



Федеральное государственное автономное образовательное учреждение высшего образования «Национальный Исследовательский Университет ИТМО»

ЛАБОРАТОРНАЯ РАБОТА №6
ПРЕДМЕТ «ТЕХНИЧЕСКОЕ ЗРЕНИЕ»
ТЕМА «МОРФОЛОГИЧЕСКИЙ АНАЛИЗ ИЗОБРАЖЕНИЙ»

Преподаватель:
Шаветов С. В.

Выполнили:
Румянцев А. А.
Чебаненко Д. А.
Овчинников П. А.

Поток: ТЕХ. ЗРЕНИЕ 2.1
Факультет: СУиР
Группа: R3241

Санкт-Петербург
2024

Содержание

1	Цель работы	2
2	Теоретические сведения	2
3	Задание 1	2
3.1	Дилатация	2
3.2	Эрозия	3
3.3	Открытие	3
3.4	Закрытие	4
3.5	Комбинации	4
3.6	Листинг для задания 1	5
4	Задание 2. Разделение объектов	5
4.1	Листинг для задания 2	7
5	Задание 3. Сегментация	8
5.1	Листинг для задания 3	9
6	Ответы на вопросы	11
6.1	Включает ли результат открытия в себя результат закрытия?	11
6.2	Какой морфологический фильтр необходимо применить, чтобы убрать у объекта выступы?	11
6.3	Каким образом с помощью морфологических операций можно найти контур объекта?	11
6.4	Что такое морфология?	11
7	Вывод	11

1 Цель работы

Освоение принципов математической морфологии в области обработки и анализа изображений.

2 Теоретические сведения

Математическая морфология в обработке изображений применяется для фильтрации шумов, сегментации объектов, выделения контуров, поиска заданного объекта на изображении, вычисления "скелета" образа и других преобразований. Далее рассмотрим базовые морфологические операции над изображением A структурным элементом B .

3 Задание 1

Выберем произвольное изображение, содержащее дефекты формы (внутренние «дырки» или внешние «выступы» объектов). Пусть это будет мухомороподобный перчаточный флаг Японии, нарисованный в Paint. Так как фон изображения белого цвета, добавим рамку вокруг картинки для видимости границ.

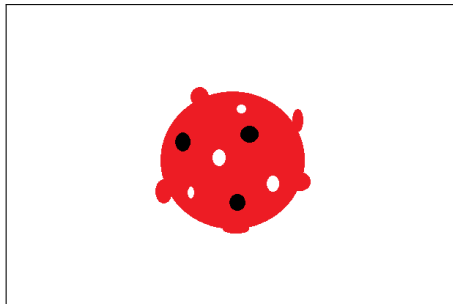


Рис. 1: Изображение для задания 1

Далее будем пробовать различные морфологические операции для устранения черных точек, белых точек и красных выступов.

3.1 Дилатация

Дилатация (расширение, наращивание): $A \oplus B$, расширяет бинарный образ A структурным элементом B . Данная операция увеличивает белые области на изображении. Применим ее к исходной картинке и посмотрим результат.

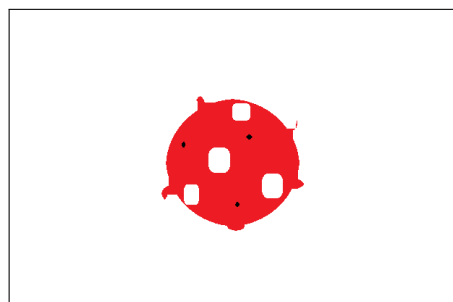


Рис. 2: Применение дилатации к исходному изображению

Видим, что белые «дырки» внутри красного круга стали больше, черные меньше. Радиус окружности уменьшился, частично срезались выступы по краям.

3.2 Эрозия

Эрозия (сжатие, сужение): $A \ominus B$, сужает бинарный образ A структурным элементом B . Эта операция уменьшает белые области на изображении. Применим ее к оригинальной картинке и сравним с ней результат.

