

Университет ИТМО

**Цифровая обработка сигналов**

**Лабораторная работа №4**

Вариант 3

Выполнила: Калугина Марина

Группа: Р3402

г. Санкт-Петербург

2020 г.

### **Задание № 1**

1. Используя полученное изображение лица человека, при помощи приведенной в разделе 2.1.1 функции осуществите его сегментацию методом центроидного связывания. Необходимо выделить лицо человека, добившись выделения максимально возможной области;
2. По результатам выполнения п.1, выведите на экран соответствующий фрагмент полутонового изображения, используя результат сегментации как маску.

Полученные изображения и программный код (за исключением программного кода реализации функции *regiongrow*) поместите в отчет).

### **Задание № 2**

Используя представленный выше программный код, осуществите квадратичное разложение полученного для данного упражнения изображения. Постарайтесь выбрать порог, обеспечивающий наиболее точное выделение основной части изображения.

Программный код и полученные результаты поместите в отчет.

### **Задание № 3**

Используя функцию *splitmerge* примените метод разделения-слияния на полученном для данного задания изображении. Постарайтесь выбрать параметры *matOj*, *minMean*, *maxMean* таким образом, чтобы добиться максимально возможного выделения основного объекта изображения.

Программный код и полученные изображения поместите в отчет.

### **Задание № 4**

Не используя функцию *edge* создайте три функции *Roberts.m*, *Sobel.m*, *Previtt.m*, реализующих работу каждого из рассмотренных выше фильтров с помощью масок по алгоритму, представленному на рисунке 8.

Выполнение функций проверьте на заранее полученных для этого задания изображениях разного типа.

Программный код и полученные изображения поместите в отчет. В отчете сделайте выводы о фильтрах Собеля и Превитта, сравнив их эффективность работы по полученным результатам.

### Текст программы:

```
f = imread("putin_512.png");
Mask = zeros(size(f));
Mask(125,287) = 1; %Выбираем точки кристаллизации
Mask(364,312) = 1;
Mask(324,193) = 1;
Mask(114,127) = 1;
Mask(278,105) = 1;
Mask(210,50) = 1;
Mask(58,134) = 1;
Mask(42,250) = 1;
[g, NR, SI]= regiongrow(f, Mask, 20); %Сегментируем
изображение
figure, imshow(f)
figure, imshow(g) %Выводим на экран результат
сегментации в виде бинарного изображения
figure, imshow(SI) %Сегментируем изображение
figure, imshow(maskout(f,g))

% Задание 2
I = imread('putin_512.png');
I = I(1:512, 1:512);
S = qtdecomp(I, .11);
blocks = repmat(uint8(0),size(S));

for dim = [512 256 128 64 32 16 8 4 2 1]
    numblocks = length(find(S==dim));
    if (numblocks > 0)
        values = repmat(uint8(1),[dim dim numblocks]);
        values(2:dim,2:dim,:) = 0;
        blocks = qtsetblk(blocks,S,dim,values);
    end
end

blocks(end,1:end) = 1;
blocks(1:end,end) = 1;
```

```
imshow(I), figure, imshow(blocks,[])
```

```
% Задание 3
```

```
Img = imread('putin_512.png');  
a = splitmerge(double(Img), 4, @splittest);  
figure, imshow(a)  
figure, imshow(maskout(Img,a))
```

```
% Задание 4
```

```
sob = sobel(double(Img), 120);  
figure, imshow(sob)  
rob = roberts(double(Img), 35);  
figure, imshow(rob)  
prev = previt(double(Img), 70);  
figure, imshow(prev)
```

### **splittest.m**

```
function flag = splittest(region)  
    sd = std2(region);  
    m = mean2(region);  
    flag = (sd > 2) & (m > 0) & (m < 255);
```

### **sobel.m**

```
function edgeImage = sobel(originalImage, threshold)  
    maskX = [-1 -2 -1; 0 0 0; 1 2 1];  
    maskY = [-1 0 1; -2 0 2; -1 0 1];  
    H = conv2(double(originalImage), maskX, 'same');  
    V = conv2(double(originalImage), maskY, 'same');  
    E = sqrt(H.*H + V.*V);  
    edgeImage = uint8((E > threshold) * 255);
```

