Оглавление

[Введение 5](#_Toc40106469)

[1. Анализ технического задания 7](#_Toc40106470)

[1.1. Обзор предметной области 7](#_Toc40106471)

[1.2. Исходные данные к проекту 9](#_Toc40106472)

[1.3. Обзор аналогичных алгоритмов 9](#_Toc40106473)

[1.4. Требования к разрабатываемому алгоритму 10](#_Toc40106474)

[2. Разработка математической модели и алгоритмов 12](#_Toc40106475)

[3. Апробация разработанных моделей и алгоритмов 20](#_Toc40106476)

[Заключение 25](#_Toc40106477)

[Список использованной литературы 26](#_Toc40106478)

[Приложение А. Исходные данные к проекту 27](#_Toc40106479)

## Введение

В данной пояснительной записке представлен процесс разработки алгоритма локализации матричной маркировки на цифровых изображениях черных металлов с использованием метода Отцу. В настоящее время, в связи с необходимостью развития промышленности, требуется введение новых наукоемких технологий по контролю движения продукции. Реализация такого контроля возможна с помощью внедрения систем автоматической идентификации и распознавания. Для контроля движения продукции используются маркировки промышленных изделий. На новейших промышленных предприятиях контроль движения продукции осуществляется специализированными средствами - системами автоматической идентификации маркировки (САИМ). Основная сложность при использовании САИМ заключается в задаче локализации промышленной продукции, которую усложняют наличие неравномерной освещенности и резкости (засвечивание изображения искусственным и солнечным светом, неверная настройка фокуса), присутствие шумов и размытия, прочих лишних объектов. Автоматизация этой сферы деятельности является мало проработанной и, чаще всего, выполняется вручную. Этим обусловлена актуальность данной работы.

Цель данного курсового проекта – разработать алгоритм на основе алгоритма Отцу, который позволял бы локализовать матричную маркировку на цифровом изображении.

Задачи данного курсового проекта:

a) провести анализ предметной области, проанализировать аналогичные алгоритмы (используя библиотеку AForge), предъявить требования к разрабатываемому алгоритму локализации;

b) проанализировать алгоритм Отцу с математической точки зрения, описать его и составить его блок-схему;

c) разработать на основе алгоритма Отцу алгоритм локализации матричной маркировки на цифровом изображении;

d) провести экспериментальные исследования, в которых сравнить работу алгоритмов из библиотеки AForge с разработанным алгоритмом.

Таким образом, общая структура курсового проекта состоит из трех глав и выглядит следующим образом:

a) в первой главе производится анализ технического задания, а именно:

- обозревается предметная область;

- определяются исходные данные к проекту;

- кратко обозреваются некоторые аналогичные алгоритмы;

- выставляются требования к разрабатываемому алгоритму;

b) во второй главе привозится математическое описание алгоритма Отцу и дополнения к нему, а также приводится блок-схема работы алгоритма с подробным описанием каждого шага и их блок-схемами;

c) в третьей главе производятся экспериментальные исследования: сравнивается работа алгоритма Отцу, представленного библиотекой AForge, и собственной разработанной программой, а также обработка исходных изображений (исходных данных к проекту).

## Анализ технического задания

## 1.1. Обзор предметной области

Маркировка красками или специальными чернилами - один из старейших способов маркировки. Наибольшее распространение сегодня маркировка красками имеет в легкой, электронной и пищевой промышленности, где используется оборудование типа IMAJE и DOMINO. Эти маркеры оборудованы специальными головками с соплами, с помощью которых возможно нанесение любого вида маркировки. Контрастность изображения обеспечивается различным количеством точек на единицу площади изображения. Новейшее оборудование для маркировки красками позволяет создавать разноцветные обозначения очень высокого качества.

Однако клеймение горячего металла красками невозможно: лучшие образцы лакокрасочной продукции выдерживают температуру не более 1000ºС. Плюс к этому в условиях металлургических предприятий и горячих цехов трудно обеспечить необходимую чистоту и приемлемый температурный режим, требуемые для безотказной работы краскоструйных сопел.

Затрудняет использование краскоструйных маркеров и характер поверхности многих металлоизделий. Окалина, пыль, неровности, влага и т.п. делают маркировку красками фрагментарной и недолговечной.

Все перечисленные факторы делают невозможным или экономически невыгодным использование краскоструйных маркеров в горячих цехах и при машинах непрерывного литья заготовок.

Однако маркировка в металлургии красками применяется - для обозначения холодных (остывших) металлоизделий. Единственное требование для использования маркировки красками - чистота и высокая адгезивность поверхности [6].

Краски используют для маркировки труб горячекатаных (на выходе из прокатного стана, когда температура изделия становится приемлемой), маркировки слябов и блюмов и т.п.

Оборудование для маркировки краской состоит из следующих узлов:

- головка с 7 или 9 соплами;

- манипулятор для перемещения головки с соплами;

- специальная панель для подготовки краски;

- пульт управления;

- терминал для оператора, с помощью которого можно вручную задавать параметры маркировки;

- электронная система управления (программное обеспечение).

Краскосопла имеют увеличенный диаметр отверстий, что препятствует их засорению, и приспособлены для операции самоочищения. Минимальная высота наносимой маркировки - 19 миллиметров, что может служить ограничением в использовании такого оборудования для некоторых видов металлоизделий.

С помощью оборудования для маркировки краской можно наносить не только буквенно-цифровые коды, но и штрих-коды. Однако, на данном оборудовании можно нанести только бинарные коды (содержащие минимум информации о металлоизделии), для считывания которых нужна специальная аппаратура (промышленные сканеры) [1].

Оборудование для нанесения маркировки краской имеет высокую степень надежности. Однако, при использовании маркировки в металлургии с помощью красок и специальных чернил необходимо учитывать, что для полного высыхания краски требуется определенное время, при этом чем горячее металлоизделие, тем больший срок требуется для высыхания краски. При высокой температуре маркируемых металлоизделий также повышаются эксплуатационные расходы на краскоструйное оборудование: увеличивается расход красок, требуется больше времени для технического обслуживания агрегата и т.п.