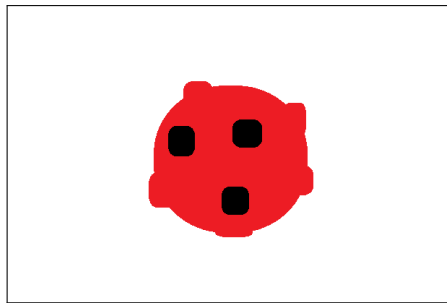


Рис. 3: Применение эрозии к исходному изображению

Можем заметить, что белые «дырки» внутри красного круга пропали, черные увеличились. Радиус круга и дефекты по его краям стали больше. Наблюдаем эффект, обратный дилатации. Результат зависит от количества итераций, то есть от того, сколько раз операция была применена к изображению. Если выбрать меньшее количество итераций в данном пункте, то белые «дырки» уменьшатся, но не пропадут. То же самое в пункте про дилатацию – если сделать больше итераций, то черные «дырки» полностью исчезнут.

3.3 Открытие

Открытие (отмыкание, размыкание, раскрытие): $(A \ominus B) \oplus B$, удаляет внешние дефекты бинарного образа A структурным элементом B . Состоит из последовательного применения эрозии и дилатации. Применим операцию к изначальному изображению.

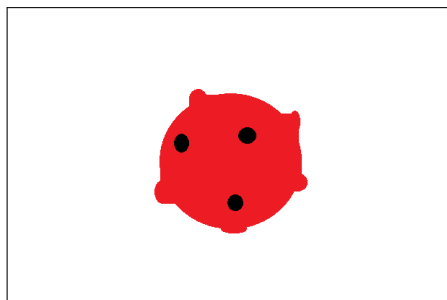


Рис. 4: Применение открытия к исходному изображению

Видим, что белые «дырки» пропали, черные остались нетронуты. Радиус красной окружности не изменился. В местах выступов круг начал сливаться с ними, что означает, что операция сгладила внешние дефекты.

3.4 Закрытие

Закрытие (замыкание): $(A \oplus B) \ominus B$, удаляет внутренние дефекты бинарного образа A структурным элементом B . Состоит из последовательного применения дилатации и эрозии. Применим данную операцию к исходному изображению и посмотрим результат.

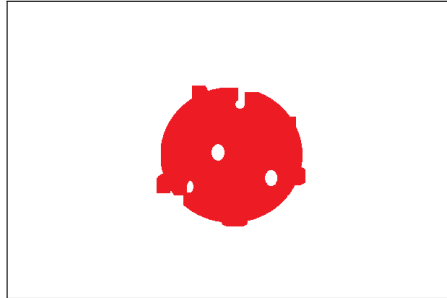


Рис. 5: Применение закрытия к исходному изображению

Черные «дырки» внутри круга исчезли, белые почти не изменились. Внешние дефекты окружности немного срезались, но вместе с некоторыми ее частями.

3.5 Комбинации

Попробуем сначала применить 4 раза эрозию, потом 9 раз дилатацию. Также применим последовательно открытие и закрытие. Посмотрим на результат.

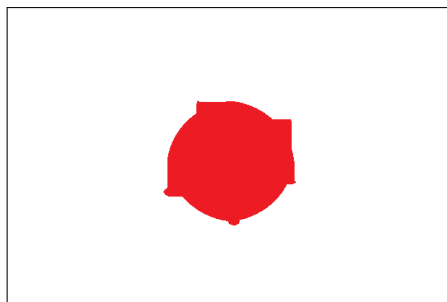


Рис. 6: Применение эрозии и дилатации к исходному изображению

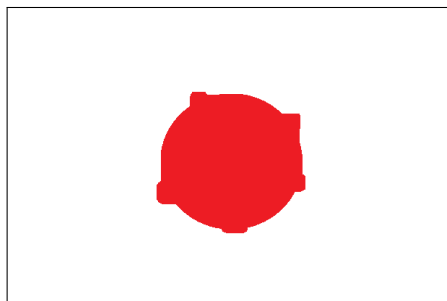


Рис. 7: Применение открытия и закрытия к исходному изображению

Как видим, оба варианта выглядят неплохо. Пропали внутренние дефекты, внешние только сглажены, так как убрать их полностью без потери большей части информации о рассматриваемом объекте (красном круге) не получается (радиус окружности становится либо сильно больше, либо сильно меньше оригинала).

