

**НИУ ИТМО**

---

---

**ФАКУЛЬТЕТ СИСТЕМ УПРАВЛЕНИЯ И РОБОТОТЕХНИКИ**

ЛАБОРАТОРНАЯ РАБОТА №6  
ПО ДИСЦИПЛИНЕ «ТЕХНИЧЕСКОЕ ЗРЕНИЕ»

Выполнили:

Гридусов Д.Д

Преподаватель:

Шаветов С.В.

Санкт-Петербург  
2024 г.

# Содержание

1	Введение	2
2	Исправить изображение морфологией	2
3	Разделение объектов	4
4	Сегментация методом управляемого водораздела	5
5	Эпилог	10

# 1 Введение

**Цель работы:** Освоение принципов математической морфологии в области обработки и анализа изображений.

## Теоритическое обоснование

Морфология - наука о форме. Методы морфологического анализа – это методы решения задач узнавания, классификации объектов, оценки параметров объектов, выделения различия в сценах по их изображениям (сигналам). Основаны на понятии формы сигнала. Математическая же морфология (долгое время эти две морфологии считались разными) основана на теории множеств. Чтобы использовать морфологию для анализа изображений вводятся морфологические фильтры - множество операторов, являющихся равносильными и увеличивающими. А также морфологические операции на бинарных изображениях (вычитание и сложение Минковского), которые позволяют используя структурный элемент ввести основные операции: дилатацию изображения и эрозию.

## 2 Исправить изображение морфологией

Возьмем два изображения: пиксельный инь и янь, и какую-то карточку. В первом случае будем пытаться убрать деление на пиксели (применим операцию открытия). А во втором случае будем решать насущную проблему - авторское право. Применив последовательно открытие и закрытие попробуем убрать надпись в правом нижнем углу.