Основные характеристики:

- постоянные метки для идентификации;

- высокая степень разборчивости символов для обычного или автоматического считывания в условиях неповреждения метки;

- разборчивая маркировка на грубых и неровных поверхностях;

- высокая скорость маркировки.

Основное оборудование, необходимое для нанесения маркировки – это многопистолетные установки нанесения краски или чернил [6].

## 1.2. Исходные данные к проекту

В качестве входных данных были определены следующие:

a) набор смоделированных исходных изображений для проверки правильности работы разрабатываемого алгоритма (см. Приложение А);

b) среда программирования Visual Studio для программной реализации алгоритма и проведения экспериментальных исследований работы алгоритма;

c) язык программирования C#, так как данный язык позволяет реализовать возможность создания графического интерфейса, что необходимо для проверки корректности работы алгоритма;

d) библиотека AForge - это библиотека с открытым исходным кодом, разработанная на языке C#, которая используется разработчиками и исследователями при решении задач, связанных с компьютерным зрением.

Выходными данными в курсовом проекте будут являться локализованные матричные маркировки на изображениях (их координаты).

## 1.3. Обзор аналогичных алгоритмов

Исходя из технического задания, для работы над созданием алгоритма локализации маркировки на изображении был выбран алгоритм Отцу, чьими преимуществами являются простота реализации, хорошая адаптация метода к различного рода изображениям за счет того, что он позволяет выбрать оптимальный порог яркости пикселей, быстрое время выполнения и отсутствие параметров [8].

Помимо Отцу, существуют также и другие методы локализации объектов на изображении, например, примитивы Хаара. Этот метод очень быстр в реализации, интуитивен и очень известен. Но этот метод имеет множество недостатков:

- неустойчивость при смене освещения;

- неустойчивость при изменении масштаба или повороте изображения;

- неустойчивость, если часть изображения — изменяющийся фон;

- низкая скорость работы.

Из-за наличия этих недостатков при работе с данным алгоритмом приходится нейтрализовать освещение нормировкой или переходом к бинаризации области, не изменять масштаб изображения и не поворачивать его, следить за равномерностью освещения фона, оптимизировать скорость поиска с большим шагом или при маленьком разрешении [5].

В ситуациях, когда результатов корреляции недостаточно — переходят к более сложным методам, таким как сравнению карт особых точек (SURF), границ, или непосредственному выделению объектов. Но эти алгоритмы в большинстве случаев достаточно медленные, их сложно написать с нуля, а также они имеют ограничения на структуру изображения.

В качестве еще одного примера можно привести каскадный алгоритм Виолы-Джонса - это метод обнаружения объектов на изображениях, основанный на признаках Хаара. Основные его особенности - высокая скорость работы и низкая частота ложных срабатываний. Изначально алгоритм был разработан для обнаружения лиц на изображениях, но его можно натренировать на обнаружение других объектов. Для своей работы он использует разбиения изображения на области, оценку яркости в этих областях и отсечения областей, где классифицируемый объект однозначно не находится [2].

## 1.4. Требования к разрабатываемому алгоритму

Требования к разрабатываемому алгоритму следующие:

1. разрабатываемый алгоритм должен оперировать со всеми входными данными и получать в результате координаты локализованной маркировки на изображении;
2. алгоритм должен состоять из отдельных элементарных шагов, работающих системно;
3. последовательность шагов алгоритма должна быть детерминирована, т.е. после каждого шага должно указываться, какой шаг следует выполнять дальше, либо указываться, когда следует работу алгоритма считать законченной;
4. алгоритм должен обладать результативностью, т.е. останавливаться после конечного числа шагов (зависящего от исходных данных) с выдачей результата;
5. в отношении времени выполнения алгоритма – он должен обладать высокой скоростью выполнения операций, что алгоритм Отцу в полной мере позволяет сделать.

## 2. Разработка математической модели и алгоритмов

Метод Отцу — это алгоритм вычисления порога бинаризации для полутонового изображения, используемый в области компьютерного распознавания образов и обработки изображений для получения чёрно-белых изображений. Процесс бинаризации представляет собой перевод цветного изображения или изображения в градациях серого в двухцветное черно-белое. Основной параметр данного преобразования – порог *t*, со значением которого затем сравнивается яркость всех. После сравнения, пикселю присваивается одно из двух возможных значений: 0 - «граница объекта» или 1 - «остальная область».

Алгоритм позволяет разделить пиксели двух классов («полезные» и «фоновые»), рассчитывая такой порог, чтобы внутриклассовая дисперсия была минимальной. В различных русскоязычных источниках можно встретить различные способы написания фамилии автора, Нобуюки Отцу, например, можно встретить «метод Оцу» и «метод Отсу» [2].

Главной целью бинаризации является радикальное уменьшение количества информации, с которой приходится работать. Удачная бинаризация сильно упрощает последующую работу с изображением.

Метод Отцу относится к глобальным методам бинаризации, то есть к таким методам, которые работают со всем изображением сразу. В ходе работы находится порог бинаризации *t*, с помощью которого происходит деление на черное и белое, причем величина порога *t* остается неизменной в течение всего процесса бинаризации [8].

Общая схема работы алгоритма представлена на рисунке 1.

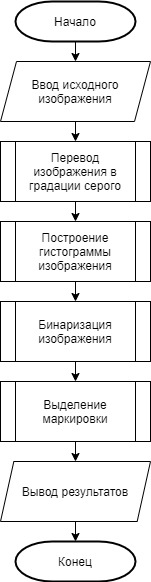


Рисунок 1 – Общая схема работы алгоритма Отцу

Как видно из блок-схемы, первой процедурой, после ввода исходного изображения, является перевод изображения в градации серого. Наглядно этот процесс представлен на рисунке 2.

