

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Таныгин Максим Олегович
Должность: и.о. декана факультета фундаментальной и прикладной информатики
Дата подписания: 21.09.2023 13:12:44
Уникальный программный ключ:
65ab2aa0d384efe848e674c6d9cc0bc175e411a

МИНОБРАЗОВАНИЯ РОССИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)

Кафедра программной инженерии



УТВЕРЖДАЮ:
проректор по учебной работе
Локтионова О.Г.
2022 г.

ОБРАБОТКА ИЗОБРАЖЕНИЙ НА ОСНОВЕ
МОРФОЛОГИЧЕСКИХ ОПЕРАТОРОВ

Методические указания для выполнения лабораторных работ по дисциплине «Методы и алгоритмы обработки изображений» для студентов направления подготовки 09.03.04 «Программная инженерия»

ОБРАБОТКА ИЗОБРАЖЕНИЙ НА ОСНОВЕ МОРФОЛОГИЧЕСКИХ ОПЕРАТОРОВ

Цель и задачи лабораторного занятия (лабораторной работы): изучение методов бинаризации изображений различных типов на основе встроенных функций системы MATLAB;

владение методами препарирования изображений, управления палитрой, повышения контраста изображений;

освоение приемами применения морфологических операций, функций по реализации морфологических операций;

приобретение практических навыков и умений их использования.

Цель работы – изучение средств по, приобретение практических навыков их использования.

Планируемые результаты обучения (формируемые знания, умения, навыки и компетенции):

Код и наименование индикатора достижения компетенции, закрепленного за дисциплиной : ПК-4.5 Разрабатывает структуры данных

Необходимые материально-техническое оборудование и материалы:

1. Класс ПЭВМ - Athlon 64 X2-2.4; Cel 2.4, Cel 2.6, Cel 800.
2. Мультимедиа центр: ноутбук ASUS X50VL PMD T2330/14"/1024Mb/ 160Gb/ сум-ка/проектор inFocus IN24+ .
3. Экран мобильный Draper Diplomat 60x60
4. Доступ в сеть Интернет.

Шкала оценивания и критерии оценивания выполненной лабораторной работы:

Форма контроля	Минимальный балл		Максимальный балл	
	балл	примечание	балл	примечание
Лабораторная работа № 1 (Работа с файлами различных типов изображений в среде MATLAB)	2	Выполнил, но «не защитил»	4	Выполнил и «защитил»

План проведения лабораторного занятия (лабораторной работы)

1. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

1. Преобразовать палитровое изображение в бинарное с использованием автоматического выбора порогового значения.

2. Преобразовать цветное изображение в бинарное, используя функцию **graythresh**.

3. Выполнить бинаризацию палитрового изображения с выбором порога по умолчанию и вычислить порог с помощью функции **graythresh**.

4. Выполнить эрозию объекта на изображении, изменяя тип структурообразующего элемента, руководствуясь данными из таблицы 2.

5. Выполнить дилатацию объекта на изображении, изменяя тип структурообразующего элемента, руководствуясь данными из таблицы 2.

6. Выделить объекты на изображении файла, используя операции замыкания и размыкания.

7. Представить отчет и листинг программы.

2. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

2.1 Бинаризация изображений

Анализ бинарных изображений включает операции, применяемые в бинарном машинном зрении, например, при анализе частиц, при обработке документов, и при анализе полутоновых изображений. Он предполагает обработку бинарных изображений, яркости элементов которых имеют значения 0 или 1, при этом 0 представляет фон, 1 представляет объект [3,4].

По признаку цветности из изображения выделяются объекты заданного цвета. В результате сегментации формируется бинарное изображение, на котором наряду с полезной информацией остаются области шума. В результате селекции связанных областей каждой связанной области назначается определенный уровень яркости (номер области). После выполнения селекции и выделения областей в некотором заданном диапазоне площадей удастся отфильтровать шум и сосчитать клетки заданного цвета, пронумеровав их [7,8].

Назначение основных операций бинарного анализа

Первая операция бинарного анализа состоит в выделении объектов на фоне из множества других объектов. Только после этого выполняется объединение элементов изображения каждого объекта. После получения бинарного изображения объекта возможно производить вычисление признаков объекта.

Состав основных операций бинарного анализа

К основным операциям бинарного анализа относится операция «*выделение признаков*», по которой можно выделить объект на фоне. В связи с этим перед бинарными операциями выполняются полутоновые преобразования, позволяющие усилить признаки объекта.

Следующая задача связана с определением порогов для выполнения бинаризации.

Пороговая бинаризация полутоновых изображений в соответствии с амплитудными характеристиками, позволяет получить бинарное изображение объекта.

Пусть f_{ij} – полутоновое изображение, t – порог и b_0 и b_1 – два бинарных значения. Результатом порогового разделения является бинарное изображение, полученное следующим образом:

$$f_{ij} = \begin{cases} b_0, & \text{если } f_{ij} \leq t \\ b_1, & \text{если } f_{ij} > t \end{cases}$$

Основной задачей является выбор порогового значения t с помощью некоторого критерия. Это значение может выбираться как одинаковым для всего изображения, так и различным для различных его частей. Если значения объекта и фона достаточно однородны по всему изображению, то может применяться одно пороговое значение для всего изображения. Использование единственного порогового значения для всех пикселей называется **глобальным пороговым значением**.

Бинаризацию полутонового изображения можно выполнить операцией *сравнения значений уровней серого с заданным порогом*

(бинаризация по порогу). Если в качестве операции сравнения выбирается операция:

«больше выбранного уровня» ($I > \mathbf{threshold}$), получается позитив;

«меньше выбранного уровня» ($I < \mathbf{threshold}$) – негатив.