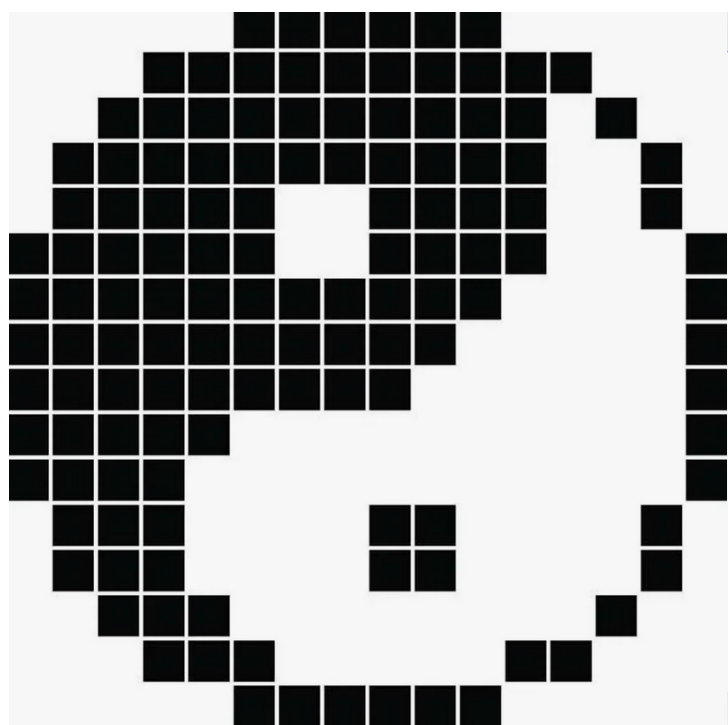


Рис. 1: Исходное изображение 1

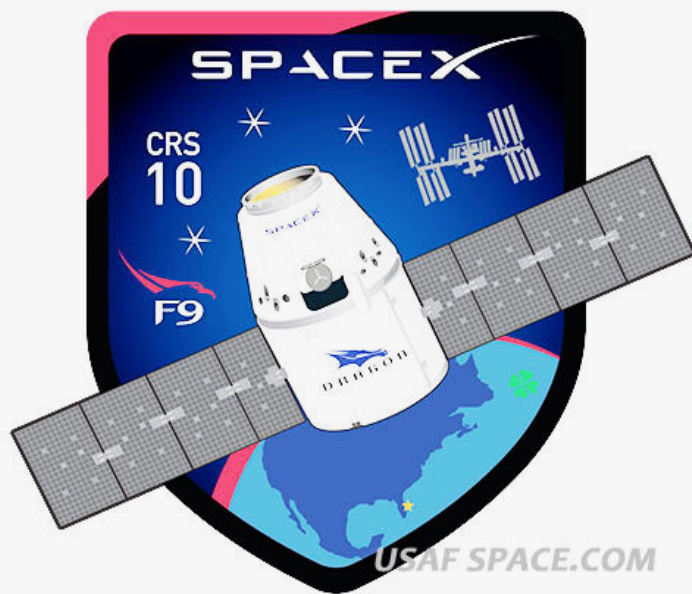


Рис. 2: Исходное изображение 2

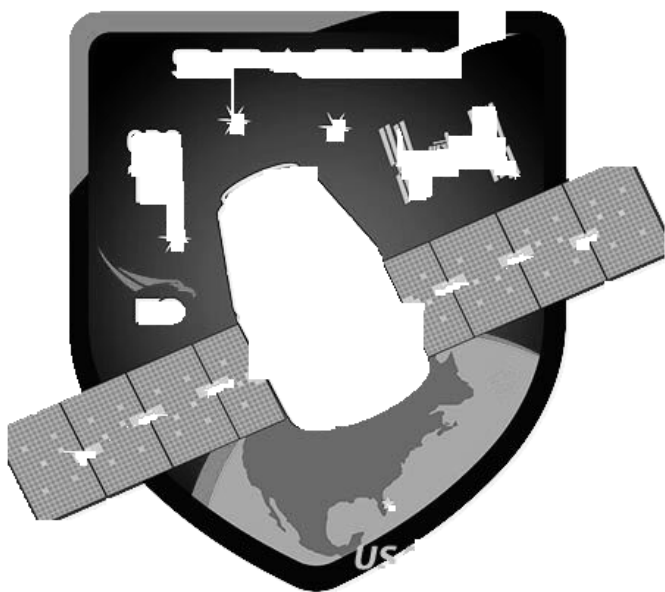


Рис. 3: Только закрытие

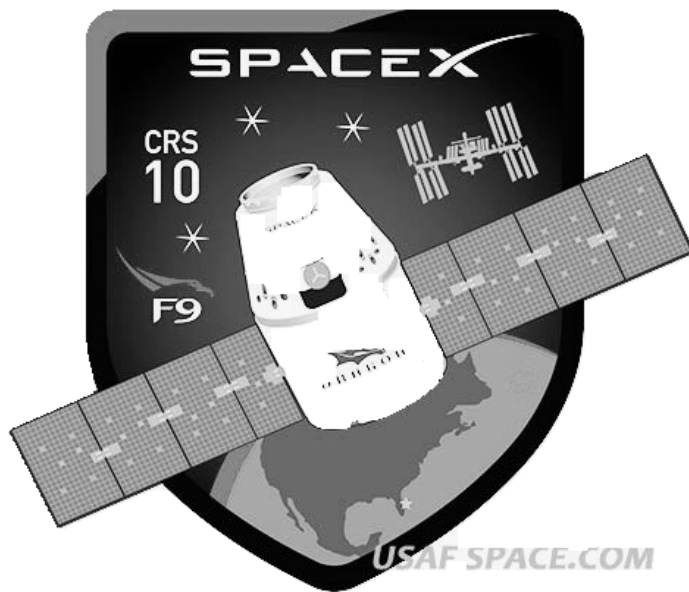


Рис. 4: Только открытие

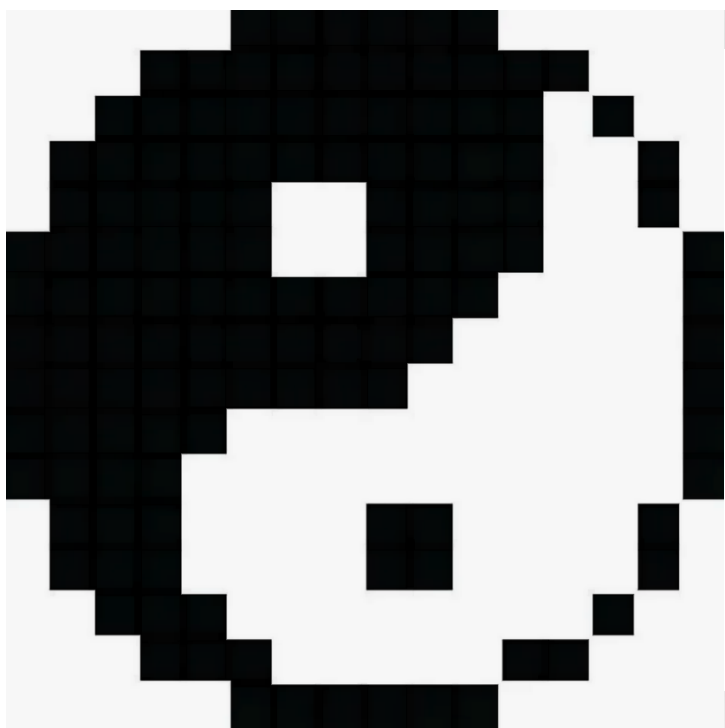


Рис. 5: Открытие первого

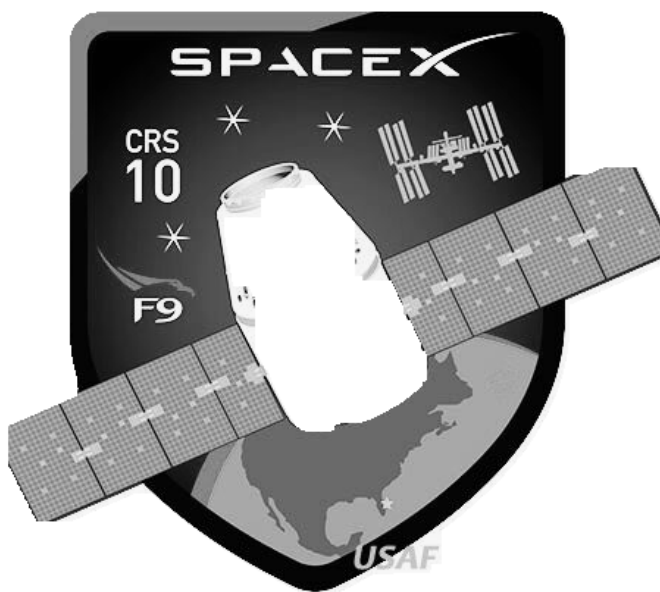


Рис. 6: Открытие + закрытие второго

После некоторого подбора параметров цели были достигнуты (ну, почти) - инь и янь теперь не настолько пиксельно выглядит (убрали внутренние дырки). А с карточкой сложнее - мы убрали большую часть надписей (осталась только та, что примыкает к изображению), но пожертвовали частью информации на карточке.

## 3 Разделение объектов

### Листинг 3.1 Разделение объектов, выделение границ

---

```
cv::Mat create_mask(cv::Mat img)
{
    cv::Mat img_gray, img_thresholded;
    cv::cvtColor(img, img_gray, cv::COLOR_BGR2GRAY);
    cv::threshold(img_gray, img_thresholded, 84, 255, cv::THRESH_BINARY_INV);
    cv::Mat B = cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(7, 7));

    cv::Mat BW2;
    cv::morphologyEx(img_thresholded, BW2, cv::MORPH_ERODE,
        B, cv::Point(-1, -1), 14, cv::BORDER_CONSTANT, cv::Scalar(0));
