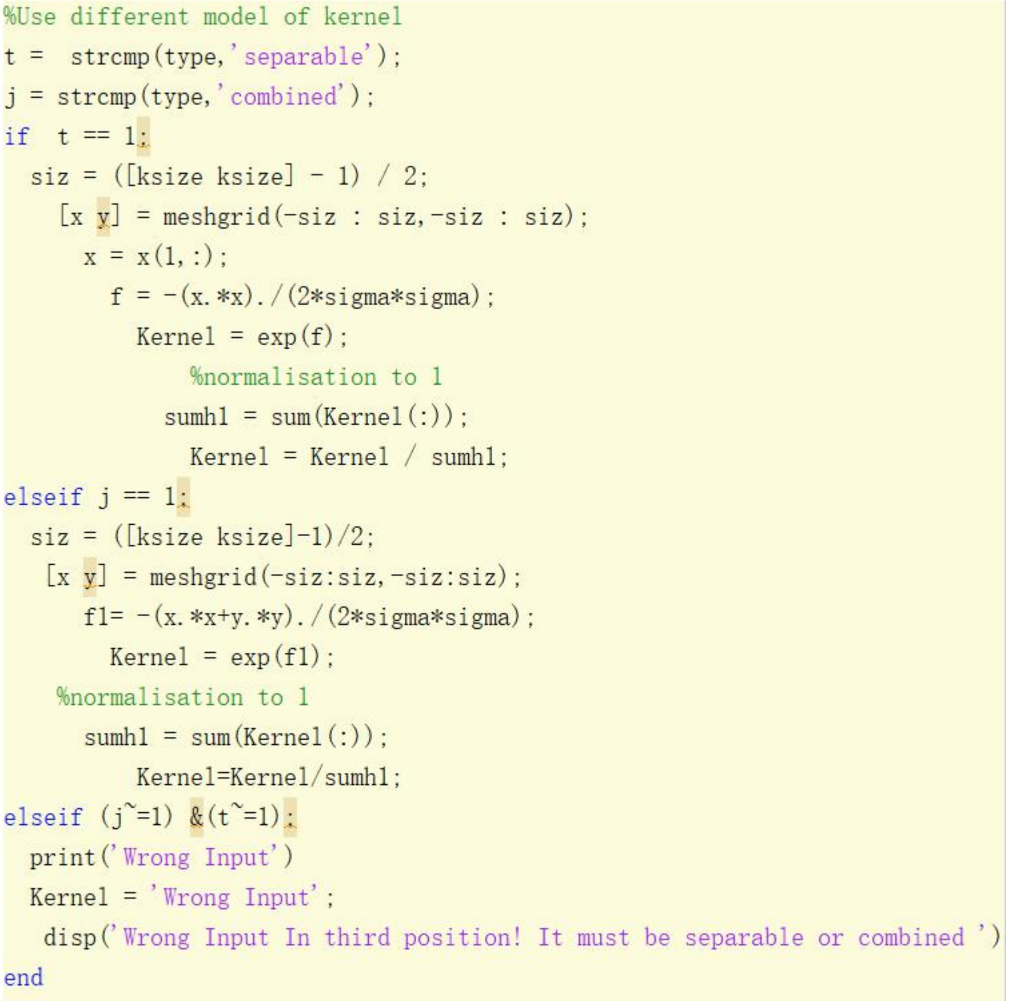
The Code Part Of My Gaussian low Pass Filter.



```
function [kernel] = Gfilter(sigma,ksize,type);
%  the variance of the filter should be positive and double style
if sigma > 0 ;
  sigma = sigma;
  sigma = double(sigma);
else
  print('Sigma should be positive');
end
% the size of the kernel should be integer
if ksize > 0;
  ksize = ksize;
  ksize = fix(ksize);
else
    print('Ksize should be positive integer');
end
```

This is the Statement Of the Input Parameter Of this Function.

The strcmp Function can distinguish the Input string parameter.