Выбор порогового значения

Существует много способов выбора порогового значения, например, разделом двух основных пиков на гистограмме яркости, усреднением функции яркости и др. Для *автоматического выбора порога* предлагается следующий алгоритм.

1. Выбрать некоторую начальную оценку для значения порога T (предлагаемая величина равна *среднему значению между минимальной и максимальной яркостью изображения*).

2. Выполнить сегментацию с помощью порога T . В результате образуется две группы пикселей: $G1$ и $G2$. Область $G1$ состоит из пикселей, яркость которых больше или равна T , а область $G2$ – из пикселей, яркость которых меньше T .

3. Вычислить среднюю яркость пикселей $S1$ и $S2$ по областям $G1$ и $G2$.

4. Вычислить новое значение порога $T = \frac{1}{2(S1 + S2)}$.

Повторить шаги со 2-го по 4-й, пока разность значений порогов T для соседних итераций не станет меньше наперед заданного значения ε .

Задание 1. Преобразовать палитровое изображение из файла Athena.bmp в бинарное с использованием автоматического выбора порога.

```
>> [X,map]=imread('c:\image\Athena.bmp');
>> I=im2double( ind2gray(X,map));
>> figure,imshow(I)
>> T = 0.5*(min(I(:)) + max(I(:)));
>> done = false;
>> while ~done
g=I>= T;
Tnext = 0.5*(double(min(I(g))) + double(max(I(~g))));
```

```
done = abs(T - Tnext) < 0.5;
T = Tnext;
end
>> bw = I > T;
```

```
>> figure, imshow(bw)
```

Для вычисления порога также можно использовать функцию `graythresh` пакета IPT, которая вычисляет порог по *методу Отса*.

Синтаксис

T = graythresh(S),

где **S** – исходное полутоновое изображение; **T** – глобальный порог, значения которого принадлежат интервалу [0 1].

Задание 2. Преобразовать цветное изображение из файла `bike.jpg` в бинарное с использованием функции `graythresh`.

```
>> rgb = imread('c:\Image\bike.jpg');
>> I = im2double(rgb2gray(rgb));
>> figure, imshow(I)
>> T = graythresh(I);
>> Bw = I > T;
>> figure, imshow(Bw)
```

Задание 3. Получить негатив с помощью бинаризации по порогу палитрового изображения, хранящегося в файле `Technlgy.bmp`.

```
>> [X,map] = imread('C:\Image\Earth.bmp' );
>> I = ind2gray(X,map);
>> figure, imshow(I)
>> T = graythresh(I);
>> BW = I < T;
>> figure, imshow(BW)
```

В системе MATLAB бинаризация изображения отсечением по порогу яркости выполняется функцией **im2bw**.

Синтаксис

BW = im2bw(I, threshold)

BW = im2bw(X, map, threshold)

BW = im2bw(RGB, threshold)

Функция **im2bw** создает бинарное изображение, используя отсечение по порогу яркости **threshold**. Для этой цели функция конвертирует полноцветные и палитровые изображения в полутоновые. Пороговое значение **threshold** должно задаваться из диапазона [0,1]. По умолчанию **threshold = 0.5**. Для вычисления порога можно воспользоваться функцией **graythresh**.

Задание 4. Выполнить бинаризацию палитрового изображения файла Technlgy.bmp:

а) выбрать порог по умолчанию:

```
>> [X,map] = imread('c:\Image\Technlgy.bmp');
>> figure,imshow(X,map)
>> BW = im2bw(X,map);
>> figure,imshow(BW)
```

б) вычислить порог с помощью функции **graythresh**:

```
>> [X,map] = imread('c:\Image\Technlgy.bmp');
>> I = im2double(ind2gray(X,map));
>> figure,imshow(I)
>> T = graythresh(I);
>> BW = im2bw(I,T);
>> figure,imshow(BW)
```

Для многих изображений глобальное пороговое значение не может применяться из-за неоднородностей внутри областей фона и объекта [5, 9]. Для таких изображений требуются различные пороговые значения для различных частей изображения. Использование различных пороговых значений для различных частей изображения называется **аддитивным или локальным пороговым разделением**.

Методика локального порогового разделения основана на разбиении первоначального изображения на меньшие части и определении порога для каждой части изображения. В результате получается бинарное изображение с разрывами серого уровня на границах фрагментов. Для устранения неоднородностей применяются сглаживающие методики [4,6,10].

2.2. Основные морфологические операции

Рассмотрим несколько простых, но важных взаимосвязей между пикселями в цифровом изображении. Рассматриваемые понятия широко применяются в обработке бинарных изображений и морфологических операциях, при выделении объектов и вычислении их признаков. Для определенности в системе MATLAB полагают, что пиксели со значениями, равными 1, относятся к объектам, а со значениями, равными 0, – к фону.

Соседние пиксели

Пиксель p с координатами (x, y) имеет четыре горизонтальных и вертикальных соседних пикселя с координатами:

$$(x + 1, y), (x - 1, y), (x, y + 1), (x, y - 1).$$

Эта группа пикселей $x1, x3, x5, x7$, называемая «четыре соседа p » или «квartetом соседей», обозначается через **N4(p)**.

Данные четыре пикселя находятся на одном расстоянии от (x, y) , а также некоторые из соседних пикселей p могут быть за пределами цифрового изображения, если (x, y) находится на границе изображения. Четыре диагональных соседних пикселя p имеют координаты

$$(x + 1, y + 1), (x + 1, y - 1), (x - 1, y + 1), (x - 1, y - 1)$$

и обозначаются через **ND(p)**. Эти точки вместе с четырьмя указанными выше называются восьмью-соседями, или октетом соседей пикселя p и обозначаются через **N8(p)**. Некоторые из точек **ND(p)** и **N8(p)** также могут выходить за пределы изображения, если (x, y) находится на границе изображения.