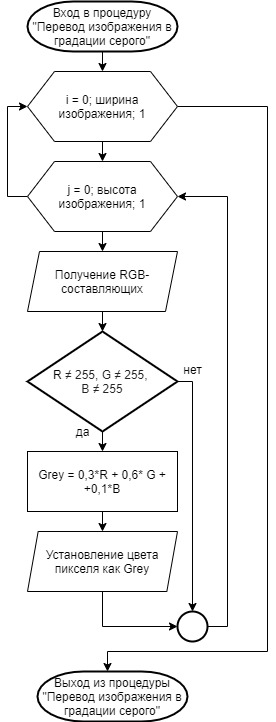


Рисунок 2 – Процедура «Перевод изображения в градации серого»

В данном случае алгоритм следующий;

a) пробег двойным циклом по каждому пикселю изображения:

b) получение RGB-составляющих каждого пикселя;

c) если ни составляющая красного канала, ни составляющая зеленого канала, ни составляющая синего канала не равны 255, то происходит присвоение каждому пикселю уровня яркости градаций серого путем сложения 30% яркости красного канала, 60% яркости зеленого канала и 10% яркости синего канала (это классическое тоновое соотношение для более равномерного распределения ступеней яркости между различными цветовыми оттенками для преобразования в градации серого).

После того, как все изображение, было переведено в градации серого, необходимо построить гистограмму уровней яркости изображения, блок-схема данного алгоритма представлена на рисунке 3.



Рисунок 3 - Процедура «Построение гистограммы изображения»

На данном этапе необходимо сделать следующее:

a) ранжирование пикселей по яркости от самого светлого к самому темному;

b) определение количества пикселей каждого уровня яркости;

c) построение гистограммы на основе полученных данных: по оси *x* -расположение уровней яркости пикселей, а по оси *y* – расположение количества пикселей данного уровня яркости.

После того, как гистограмма построена, можно приступать к бинаризации изображения. Блок-схема данной процедуры представлена на рисунке 4.

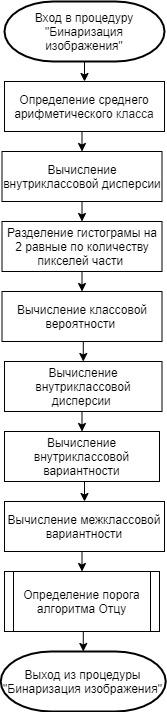


Рисунок 4 – Процедура «Бинаризация изображения»

На данном этапе производится присваивание пикселям одной из двух яркостей: либо 0, либо 255. Происходит это следующим образом:

a) определение среднего арифметического класса  по формуле 1:

 (1)

где *n* – общее количество пикселей изображения;

*i* – уровень яркости пикселя;

*p* – количество пикселей каждого уровня яркости.

b) вычисление внутриклассовой дисперсии  по формуле 2:

 (2)

c) разделение гистограммы на 2 класса, приблизительно равные по количеству пикселей (вдоль по оси *x*);

d) вычисление классовой вероятности для класса 1 (*W*1) и класса 2 (*W*2) по формуле 3:

; (3)

e) вычисление внутриклассовой дисперсии для каждого класса по формуле 2;

f) вычисление внутриклассовой вариантности  по формуле 4:

; (4)

g) вычисление межклассовой вариантности  по формуле 5:

; (5)

h) определение порога алгоритма Отцу (алгоритм представлен на рисунке 5), после чего процесс бинаризации изображения можно считать завершенным.



Рисунок 5 – Процедура «Определение порога алгоритма Отцу»

a) построение таблицы из 3-х строк и количества столбцов *n*, где *n* – количество уровней яркости пикселей; 1 строка – пороговый уровень от 0 до *n*, 2 и 3 строки – это коэффициенты  и , соответствующие данному уровню порога;

b) поиск порога, чей коэффициент  максимально приближен к , рассчитанной в результате разделения гистограммы на 2 класса ;

c) вывод результата (порога *t*).

Регулируя порог при помощи бегунка, становится возможным уточнить местоположение маркировки и исключить некоторые помехи на изображении.

Следующий этап – выделение маркировки. Для того, чтобы это осуществить, необходимо построить две гистограммы изображения (исходя из алгоритма на рисунке 3). Одна гистограмма отражает местоположение метки по оси *x* (тогда можно будет определить координаты *x1* и *x2*), а вторая – местоположение метки по оси y (следовательно, определяются координаты *y1* и *y2*).

Таким образом, после определения этих координат, становится возможным определение координат маркировки. Так координаты (*x1; y1*) отвечают за левый верхний угол маркировки, а координаты (*x2; y2*) – за правый нижний угол. Затем происходит вывод координат.

Наглядно процесс выделения маркировки представлен на рисунке 6.



Рисунок 6 – Процедура «Выделение маркировки»

Достоинствами метода Отцу являются:

- простота реализации;

- адаптация к различного рода изображениям, при помощи выбора оптимального порога;

- быстрое время выполнения [1].

Наряду с достоинствами, этот метод имеет и недостатки, ведь сама по себе пороговая бинаризация чувствительна к неравномерной яркости изображения. Решением такой проблемы - ручное регулирование порога. В программе, описывающейся в данном курсовом проекте это реализовано при помощи бегунка, который принимает первоначальное положение согласно порогу, вычисленному алгоритмом. Передвигая бегунок, появляется возможность увеличения или уменьшения яркости пикселей, при которой определяется, станет пиксель белым или черным при бинаризации.

## 3. Апробация разработанных моделей и алгоритмов

Главной целью данного курсового проекта является сравнение разработанного алгоритма и алгоритма Отцу из библиотеки AForge при локализации матричной маркировки.

Результаты апробации разработанного алгоритма представлены на рисунках 7, 8, 9, 10 и 11.

