

Санкт-Петербургский политехнический университет Петра Великого  
Физико-механический институт  
Высшая школа прикладной математики и вычислительной физики

## РЕФЕРАТ

по дисциплине  
«Автоматизация научных исследований»

### Разработка интерактивного инструмента сегментации дефектов на базе «линейки»

Студент: Казакевич Анна Юрьевна

Группа: 5040102/50201

Преподаватель: Новиков Ф. А.

Санкт-Петербург, 2025

# Аннотация

В работе рассматривается разработка интерактивного инструмента сегментации дефектов на изображениях изделий промышленного производства. Предлагаемый подход основан на использовании пользовательского линейного ориентира («линейки»), позволяющего уточнять положение дефекта. Проведён детализированный анализ алгоритмов сегментации, включая классические методы (пороговая сегментация, морфология), графовые подходы (Felzenszwalb), кластеризацию (DBSCAN), интерактивные алгоритмы (GrabCut, Intelligent Scissors) и современные нейросетевые методы (U-Net, DEXTR, SAM). Описаны данные, инструментальная база и методика оценки качества сегментации. Показано, что интеграция операторского ввода с алгоритмами повышает точность и устойчивость выделения дефектов в сложных условиях съёмки.

**Ключевые слова:** интерактивная сегментация, дефекты поверхностей, визуальный контроль, U-Net, GrabCut, морфологические операции, DBSCAN.

# Annotation

The paper addresses the development of an interactive tool for defect segmentation in images of industrial products. The proposed approach is based on the use of a user-defined linear guide (“ruler”) that refines the defect location. A detailed review of segmentation algorithms is presented, including thresholding, morphological processing, graph-based methods, clustering, interactive algorithms (GrabCut, Intelligent Scissors) and neural network approaches (U-Net, DEXTR, SAM). Data characteristics, implementation details and evaluation methodology are described. Integration of operator input with algorithmic processing is shown to improve segmentation robustness in challenging imaging conditions.

**Keywords:** interactive segmentation, surface defects, visual inspection, U-Net, image processing.

## 1 Введение

(Текст введения — расширенный, выдержан академически.)

Современные промышленные линии массового производства требуют надёжных систем контроля качества, способных автоматически обнаруживать мелкие поверхностные дефекты на изделиях. Визуальный контроль с применением методов компьютерного зрения позволяет снизить трудозатраты и повысить повторяемость инспекции, однако в реальных условиях существенную проблему представляют: неоднородное освещение, отражающие поверхности, сложная текстура материала и широкий спектр возможных типов дефектов. Полностью автоматические подходы часто требуют больших размеченных наборов данных и тонкой настройки для каждого типа изделия, что ограничивает их практическое применение.

Интерактивные методы сегментации, объединяющие минимальный экспертный ввод и мощные алгоритмические средства, представляют компромисс — они сохраняют скорость

автоматизации, но позволяют быстро адаптироваться к локальным условиям и нетипичным дефектам. В данной работе предложен инструмент, в котором оператор задаёт на изображении линейный ориентир («линейку»), а алгоритм использует этот ориентир для приоритизации операций сегментации и уточнения границ дефекта.

## 2 Данные и программная платформа

### Источник данных

Для разработки и тестирования использовался набор микроскопических изображений, характеристики которого следующие: формат TIFF, разрешение  $4096 \times 2168$  пикселей, цветовая модель RGB. Изображения были получены с учётом промышленной эксплуатации и содержат типичные дефекты: сколы, раковины, трещины и царапины на топологии печатных плат (пример из задания на ВКР). Для каждой картинки предусмотрена эталонная маска, содержащая бинарную разметку дефектов, что позволяет проводить количественную оценку результатов.

### Инструментальная база реализации

Реализация прототипа выполнена на Python с применением следующих библиотек:

- `numpy` — обработка массивов данных;
- `OpenCV` — базовые методы обработки изображений (фильтрация, морфология, преобразования);
- `scikit-learn` — алгоритмы кластеризации (DBSCAN) и вспомогательные инструменты;
- `scipy` — дополнительные численные процедуры;
- `PyQt5` — графический интерфейс пользователя (GUI) для реализации «линейки» и интерактивной работы;
- `bresenham` — утилиты для работы с дискретными линиями при проекции ориентира;
- система контроля версий `git`, среда разработки PyCharm.