Связи пикселей

Объект называется *четырёхсвязный*, если для каждого пикселя объекта среди квартета соседних пикселей существует хотя бы один, равный 1 и принадлежащий этому объекту.

Объект называется *восьмисвязный*, если для каждого пикселя объекта среди октета соседних пикселей существует хотя бы один, равный 1 и принадлежащий этому объекту.

Аналогичные определения связности можно ввести для фона. Четырёхсвязность фона автоматически означает восьмисвязность объектов и наоборот. Среди связных областей объектов могут встречаться связные области из пикселей фона. Они называются дырами.

Морфологические операции

Математическая морфология предлагает единый мощный подход для решения многочисленных задач обработки и анализа изображений [2,4,7,8]. В математической морфологии используется язык теории множеств. В математической морфологии изображения представляются множествами, элементами которых являются пиксели с атрибутами координат и яркости или координат и R, G, B – коды. Приведем основные базовые понятия теории множеств, лежащие в основе математической морфологии.

Теория морфологии (морфологии – наука о форме) рассматривает бинарное изображение в виде множеств его пикселей переднего плана (со значениями 1), которое лежит в пространстве Z^2 (двухмерное пространство). Над бинарными изображениями можно совершать операции как над множествами с помощью следующих логических операций MATLAB [1,2,5]:

OR (\cup) – логическое сложение – объединение множеств $A \cup B$;

AND ($\&$) – логическое умножение – пересечение множеств $A \cap B$;

NOT (\sim) – отрицание – дополнение множества A^c ;

DIFFERENCE ($A \setminus B$) – разность множеств $A \setminus B$;

Логические операции MATLAB, применяемые к бинарным изображениям [3], приведены в таблице 1.

Таблица 1

Логические операции Matlab для бинарных изображений

Операции	Выражение Matlab для бинарных выражений	Имя
$A \cup B$	$A B$	OR
$A \cap B$	$A \& B$	AND
A^C	\tilde{A}	NOT
$A \setminus B$	$A \& \tilde{B}$	DIFFERENCE

Пусть A – множество в пространстве Z^2 . Если x является элементом множества A , то обозначается $x \in A$, иначе $x \notin A$. *Пустое* множество – это множество, не содержащее ни одного элемента, обозначается символом \emptyset .

Если элементы множества A являются элементами другого множества B , то говорят, что A является *подмножеством* множества B и обозначается $A \subset B$.

Объединение двух множеств A и B , которое обозначается $C = A \cup B$, есть по определению множество всех элементов, принадлежащих либо множеству A , либо множеству B , либо обоим множествам одновременно.

$$A \cup B = \{x | x \in A \text{ или } x \in B\}.$$

Пересечение двух множеств A и B , которое обозначается $D = A \cap B$, есть по определению множество, состоящее из тех и только тех элементов, которые принадлежат одновременно обоим множествам.

$$A \cap B = \{x | x \in A \text{ и } x \in B\}.$$

Непересекающимися назовем два множества A и B , если у них нет общих элементов, в этом случае $A \cap B = \emptyset$.

Дополнение множества A есть множество элементов, не содержащихся в A :

$$A^C = \{x | x \notin A\}.$$

Разность двух множеств A и B обозначается $A \setminus B$ и определяется как

$$A \setminus B = \{x | x \in A, x \notin B\} = A \cap B^C.$$

Центральное отражение множества A обозначается \hat{A} и определяется в виде

$$\hat{A} = \{w \mid w = -x, \quad x \in A\}.$$

Параллельный перенос (или сдвиг) множества A в точку $z = (z_1, z_2)$ обозначается $(A)_z$ и определяется по правилу

$$(A)_z = \{c \mid c = x + z, \quad x \in A\}.$$

Одним из основных понятий математической морфологии является понятие *структурообразующего, или структурного элемента*. Структурный элемент B – это множество, состоящее из двух непересекающихся подмножеств B_1 и B_2 , для которых определено общее начало.

Функция создания структурообразующего элемента **strel**

Функция **strel** строит структурообразующие элементы различных форм и размеров.

Синтаксис

se = strel(shape, parameters),

где **se** – структурообразующий элемент; **shape** – строка, задающая форму структурообразующего элемента; **parameters** – дополнительные параметры, которые уточняют информацию о его форме.

В таблице 2 приведены некоторые варианты задания формы структурообразующего элемента.

Результатом выполнения функции **strel** является так называемый **strel**- объект. Для его разложения используется функция **getsequence**.

Таблица 2

Способы определения структурообразующего элемента

Вызов функции strel	Форма структурообразующего элемента	Дополнительные параметры
se = strel('diamond', R)	ромб	R – расстояние от центра структурообразующего элемента до крайней точки ромба
se = strel('disk', R)	круг	R – радиус структурообразующего

		элемента
se = strel('line', LEN, DEG)	линейный элемент	LEN – длина, DEG – угол наклона линии против часовой стрелки от горизонтальной оси (в градусах)
se = strel('pair', OFFSET)	две точки: первая точка находится в центре, а вторая смещена от первой на вектор OFFSET	OFFSET – двумерный вектор с неотрицательными целыми компонентами
se = strel('rectangle', MN)	прямоугольник	MN – двумерный вектор с неотрицательными целыми компонентами. MN(1) – число строк, MN(2) – число столбцов
se = strel(NHOOD)	элемент произвольной формы	NHOOD – матрица из нулей и единиц, задающая его форму

Синтаксис

decomp = getsequence(se)

Структурообразующие элементы разложения можно получить, индексируя переменную **decomp**.

Морфологические операции *дилатации* и *эрозии* имеют основополагающее значение при морфологической обработке изображений.