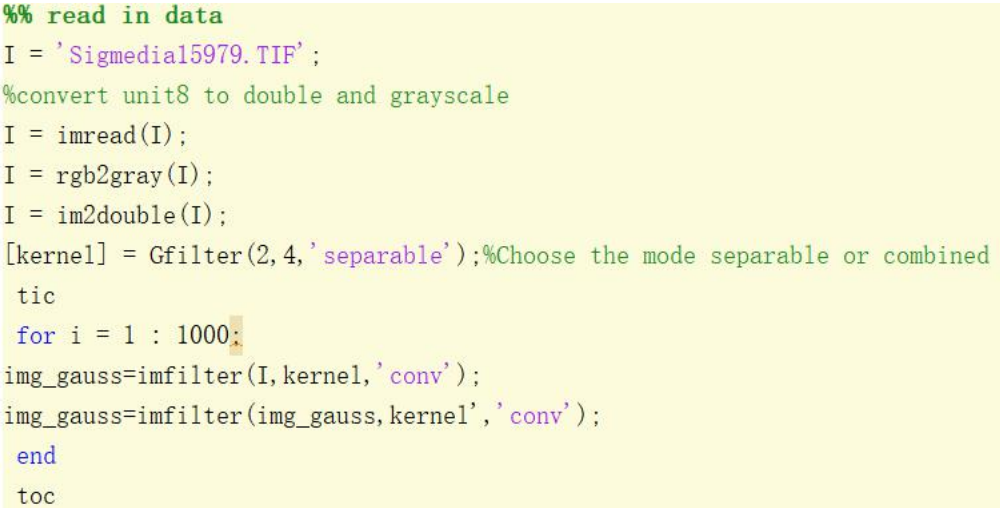


```
%Use different model of kernel
t =  strcmp(type,'separable');
j = strcmp(type,'combined');
if  t == 1;
  siz = ([ksize ksize] - 1) / 2;
    [x y] = meshgrid(-siz : siz,-siz : siz);
      x = x(1,:);
        f = -(x.*x)./(2*sigma*sigma);
          Kernel = exp(f);
              %normalisation to 1
            sumh1 = sum(Kernel(:));
              Kernel = Kernel / sumh1;
elseif j == 1;
  siz = ([ksize ksize]-1)/2;
   [x y] = meshgrid(-siz:siz,-siz:siz);
      f1= -(x.*x+y.*y)./(2*sigma*sigma);
        Kernel = exp(f1);
    %normalisation to 1
      sumh1 = sum(Kernel(:));
          Kernel=Kernel/sumh1;
elseif (j~=1) &(t~=1);
  print('Wrong Input')
  Kernel = 'Wrong Input';
   disp('Wrong Input In third position! It must be separable or combined ')
end
```

The meshgrid Function can make two matrix that can help to build the ‘h’ and ‘k’ in the Gaussian equation. It also accept the odd number and the even number to build the matrix.
When the input is a even number, the Function will let the centre of the matrix have the same value to fit the Gaussian distribution. The centre have the biggest value.
After building the Kernel by Gaussian equation, Normalizing the kernel.(sum to 1).
The Normalization is to balance the Lightness. If the Kernel Sum is bigger than 1, then after doing the Convolution Calculation of the Image, the Result will be lighter. For the Sum is smaller than 1, it cause the image become darker. The Purpose for the Gaussian Filter is to reducing the high-frequency components of the image and reduce the noisy and details.

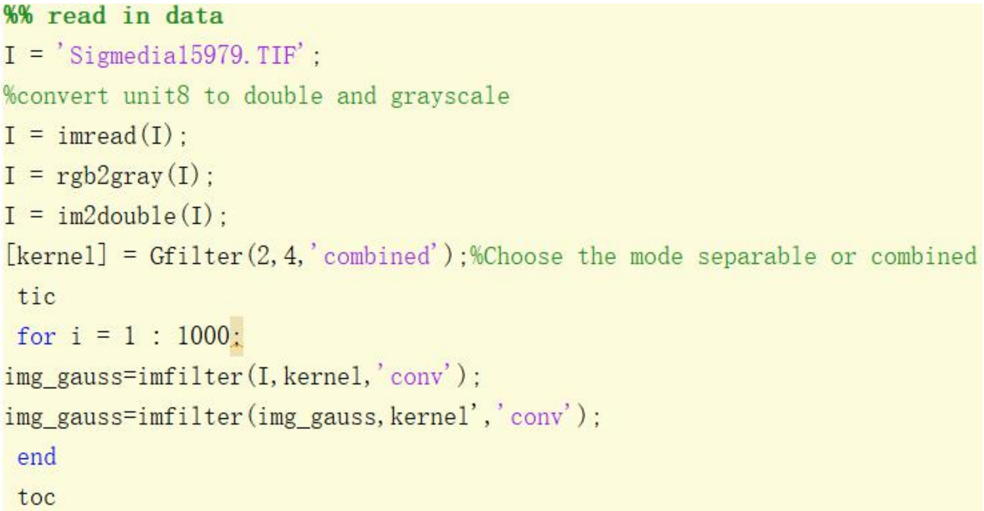
The Gaussian filter is a linear smoothing filter, and the template of the filter is obtained by discretizing the two-dimensional Gaussian function. Because the centre value of the Gaussian template is the largest and the adjacent steps are gradually shifted, the filtered result is better than the mean filter.
The most important parameter of the Gaussian filter is the standard deviation σ of the Gaussian distribution. The standard deviation and the smoothing ability of the Gaussian filter have a great ability. it is good. By adjusting the σ parameter, the suppression of image noise and the blur of image can be balanced.

