

Санкт-Петербургский государственный университет

Кафедра системного программирования

Ярош Дмитрий Сергеевич

# Выбор фокусной серии в видеопотоке

Курсовая работа

Научный руководитель:  
ст. преп. Я. А. Кириленко

Консультант:  
вед. прогр. ООО «КиберТех Лабс» А. В. Корнилова

Санкт-Петербург  
2020

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1. Постановка задачи</b>	<b>7</b>
<b>2. Обзор</b>	<b>9</b>
2.1. Фокус-стекинг: алгоритмические подходы . . . . .	9
2.1.1. Pixel-based approach . . . . .	9
2.1.2. Neighbour-based approach . . . . .	9
2.1.3. Transform-based approach . . . . .	10
2.2. Фокус-стекинг: существующие решения . . . . .	12
2.2.1. Коммерческие решения . . . . .	13
2.2.2. Open-source решения . . . . .	15
2.3. Выбор сфокусированных изображений . . . . .	15
2.3.1. Операторы фокусной меры . . . . .	16
2.3.2. Зашумленные изображения . . . . .	17
2.4. Датасет . . . . .	18
<b>3. Прототип системы автоматического отбора сфокусированных кадров</b>	<b>20</b>
3.1. Сравнение и тестирование алгоритмов . . . . .	21
3.1.1. Устойчивость алгоритмов фокус-стекинга на большом стеке . . . . .	21
3.1.2. Метрика для сравнения алгоритмов выбора сфокусированных кадров . . . . .	22
3.1.3. Сравнение фокусных операторов . . . . .	23
3.2. Алгоритм отбора сфокусированных кадров . . . . .	24
3.2.1. Описание . . . . .	25
3.2.2. Тестирование . . . . .	28
3.2.3. Оптимальные параметры алгоритма . . . . .	29
<b>4. Применение</b>	<b>31</b>
4.1. Кроссплатформенная мобильная библиотека . . . . .	31

<b>Заключение</b>	<b>32</b>
<b>Список литературы</b>	<b>33</b>

# Введение

С развитием технологий оптической микроскопии стоимость микроскопов стала достаточно низкой для их массового использования дома и в образовательных учреждениях. Однако оптические микроскопы имеют ряд проблем, основная из которых — отсутствие возможности рассмотреть образец целиком. Это связано с тем, что оптические системы данных микроскопов позволяют лишь части объемного объекта находиться в фокусе из-за использования линз с малой глубиной резкости. Для того чтобы получить представление об объекте целиком, необходимо рассматривать его поэлементно, постоянно меняя фокусное расстояние микроскопа. Таким образом, нет возможности без использования сторонних средств получить полное изображение образца.

Основным решением данной проблемы на текущий момент является цифровая обработка кадров, полученных с микроскопа. Но в большинстве случаев бюджетные микроскопы не комплектуются камерой для захвата изображения. Одним из всегда имеющихся под рукой устройств для захвата цифрового изображения и дальнейшей его обработки является смартфон. Так, Arunan Skandarajah et. al [23] показали, что разрешений камер современных смартфонов достаточно для захвата изображения с оптического микроскопа без потери данных, что подтверждает целесообразность использования смартфона для данной задачи.

Среди методов цифровой обработки изображений, повышающих качество, особое место занимает фокус-стекинг — метод, который объединяет несколько изображений, сделанных с разными фокусными расстояниями. В результате получается изображение с большей глубиной резкости, чем у любого из отдельных исходных изображений (рис. 1). Одним из значимых свойств алгоритма фокус-стекинга является его **устойчивость** — сохранение качества получаемого изображения при возрастании числа кадров во входном наборе изображений.

Однако существующие решения в области фокус-стекинга, такие как Zerene Stacker [26], Affinity Photo [25], Helicon Focus [11], требуют от пользователя самостоятельного выбора кадров. Отбор кадров



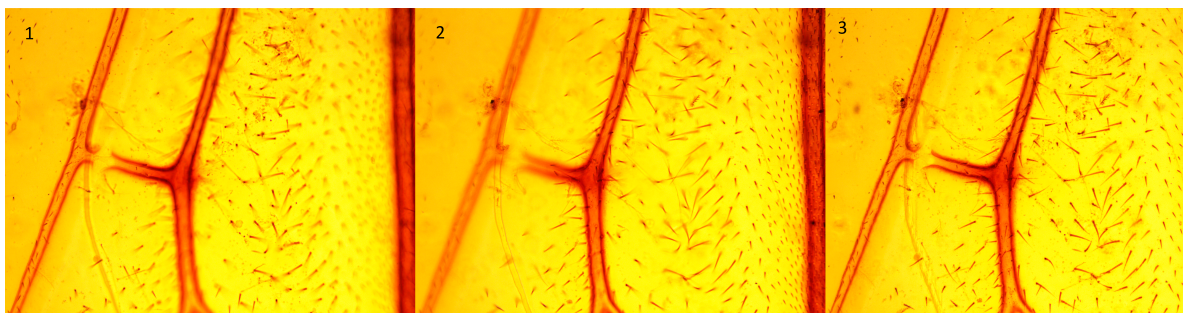


Рис. 1: Пример фокус-стекинга: первые два изображения — исходные, третье — результат фокус-стекинга