Операция *эрозии* «ужимает» или «утончает» объекты двоичных изображений. Операция *дилатации* «наращивает» или «утолщает» на объекты двоичных изображениях. *Способ и степень этих преобразований контролируется формой структурного элемента.*

Эрозия

Эрозией множества X по B называется множество

$$Y = X \ominus B = \{x : (B)_x \cap X^c = \emptyset\}$$

где \emptyset – пустое множество.

Эрозия X по B состоит из пикселей с такими координатами, для которых сдвиг множества в эту точку не пересекается с фоном изображения X .

Эрозия выполняется функцией **imerode** пакета IPT.

Синтаксис

D = imerode(S, se),

где **S** – это двоичное или полутоновое изображение; **se** – матрица из нулей и единиц, которая определяет структурообразующий элемент.

Задание 5. Выполнить эрозию изображения из файла `erode.bmp`. Удалить тонкие провода с изображения с сохранением всех остальных структур. Необходимо выбрать достаточно малый структурный элемент, чтобы он помещался внутри центрального квадрата, а также внутри толстых полосок у границ, и достаточно большим, чтобы он не помещался внутри удаляемых проводков.

```
>> [x,map] = imread('c:\image\erode.bmp');
>> I=im2double(ind2gray(x,map));
>> figure,imshow(I)
>> T=graythresh(I);
>> BW = I>T;
>> figure,imshow(I)
>> se = ones(15);
>> se(15,15) = 0;
>> se(15,1) = 0;
>> se(1,15) = 0;
>> se(1,1) = 0;
>> BW1 = imerode(BW,se);
>> figure,imshow(BW1)
>> se = ones(18);
>> se(18,18) = 0;
>> se(18,1) = 0;
>> se(1,18) = 0;
>> se(1,1) = 0;
```

```

>> BW1 = imerode(BW,se);
>> figure,imshow(BW1)
>> se = ones(60);
>> se(60,60) = 0;
>> se(60,1) = 0;
>> se(1,60) = 0;
>> se(1,1) = 0;
>> BW1 = imerode(BW,se);
>> figure,imshow(BW1)

```

Дилатация

Операцией, двойственной к эрозии, является *дилатация (dilatation)*, которая определяется следующим образом:

$$Y = X \oplus B = \{x : (\hat{B})_x \cap X \neq \emptyset\}$$

Дилатация выполняется функцией **imdilate** пакета IPT.

Синтаксис

D = imdilate (S, se),

где **S** – это двоичное или полутоновое изображение, **se** – матрица из нулей и единиц, которая определяет структурообразующий элемент.

Задание 6. Выполнить дилатацию изображения из файла TextRoman.bmp.

```

>> I = imread('c:\Image\TextRoman.bmp');
>> bw = I>150;
>> figure,imshow(bw)
>> se = [0 1 0;1 1 1;0 1 0];
>> bw1 = imdilate(bw,se);
>> figure,imshow(bw1)

```

Эрозия и дилатация – операции, предназначенные в первую очередь для выявления различных морфологических особенностей изображения с использованием различных структурных элементов. Так, эрозия посредством круга с радиусом r позволяет найти в изображении объекты, минимальный поперечный размер которых превышает $2r$. Если в качестве структурного элемента взять две точки, смещение между которыми определяется вектором h , эрозия позволит выделить объекты, имеющие соседей в направлении и на расстоянии, заданных этим вектором.

Задание 7. Найти объекты на изображении файла `cgc.bmp`, поперечный размер которых превышает 30 пикселей с помощью эрозии посредством круга с радиусом $r = 15$.

```
>> f1 = imread('c:\Image\cgc.jpg');
>> I=im2double(rgb2gray(f1));
>> T=graythresh(I);
>> bw=I<T;
>> figure,imshow(bw)
>> title('original')
>> r=15;
>> se = strel('disk', r);
>> bw1 = imerode(bw,se);
>> figure,imshow(bw1)
>> title('rezult')
```

Операции замыкание и размыкание являются комбинированием дилатации и эрозии.

Морфологические операции размыкание и замыкание

Морфологическое размыкание A по B обозначается $A \circ B$ и определяется как эрозия A по B , после которой выполняется дилатация результата по B :

$$A \circ B = (A \ominus B) \oplus B.$$

Операция **размыкания** допускает простую геометрическую интерпретацию (рис. 2.1). Предположим, что примитив B имеет форму круга. Тогда границу множества $A \circ B$ образуют максимально близкие к границе A точки множества B при «обкатывании» примитивом B этой границы изнутри.

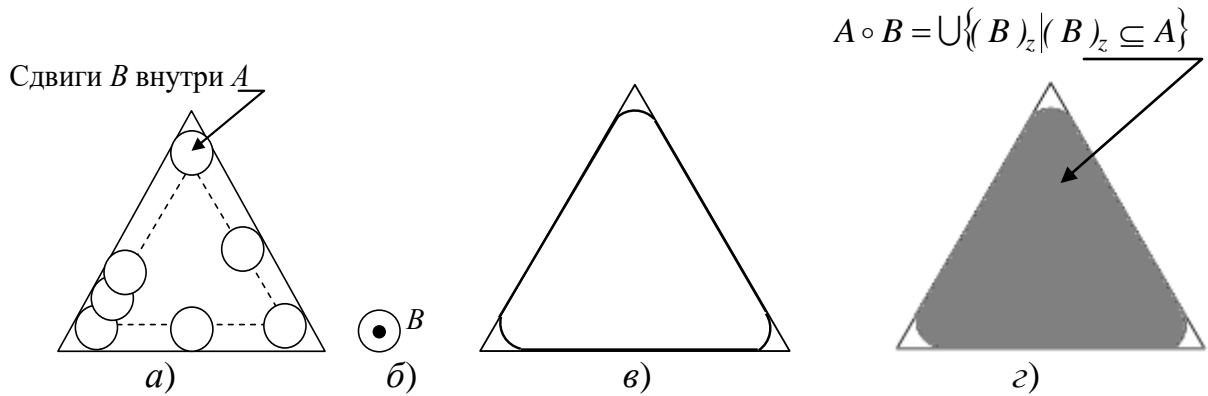


Рисунок 2.1- Иллюстрация операции размыкания:
a – примитив *B*, окатывающий изнутри границу множества *A*; *б* – примитив *B*, положение его центра указано точкой; *в* – жирная линия показывает внешнюю границу размыкания; *г* – все множество, получаемое в результате размыкания

Морфологическое замыкание множества *A* по примитиву *B* обозначается $A \bullet B$ и определяется как

$$A \bullet B = (A \oplus B) \ominus B$$

то есть как дилатация множества *A* по *B*, затем следует эрозия по тому же примитиву *B*.