    cv::Mat D, C, S;
    cv::Mat T = cv::Mat::zeros(img_thresholded.size(), img_thresholded.type());
    int pix_num = img_thresholded.rows * img_thresholded.cols;
    while (cv::countNonZero(BW2) < pix_num)
    {
        cv::morphologyEx(BW2, D, cv::MORPH_DILATE,
            B, cv::Point(-1, -1), 1,
            cv::BORDER_CONSTANT, cv::Scalar(0));

        cv::morphologyEx(D, C, cv::MORPH_CLOSE,
            B, cv::Point(-1, -1), 1,
            cv::BORDER_CONSTANT, cv::Scalar(0));

        S = C - D;
        cv::bitwise_or(S, T, T);
        BW2 = D;
    }

    cv::morphologyEx(T, T, cv::MORPH_CLOSE, B, cv::Point(-1, -1), 14, cv::BORDER_CONSTANT,
        cv::Scalar(255));
    cv::bitwise_and(~T, img_thresholded, img_thresholded);

    return img_thresholded;
}

cv::Mat find_borders(cv::Mat img)
{
    cv::Mat img_gray, img_thresholded;
    cv::cvtColor(img, img_gray, cv::COLOR_BGR2GRAY);
    cv::threshold(img_gray, img_thresholded, 80, 255, cv::THRESH_BINARY_INV);

    cv::Mat img_erode;
    cv::Mat B = cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(3, 3));
    cv::erode(img_thresholded, img_erode, B, cv::Point(-1, -1), 1);
    cv::Mat border;
```

```

border = img_thresholderd - img_erode;
return border;
}