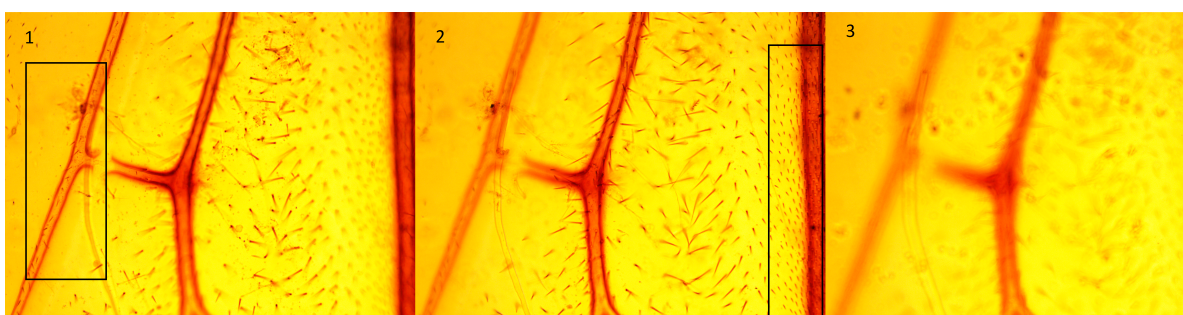


Рис. 2: Пример отбора кадров

необходим, так как заранее неизвестно на каком из кадров хотя бы один фрагмент изучаемого образца окажется в фокусе. Также, в случае когда фокусное расстояние микроскопа меняется медленно, в фокусе остаются те же фрагменты изучаемого образца, поэтому данные повторы необходимо вручную удалять. На некоторых кадрах в фокусе могут оказаться пылинки, фрагменты грязи или воды, находящиеся на линзе микроскопа (особенно часто такие проблемы возникают в случае бюджетных моделей) — такие кадры тоже необходимо отфильтровать, так как они не несут информации об изучаемом образце. Пример отбора кадров представлен на рис. 2. На первых двух кадрах в фокусе находятся выделенные сектора, а третий кадр не содержит ни одного из фрагментов образца в фокусе, он будет отфильтрован и не попадет в итоговый стек в отличие от первых двух.

В связи с данными сложностями ручной процесс обработки видео одного образца занимает значительное количество времени, поэтому необходима его автоматизация. Однако даже при идеальном выборе фокусной серии не все алгоритмы фокус-стекинга дают удовлетвори-

тельными результатами. Данные проблемы подробно описали G. Wang, W. Li, X. Chen [3]. В большинстве случаев такие искажения возникают на изображениях низкого разрешения, что не позволяет свободно применять все алгоритмы фокус-стекинга в нашем случае — записи видео с микроскопа на смартфон при медленном сканировании образца.

В данной работе будет предложено комплексное решение автоматизированного выбора фокусной серии с использованием цифровой обработки изображений и алгоритм фокус-стекинга, который будет давать качественное изображение на выбранной фокусной серии. Это позволит обрабатывать даже большие наборы кадров без потерь производительности и качества.

# 1. Постановка задачи

Цель работы — разработать решение в рамках кроссплатформенной библиотеки для смартфона, которое будет включать в себя алгоритм фокус-стекинга и автоматизированный отбор кадров для данного алгоритма. Общими требованиями к решению являются высокое качество итогового изображения, производительность на мобильных платформах и устойчивость на стеках произвольного размера.

Требованиями к реализации алгоритма выбора фокусной серии являются:

- минимизация размера стека, при сохранении в нем всех элементов образца в сфокусированном виде;
- удаление полностью не сфокусированных кадров;
- удаление кадров, содержащих в фокусе пыль или грязь.

Требованиями к реализации алгоритма фокус-стекинга являются:

- наличие всех сфокусированных элементов образца на итоговом изображении;
- их правильное расположение относительно друг друга.

Для достижения цели были поставлены следующие задачи:

1. сделать обзор существующих алгоритмов фокус-стекинга;
2. сравнить их устойчивость на большом стеке в случае видео, снятого на камеру смартфона;
3. сделать обзор существующих алгоритмов выбора фокусной серии;
4. выработать метрики для оценки качества фокусной серии и провести сравнение алгоритмов;
5. выбрать из существующих или разработать свои алгоритмы фокус-стекинга и выбора фокусной серии, оптимальные для решаемой задачи;

6. реализовать выбранные методы фокус-стекинга и выбора фокус-ной серии в кроссплатформенной мобильной библиотеке.

## 2. Обзор

В данной главе будут рассмотрены различные подходы к решению задачи фокус-стекинга, инструменты, реализующие их, а также способы и алгоритмы, позволяющие проводить выбор сфокусированных кадров из большого стека изображений.

### 2.1. Фокус-стекинг: алгоритмические подходы

M. S. Sigdel et. al [10] выделили три ключевых подхода к фокус-стекингу: pixel based approach, neighbour based approach, transform based approach.