Замыкание допускает сходную геометрическую интерпретацию, но в этом случае примитив *B* обкатывает границу *A* снаружи (рис. 2.2).

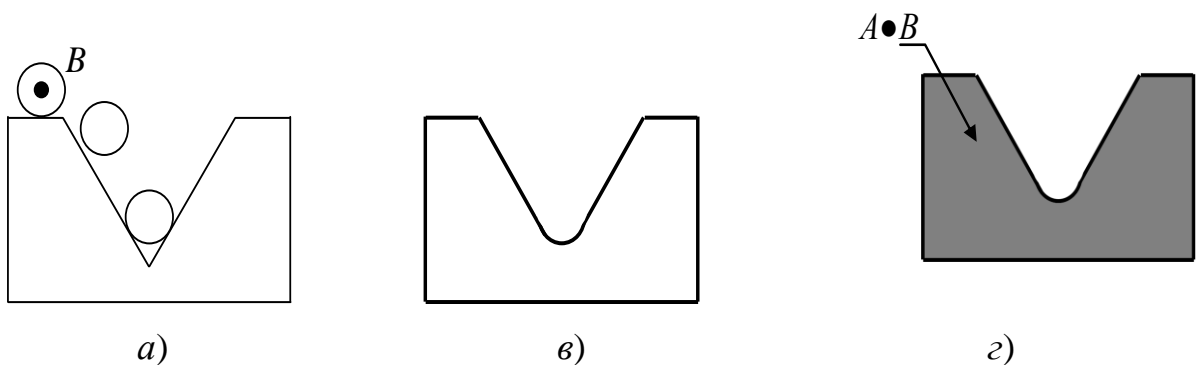


Рисунок 2.2- Иллюстрация операции замыкания:
a – примитив *B*, обкатывающий снаружи границу множества *A*; *б* – внешняя граница замыкания; *в* – все множество, получаемое в результате замыкания

Преобразование *размыкания* и *замыкания* реализовано функциями **imopen** и **imclose** соответственно.

Синтаксис

D = imopen(S, se)

D = imclose(S, se),

где *S* – это двоичное или полутоновое изображение; *se* – матрица из нулей и единиц, которая определяет структурообразующий элемент.

Задание 8. Выделить объекты на изображении файла *Ex4.bmp*, используя операции замыкания и размыкания.

```
>> [x,map]= imread('C:\Image\Ex4.bmp');
>> figure,imshow(x,map)
>> I=im2double(ind2gray(x,map));
>> T=graythresh(I);
>> bw=im2bw(I,T);
>> figure,imshow(bw), title('bw')
>> R=18, se = strel('disk', R);
>> bwo=imopen(bw,se);
>> figure,imshow(bwo)
>> r=int2str(R);
>> s=cat(2,'bwo-',r)
>> title(s)
>> R=5;
>> se = strel('disk', R);
>> bwcl=imclose(bwo,se);
>> figure,imshow(bwcl)
>> r=int2str(R);
>> s=cat(2,'bwcl-',r)
>> title(s)
```

3. ПРИМЕРЫ ВЫПОЛНЕНИЯ ЗАДАНИЙ

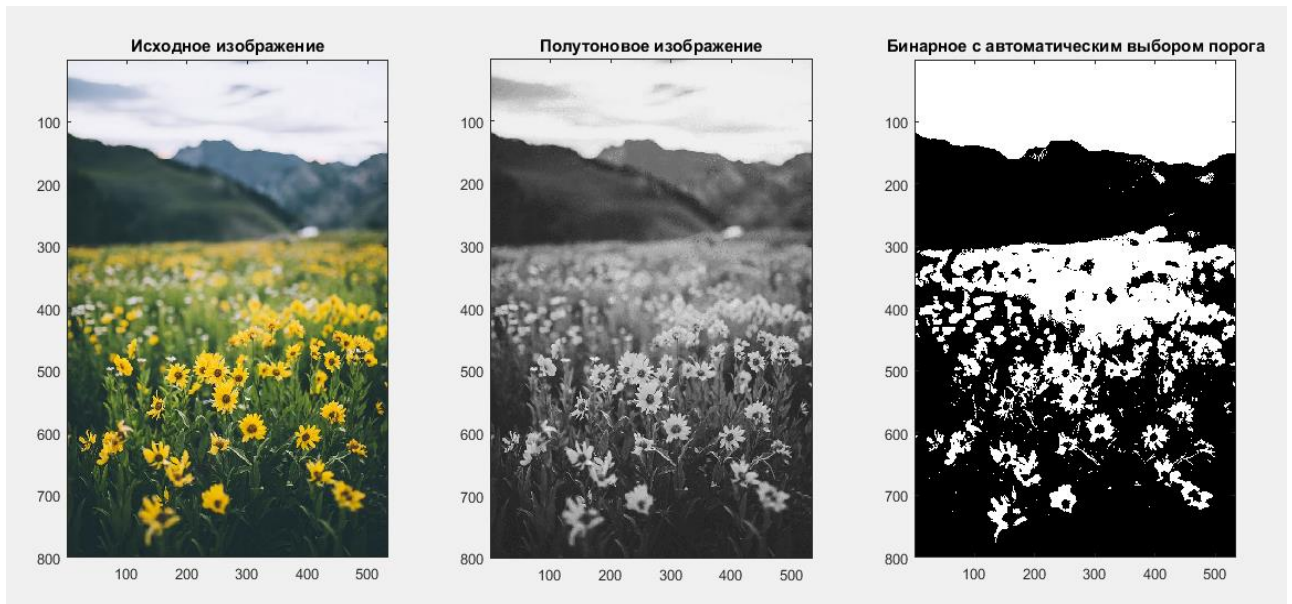


Рисунок 3.1 – Результат преобразования палитрового изображения в бинарное с использованием автоматического выбора порогового значения