Explain why it is necessary that all the coefficients of the low pass filter sum to one. IMAGE PROCESSING 4 3. Changing the value of N to 21 and using the matlab functions tic and toc, record the time that it takes to apply the system to the image 1000 times. The filter should be applied to both the rows and the columns of the image. In your answer include the code you use and write down the time taken. 4. Change your m-file to use the full 2D implementation of the filter (ie. not separably). Include your code and write down the time it takes to apply this separable implementation 1000 times as before. 5. Is there any difference in execution time between these two implementations? Explain your findings.



```
%% read in data
I = 'Sigmedia15979.TIF';
%convert unit8 to double and grayscale
I = imread(I);
I = rgb2gray(I);
I = im2double(I);
[kernel] = Gfilter(2,4,'separable');%Choose the mode separable or combined
 tic
 for i = 1 : 1000;
img_gauss=imfilter(I,kernel,'conv');
img_gauss=imfilter(img_gauss,kernel','conv');
 end
 toc
```

Elapsed time is 6.440618 seconds.(Separable Mode)



```
%% read in data
I = 'Sigmedia15979.TIF';
%convert unit8 to double and grayscale
I = imread(I);
I = rgb2gray(I);
I = im2double(I);
[kernel] = Gfilter(2,4,'combined');%Choose the mode separable or combined
 tic
 for i = 1 : 1000;
img_gauss=imfilter(I,kernel,'conv');
img_gauss=imfilter(img_gauss,kernel','conv');
 end
 toc
```

Elapsed time is 9.557023 seconds.(Combined Mode)

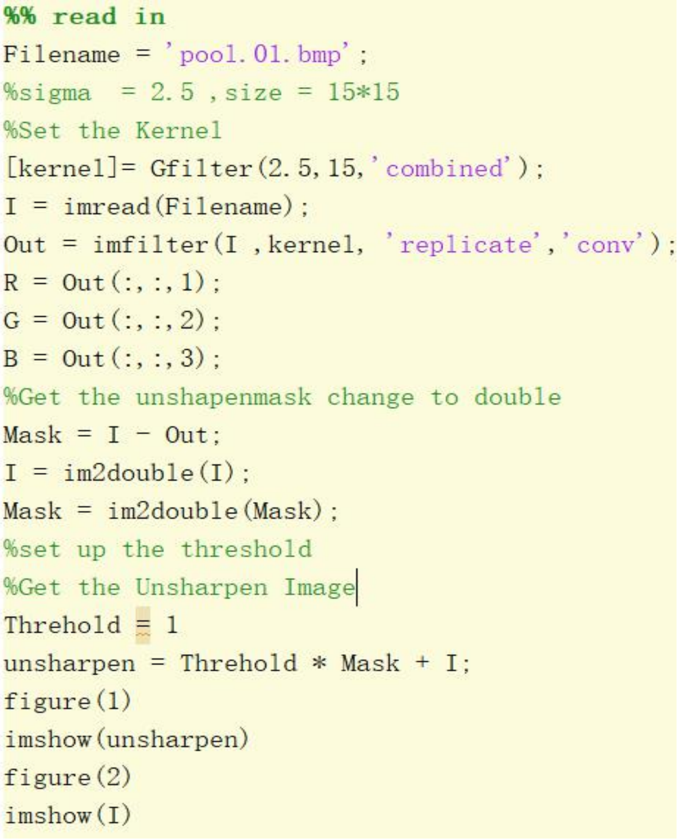
Answer: The two different ways have different results. The Separable Mode take less time than the Combined Mode. The Combined Mode time is almost as long as one and a half times Of Separable Mode.

However, The image result shows the difference to time. As the Pictures shown on Figure.3 and Figure.4, the Combined Mode have a better effect in reducing high-frequency components of the image.

The difference will be very small when the kernel size is not big enough, when the kernel size comes to bigger size, the time have a big difference. That is because the calculation cost more time when the size become bigger.