#### 2.1.1. Pixel-based approach

Данный подход основан на одиночных пикселях. Для каждой позиции на изображении-результате сравниваются пиксели изображений из входного набора кадров и выбирается наиболее сфокусированный пиксель. Фокусировка пикселя на изображениях определяется с помощью специальной функции (например, оператор Лапласа), затем с помощью критерия (например, максимум) выбирается самый подходящий пиксель. Данный подход не работает с изображением как с единым целым, в результате чего является высоко зависимым от исходного стека. При этом pixel based approach отличается высокой скоростью обработки данных, так как производит минимум операций над пикселями среди всех остальных подходов. В большинстве случаев для определения четких сегментов используются свертки Лапласа и Собеля, для которых существует большое количество реализаций, включающих в себя машинно-зависимые оптимизации.

#### 2.1.2. Neighbour-based approach

Данный подход основан на соседях пикселя. Метод улучшает предыдущий, решая проблему пикселей на границе объектов. Там образуются

Подход	Скорость	Качество итог. изобр.	Устойчивость
Осн. на пикселях	выс.	низ	низ.
Осн. на соседях пикселя	средн.	средн.	средн.
Веществ. вейвлеты	низ.	выс.	выс.
Компл. вейвлеты	низ.	выс.	выс.

Таблица 1: Характеристики алгоритмов фокус-стекинга

разрывы — пиксели берутся с разных изображений и находятся вперемешку. Данная проблема продемонстрированы на рис. 3. При итоговом слиянии изображений учитывается не только значение пикселя (например, полученное с помощью оператора Лапласа), но и значения его соседей. Помимо ставшего классическим оператора Лапласа в алгоритмах, основанных на использовании оператора, можно применять меру Харриса. M. S. Sigdel et. al [10] описали улучшение меры Харриса, позволяющее использовать ее в фокус-стекинге. Авторы предлагают алгоритм фокус стекинга, основанный на использовании соседей пикселя и их модификации меры Харриса. Также они приводят краткий обзор проблем, с которыми сталкиваются различные реализации алгоритмов фокус-стекинга, такие как:

- контрастность и яркость в фрагменте изображения, находящемся в фокусе, может быть ниже, чем в этом же фрагменте находящемся вне фокуса;
- возможность изменения освещения объектов в течение съемки набора кадров, так как она происходит не мгновенно.

### 2.1.3. Transform-based approach

Данный подход основан на преобразованиях. Этот метод подразумевает преобразование изображений (обычно используются различные вейвлет-преобразования), преобразованные изображения сравниваются

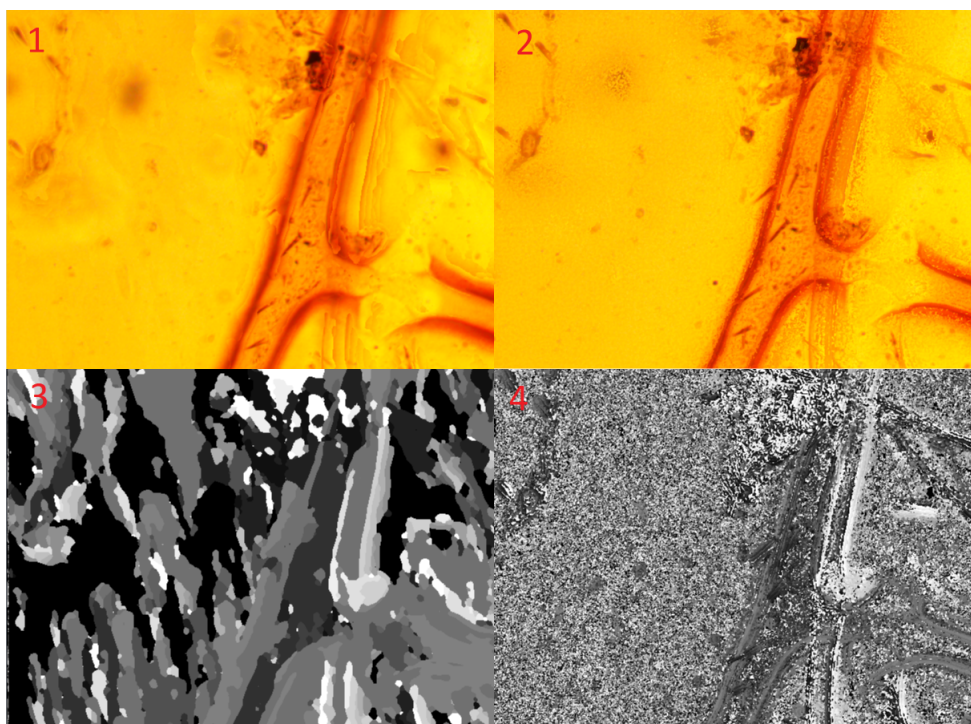


Рис. 3: Пример разрывов на границах объектов. Для полученных с помощью алгоритмов фокус-стекинга изображений приведены карты глубин. Значение пикселя на карте глубин соответствует номеру изображения в стеке, из которого пиксель в данной позиции был вставлен в результат фокус-стекинга. 1. Neighbour based approach, результат. 2. Pixel based approach, результат. 3. Neighbour based approach, карта глубин. 4. Pixel based approach, карта глубин.