### **roberts.m**

```
function edgeImage = sobel(originalImage, threshold)  
    maskX = [0 0 0; 0 -1 0; 0 0 1];  
    maskY = [0 0 0; 0 0 -1; 0 0 1];  
    H = conv2(double(originalImage), maskX, 'same');  
    V = conv2(double(originalImage), maskY, 'same');  
    E = sqrt(H.*H + V.*V);  
    edgeImage = uint8((E > threshold) * 255);
```

### **previtt.m**

```
function edgeImage = sobel(originalImage, threshold)  
    maskX = [-1 -1 -1; 0 0 0; 1 1 1];  
    maskY = [-1 0 1; -1 0 1; -1 0 1];  
    H = conv2(double(originalImage), maskX, 'same');  
    V = conv2(double(originalImage), maskY, 'same');  
    E = sqrt(H.*H + V.*V);  
    edgeImage = uint8((E > threshold) * 255);
```

## Результаты

### Задание 1.

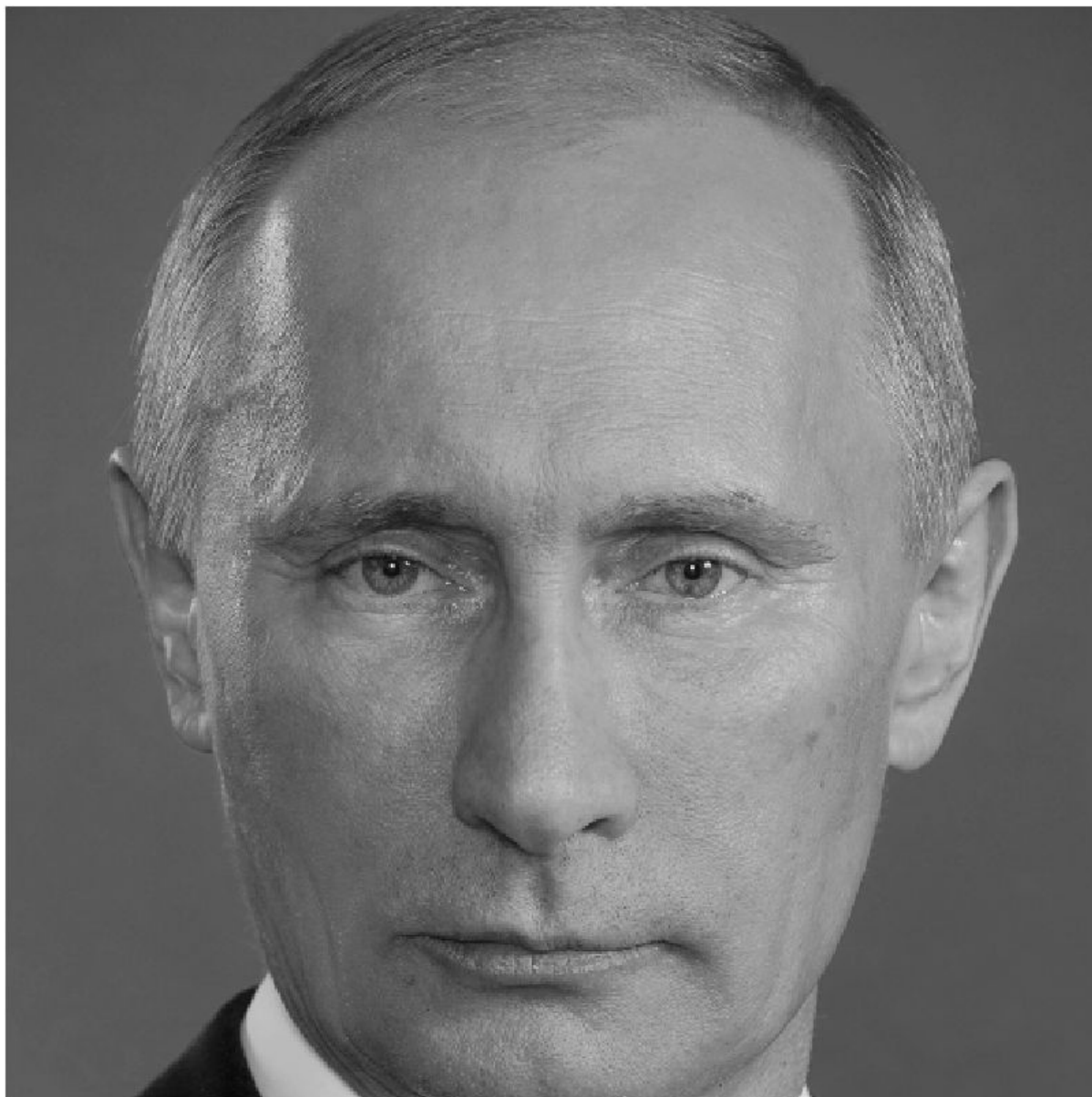


Рисунок 1 - исходное изображение

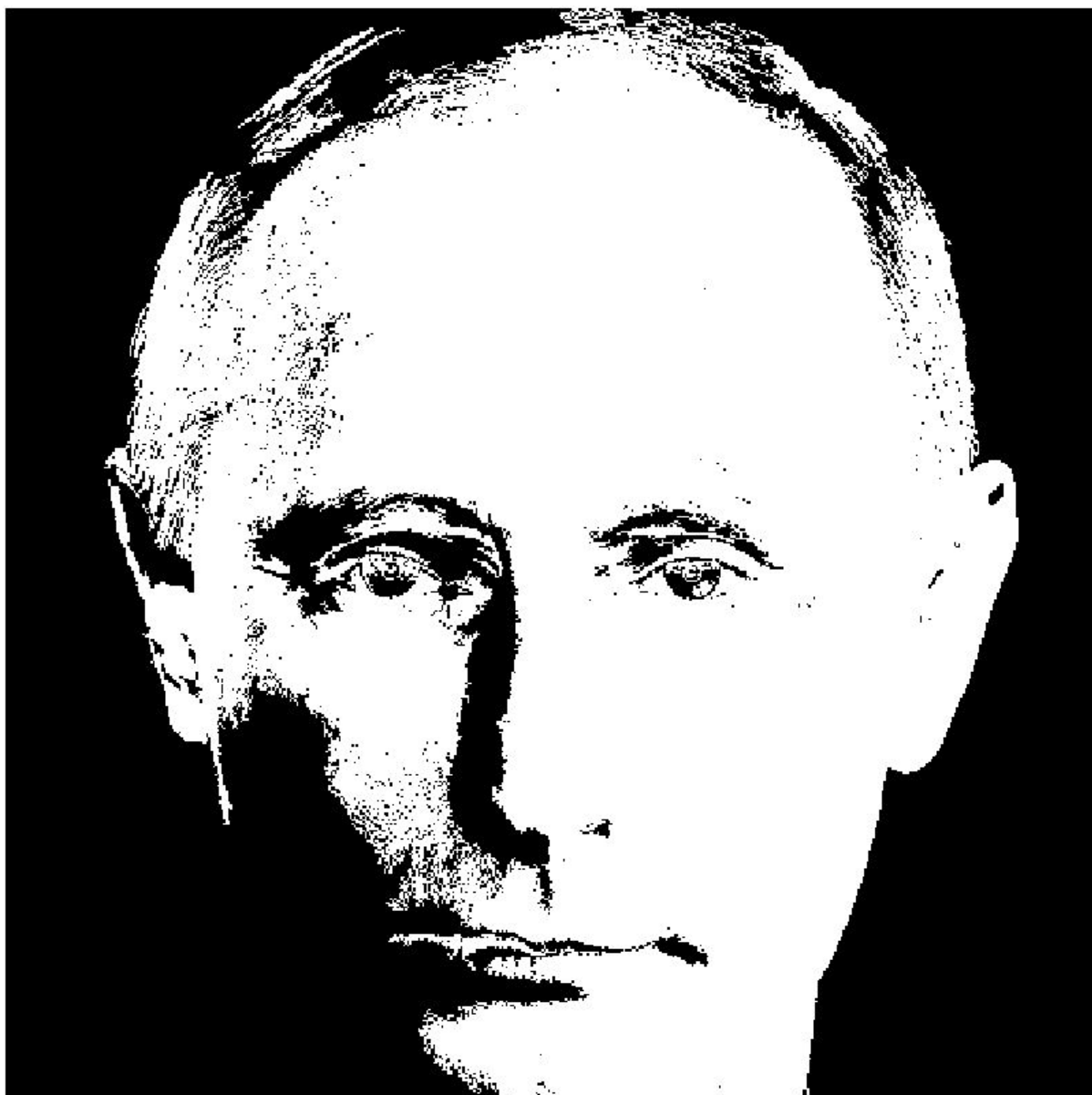


Рисунок 2 - результат сегментации методом центроидного связывания.