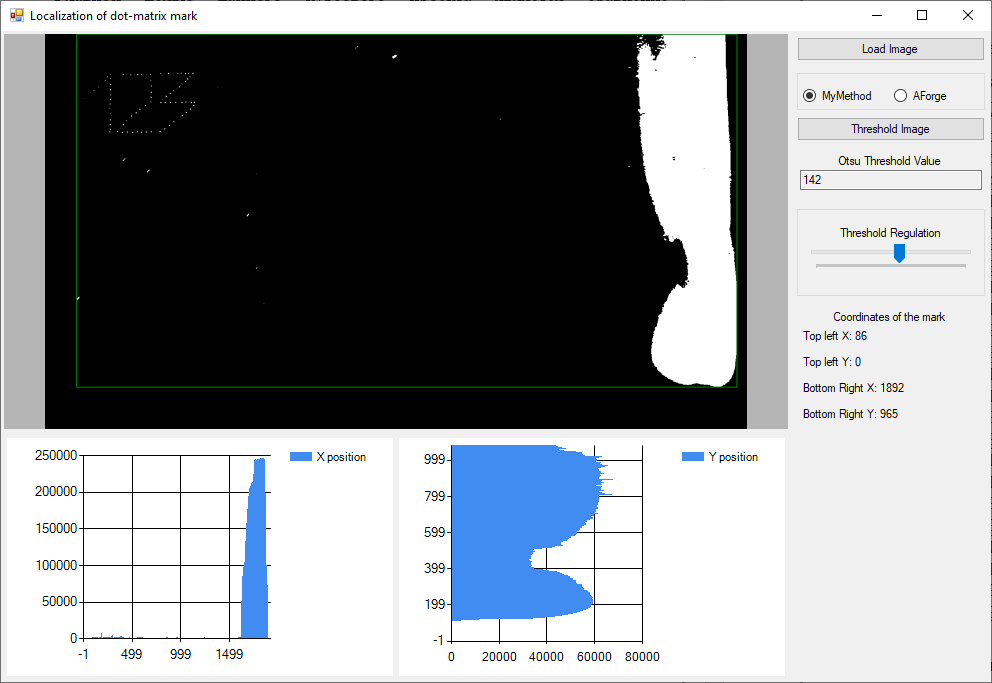


Рисунок 7 – Тест первого исходного изображения

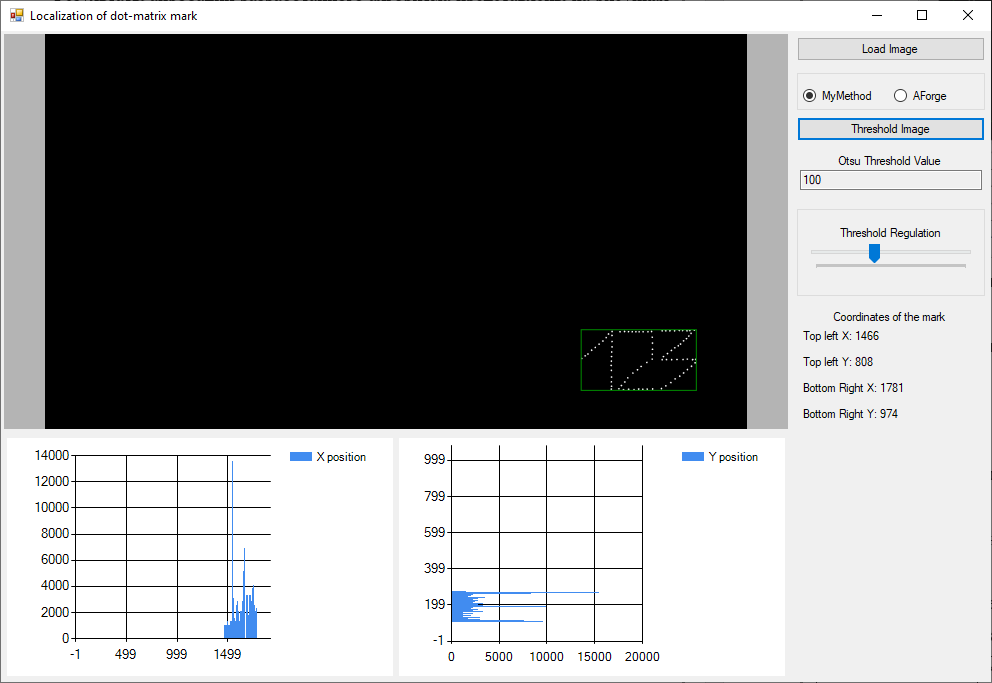


Рисунок 8 – Тест второго исходного изображения

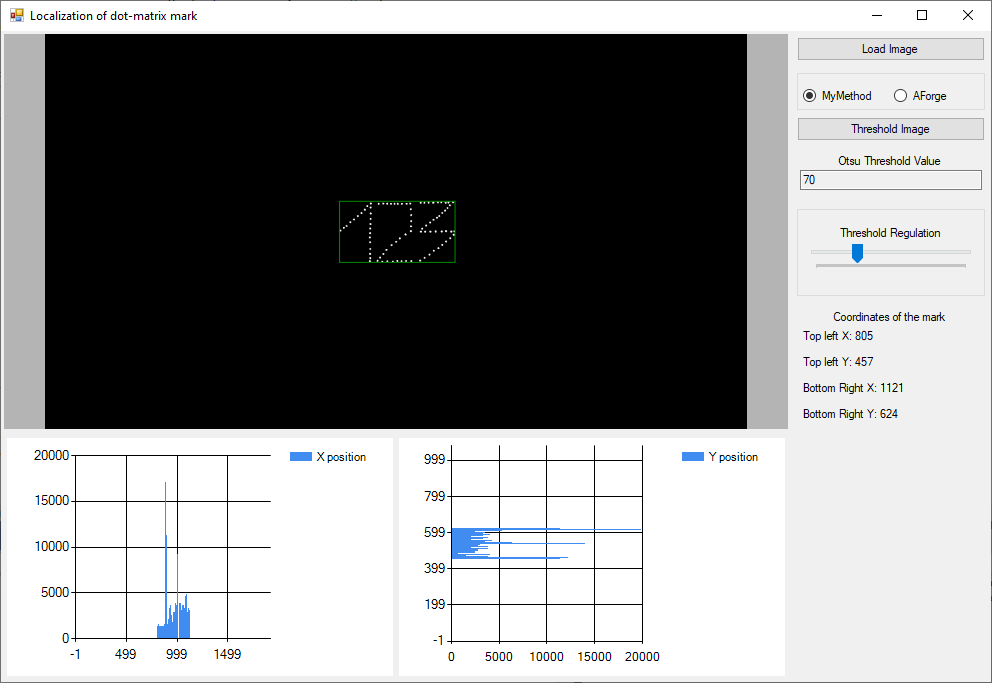


Рисунок 9 – Тест третьего исходного изображения

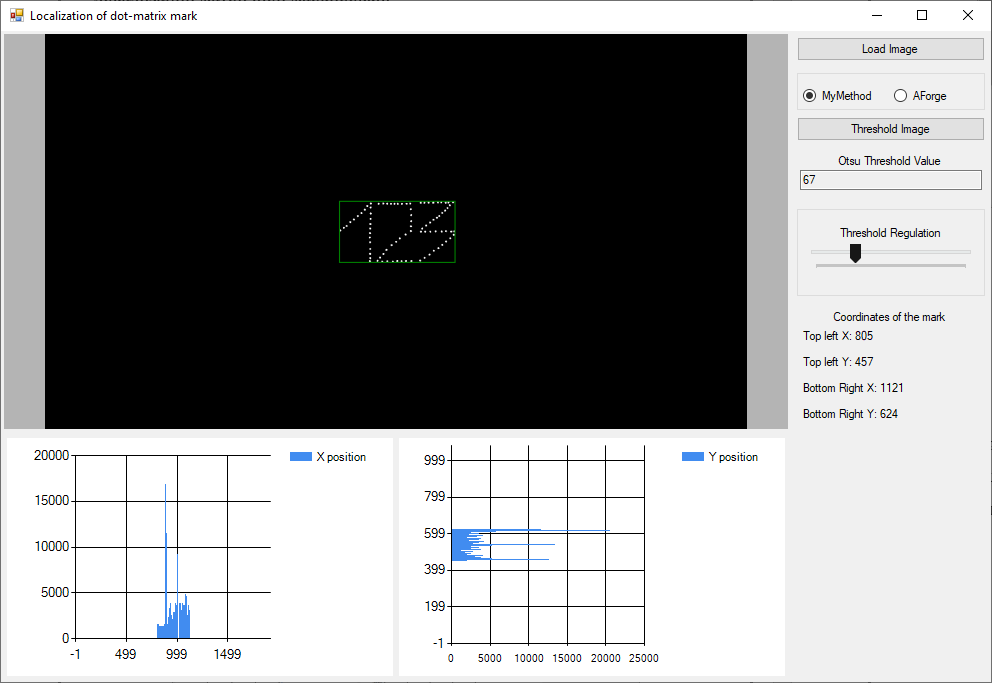