## Q3. Unsharp Masking

Along with your code the answer should explain how the Unsharp masking filter is implemented. The code that you submit for this section will be partially assessed on its neatness, clarity and computational efficiency.



```
%% read in
Filename = 'pool.01.bmp';
%sigma  = 2.5 ,size = 15*15
%Set the Kernel
[kernel]= Gfilter(2.5,15,'combined');
I = imread(Filename);
Out = imfilter(I ,kernel, 'replicate','conv');
R = Out(:,:,1);
G = Out(:,:,2);
B = Out(:,:,3);
%Get the unshapenmask change to double
Mask = I - Out;
I = im2double(I);
Mask = im2double(Mask);
%set up the threshold
%Get the Unsharpen Image
Threhold = 1
unsharpen = Threhold * Mask + I;
figure(1)
imshow(unsharpen)
figure(2)
imshow(I)
```

Answer:

As the flow chart shown in the Question 4, it can be described by this equation

$$\mathbf{y(n,m)= x(n,m)+ \lambda z(n,m)} \tag{1}$$

Where y(n,m) is stand for the output image, x(n,m) is the input image, λ is the threshold, Z(n,m) is the Mask, (n,m) is the location of the image.

The First step is to get a result from the Low Pass Filter and do the Subtraction operation between the Original image and the result from the filter to generate the Mask. The Second step is to add the Mask to the Original image, Before that, It have a threshold to control this Mask.

When the threshold is set by one, the outcome have a best appearance.

It have three channels in the picture, they all show different appearance.