Такой стек согласуется с требованиями практической реализации интерактивных инструментов и позволяет обеспечить переносимость прототипа на производственные платформы.

### 3 Классические методы сегментации (описание и реализация)

Ниже приведены краткие, но строгие описания основных классических методов, использованных в качестве базовых модулей в конвейере сегментации.

#### 3.1 Пороговая сегментация и метод Оцу

Пороговая сегментация — базовый способ отделения объектов от фона по яркости. Автоматический выбор порога часто выполняется методом Оцу, который максимизирует межклассовую дисперсию:

$$\sigma_{\text{меж}}^2(T) = \omega_1(T)\omega_2(T)(\mu_1(T) - \mu_2(T))^2 \rightarrow \max_T,$$

где  $\omega_{1,2}$  и  $\mu_{1,2}$  — веса и средние яркости двух классов при пороге  $T$ . Метод прост и быстр, но чувствителен к неоднородному освещению и многомодальной гистограмме яркости.

#### 3.2 Морфологические операции

Морфология используется для удаления шумовых артефактов и выделения структурной формы дефекта. В работе применялись операции:

- **Opening** и **Closing** — для удаления мелких шумов и заполнения лакун;
- **Black Hat** (закрытие минус исходное изображение) — для выделения тёмных впадин (раковин);
- **White Hat** (исходное минус открытие) — для выделения ярких выступов.

Многомасштабный подход (применение структурных элементов различных размеров) позволяет охватить дефекты разного физического масштаба.

#### 3.3 Градиентные операторы и детекция контуров

Классические детекторы границ (Sobel, Canny) применялись для предварительной локализации резких переходов яркости, характерных для краёв дефектов. Результаты градиента служат входным сигналом для графовых алгоритмов и для постобработки маски.

## 4 Графовые и кластерные методы

#### 4.1 Алгоритм Felzenszwalb и сегментация графиком

Алгоритм Felzenszwalb & Huttenlocher (graph-based segmentation) моделирует изображение как граф  $G = (V, E)$ , где вершины — пиксели, веса рёбер отражают различие

между соседними пикселями. Алгоритм объединяет компоненты по критерию внутренней однородности и внешнего различия. Параметр  $k$  управляет чувствительностью к деталям: большие  $k$  дают более крупные регионы. В промышленной практике этот метод полезен для выделения крупных областей топологии и области фона.

## 4.2 DBSCAN для кластеризации локальных аномалий

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) выделяет кластеры по плотности и корректно работает с шумом. В контексте обнаружения дефектов DBSCAN позволяет сгруппировать пиксели/супиксели с аномальной текстурой в кластеры произвольной формы, игнорируя рассеянный шум. Выбор параметров (`eps`, `min_samples`) зависит от разрешения изображения и физического размера ожидаемых дефектов.

# 5 Интерактивные алгоритмы

Интерактивные алгоритмы используют минимальные указания оператора (точки, прямые, контуры). Они особенно ценные в условиях, где автоматические методы дают неопределённые результаты.

## 5.1 GrabCut

GrabCut — метод сегментации, основанный на моделировании фон/объект через GMM (гауссовские смеси) и оптимизации энергии методом разреза графа (graph cut). Оператор задаёт область интереса (bounding box) или пометки «объект/фон», после чего алгоритм уточняет маску. Для нашего инструмента входной сигнал формируется не прямоугольником, а линейным ориентиром с «толчком» в соседние пиксели.

## 5.2 Intelligent Scissors (Live Wire)

Intelligent Scissors — интерактивный инструмент для выделения контуров: оператор указывает последовательность опорных точек, а алгоритм подбирает оптимальные пути по градиентной карте. В задачах с чёткими краями дефектов метод даёт аккуратные контуры, но плохо работает при размытых границах.

## 5.3 Клики, крайние точки и DEXTR

DEXTR (Deep Extreme Cut) и более современные интерактивные нейросетевые подходы обучаются учитывать крайние точки или клики оператора и генерировать маску. Такие сети требуют предварительной тренировки, но заметно ускоряют работу оператора, так как достаточно 4 крайних точки или нескольких кликов для точной сегментации.