Рисунок 10 – Тест четвертого исходного изображения

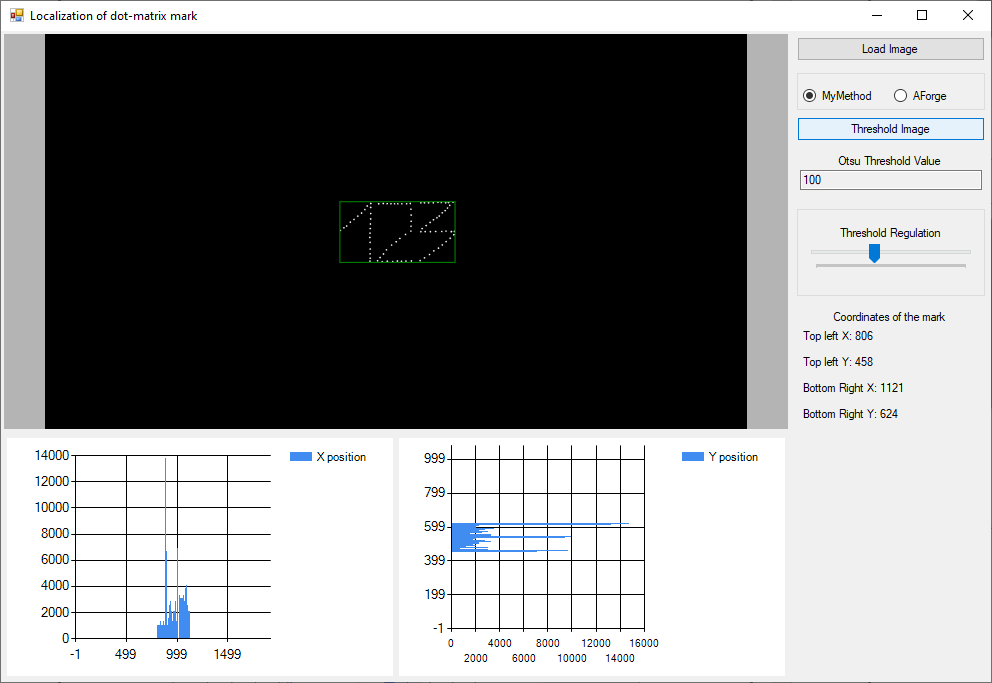


Рисунок 11 – Тест пятого исходного изображения

С этими же изображениями был проведен тест по определению локализации матричной маркировки с использованием метода Отцу из библиотеки AForge. Результаты этого теста представлены на рисунках 12, 13,14, 15 и 16.