с помощью специальной функции и критерия, а после к результату применяется обратное преобразование. Brigitte Forster-Heinlein et. al [5] описывают алгоритм фокус-стекинга, в котором для поиска области фокуса применяется комплексное вейвлет-преобразование. Главное преимущество вейвлет-преобразования в том, что, в отличие от других преобразований (например, преобразования Фурье), это преобразование подходит для поиска фрагментов различных размеров (области фокуса) без каких либо модификаций. В качестве вейвлетов в данной статье алгоритме комплексные вейвлеты Добеши [16]. Комплексное вейвлет-преобразование выполняется путем фильтрации с помощью комплексных фильтров низких и высоких частот конечной длины и последующей понижающей дискретизации. Процесс работы метода:

1. преобразование вектора в скаляр: многоканальные данные кон-

- вертируются в одноканальные — для изображений это черно-белое преобразование;
2. комплексное целочисленное вейвлет-преобразование;
  3. применение правила выбора (выбор наибольшего) и проверки целостности результата;
  4. обратное комплексное целочисленное вейвлет-преобразование;
  5. преобразование скаляра в вектор: восстановление цвета из наиболее близкого к результату в этой области изображения.

## **2.2. Фокус-стекинг: существующие решения**

Существующие решения делятся на две категории — аппаратные решения и программные решения. Аппаратные решения представляют собой электронные микроскопы с встроенным микропроцессором и системой предобработки изображений, позволяющей показывать пользователю полностью сфокусированные изображения образцов. Программные решения подразумевают наличие уже отснятых кадров с оптического микроскопа и лишь реализуют различные алгоритмы фокус-стекинга.

Главным недостатком аппаратных решений является их высокая цена (более 1000\$ при цене простого оптического микроскопа в 50\$) и сложная настройка, т.к. такие микроскопы считаются профессиональными и рассчитаны на использование учеными в специально оборудованных лабораториях. В связи с тем, что данная работа подразумевает использование оптического микроскопа и смартфона, данные решения в дальнейшем рассматриваться не будут.

Программные решения предполагают наличие ПК для обработки изображений. В данных решениях отсутствует фильтрация изображений, предполагается, что пользователь сам отсеет лишние кадры (полностью расфокусированные или дублирующиеся).



### 2.2.1. Коммерческие решения

Большинство коммерческих решений представляют собой фоторедакторы, одной из опций которых является фокус-стекинг. Рассмотрим ключевые приложения на данном рынке.

- Zerene Stacker [26] — коммерческий инструмент, имеющий два алгоритма фокус-стекинга, включает в себя инструмент для ручного объединения сфокусированных частей фотографий в итоговое изображение и подразумевает, что пользователь сам объединит два изображения, полученных этими алгоритмами. Оба алгоритма фокус-стекинга используют подход, основанный на соседях пикселя. Главный минус — неполная автоматизация процесса получения итогового изображения.
- Helicon Focus [11] — коммерческий инструмент, имеющий три алгоритма фокус-стекинга, имеет ограничение на размер входного набора кадров — не более 255. Алгоритмы фокус-стекинга используют подходы, основанные на соседях пикселя и одиночных пикселях. Главные недостатки — высокая цена (115\$ за самое простое издание) и наличие большого числа посторонних функций, т.к. это полноценный фоторедактор.
- Affinity Photo [25] и Adobe Photoshop [1] — большие коммерческие фоторедакторы, не предназначенные для работы с изображениями в области микроскопии. Имеют высокую стоимость и большие требования к ПК.

Результаты тестирования описанных выше коммерческих решений на тестовом датасете представлены на рис. 4. Отдельное исследование-сравнение [6] было проведено канадцем Робертом Берданом, в нем на реальных изображениях сравниваются три инструмента: Adobe Photoshop, Zerine Stacker, Helicon Focus. Автор утверждает, что Adobe Photoshop выдает приемлемый результат, однако имеет мало настроек, в то же время Helicon Focus предоставляет дружелюбный пользовательский интерфейс и поддерживает raw-файлы. Главным минусом Zerene Stacker