```

---

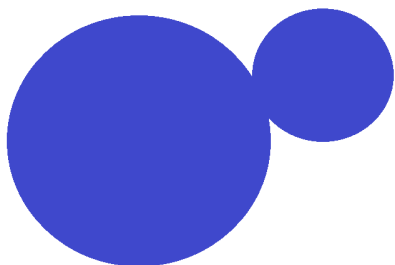


Рис. 7: Исходное изображение

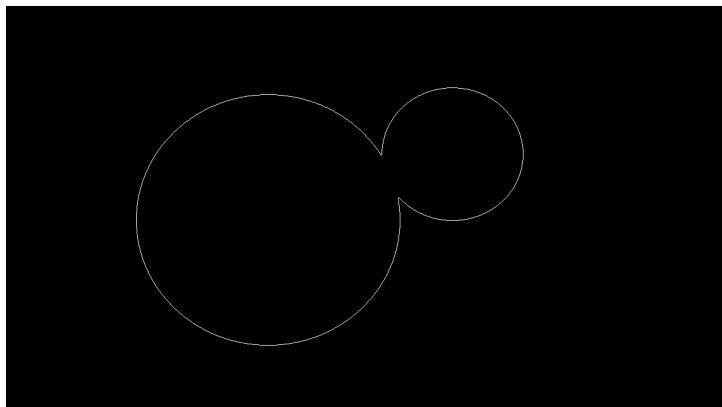


Рис. 8: Исходная граница

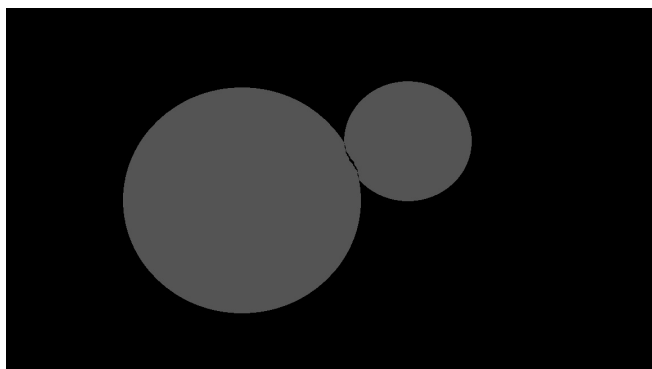


Рис. 9: Разделенные объекты

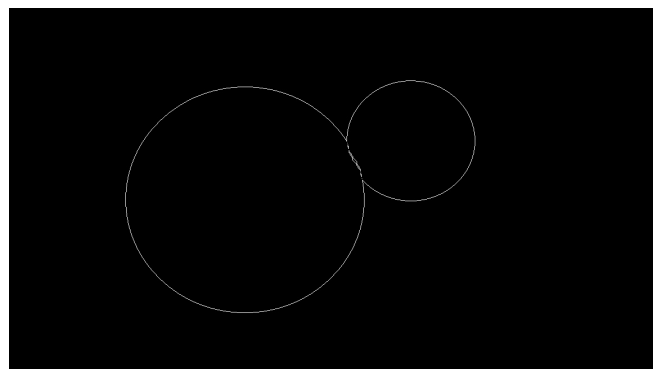


Рис. 10: Новая граница

В этом пункте мы рассмотрели применение морфологических фильтров сразу для двух целей: разделения объектов и выделения их границ. Все получилось достаточно успешно, кроме этапа подбора изображения - пришлось переключаться на Windows и вспоминать кто такой Paint (ни одной работы без того, чтобы пожаловаться - ужас).