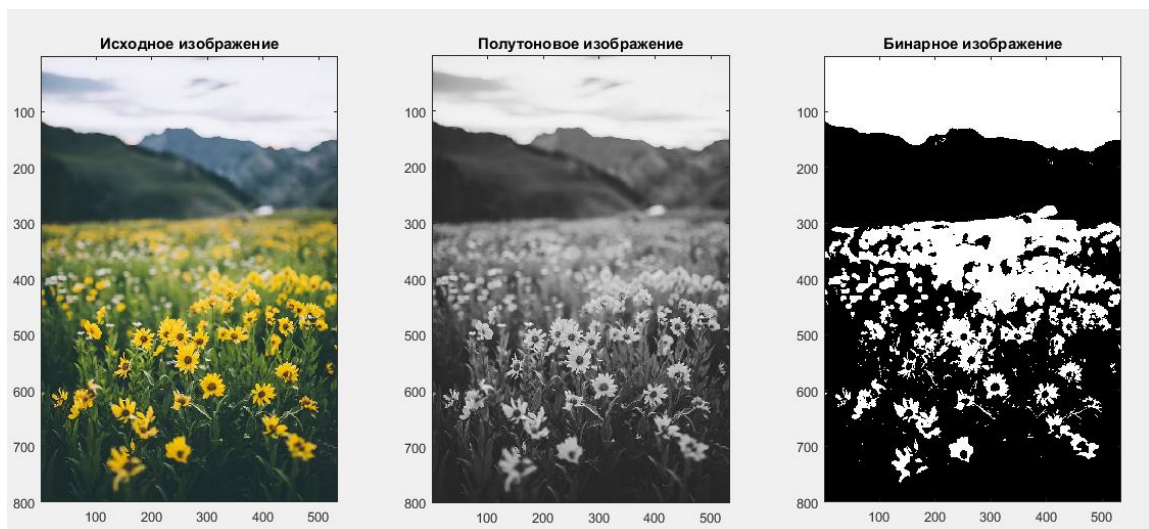


Рисунок 3.2 – Результат преобразовать цветное изображение в бинарное, используя функцию graythresh

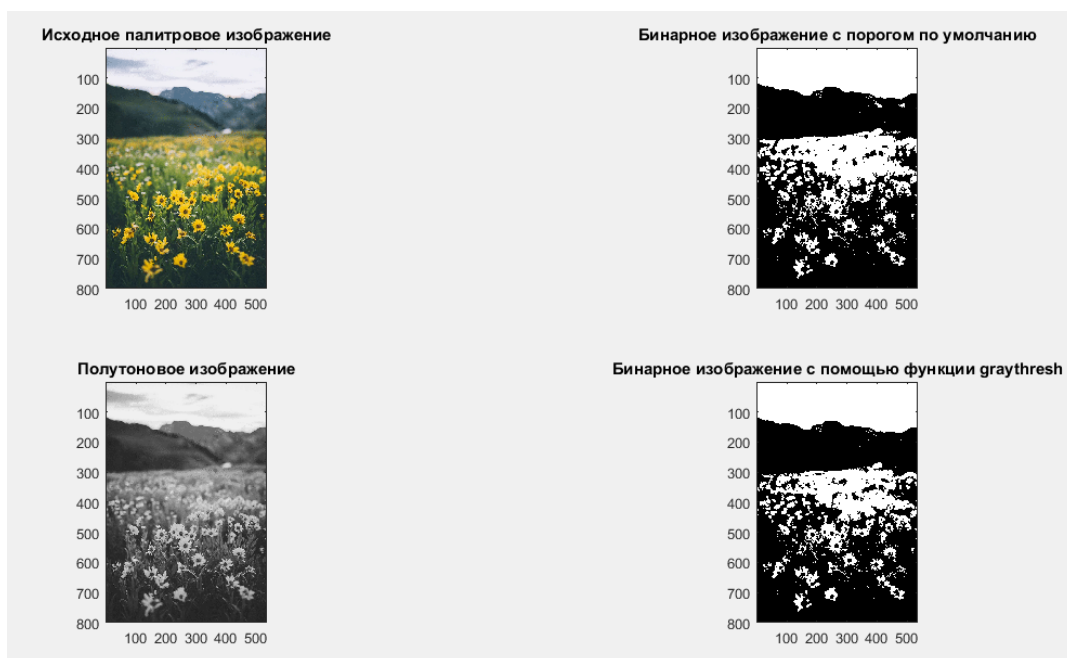


Рисунок 3.3 – Результат выполнения бинаризации палитрового изображения с выбором порога по умолчанию и вычислить порог с помощью функции `graythresh`