Рисунок 3 - точки кристаллизации

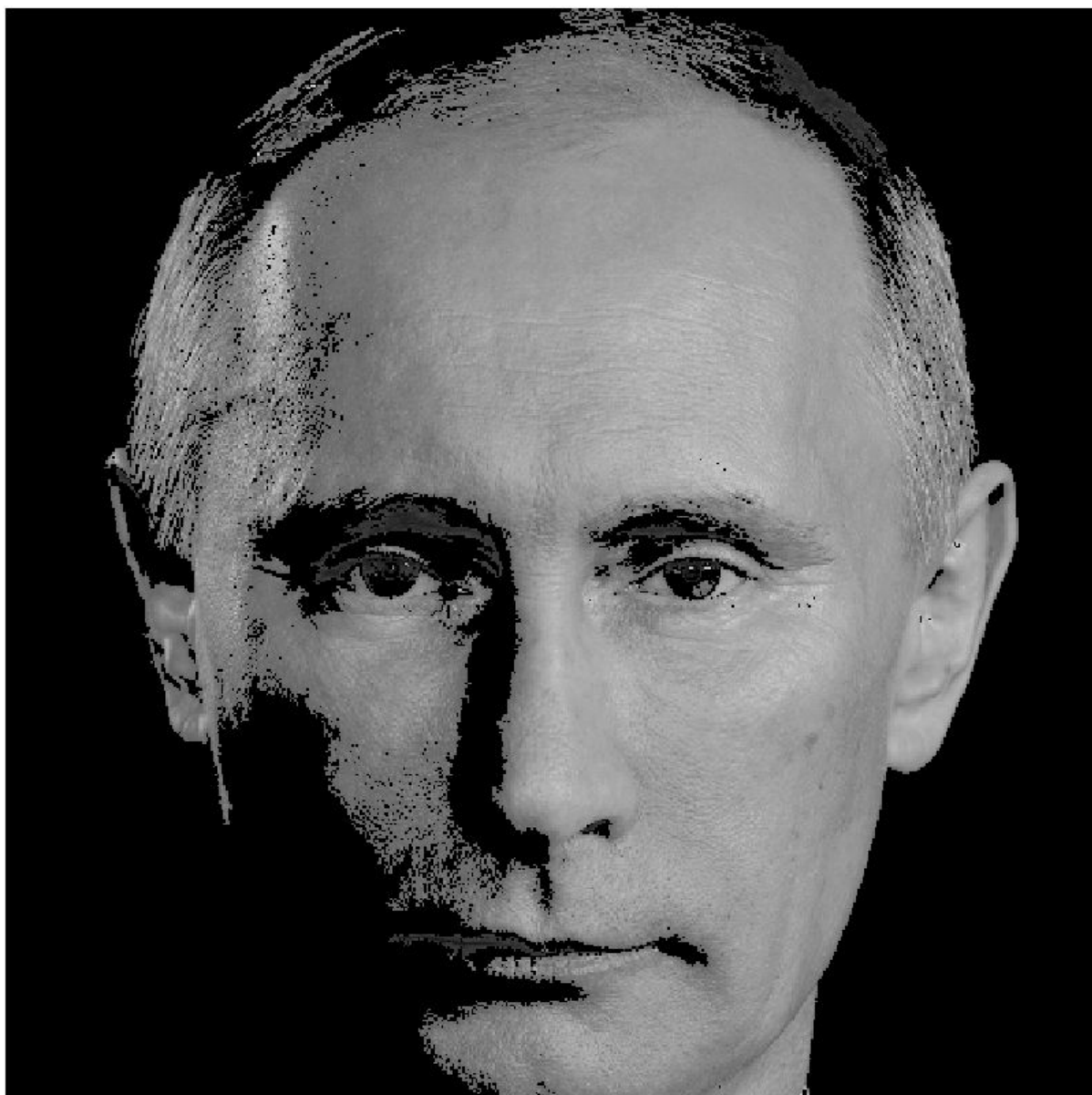


Рисунок 4 - фрагмент изображения после применения результата сегментации



Задание 2.