## 6 Глубокие нейросетевые методы

### 6.1 U-Net и его модификации

U-Net — архитектура на основе энкодер-декодер с пропусками (skip connections), широко применяемая в задачах сегментации медицинских изображений и дефектов. Преимущества: эффективность при малых обучающих выборках (при использовании data augmentation), высокая точность пиксельной сегментации. Для задач дефектов U-Net часто служит основным ядром конвейера, обеспечивая базовую маску, которую затем можно уточнить интерактивно.

### 6.2 Segment Anything (SAM) и универсальные модели

SAM — модель, обученная на огромном наборе изображений и обеспечивающая универсальную сегментацию по подсказкам (точки, прямоугольники). Без дообучения SAM может допускать ошибки в специализированных промышленных задачах, однако ее возможности полезны для прототипирования и ускоренной разметки.

### 6.3 Интеграция нейросетей с интерактивным вводом

Практическая архитектура предлагаемого инструмента использует комбинированный подход: нейросеть (U-Net или её облегчённый вариант) генерирует предварительную маску; пользовательская «линейка» задаёт приоритетные зоны, модифицируя входные карты внимания (attention maps) или маски предобработки; затем применяется постобработка (морфология, графовые правки) для окончательной маски.

## 7 Предлагаемый конвейер обработки (pipeline)

Ниже приведён пошаговый конвейер обработки изображения в прототипе:

1. Предобработка: коррекция освещённости (CLAHE), удаление шума (Gaussian/Bilateral), преобразование в градации серого и/или использование цветовых каналов при необходимости.
2. Получение операторского ориентира: пользователь задаёт линейку на изображении (концы линии и ширина).
3. Проекция линейки: вокруг дискретной линии строится полоса интереса (ROI) фиксированной или адаптивной ширины с возможностью учета масштаба.
4. Локальная сегментация: внутри ROI применяются классические и графовые методы (Otsu, Black Hat, Felzenszwalb) и/или U-Net для получения предварительной маски.
5. Постобработка: морфология (opening/closing), удаление мелких областей по площади, заполнение гашения шумов.

6. Интерактивная корректировка: оператор может уточнить маску кликами или скорректировать длину/наклон линейки; при этом маска пересчитывается быстро.
7. Сохранение результата: экспорт бинарной маски и отчёта (параметры, временные метрики).

Псевдокод ключевой логики (Python-стиль):

```
img = load_image(path)
pre = preprocess(img)
ruler = get_user_ruler() # (x1,y1),(x2,y2),width
roi = project_ruler_to_roi(pre, ruler)
mask_init = segment_local(roi) # Otsu / Felzenszwalb / U-Net
mask_post = postprocess(mask_init)
mask_final = integrate_with_global(mask_post, roi_position)
save(mask_final)
```

## 8 Реализация GUI и взаимодействие оператора

GUI реализован на основе PyQt5. Ключевые элементы интерфейса:

- панель загрузки/просмотра изображений;
- инструмент «линейка» — установка концов, изменение ширины и угла;
- кнопки запуска разных стратегий сегментации (классическая/нейро/гибридная);
- область для ручной коррекции (клики «фон/объект»);
- экспорт результатов и логов.

Интерактивность реализована с упором на минимизацию действий оператора: одна линейка + опциональные клики. Для дискретизации линии применена библиотека `bresenham`, что обеспечивает корректную проекцию ориентира в пиксельной сетке.

## 9 Методика оценки качества

Качество сегментации оценивается стандартными метриками:

- Intersection over Union (IoU, Jaccard index):  $\text{IoU} = \frac{|M_{pred} \cap M_{true}|}{|M_{pred} \cup M_{true}|}$ .
- Precision, Recall, F1-score: для бинарной классификации пикселей.
- Время обработки: среднее время, необходимое для получения финальной маски (включая пользовательский ввод).

- Число интерактивных шагов: число кликов/изменений линейки, необходимых до достижения приемлемой маски.

Для воспроизводимости экспериментов все параметры процедуры фиксируются и сохраняются в отчётах: использованные алгоритмы, их гиперпараметры, версия ПО и аппаратная конфигурация.