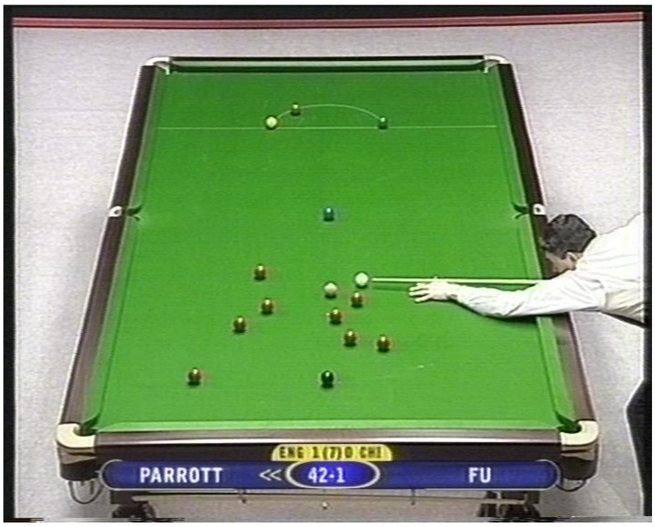


Figure.5 The Original Picture

**R Channel Test Result** :

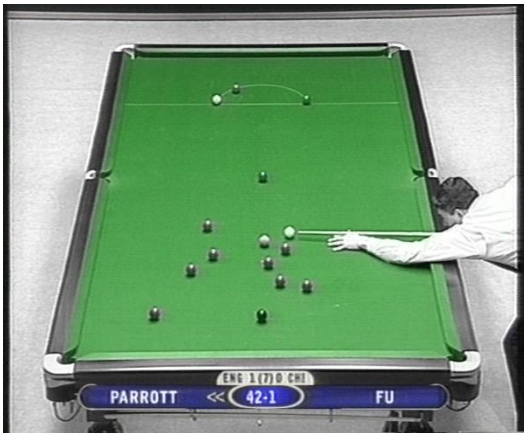


Figure.6 Threshold = 1

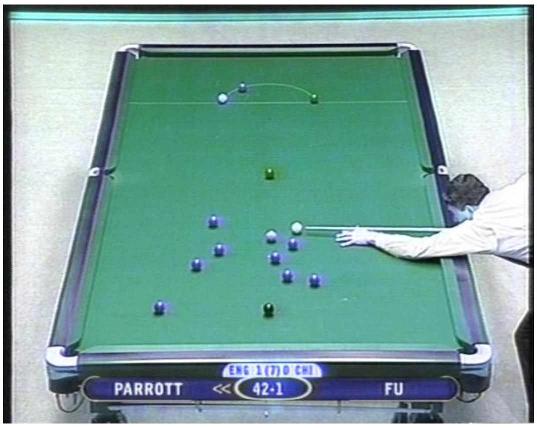


Figure.7 Threshold = 3

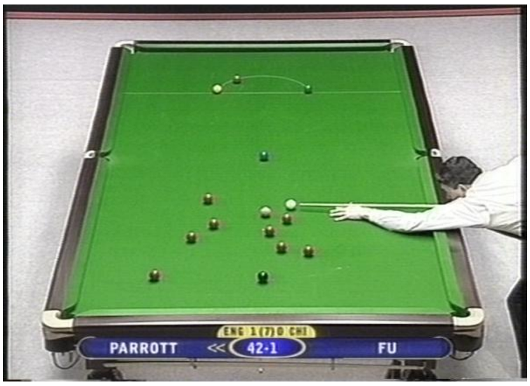


Figure.8 Threshold = 0.3

**G Channel Test Result** :

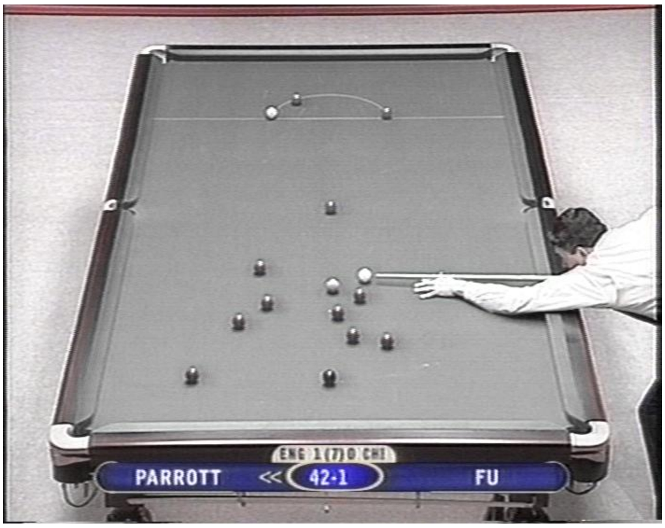


Figure.9 Threshold = 1