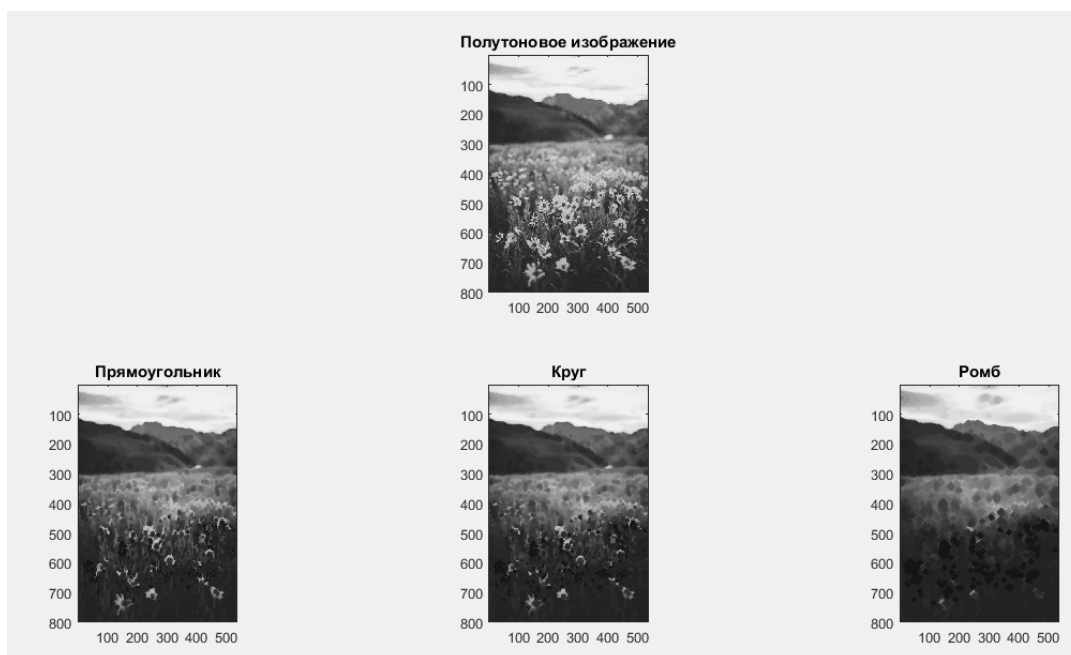


Рисунок 3.4 – Результат выполнения эрозии объекта на изображении, изменяя тип структурообразующего элемента

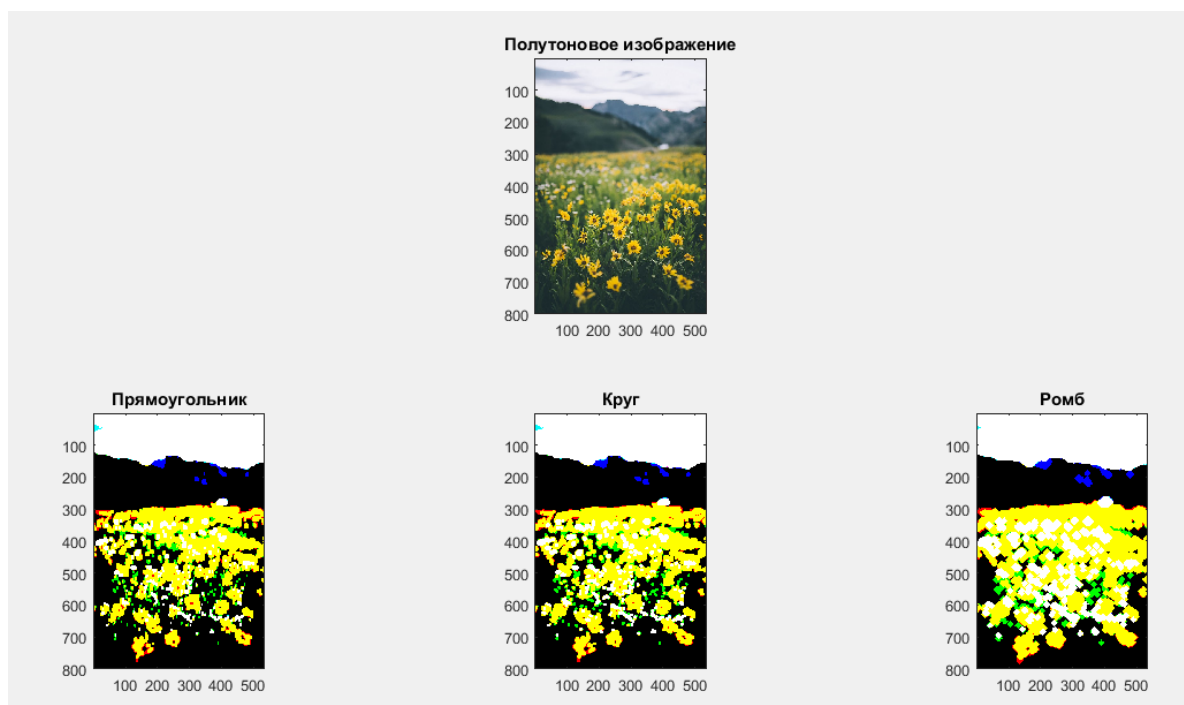


Рисунок 3.5 – Результат выполнения дилатации объекта на изображении, изменяя тип структурообразующего элемента