3.6 Листинг для задания 1

Задание было выполнено на языке программирования Python с использованием библиотек `numpy` и `opencv-python`.

```
1  import cv2
2  import numpy as np
3
4  def dilate(I, ker_sz=(5, 5), iters=1):
5      ker = np.ones(ker_sz, np.uint8)
6      return cv2.dilate(I, ker, iterations=iters)
7
8  def erode(I, ker_sz=(5, 5), iters=1):
9      ker = np.ones(ker_sz, np.uint8)
10     return cv2.erode(I, ker, iterations=iters)
11
12     def opening(I, ker_sz=(5, 5)):
13         ker = np.ones(ker_sz, np.uint8)
14         return cv2.morphologyEx(I, cv2.MORPH_OPEN, ker)
15
16     def closing(I, ker_sz=(5, 5)):
17         ker = np.ones(ker_sz, np.uint8)
18         return cv2.morphologyEx(I, cv2.MORPH_CLOSE, ker)
19
20     if __name__ == '__main__':
21         path = 'tv_lab6'
22         src = 'source'
23         render = 'renders'
24
25         bm = cv2.imread(f'{path}/{src}/base_morf.png', cv2.IMREAD_COLOR)
26
27         dil = dilate(bm, iters=3)
28         er = erode(bm, iters=4)
29         op = opening(bm, ker_sz=(15, 15))
30         cl = closing(bm, ker_sz=(18, 18))
31         cv2.imwrite(f"{path}/{render}/dil_bm.png", dil)
32         cv2.imwrite(f"{path}/{render}/er_bm.png", er)
33         cv2.imwrite(f"{path}/{render}/op_bm.png", op)
34         cv2.imwrite(f"{path}/{render}/cl_bm.png", cl)
35
36         ans = dilate(erode(bm, iters=4), iters=9)
37         ans2 = closing(opening(bm, (15, 15)), (18, 18))
38         cv2.imwrite(f"{path}/{render}/er_then_dil_bm.png", ans)
39         cv2.imwrite(f"{path}/{render}/op_then_cl_bm.png", ans2)
```

Листинг 1: Программа для базовых морфологических операций

4 Задание 2. Разделение объектов

Морфологические операции можно использовать для задачи разделения склеившихся на изображении объектов. Она будет решена с достаточной степенью точности при помощи последовательного выполнения нескольких раз фильтра сжатия, а затем максимально возможного расширения полученного результата. Пересечение исходного изображения с обработанным позволит разделить склеенные объекты.

Выберем произвольное бинарное изображение, содержащее перекрывающиеся объекты. Пусть это будет набор различных геометрических фигур, которые есть в стандартном

Paint. Фигуры сделаем черными, а фон белым. Добавим рамку для картинки, чтобы видеть ее края.

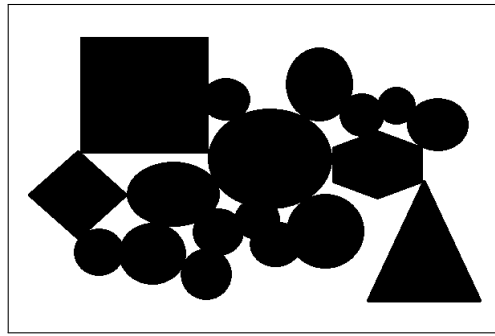


Рис. 8: Исходное изображение для задания 2

Разделим наши фигуры морфологическими операциями. Далее будут приведены полученное изображение и оно же, но с выделенными краями. Так мы сможем увидеть, насколько хорошо фигуры отделились друг от друга.