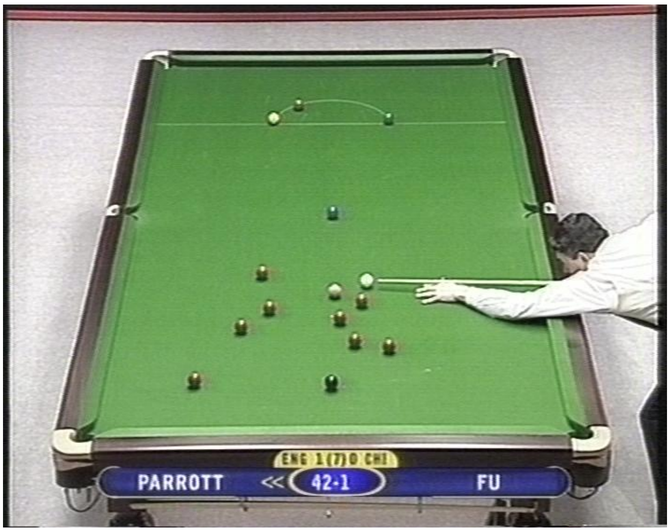


Figure.10 Threshold = 0.3

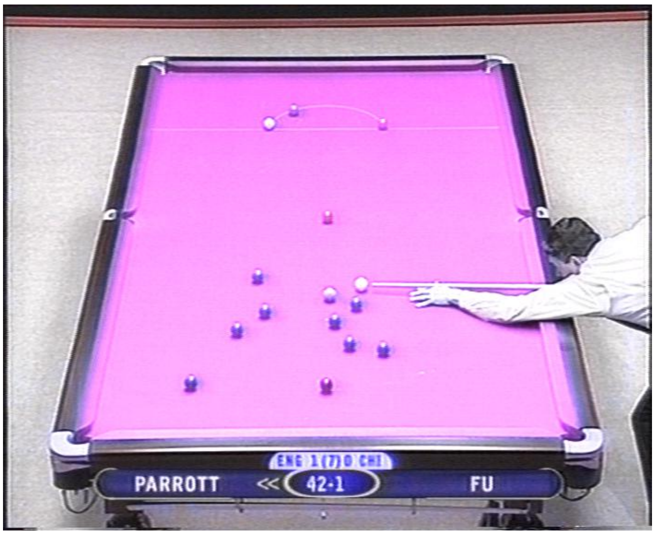


Figure.11 Threshold = 3

**B Channels Test Result** :

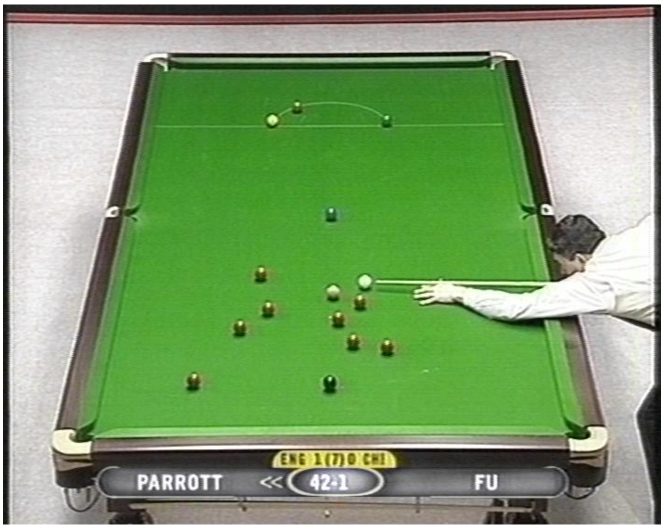


Figure.12 Threshold = 1

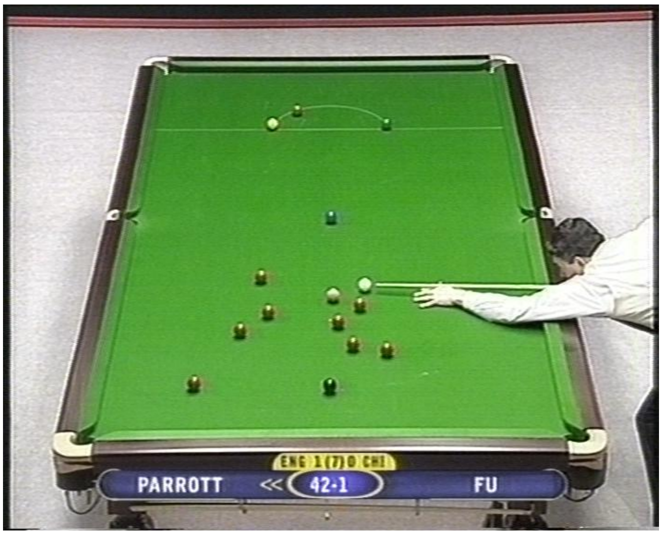


Figure.13 Threshold = 0.3

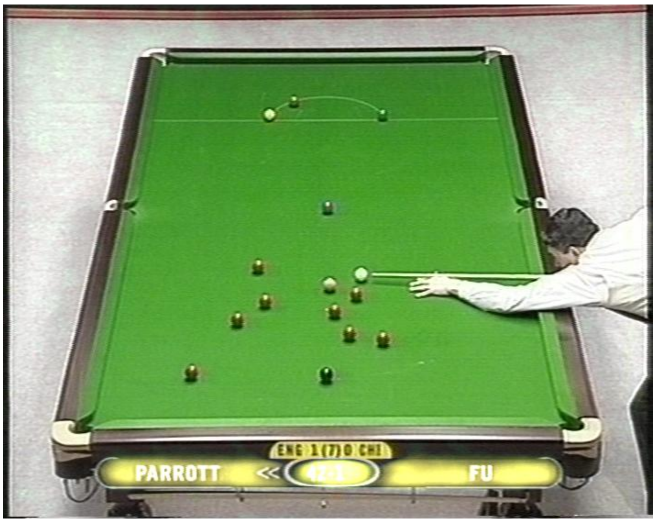


Figure.14 Threshold = 3

RGB Channels Picture :