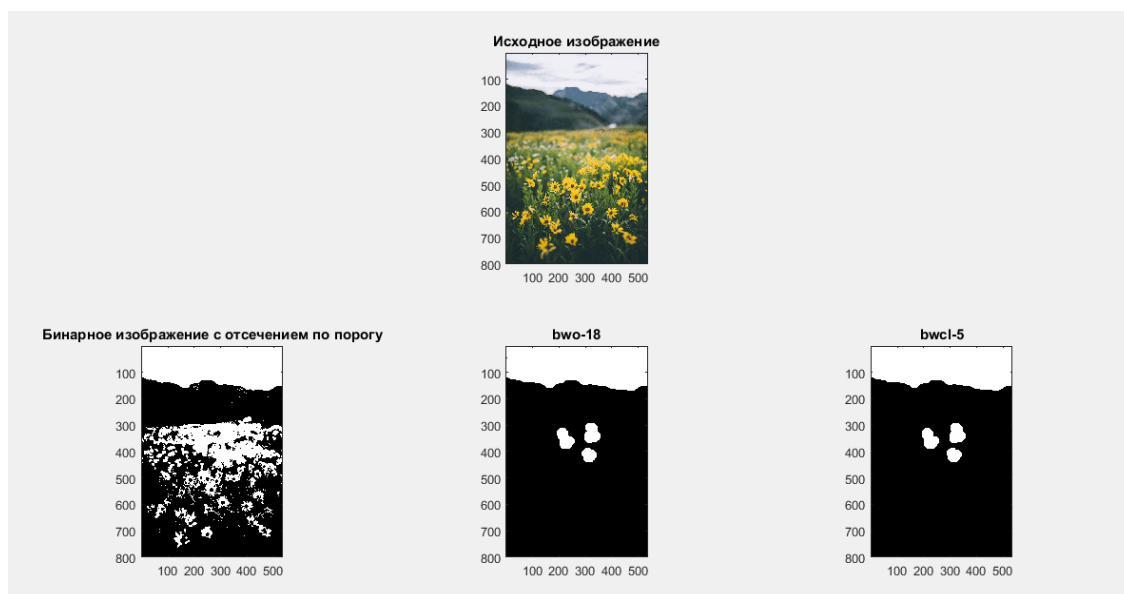


Рисунок 3.6 – Результат выполнения выделения объектов на изображении файла, используя операции замыкания и размыкания

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Как можно получить бинарное изображение?
2. Чем отличаются глобальное и локальное пороговое разделение?
3. Какие функции используются для бинаризации изображения в системе MATLAB?
4. В чем заключается алгоритм автоматического выбора порога?
5. Как определяются координаты 4-х соседних пикселей изображения?
6. Как определяются 4-х связный объект?
7. Какие морфологические операции вы знаете?
8. Дайте определение понятия эрозии.
9. Дайте определение понятия дилатации.
10. В чем заключается действие операции размыкание?
11. В чем заключается действие операции замыкание?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Томакова, Р. А. Методы и алгоритмы цифровой обработки изображений : учебное пособие для студентов всех форм обучения по направлениям подготовки 09.03.03 Прикладная информатика; 09.03.04 «Программная инженерия», 09.04.04 «Программная инженерия» (профиль «Разработка информационно-вычислительных систем») / Р. А. Томакова, Е. А. Петрик ; Юго-Зап. гос. ун-т. - Курск : Университетская книга, 2020. - 310 с. - Загл. с титул. экрана. – Текст : электронный.

3. Яне, Бернд. Цифровая обработка изображений : [учебное пособие] / пер. с англ. А. М. Измайловой. - М. : Техносфера, 2007. - 584 с. : ил. + 1 эл. опт. диск (CD-ROM). - (Мир цифровой обработки. XI. 06). - Библиогр.: с. 575-583 (221 назв.). - ISBN 978-5-94836-1 22-2 : 285.00 р. - Текст : непосредственный.

4. Красильников, Н. Н. Цифровая обработка 2D- и 3D-изображений : учебное пособие / Н. Н. Красильников. - СПб. : БХВ-Петербург, 2011. - 608 с. : ил. - ISBN 978-5-9775-07 00-4 : 372.50 р. - Текст : непосредственный.

5. Томакова, Римма Александровна. Интеллектуальные технологии сегментации и классификации биомедицинских изображений : монография / Р. А. Томакова, С. Г. Емельянов, С. А. Филист ; Юго-Западный государственный университет. - Курск : ЮЗГУ, 2012. - 222 с. - Текст : электронный.

6. Томакова, Римма Александровна . Теоретические основы и методы обработки и анализа микроскопических изображений биоматериалов : монография / Р. А. Томакова, С. А. Филист, С. Г. Емельянов ; МИНОБРНАУКИ РФ, Юго-Западный государственный университет. - Курск : ЮЗГУ, 2011. - 202 с. - Текст : электронный.

7. Методы цифровой обработки изображений : учебное пособие / А. Е. Архипов, С. В. Дегтярев, С. С. Садыков, С. Н. Серeda, В. С. Титов. - Курск : КГТУ, 2002 - Ч. 2. - 115 с. - Текст : непосредственный.

9. Технологии сетей связи. Особенности кодирования цифровых факсимильных сообщений : учебное пособие / А. И. Атакищев [и др.] ; Министерство образования Российской Федерации, Курский государственный технический университет. -

Курск :КурскГТУ, 2002. - 159 с. :ил.табл. - Имеется электрон. аналог. - ISBN 5-7681-0111-X : 62.00 р. - Текст : непосредственный.

10. Гонсалес, Р. Цифровая обработка изображений: практические советы : монография / Р. Гонсалес, Р. Вудс ; пер. П. А. Чочиа, Л. И. Рубанова. – 3-е изд., испр. и доп. – Москва :Техносфера, 2012. – 1104 с. :- (Мир цифровой обработки).– URL: <https://biblioclub.ru/index.php?page=book&id=233465> (дата обращения: 23.03.2022).

11. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. - М. :Техносфера, 2006. - 1072 с. - (Мир цифровой обработки). - ISBN 5-94836-028-8 : 394.66 р. - Текст : непосредственный.

12. Фисенко, В.Т. Компьютерная обработка и распознавание изображений / В.Т. Фисенко, Т.Ю. Фисенко. – СПб.: СПбГУ ИТМО, 2008. – 192 с.

13. Гонсалес, Р. Цифровая обработка изображений в среде Matlab / Р. Гонсалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.