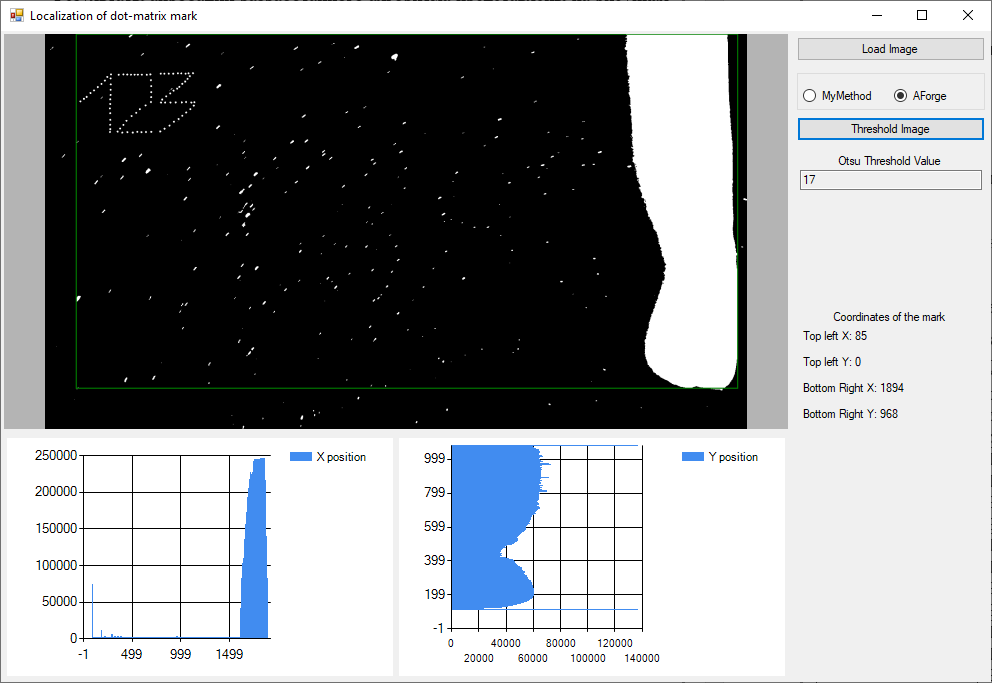


Рисунок 12 - Тест первого исходного изображения

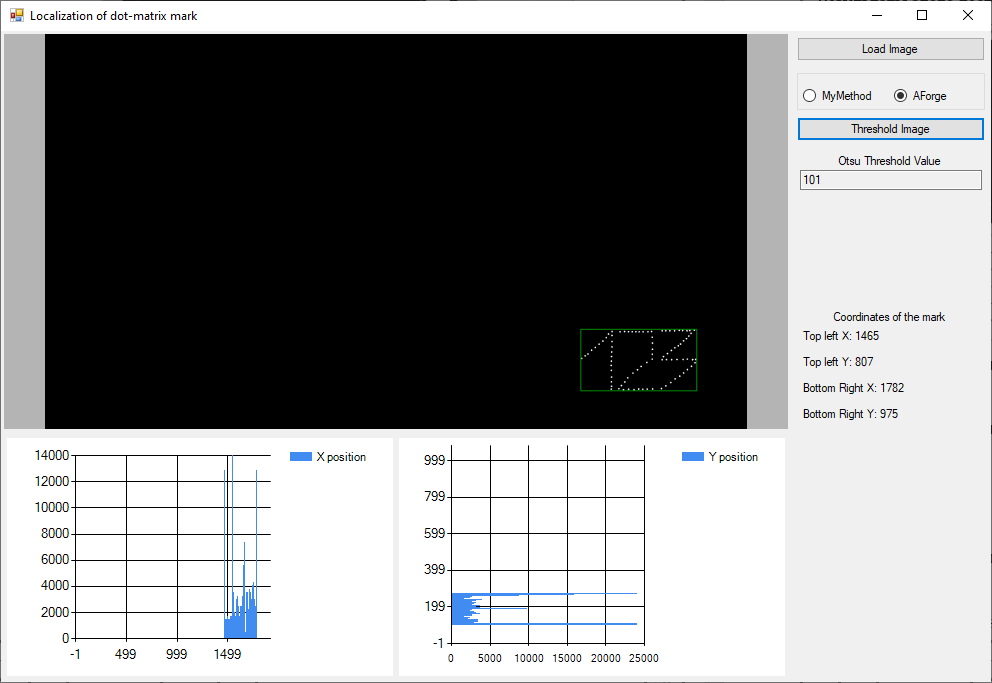


Рисунок 13 - Тест второго исходного изображения

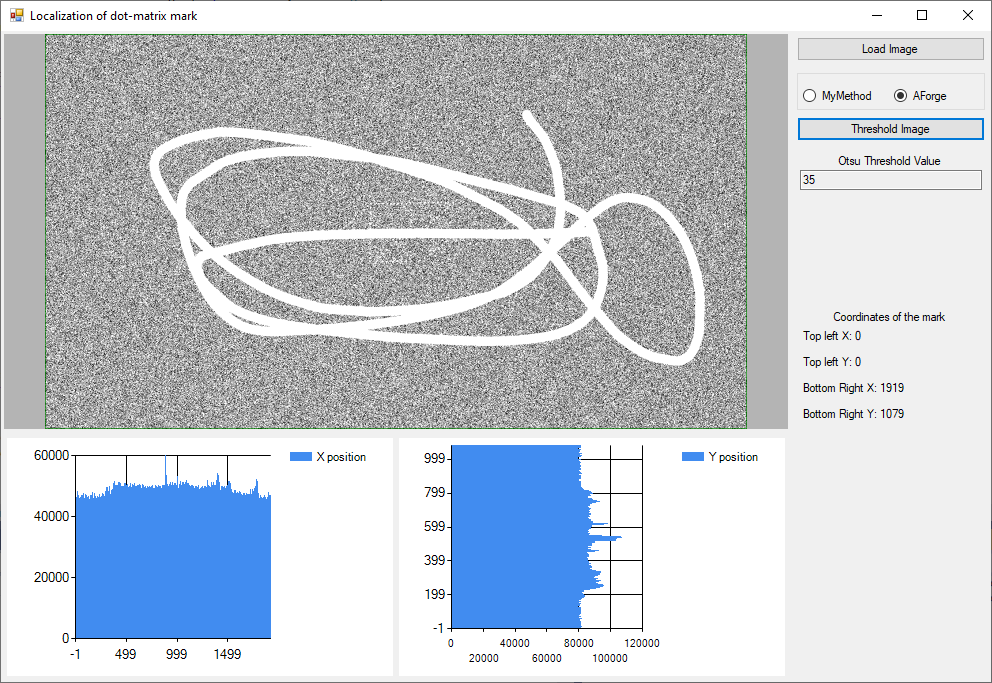


Рисунок 14 - Тест третьего исходного изображения

