|  |
| --- |
| УНИВЕРСИТЕТ ИТМО |
| Лабораторная работа №4 по дисциплине «Цифровая обработка сигналов» |
| Сегментация изображений в системе Matlab |
| Группа Р3402 |
| **Выполнила: Орлова Кристина Александровна** |
| **Преподаватель: Тропченко Андрей Александрович** |

|  |
| --- |
| *21.01.20* |

**Цель**

* Реализация методов сегментации изображений в системе Matlab
* Исследование эффективности методов сегментации для различных видов изображений
* Реализация градиентных фильтров для выделения границ изображения

**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.

Исходный код

f = imread('face1.pgm');

seeds = zeros(size(f));

seedx = [175 135 190 216];

seedy = [30 102 50 102];

seeds(sub2ind(size(f), seedy, seedx)) = 1;

[g, NR, SI] = regiongrow(f, seeds, 40);

masked = f;

masked(g == 0) = 0;

figure, imshow(f);

figure, imshow(g);

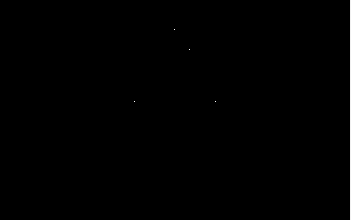
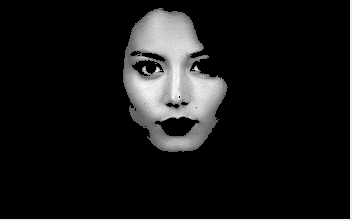
figure, imshow(masked);

figure, imshow(SI);

Результаты



а) б)



в) г)

*Рисунок 1 - Центроидное связывание (а - Исходное изображение, б - Маска, в - Сегментированное изображение, г - Точки кристаллизации)*

**Задание №2**

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.

Исходный код

I = imread('face2.pgm');

S = qtdecomp(I, 0.1);

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

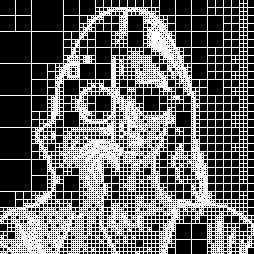
blocks(end, 1:end) = 1;

blocks(1:end, end) = 1;

figure, imshow(I);

figure, imshow(blocks, []);

Результаты



а) б)

*Рисунок 2 - Квадратичное разложение (а - Исходное изображение, б - Разложенное по квадродереву изображение)*

**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.

Исходный код

**\*predicate.m\***

function flag = predicate(region)

matOj = 4;

minMean = 0;

maxMean = 255;

sD = std2(region);

average = mean2(region);

flag = (sD >= matOj) & (average >= minMean) & (average <= maxMean);

end

**\*task3.m\***

I = imread('face3.pgm');

%disp([max(img(:)), min(img(:))]);

a = splitmerge(I, 3, @predicate);

figure, imshow(I);

figure, imshow(a);

Результаты



а) б)

*Рисунок 3 - Метод разделения-слияния (а - Исходное изображение, б - Фрагментированное изображение с блоками минимального размера 3х3)*

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

Исходный код

**\*Roberts.m\***

function filtered = Roberts(img, t)

FX = [0 0 0; 0 -1 0; 0 0 1];

FY = [0 0 0; 0 0 -1; 0 1 0];

filtered = sqrt(conv2(img, FX, 'same').^2 + conv2(img, FY, 'same').^2 );

filtered = filtered ./ max(max(filtered));

filtered(filtered < t) = 0;

filtered = filtered .\* 255;

end

**\*task4.m\***

I = imread('face4.pgm');

IR = Roberts(I, 0.2);

IS = Sobel(I, 0.2);

IP = Prewitt(I, 0.2);

subplot(1,4,1), imshow(I), title('Original');

subplot(1,4,2), imshow(IR), title('Roberts');

subplot(1,4,3), imshow(IS), title('Sobel');

subplot(1,4,4), imshow(IP), title('Prewitt');

Файлы Sobel.m, Prewitt.m аналогичны Roberts.m с отличием в значения матриц:

**\*Sobel.m\***

FX = [-1 -2 -1; 0 0 0; 1 2 1];

FY = [-1 0 1; -2 0 2; -1 0 1];

**\*Prewitt.m\***

FX = [-1 -1 -1; 0 0 0; 1 1 1];

FY = [-1 0 1; -1 0 1; -1 0 1];

Результаты



*Рисунок 4 - Выделение границ объектов изображения*

**Выводы**

В результате данной лабораторной работы были рассмотрены методы сегментирования изображения:

* Центроидное связывание, которое работает за счет самостоятельного выбора пользователем точек кристаллизации, по которым уже будет просчитываться конечное изображение. Этот способ показался не очень удобным, так как любая малая погрешность в координатах точек может вызвать в качестве результата непредвиденное изображение.
* Квадратичное разложение, которое разбивает заданное изображение на области, представляя в качестве конечного изображения собранную из областей картинку. Минус квадратичного разложения состоит в том, что оно принимает квадратные изображения шириной и высотой в степень числа 2.
* Метод разделения-слияния, который разбивает изображение на области, исходя из яркости пикселов.
* Выделение границ объектов изображения с помощью градиентных фильтров. В работе были рассмотрены 3 вида фильтров: Робертса, Собеля, Превитта, которые отличаются масками фильтров. По результатам обработки изображения видно, что фильтры Собеля и Превитта выдают идентичный результат, а фильтр Робертса отстает от них как минимум на 0.1 порога бинаризации сегментированного изображения.