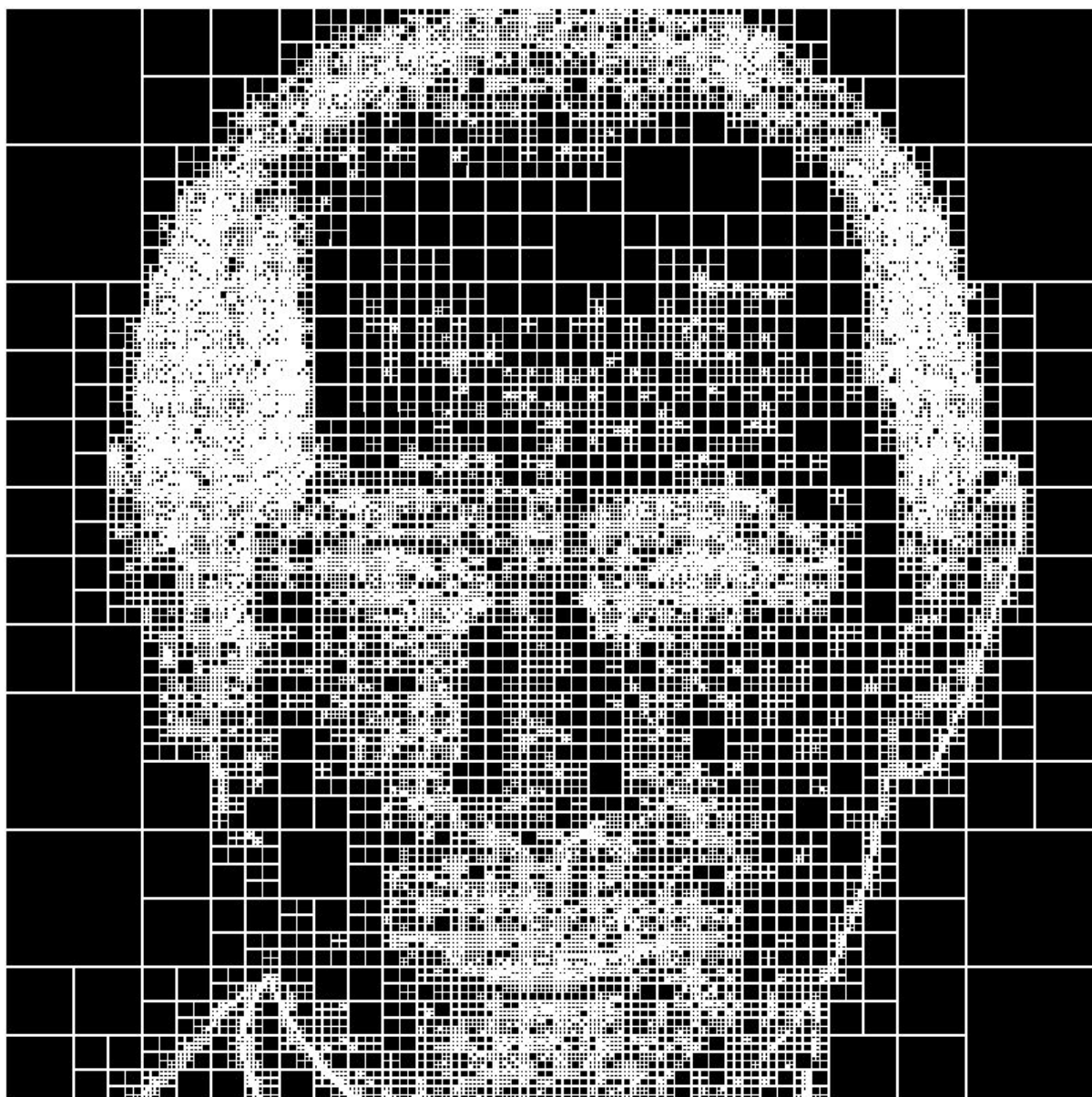


Рисунок 5 - результат квадратического разложения

Задание 3.