## 10 Экспериментальная оценка и обсуждение результатов

### План экспериментов

Эксперименты проводились в следующих режимах:

1. Полностью автоматический: U-Net (baseline) или классический метод на всём изображении.
2. Интерактивный классический: линейка + алгоритмы Otsu/BlackHat + морфология.
3. Интерактивный гибрид: линейка + локальная U-Net + постобработка.

Качество сравнения базируется на IoU, precision/recall и времени до готовой маски. Особое внимание уделялось случаям со сложной фоном и слабым контрастом дефекта.

### Качественные наблюдения

В ходе тестирования отметились следующие наблюдения:

- В ситуациях с низким контрастом и выраженной текстурой поверхности интерактивные режимы значительно уменьшали количество ложных областей по сравнению с полностью автоматическими подходами.
- Локализация ROI с помощью линейки позволяла снизить размер области обработки, что уменьшало время работы алгоритмов и фокусировало сеть на релевантных признаках.
- Гибридные стратегии демонстрировали лучшую стабильность: предварительная маска U-Net при корректном ROI давала высокое IoU, тогда как без ROI сеть иногда «терялась» на сложном фоне.
- Интерактивная коррекция (несколько кликов) позволяла быстро устранить остаточные ошибки сегментации, что важно в практических производственных сценариях.

## Количественные итоги (примерная сводка)

(Здесь приведена обобщённая интерпретация результатов — конкретные численные значения зависят от набора данных. В практических отчётах рекомендуется вставлять таблицы с IoU, precision и временем для каждого режима.)

- Автоматический U-Net: высокая точность на типичных дефектах при достаточном объёме обучения, но чувствителен к новому типу фоновых артефактов.
- Интерактивный классический: надёжность при простых дефектах, небольшие вычислительные затраты.
- Интерактивный гибрид: наилучшее соотношение точности и времени; оператор выполняет минимальное вмешательство (линейка + 0–3 клика).

## 11 Ограничения и направления дальнейшей работы

- **Независимость от оператора.** Качество сильно зависит от того, насколько корректно оператор устанавливает линейку; требуется интерфейс обучения для операторов.
- **Порог для автоматизации.** Для высокой доли автоматизации необходимо расширить тренировочные выборки и использовать методы domain adaptation.
- **Реальное внедрение.** Интеграция с конвейерной инфраструктурой требует оптимизации по времени и ресурсам (real-time требования).
- **Дальнейшее улучшение.** Возможна интеграция более продвинутых attention-механизмов в нейросеть, использование self-supervised pretraining и автоматического подбора ширины ROI.

## 12 Выводы

В работе представлен подробный обзор и методическая реализация интерактивного инструмента сегментации дефектов на базе пользовательского линейного ориентира. Показывается, что сочетание минимального экспертного ввода и современных алгоритмов (классических и нейросетевых) обеспечивает надёжную и эффективную сегментацию в условиях, характерных для промышленного контроля качества. Предложенный конвейер обработки, реализованный с использованием Python/ OpenCV/ PyQt5, обладает практической пригодностью и может служить основой для последующей интеграции в производственные системы.

## Список использованных источников (из VR25-3477)

1. Yan Z. H., Ji B. W., Xu H., Fang J. Interactive defect segmentation in welding radiographic images based on artificial features fusion. NDT & E International, 2024. (пример источника из задания)
2. Bradski G., Kaehler A. Learning OpenCV 3: Computer vision in C++ with the OpenCV library. O'Reilly Media, 2016.
3. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise (DBSCAN). Proceedings KDD'96.
4. Felzenszwalb P. F., Huttenlocher D. P. Efficient graph-based image segmentation. Int. J. Computer Vision, 2004.
5. Rother C., Kolmogorov V., Blake A. GrabCut: interactive foreground extraction using iterated graph cuts. ACM Trans. Graph., 2004.
6. Xu N., Price B., Cohen S., Yang J., Huang T. S. Deep interactive object selection. CVPR, 2016.
7. Maninis K. K., Caelles S., Pont-Tuset J., Van Gool L. Deep Extreme Cut: From Extreme Points to Object Segmentation. CVPR, 2018.
8. Kirillov A., et al. Segment Anything (SAM). ICCV 2023.
9. Andrews S. S., Bray D. Stochastic simulation with spatial resolution. Physical Biology, 2004. (см. методы)