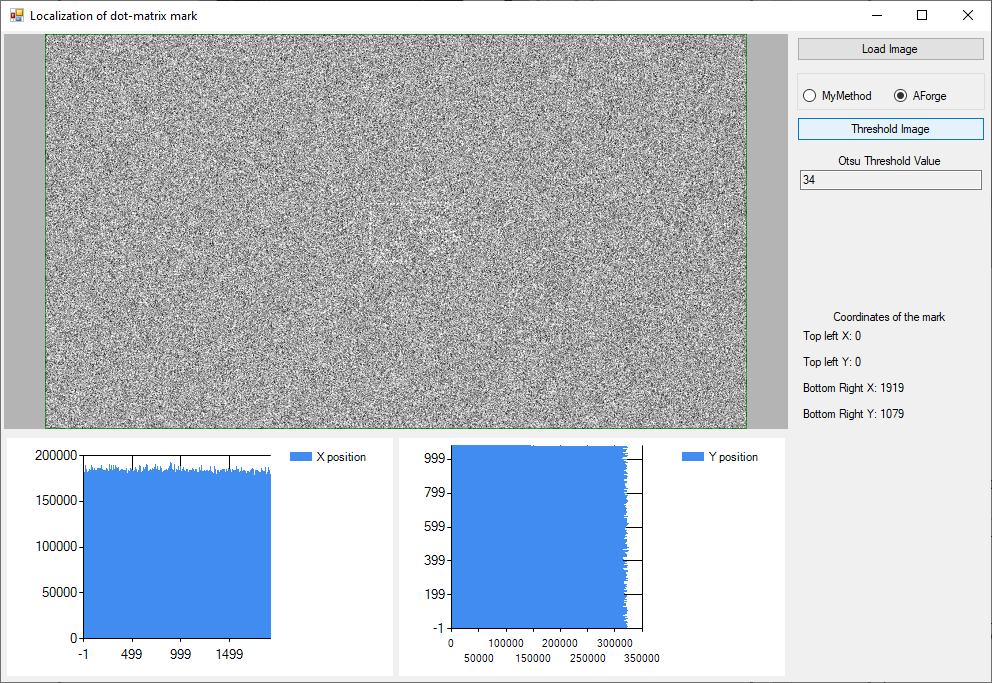


Рисунок 15 – Тест четвертого исходного изображения

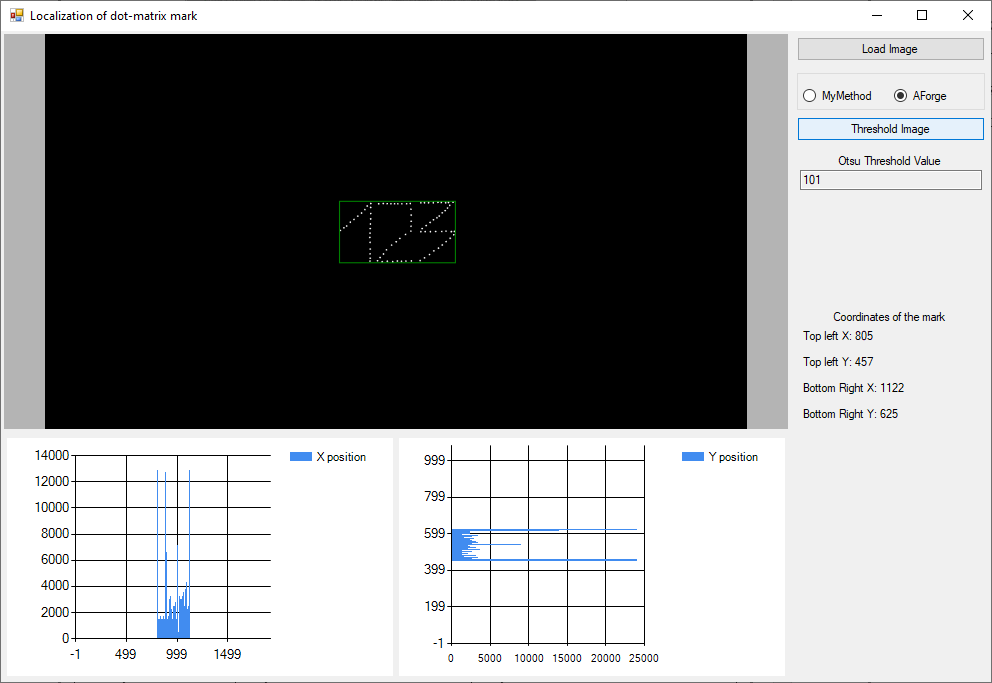


Рисунок 16 – Тест пятого исходного изображения

Все полученные данные можно свести в таблицу 1.