Рисунок 6 - результат функции splitmerge при параметрах  $matOj=2$ ,  $minMean=0$ ,  $maxMean=255$ ,  $mindim = 4$



Рисунок 7 - фрагмент изображения после применения результата сегментации методом splitmerge

Задание 4.

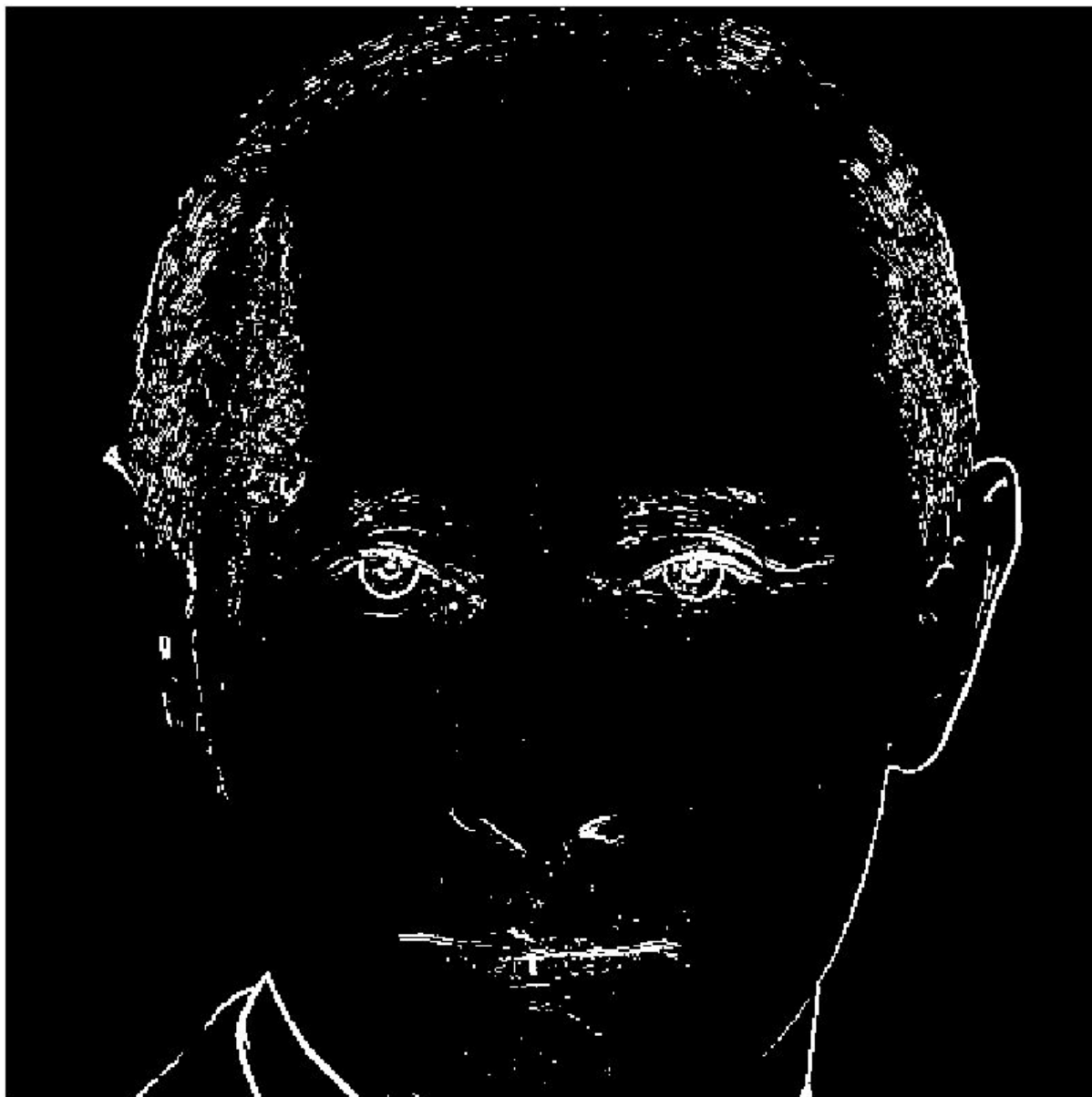


Рисунок 8 - результат применения фильтра Собеля.