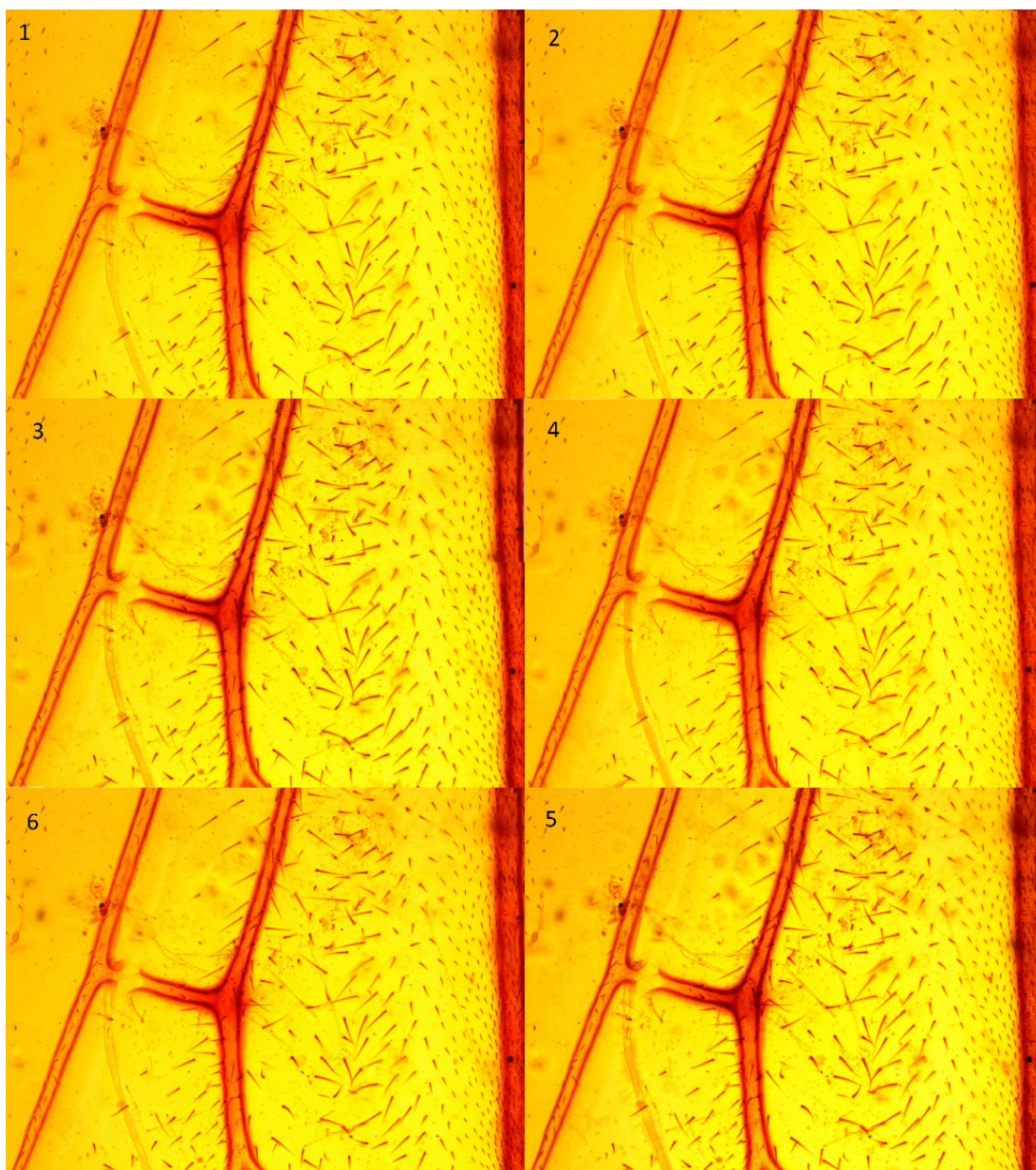


Рис. 4: Результаты запуска коммерческих решений на тестовом датасете. 1. Helicon Focus, алгоритм А. 2. Helicon Focus, алгоритм В. 3. Helicon Focus, алгоритм С. 4. Affinity Photo. 5. Zerene Stacker, алгоритм PMax. 6. Zerene Stacker, алгоритм DMap.

названа его высокая цена и отсутствие поддержки формата-raw. Поддержка файлов формата raw важна, так как данный формат позволяет хранить изображения очень высокого разрешения, полученные с использованием профессиональных фотокамер.

### 2.2.2. Open-source решения

- Extended Depth of Field Plugin for ImageJ [8]. Плагин для редактора изображений с открытым исходным кодом ImageJ [17]. Код плагина также выложен на Github. Содержит реализации нескольких алгоритмов фокус-стекинга, в том числе и основанных на преобразованиях: используются вещественные и комплексные вейвлет-преобразования. К сожалению, т.к. этот инструмент является научным, он имеет довольно недружелюбный интерфейс и большое число параметров, которые пользователь должен задать перед запуском. Для правильного ввода этих данных необходимо знание устройства алгоритмов фокус-стекинга.
- Решение от rAum<sup>1</sup> [22]. Небольшое приложение написанное на C++ и C#. Реализует несколько вариаций pixel based approach. В отличие от остальных решений не использует готовой библиотеки для работы с изображениями. Для правильной работы данное приложение требует, чтобы входной стек был упорядочен по фокусному расстоянию кадра.
- Решение от cmcguinness<sup>2</sup> [4]. Небольшое консольное приложение написанное на языке Python. Реализует pixel based approach, в качестве функции для определения фокусировки пикселя используется оператор Лапласа. Перед запуском алгоритма фокус-стекинга к стеку применяется фильтр Гаусса (размера 5 на 5) для подавления шумов.

Результаты тестирования описанных выше open-source решений на тестовом датасете представлены на рис. 5.