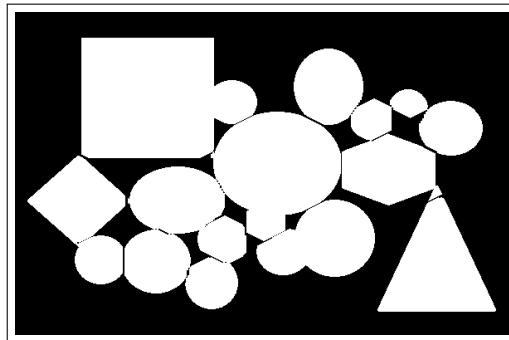


Рис. 9: Результат разделения фигур морфологическими операциями

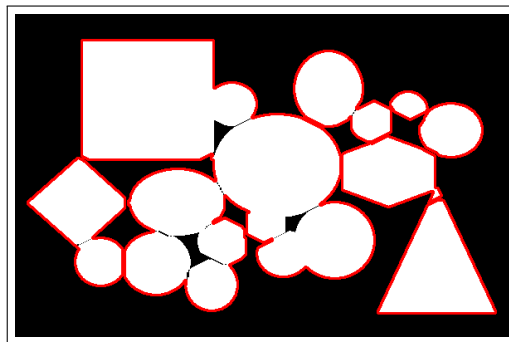


Рис. 10: Выделение контура

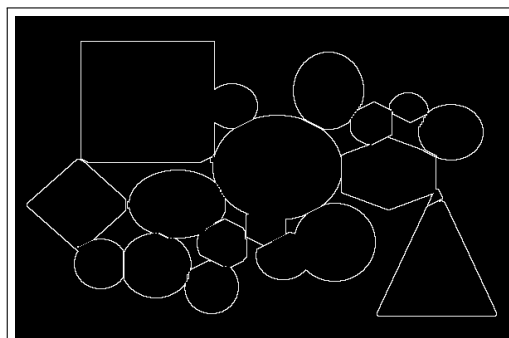


Рис. 11: Выделение контура подходом $C = (A \oplus B) - A$

Видим, что лучше всего разделяются окружности или овалы. Разделение «склеенных» объектов получилось неплохо, но некоторые фигуры деформировались или на них появились лишние линии разделения.

4.1 Листинг для задания 2

```
1  import cv2
2  import numpy as np
3
4  def separate_objs(I, iters, mellipse=(5,5)):
5      ret, Inew = cv2.threshold(I, 160, 255, cv2.THRESH_BINARY_INV)
6      B = cv2.getStructuringElement(cv2.MORPH_ELLIPSE, mellipse)
7
8      BW2 = cv2.morphologyEx(Inew,
9                             cv2.MORPH_ERODE,
10                            B,
11                            iterations=iters,
12                            borderType=cv2.BORDER_CONSTANT,
13                            borderValue=0)
14
15      T = np.zeros_like(Inew)
16      while cv2.countNonZero(BW2) < BW2.size:
17          D = cv2.dilate(BW2, B, borderType=cv2.BORDER_CONSTANT,
18                        borderValue=0)
19          C = cv2.morphologyEx(D,
20                              cv2.MORPH_CLOSE,
21                              B,
22                              borderType=cv2.BORDER_CONSTANT,
23                              borderValue=0)
24          S = C - D
25          T = cv2.bitwise_or(S, T)
26          BW2 = D
27
28      T = cv2.morphologyEx(T,
29                          cv2.MORPH_CLOSE,
30                          B,
31                          iterations=iters,
32                          borderType=cv2.BORDER_CONSTANT,
33                          borderValue=255)
34
35      Inew = cv2.bitwise_and(~T, Inew)
36      return Inew
37
38  def find_counters(Inew, col=(0, 0, 255), th=2):
39      contours, _ = cv2.findContours(Inew, cv2.RETR_EXTERNAL,
40                                   cv2.CHAIN_APPROX_SIMPLE)
41      contour_I = cv2.cvtColor(Inew, cv2.COLOR_GRAY2BGR)
42      cv2.drawContours(contour_I, contours, -1, col, th)
43      return contour_I
44
45  def outer_contour(A):
46      B = cv2.getStructuringElement(cv2.MORPH_RECT, (3, 3))
47      dilation = cv2.dilate(A, B)
48      outer_contour = dilation - A
49      return outer_contour
50
51  if __name__ == '__main__':
52      path = 'tv_lab6'
53      src = 'source'
```



```
53     render = 'renders'
54
55     binary_I = cv2.imread(f"{path}/{src}/bin.png", cv2.IMREAD_GRAYSCALE)
56
57     binary_Inew = separate_objs(binary_I, 10)
58     c_im = find_counters(binary_Inew)
59     c_im2 = inner_contour(binary_Inew)
60     cv2.imwrite(f"{path}/{render}/bin_new.png", binary_Inew)
61     cv2.imwrite(f"{path}/{render}/bin_new_c.png", c_im)
62     cv2.imwrite(f"{path}/{render}/bin_new_c2.png", c_im2)
```