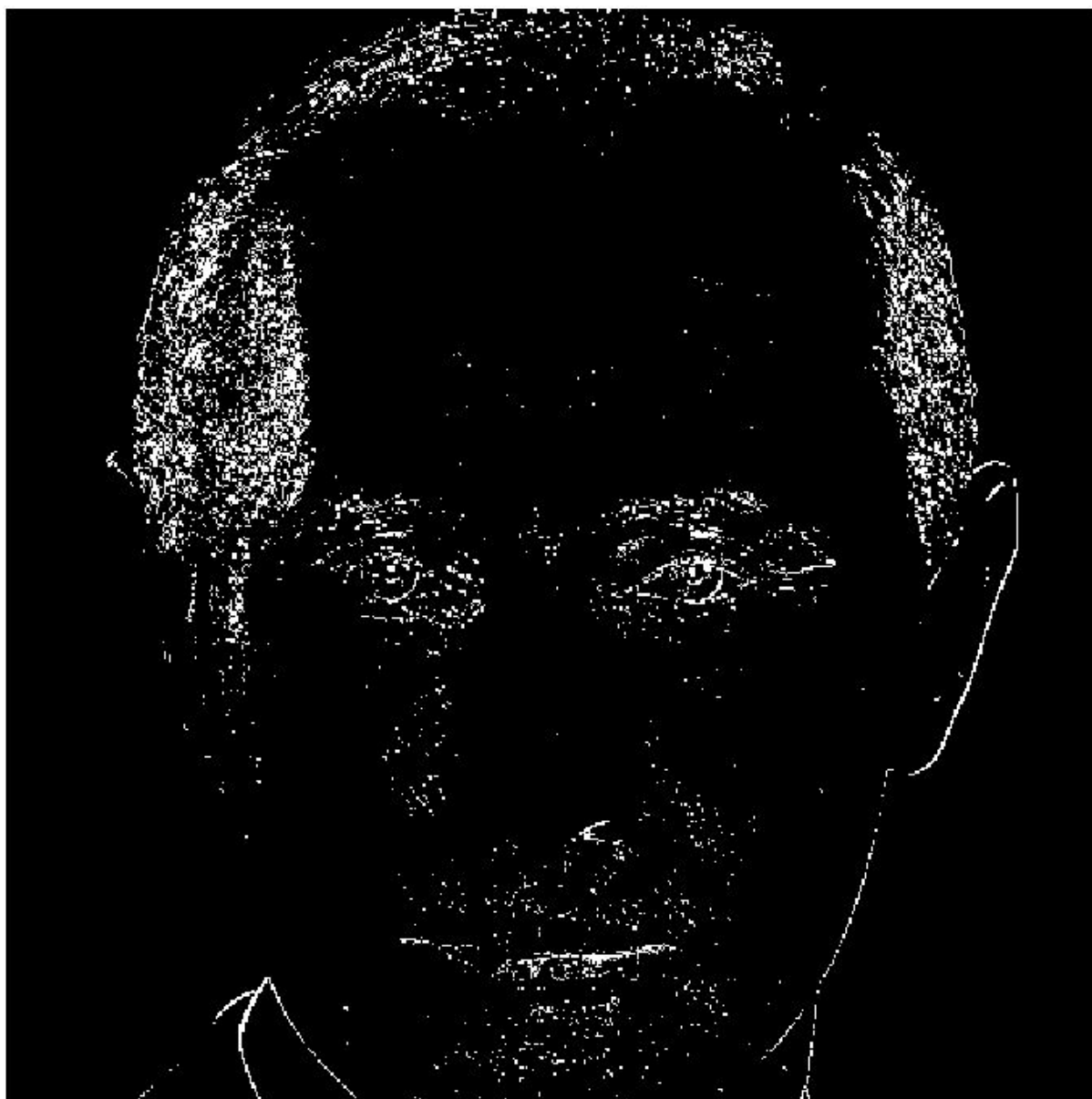


Рисунок 9 - результат применения фильтра Робертса.