## 2.3. Выбор сфокусированных изображений

Отбор наиболее сфокусированных изображений необходим для сокращения времени на работу алгоритмов фокус-стекинга, а также для

---

<sup>1</sup><https://github.com/rAum>

<sup>2</sup><https://github.com/cmcguinness>

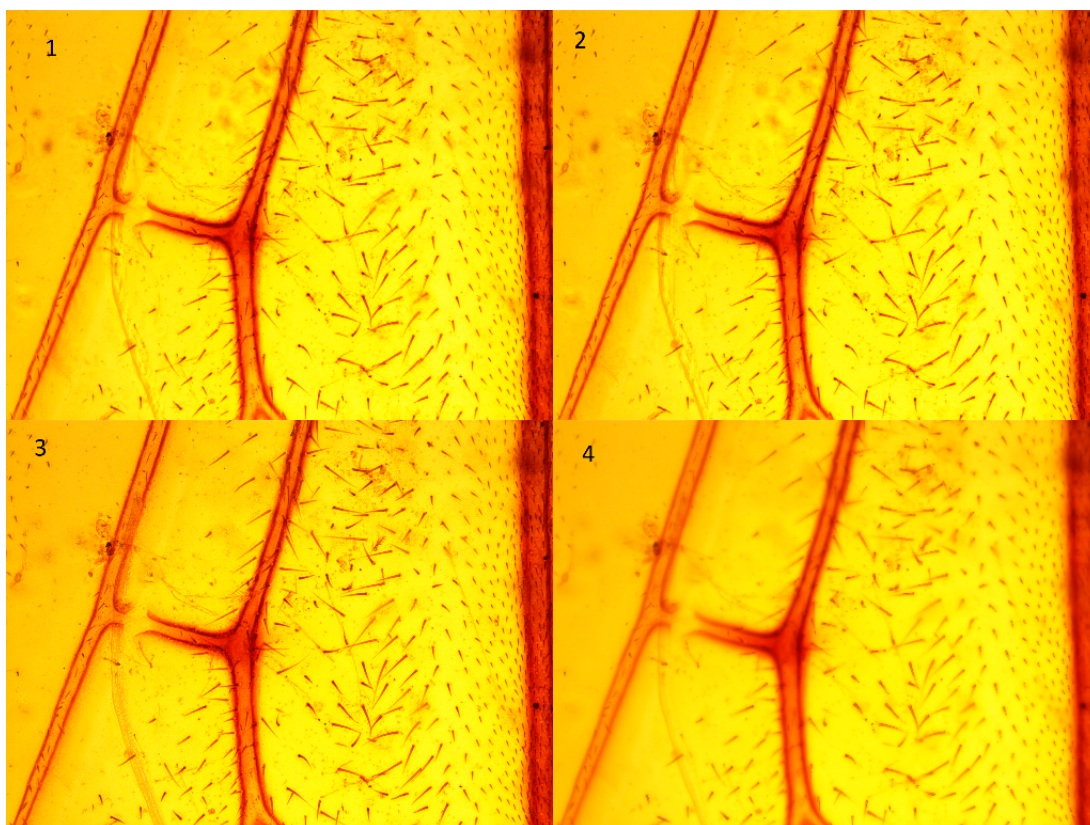


Рис. 5: Результаты запуска open-source решений на тестовом датасете. 1. Extended Depth of Field Plugin for ImageJ, вещественные вейвлеты. 2. Extended Depth of Field Plugin for ImageJ, комплексные вейвлеты. 3. Решение от rAum. 4. Решение от smcguinness.

уменьшения количества шума в итоговом изображении. Появление шума может быть связано с небольшими сдвигами образца или изменениям освещения во время захвата кадров с микроскопа.

В качестве базового инструмента поиска сфокусированных изображений применяются операторы фокусной меры, с помощью которых возможно найти максимально сфокусированные изображения.

### 2.3.1. Операторы фокусной меры

Для определения сфокусированности изображения используются операторы фокусной меры. Они делятся на несколько групп:

- основанные на градиенте;
- основанные на операторе Лапласа;



- основанные на вейвлет-преобразованиях;
- статистические;
- смешанные.

Подробный обзор таких операторов был представлен Pertuz Said et. al [21]. Также авторы провели сравнение производительности операторов при различных аспектах (зашумленные изображения, различный размер окна оператора).

Использование операторов фокусной меры для автофокуса в микроскопии было изучено O. A. Osibote et. al [2]. Авторы предлагают использовать операторы фокусной меры и выбирать максимум из результатов их применения на всех кадрах, снятых микроскопом, — самое сфокусированное положение. Также авторами было проведено сравнения операторов фокусной меры по качеству работы, точности и скорости поиска фокуса с их помощью в области микроскопии. На основе исследования сделан вывод, что наиболее подходящими операторами для применения в данной области являются: нормализованная дисперсия (the normalized variance), градиент Бреннера (the Brenner gradient), модифицированный оператор Лапласа (the sum-modified Laplacian), энергия оператора Лапласа (the energy of the Laplacian), Vollath F4 и оператор Тененграда (Tenengrad operator).