## 4 Сегментация методом управляемого водораздела

### Листинга 4.1 Метод управляемого водораздела

```

cv::Mat segmentation_watershed(cv::Mat img)
{
    cv::Mat img_gray, img_bw;
    cv::cvtColor(img, img_gray, cv::COLOR_BGR2GRAY);
    cv::threshold(img_gray, img_bw, 0, 255, cv::THRESH_BINARY + cv::THRESH_OTSU);
    bwareaopen(img_bw, img_bw, 10, 4);

    cv::Mat B = cv::getStructuringElement(cv::MORPH_ELLIPSE, cv::Size(11, 11));
    cv::morphologyEx(img_bw, img_bw, cv::MORPH_CLOSE, B);

    cv::Mat img_fg;

```

```

double img_fg_min, img_fg_max;
cv::distanceTransform(img_bw, img_fg, cv::DIST_L2, 5);
cv::minMaxLoc(img_fg, &img_fg_min, &img_fg_max);
cv::threshold(img_fg, img_fg, 0.6 * img_fg_max, 255, 0);
img_fg.convertTo(img_fg, CV_8U, 255.0 / img_fg_max);
cv::Mat markers;
int num = cv::connectedComponents(img_fg, markers);

cv::Mat img_bg=cv::Mat::zeros(img_bw.size(), img_bw.type());
cv::Mat markers_bg=markers.clone();
cv::watershed(img, markers_bg);
img_bg.setTo(cv::Scalar(255), markers_bg==-1);

cv::Mat img_unk;
cv::bitwise_not(img_bg, img_unk);
cv::subtract(img_unk, img_fg, img_unk);

markers += 1;
markers.setTo(cv::Scalar(0), img_unk == 255);
cv::watershed(img, markers);

cv::Mat markers_jet;
markers.convertTo(markers_jet, CV_8U, 255.0/(num+1));
cv::applyColorMap(markers_jet, markers_jet, cv::COLORMAP_JET);

img.setTo(cv::Scalar(255, 255, 0), markers==-1);
return img;
}

void bwareaopen(const cv::Mat &A, cv::Mat &C, int dim, int conn)
{
    if (A.channels() != 1 && A.type() != A.type() != CV_8U && A.type() != CV_32F)
    {
        return;
    }

    cv::Mat labels, stats, centers;
    int num = cv::connectedComponentsWithStats(A, labels, stats, centers, conn);

    C = A.clone();
    std::vector<int> td;
    for (int i = 0; i < num; i++)
    {
        if (stats.at<int>(i, cv::CC_STAT_AREA) < dim)
        {
            td.push_back(i);
        }
    }
    if (td.size() > 0)
    {
        if (A.type() == CV_8U)
        {
            for (int i = 0; i < C.rows; i++)