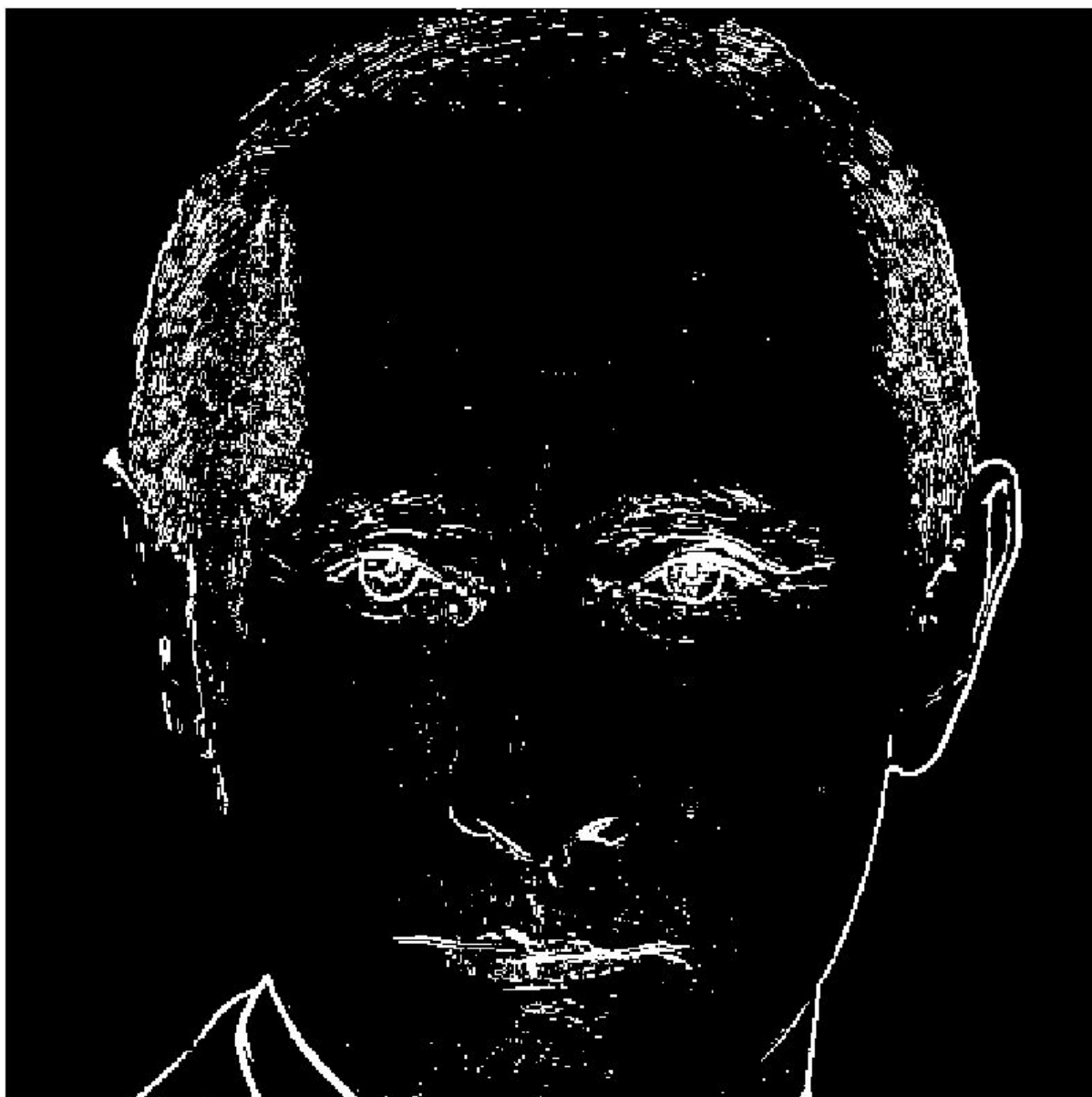


Рисунок 10 - результат применения фильтра Превитта.

## Вывод

При выполнении данной лабораторной работы было произведено сегментирование изображения методами центроидного связывания и разделения-слияния. Также было произведено квадратическое разложение изображения, реализованы фильтры Собеля, Превитта и Робертса. Было произведено выявление границ на изображении с использованием данных фильтров.

При сравнении фильтров Собеля и Превитта было определено, что при использовании одинакового порогового значения, фильтр Собеля обладает большей чувствительностью, что приводит к выявлению большего количества границ на изображении, однако вносит помехи. Варьируя граничное значение для фильтров Собеля и Превитта можно добиться получения одинаковых результатов работы фильтров.