### **2.3.2. Зашумленные изображения**

В ходе применения операторов фокусной меры можно столкнуться с проблемой наличия грязи на изображениях — элементов, не являющихся частью изучаемого образца, однако попадающих в кадр. Так как грязь обычно находится на приборном стекле или линзе микроскопа, то она оказывается в фокусе в те моменты, когда основной образец размыт. В результате возможно ложное определение кадра как сфокусированного из-за наличия такой грязи в фокусе на этом кадре. Одно из возможных решений этой проблемы предлагает V. Hilsenstein [12]. Автор принимает во внимание особенности грязи относительно образ-

ца (ее форму и размер) и приводят фильтры способные обнаружить объекты, которые мы классифицируем как грязь на фотографиях и удалить такие кадры из исходного набора. Причиной появления шума на изображении также может стать сдвиг образца во время захвата кадров. Решение данной проблемы предлагают С. Zhang et. al [9]. Авторы используют аффинное преобразование и сопоставление позиций одинаковых элементов на изображениях. К сожалению, данный метод не может быть эффективно использован на изображениях с большим количеством мелких повторяющихся объектов (например, клеток в микроскопии) в связи с возможной ошибкой сопоставления позиций таких объектов.

## 2.4. Датасет

Для тестирования алгоритмов фокус-стекинга, а также операторов фокусной меры необходим датасет, содержащий видеозаписи образцов, записанные с помощью смартфона и микроскопа. Использование смартфона для записи датасета важно, т.к. одной из особенностей съемки на телефон является эффект затемнения краев изображения. Это связано с несовпадением геометрической формы сенсора камеры смартфона и окуляра микроскопа. В итоге на изображении образец находится внутри круга в центре, а остальная часть изображения - темная. Пример можно увидеть на рис. 6. Были изучены датасеты представленные в Cell Image Library, а также остальные датасеты, рекомендуемые для использования в области микроскопии [18]. К сожалению, найти готовый датасет не удалось, т.к. большинство датасетов в области микроскопии ориентированы на задачи, связанные с распознаванием и обработкой снимков клеток. В связи с этим они отсняты с использованием профессиональной фотоаппаратуры для захвата кадров и, как следствие, серьезно отличаются по качеству изображения, а также не имеют характерного для мобильных телефонов эффекта затемненных краев.

Принимая во внимание данные проблемы, компания MELScience предоставила оборудование (смартфон Nexus 6P и микроскоп) для за-



Рис. 6: Пример эффекта затемненных краев.

писи необходимого числа образцов. В результате был создан датасет [14], содержащий видеозаписи для 30 образцов. Из них 15 имели сложную структуру - их нельзя рассмотреть полностью без применения фокус-стекинга. Для каждого образца были сделаны 4 видеозаписи, отличающиеся скоростью движения предметного столика во время исследования образца.

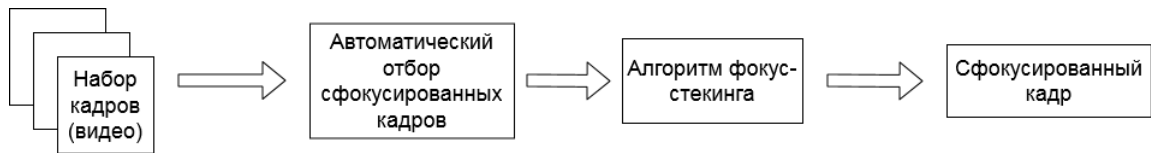


Рис. 7: Конвейер для получения сфокусированного изображения.

### 3. Прототип системы автоматического отбора сфокусированных кадров

Как было выяснено в предыдущих разделах, несмотря на наличие большого количества алгоритмов фокус-стекинга и операторов фокусной меры не существует готового решения для мобильных устройств в области микроскопии, способного проводить автоматический отбор сфокусированных кадров в видеозаписи и применять алгоритм фокус-стекинга к выбранным кадрам.

В данной главе будет описан прототип системы автоматического отбора сфокусированных кадров и получения изображения, содержащего все сфокусированные элементы образца, реализованный для тестирования алгоритмов фокус-стекинга и операторов фокусной меры.

Для реализации прототипа был выбран язык программирования Python, т.к. он позволяет быстро прототипировать и реализовывать эксперименты, связанные с компьютерным зрением. Также при реализации прототипа использовались библиотеки OpenCV [20] (для эффективной работы с изображениями), SciPy [24] (для использования эффективных функций свертки), NumPy [19] (для работы с массивами) и Matplotlib [13] (для построения графиков во время проведения экспериментов).

Архитектура прототипа представляет собой конвейер (рис. 7), в котором видео или набор кадров вначале обрабатываются с помощью автоматического отбора сфокусированных кадров, а затем подаются на вход алгоритму фокус-стекинга. Для проверки различных комбинаций алгоритмов фокус-стекинга и операторов фокусной меры в прототипе были реализованы следующие алгоритмы фокус-стекинга.

- Основанный на одиночных пикселях. В качестве оператора можно



использовать оператор Лапласа или оператор Собеля. В результирующее изображение выбираются пиксели с максимальным значением, полученном при применении оператора к изображению.

- Основанный на соседях пикселя. В качестве оператора используется оператор Харриса.
- Основанный на преобразованиях. В качестве преобразований выбраны вещественные и комплексные вейвлет-преобразования.

Опираясь на изученные научные статьи, для использования в прототипе было решено выбрать следующие операторы:

- Laplacian Variance;
- Tenengrad operator;
- Vollaht F4;
- Modified Laplacian.

Данный выбор сделан с учетом того, что необходимо получить высокопроизводительные операторы для обработки изображений на телефоне.