Таблица 1 – Сравнение результатов работы алгоритмов

|  |  |  |
| --- | --- | --- |
|  | Координаты, полученные при помощи бинаризации изображения разработанным алгоритмом | Координаты, полученные при помощи бинаризации изображения алгоритмом Отцу из библиотеки AForge |
| Изображение 1 | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: |
| Изображение 2 | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: |
| Изображение 3 | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: |
| Изображение 4 | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: |
| Изображение 5 | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: | Top left X:  Top left Y:  Bottom right X:  Bottom right Y: |

На основании проведенных тестов можно сделать вывод о том, что с помощью собственного разработанного алгоритма локализация маркировок происходит за счет того, что имеется возможность ручного регулирования порога, но локализация невозможна при наличии на исходных изображениях сильных засветов. В дальнейшем планируется проведение исследования удаления бликов и засветов на изображениях.

## Заключение

В процессе курсового проектирования был разработан алгоритм на основе алгоритма Отцу, который позволял бы локализовать матричную маркировку на цифровом изображении. Были соблюдены требования к работе алгоритма: он оперирует со всеми входными данными и получает в результате координаты локализованной маркировки на изображении, состоит из детерминированной последовательности отдельных элементарных шагов, работающих системно, обладает результативностью и высокой скоростью выполнения операций.

В первую очередь был проведен анализ предметной области, проанализированы аналогичные алгоритмы (используя библиотеку AForge), предъявлены требования к разрабатываемому алгоритму локализации. Затем был проанализирован алгоритм Отцу с математической точки зрения, а также составлена его блок-схема. На основе алгоритма Отцу был разработан алгоритм локализации матричной маркировки на цифровом изображении. В завершение были проведены экспериментальные исследования, сравнивающие работу алгоритмов из библиотеки AForge с разработанным алгоритмом и сделаны выводы о том, что разработанный алгоритм точнее локализует маркировки, но оба алгоритма не справляются с локализацией маркировок на изображениях с сильными засветами.

Таким образом, задачи данного курсового проекта были выполнены в полном объеме.

## Список использованной литературы

1. Автоматический выбор порогов для сегментации изображений на основе градиентных структурных тензоров [электронный ресурс] / Грузман // Доклады Академии наук высшей школы Российской Федерации. - 2013. - №2. - с.67-75. - Режим доступа: https://rucont.ru/efd/236309.

2. Гонсалес, Р., Вудс, Р. Цифровая обработка изображений: монография. - Москва: Техносфера, 2005. - 1072 с.

3. Грузман, И.С. Цифровая обработка изображений в информационных системах: учеб. пособие. – Новосибирск: Изд-во НГТУ, 2000. – 168 с.

4. Кравцова, Т.А. Сравнительное исследование методов адаптивной бинаризации в задаче автоматизированного анализа изображений клеток в иммуноцитохимии. Молодежный научно-технический вестник, 2015.

5. Приоров, А.Л. Цифровая обработка изображений: учеб. пособие / Ярославский государственный университет. – Ярославль: ЯрГУ, 2007. – 235 с.

6. Федоров, А. Бинаризация черно-белых изображений: состояние и перспективы развития [электронный ресурс]. Режим доступа: http://itclaim.ru/Library/Books/ITS/wwwbook/ist4b/its4/fyodorov.htm/ (дата обращения: 31.03.2020).

7. Фисенко, В.Т., Фисенко, Т.Ю. Компьютерная обработка и распознавание изображений: учеб. пособие. – СПб.: СПбГУ ИТМО, 2008. – 192 с.

8. Янковский, А.А., Бугрий, А.Н. Критерии выбора метода бинаризации при обработке изображений лабораторных анализов // АСУ и приборы автоматики [электронный ресурс], 2010. № 153. Режим доступа: http://cyberleninka.ru/article/n/kriterii-vybora-metoda-binarizatsiipri-obrabotke-izobrazheniy-laboratornyh-analizov/ (дата обращения: 31.03.2020).

## Приложение А

Исходные данные к проекту



Рисунок А1 – Исходное изображение 1



Рисунок А2 – Исходное изображение 2

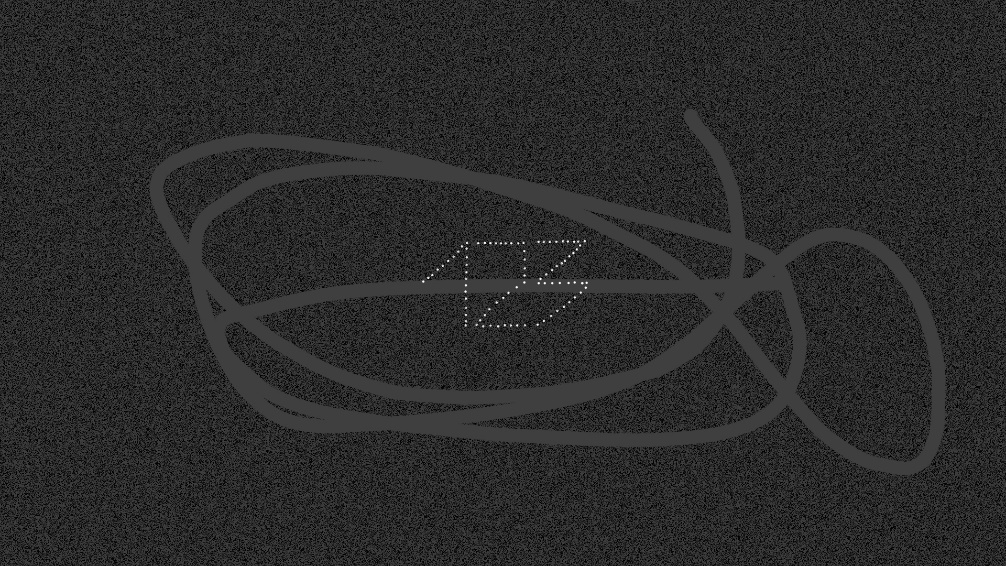


Рисунок А3 – Исходное изображение 3



Рисунок А4 – Исходное изображение 4



Рисунок А5 – Исходное изображение 5