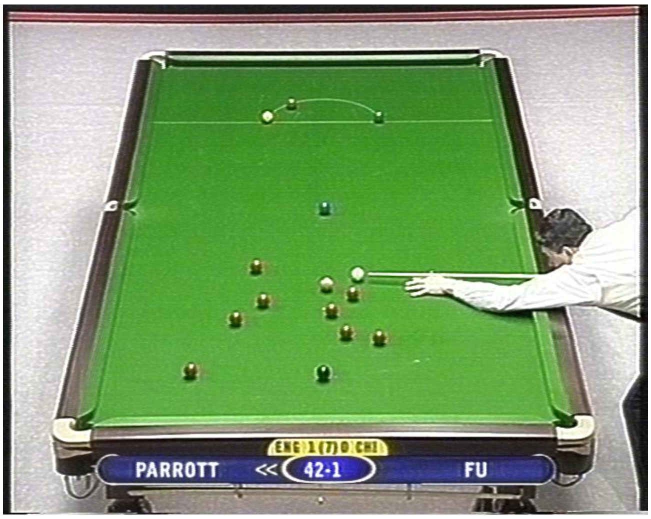


Figure.15 Threshold = 1

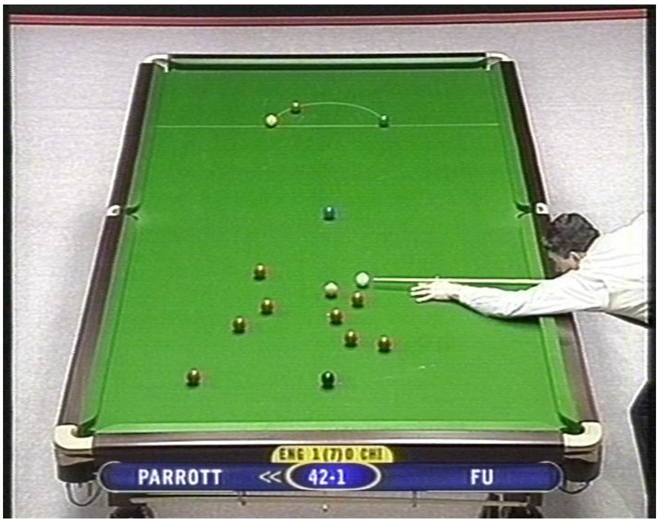


Figure.16 Threshold = 0.3

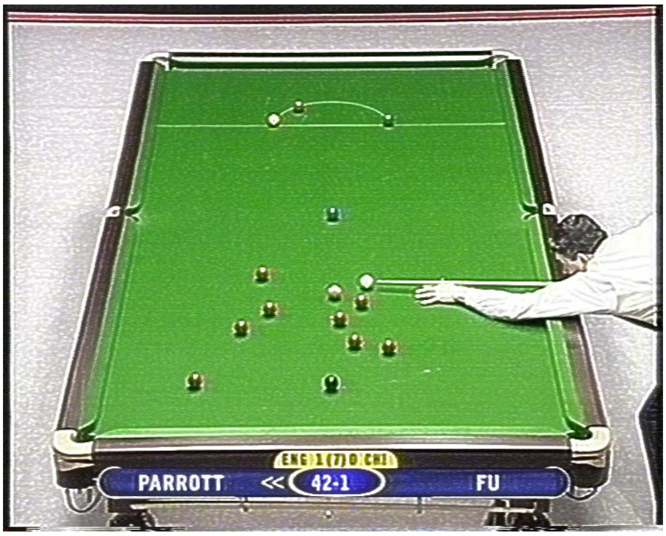


Figure.17 Threshold = 0.3

Answer :

These Output image of different threshold have a very big difference, to the original But it shows a Same Conversion in each Colour Channel. It can be clearly observed that If the Value of the Threshold have a smaller number, then it will have less difference to the original image. On the Contrast, If the Value of Threshold have a bigger number, the outcome shows a big difference.

When the Threshold shows a bigger number in each channel, It will keep its colour of the other two Channel and reduce the the colour of its own Channel.

When the Threshold equal to one, the image Can have the best effect.

Change the Image to sigmedia06907.tif

The RGB Channels

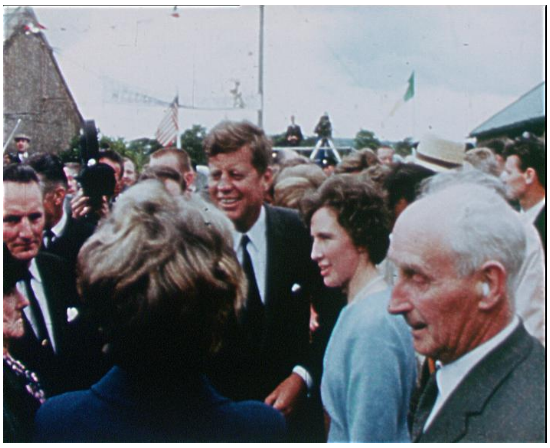


Figure.18 Threshold = 1

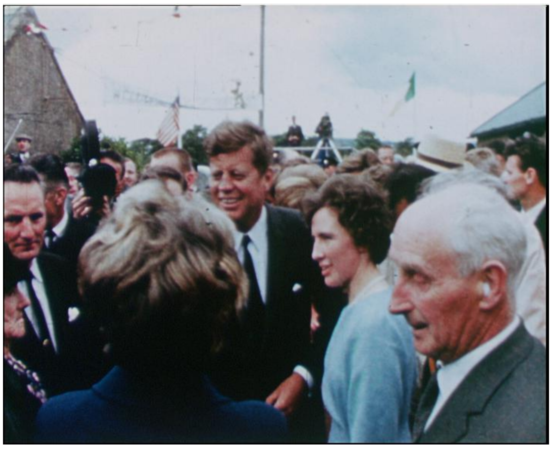


Figure.19 Threshold = 0.3

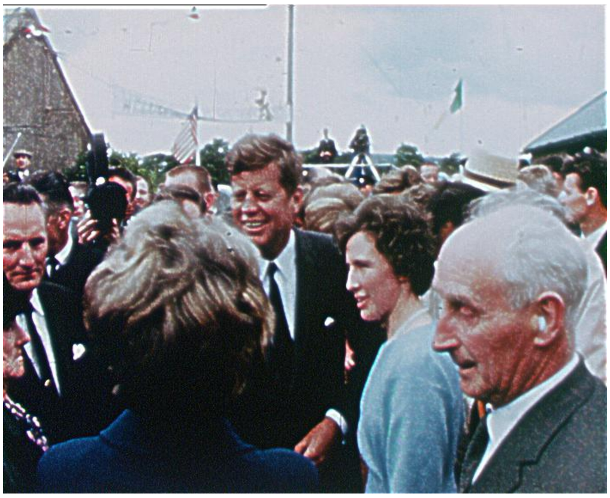


Figure.20 Threshold = 0.3

R Channel :

