**4ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4

по дисциплине «Цифровая обработка сигналов»

Выполнил:Ларочкин Г.И

Группа: P3400

Преподаватель: Тропченко А.А.

Санкт-Петербург

2020 г.

## Постановка задачи

**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 разложения 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.

## Решение задачи

### Задание №1

Выделяем лицо человека, используя метод центроидного связывания. Я выбрал 4 точки (лоб, переносица, левая и правая щека) кристаллизации.

#### Код task1.m:

f = imread('foto.jpg');

f = f(:,:,2);

Mask = zeros(size(f));

Mask(24, 69) = 1;

Mask(97, 87) = 1;

Mask(40, 60) = 1;

Mask(80, 50) = 1;

[g, NR, SI]= regiongrow(f, Mask, 30);

figure, imshow(f)

figure, imshow(g)

figure, imshow(SI)

f(~g) = 0;

figure, imshow(f)



Рис. 1 – исходное изображение



Рис. 2 – точки кристаллизации

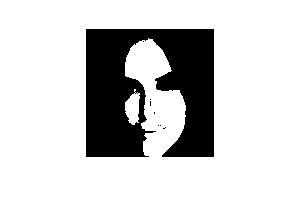


Рис. 3 – результат центроидного связывания

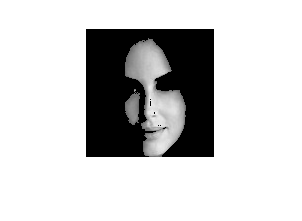


Рис. 4 – полученная маска на исходное изображение

### Задание №2

Выделим лицо с помощью квадратичного разложения. Исходное изображение – квадратное с размером 128x128.

#### Код task2.m:

I = imread('foto.jpg');

I = I(:,:,2); % select only 2 dim from jpg

S = qtdecomp(I,0.5);

blocks = repmat(uint8(0),size(S));

for dim = [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,[])

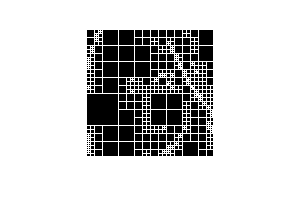


Рис. 5 – результат метода квадратичного разложения

### Задание №3

Реализуем функцию splitmerge и применим метод разделения-слияния.

#### Код splitmerge.m:

function g = splitmerge(f, mindim, mathOj, minMean, maxMean)

fun = @(region) predicate(region, mathOj, minMean, maxMean);

Q = 2^nextpow2(max(size(f)));

[M, N] = size(f);

f = padarray(f, [Q - M, Q - N], 'post');

S = qtdecomp(f, @split\_test, mindim, fun);

Lmax = full(max(S(:)));

g = zeros(size(f));

MARKER = zeros(size(f));

for K = 1:Lmax

[vals, r, c] = qtgetblk(f, S, K);

if ~isempty(vals)

for I = 1:length(r)

xlow = r(I); ylow = c(I);

xhigh = xlow + K - 1; yhigh = ylow + K - 1;

region = f(xlow:xhigh, ylow:yhigh);

flag = feval(fun, region);

if flag

g(xlow:xhigh, ylow:yhigh) = 1;

MARKER(xlow, ylow) = 1;

end

end

end

end

g = bwlabel(imreconstruct(MARKER, g));

g = g(1:M, 1:N);

function v = split\_test(B, mindim, fun)

k = size(B, 3);

v(1:k) = false;

for I = 1:k

quadregion = B(:, :, I);

if size(quadregion, 1) <= mindim

v(I) = false;

continue

end

flag = feval(fun, quadregion);

if flag

v(I) = true;

end

end

function flag = predicate(region, mathOj, minMean, maxMean)

sd = std2(region);

m = mean2(region);

flag = (sd > mathOj) & (m > minMean) & (m < maxMean);

#### Код task3.m:

f = imread('foto.jpg');

f = f(:,:,2);

q = splitmerge(f, 2, 10, 15, 255);

figure, imshow(f)

figure, imshow(q)



Рис. 6 – результат метода разделения-слияния

### Задание №4

Реализуем градиентные фильтры Робертса, Собеля и Превитта, не используя функцию edge.

#### Код gradient.m:

function g = imgradient(f, T, x\_mask, y\_mask)

f = im2double(f);

g = zeros(size(f));

for i = 2:1:size(f,1)-1

for j = 2:1:size(f,2)-1

mask = f(i-1:i+1,j-1:j+1);

gx = sum(sum(mask .\* x\_mask)).^2;

gy = sum(sum(mask .\* y\_mask)).^2;

t = sqrt(gx + gy);

if (t > T)

g(i,j) = 1;

end

end

end

end

#### Код task4.m:

f = imread('foto.jpg');

f = f(:,:,2);

r = roberts(f, 0.1);

s = sobel(f, 0.51);

p = previtt(f, 0.3);

figure, imshow(f)

figure, imshow(r)

figure, imshow(s)

figure, imshow(p)

function g = roberts(f, T)

x\_mask = [

0 0 0

0 -1 0

0 0 1

];

y\_mask = [

0 0 0

0 0 -1

0 1 0

];

g = imgradient(f, T, x\_mask, y\_mask);

end

function g = sobel(f, T)

x\_mask = [

-1 -2 -1

0 0 0

1 2 1

];

y\_mask = [

-1 0 1

-2 0 2

-1 0 1

];

g = imgradient(f, T, x\_mask, y\_mask);

end

function g = previtt(f, T)

x\_mask = [

-1 -1 -1

0 0 0

1 1 1

];

y\_mask = [

-1 0 1

-1 0 1

-1 0 1

];

g = imgradient(f, T, x\_mask, y\_mask);

end



Рис. 7 – результат фильтра Превитта



Рис. 8 – результат фильтра Робертса



Рис. 9 – результат фильтра Собеля

## Вывод

В ходе данной лабораторной работы я познакомился с методами сегментации и выделения границ изображений. Были использованы методы центроидного связывания, квадратичного разложения, разделения-слияния и наложение градиентных фильтров. Метод центроидного связывания полезен, если мы знаем координаты точек кристаллизации. Метод квадратичного разложения показал плохой результат из-за маленького размера фото. Метод разделения-слияния полезен, если мы знаем диапазон яркости пикселей, который необходим. У фильтра Превитта получились довольно жирные границы по сравнению с остальными. Фильтр Робертса имеет довольно резкие линии и результат хуже, чем у остальных фильтров. Фильтр Собеля имеет достаточно жирные линии, но не так сильно, по сравнению с фильтром Превитта и по сравнению с остальными имеет наилучший результат.