Листинг 2: Программа для разделения «склеенных» объектов

5 Задание 3. Сегментация

В подходе сегментации по водоразделам изображение рассматривается как карта высот, на котором интенсивности пикселей описывают высоты относительно некоторого уровня. На такую «высотную местность» «льет дождь», образуя множество водосборных бассейнов. Постепенно вода из переполненных бассейнов переливается, и бассейны объединяются в более крупные. Места объединения бассейнов отмечаются как линии водораздела. Если «дождь» остановить рано, тогда изображение будет сегментировано на мелкие области, а если поздно – на крупные.

В таком подходе все пиксели подразделяются на три типа:

1. локальные минимумы.
2. находящиеся на склоне (с которых вода скатывается в один и тот же локальный минимум).
3. локальные максимумы (с которых вода скатывается более чем в один минимум).

При реализации данного метода необходимо определить водосборные бассейны и линии водораздела путем обработки локальных областей и вычисления их характеристик.

Алгоритм сегментации состоит из следующих шагов:

1. Вычисление функции сегментации. Как правило, для этого используется градиентное представление изображения.
2. Вычисление маркеров переднего плана на основании связности пикселей каждого объекта.
3. Вычисление маркеров фона, представляющих пиксели, не являющиеся объектами.
4. Модифицирование функции сегментации с учетом взаиморасположения маркеров переднего плана и фона.

В результате работы алгоритма будет получена маска, где пиксели одинаковых сегментов будут помечены одинаковыми метками и будут образовывать связную область.

Выберем произвольное изображение, содержащее небольшое число локальных минимумов, после чего выполним сегментацию по водоразделам. Пусть это будет картинка бананов, взятая из интернета:

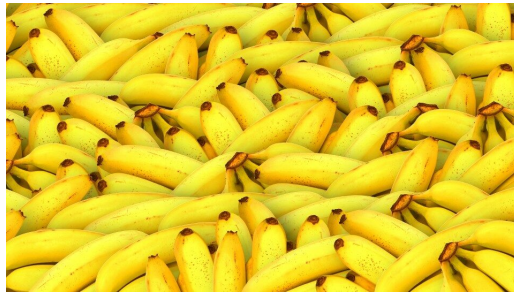


Рис. 12: Исходное изображение для задания 3

Применим сегментацию к изображению и получим следующее:

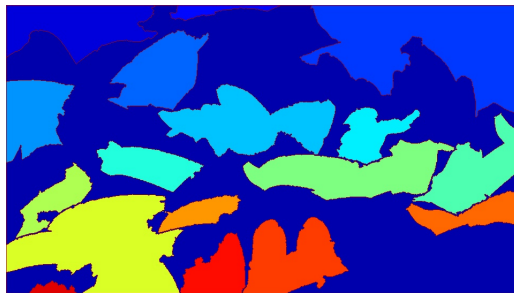


Рис. 13: Маркеры и границы, наложенные на изображение с использованием цветовой карты JET