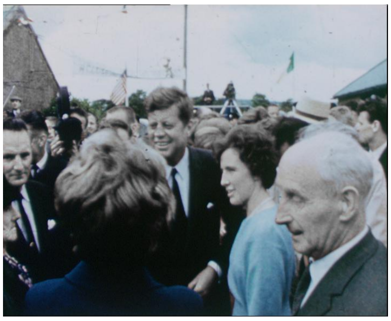


Figure.21 Threshold = 1

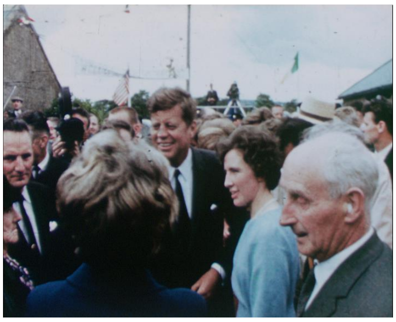


Figure.22 Threshold = 0.3

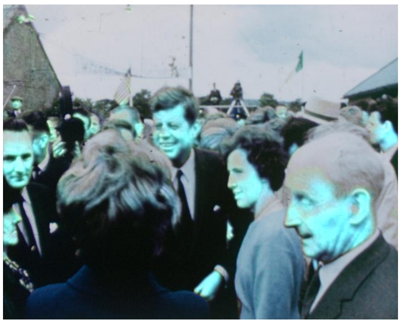


Figure.23 Threshold = 3

B Channel :

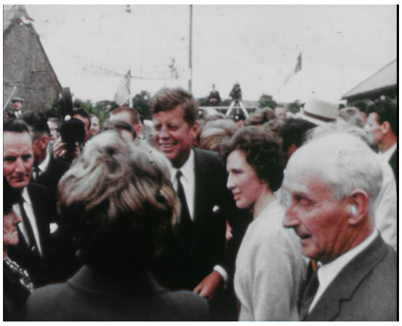


Figure.24 Threshold = 1

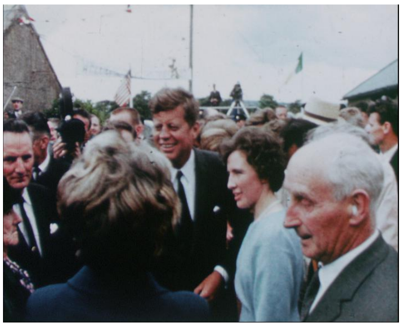


Figure.25 Threshold = 0.3

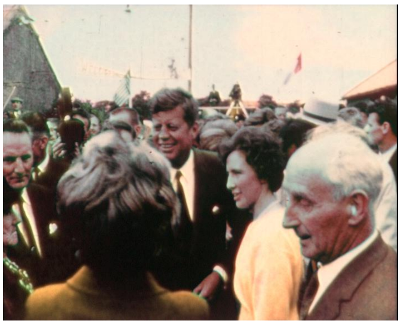


Figure.26 Threshold = 3

G Channel :

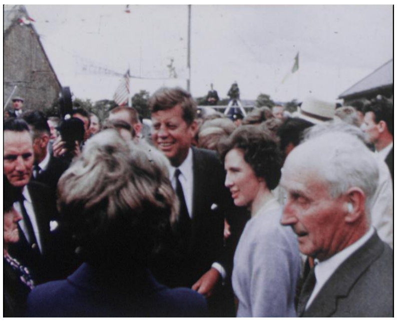


Figure.27 Threshold = 1

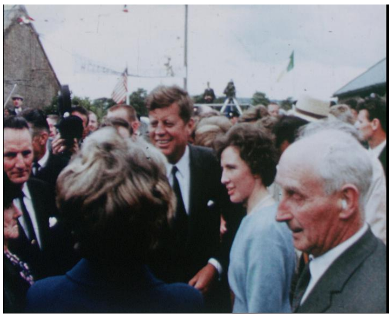


Figure.28 Threshold = 0.3

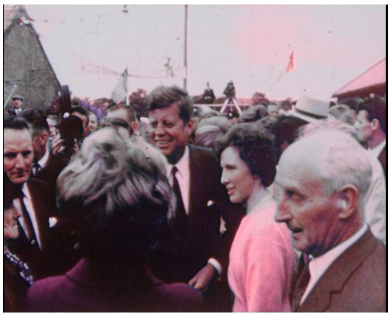


Figure.29 Threshold = 3

Answer : The finding is the same as the finding in Pool.01, But this image have more complex consist of colour, So It can be easily to find the difference between each Colour Channels.