```







Рис. 11: Оригинальное изображения

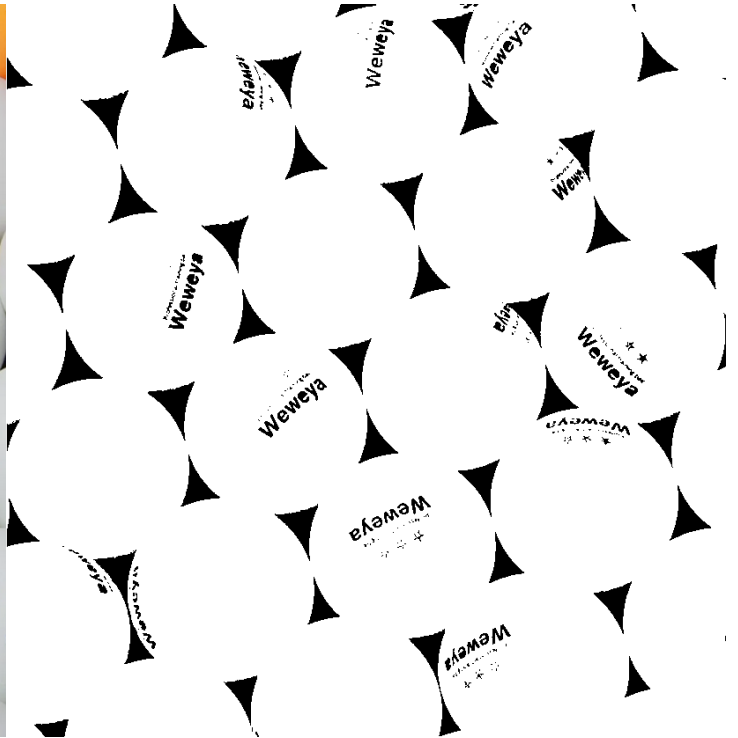


Рис. 12: Бинаризация + bwatopen

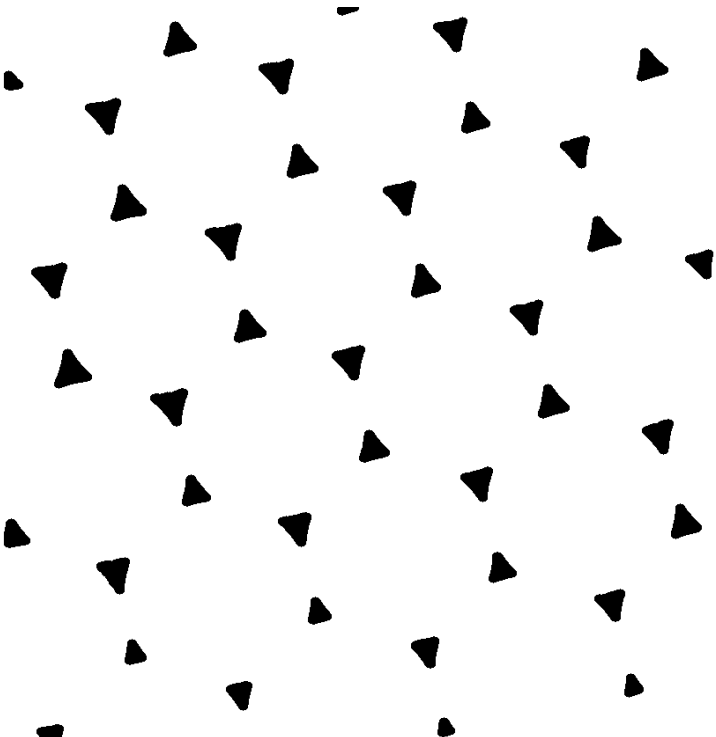


Рис. 13: MORPH\_CLOSE

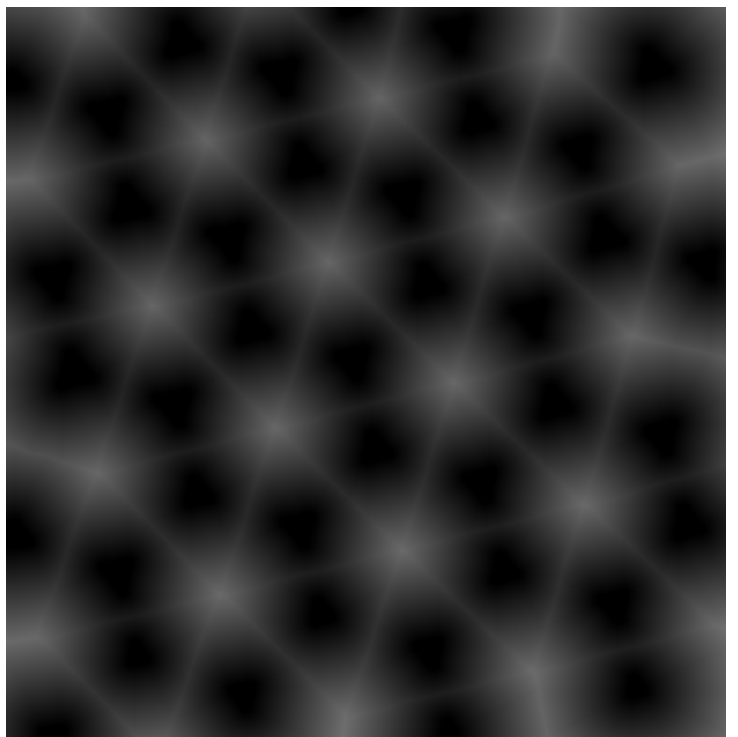


Рис. 14: Преобразование L2 метрики



Рис. 15: Маркеры переднего плана

Рис. 16: Маркеры заднего плана



Рис. 17: Сегментация

Рис. 18: Сегментированное изображение

Алгоритм работает очень даже неплохо, но важно правильно подобрать параметры (размер структурного элемента, количество повторений в методе `morphologyEx`).

## 5 Эпилог

**Вывод:** я освоил основные принципы математической морфологии в области обработки и анализа изображений. Действительно, появилось какое-то понимание, что это за методы, когда их применять. Что не может не радовать - я сдал эту работу в дедлайн (даже заранее, можно сказать). Курс был интересным и насыщенным, C++ - one love, спасибо!

### Ответы на вопросы

1. Судя по экспериментальным данным и примеру из лекции - нет, не содержит.
2. Необходимо использовать закрытие (дилатация + эрозия)
3. Нужно из объекта убрать его эрозию
4. Наука о форме, применяемая во многих областях. В анализе изображений позволяет решать, например, задачи узнавания и классификации.