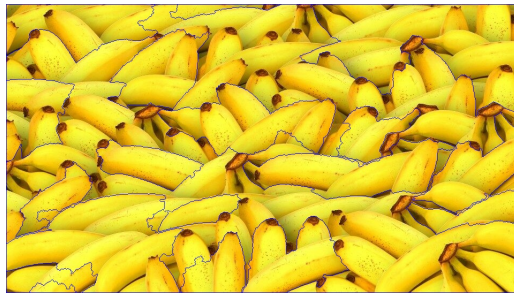


Рис. 14: Сегментация исходного изображения (маркеры и границы, наложенные на исходное изображение)

5.1 Листинг для задания 3

```
1  import cv2
2  import numpy as np
3
4  def bwareaopen(A, dim, conn=8):
5      if A.ndim > 2:
6          return None
7
8      num, labels, stats, centers = \
9          cv2.connectedComponentsWithStats(A,
10             connectivity = conn)
11
12     for i in range(num):
13         if stats[i, cv2.CC_STAT_AREA] < dim:
14             A[labels == i] = 0
15     return A
16
17
```

```
18 def segmentation(I, dim, conn=8, mellpise=(5, 5), dis=5, coeff=0.6,
19 color=(255, 0, 0)):
20     I_gray = cv2.cvtColor(I, cv2.COLOR_BGR2GRAY)
21     ret, I_bw = cv2.threshold(I_gray, 0, 255,
22                               cv2.THRESH_BINARY + cv2.THRESH_OTSU)
23     I_bw = bwareaopen(I_bw, dim, conn)
24     B = cv2.getStructuringElement (\
25         cv2.MORPH_ELLIPSE, mellpise)
26     I_bw = cv2.morphologyEx (I_bw,\
27         cv2.MORPH_CLOSE, B)
28
29     I_fg = cv2.distanceTransform(I_bw, cv2.DIST_L2, dis)
30     ret, I_fg = cv2.threshold(I_fg, coeff * I_fg.max(), 255, 0)
31     I_fg = I_fg.astype(np.uint8)
32     ret, markers = cv2.connectedComponents(I_fg)
33
34     I_bg = np.zeros_like(I_bw)
35     markers_bg = markers.copy()
36     markers_bg = cv2.watershed(I, markers_bg)
37     I_bg[markers_bg == -1] = 255
38
39     I_unk = cv2.subtract(~I_bg, I_fg)
40
41     markers = markers + 1
42     markers[I_unk == 255] = 0
43
44     markers = cv2.watershed(I, markers)
45     markers_jet = cv2.applyColorMap(
46         (markers.astype(np.float32) * 255 / (ret + 1)).astype(np.uint8),
47         cv2.COLORMAP_JET)
48     I[markers == -1] = color
49     return I, markers_jet
50
51 if __name__ == '__main__':
52     seg_I = cv2.imread(f"{path}/{src}/seg.jpg", cv2.IMREAD_COLOR)
53
54     seg_Inew, smj = segmentation(seg_I, dim=20, conn=4, mellpise=(5, 5),
55                                coeff=0.6)
56     cv2.imwrite(f"{path}/{render}/seg_new.jpg", seg_Inew)
57     cv2.imwrite(f"{path}/{render}/seg_new_mj.jpg", smj)
```

Листинг 3: Программа для сегментации изображения

6 Ответы на вопросы

- 6.1 Включает ли результат открытия в себя результат закрытия?
- 6.2 Какой морфологический фильтр необходимо применить, чтобы убрать у объекта выступы?
- 6.3 Каким образом с помощью морфологических операций можно найти контур объекта?
- 6.4 Что такое морфология?

7 Вывод