### **3.1. Сравнение и тестирование алгоритмов**

В данном разделе будут описаны эксперименты, проведенные для оценки качества работы существующих алгоритмов в условиях решаемой задачи, и использованные для этого метрики.

#### **3.1.1. Устойчивость алгоритмов фокус-стекинга на большом стеке**

При записи данных с микроскопа с помощью смартфона получается видеозапись длительностью около минуты. Это обусловлено необходимостью изменять фокусное расстояние микроскопа во время съемки

для просмотра всего образца. При этом такая фокусировка микроскопа должна проводиться достаточно медленно для захвата всей доступной информации об образце без размытий, возникающих из-за быстрого движения приборного столика. Для проверки эффективности использования алгоритмов фокус-стекинга на таком видео были проведены замеры производительности алгоритмов фокус-стекинга. При тестировании измерялось среднее время обработки одного кадра алгоритмом и общее время работы алгоритма. В качестве видео была взята видеозапись длительностью 1 минута, состоящая из 1800 кадров. Результаты представлены в Таблице 2. Из них видно, что все алгоритмы кроме основанного на пикселях работают слишком долго (из-за большого времени обработки одного кадра). В то же время оставшийся алгоритм показывает низкое качество итогового изображения, не позволяя использовать его для получения данных об образце. Таким образом, данные алгоритмы не способны получить качественное итоговое изображение на большом стекке за требуемое время. Однако, специфика области применения позволяет сократить размер обрабатываемого стека, т.к. все биологические образцы, изучаемые с помощью микроскопа, обладают малой толщиной. Это позволяет получить полную информацию об образце из 3-5 правильно отобранных кадров.

Алгоритм	Осн. на пикселях	Осн. на соседях пикселя	Веществ. вейвлеты	Компл. вейвлеты
Время общее, мин	< 0,2	37,7	36	56,2
Время на кадр, мс	5	1259	1200	1873

Таблица 2: Время работы алгоритмов фокус-стекинга на видео (1 минута, 1800 кадров).

### 3.1.2. Метрика для сравнения алгоритмов выбора сфокусированных кадров

В качестве метрики для сравнения результатов работы операторов фокусной меры была выработана следующая система оценки результата. Подсчитывалось общее количество кадров в отобранном стекке,

Название видео	arteria			
	super_slow	slow	medium	fast
Длинное	+	-	-	-
Многоуровневое	-	-	-	-
Отобраны все части	+	+	+	+
Кол-во дубликатов	0	0	0	0
Кол-во заблюренных	0	0	1	0
Размер финального стека	2	2	2	1
Качество pixel	5	5	5	5
Качество neighbour	5	5	5	5
Качество complex wavelet	5	5	5	5
Качество real wavelet	5	5	5	5
Выбранные кадры	785,791	249,25	73,74	42
Общее число кадров	2237	386	222	65

Рис. 8: Пример оценки обработки видеофайла по разработанной метрике.

количество повторяющихся кадров среди них, количество кадров, не содержащих информации об образце, а также наличие каждой детали образца хотя бы на одном изображении в итоговом стеке. Оценкой алгоритма по данной метрике является количество образцов для которых все учитываемые метрикой параметры находятся в диапазонах оптимальных значений (1-5 кадров в стеке, отсутствие кадров-дубликатов и кадров, не содержащих информации об образце, наличие всех деталей образца в стеке). Лучшим алгоритмом по данной метрике считается алгоритм с наибольшей оценкой. Пример описания приведен на рис. 8

### 3.1.3. Сравнение фокусных операторов

Для тестирования фокусных операторов был реализован наивный алгоритм отбора сфокусированных изображений - в итоговый стек попадают 5 кадров, имеющих максимальное значение фокусного оператора. Для тестирования использовались все операторы фокусной меры, реализованные в рамках прототипа. Тестирование проводилось на датасете, описанном выше. Сводные результаты представлены в Таблице 3. Полный результаты в виде таблицы находятся в данном документе. Лучший результат в ходе проведенного тестирования показал оператор Тененграда. Однако в ходе проведенного тестирования был выявлен недостаток наивного алгоритма отбора сфокусированных изображений.

В случае наличия на изображении крупного объекта, находящегося в расфокусированном состоянии, и нескольких мелких объектов, которые находятся в фокусе, любой из протестированных операторов фокусной пропускал такое изображение, тем самым теряя часть информации об образце. Пример такого случая приведен на рис. 9. На нем видно что некоторые мелкие части образца, которые присутствуют в фокусе на одном из кадров видеозаписи, отсутствуют на итоговом изображении, т.к. кадры на которых эти детали находятся в фокусе не были отобраны алгоритмом.

Оператор фокусной меры	Laplacian Variance	Tenengrad operator	Vollath F4	Modified Laplacian
Оценка по метрике	6	15	8	10

Таблица 3: Сравнение операторов фокусной меры с использованием разработанной метрики.

Таким образом итоговый стек содержит не все сфокусированные части образца. Также в стеке могут присутствовать кадры-дубликаты (кадры, не имеющие отличий для человеческого глаза, однако немного различающиеся значениями пикселей), а также кадры, которые содержат в фокусе грязь или пыль, находящуюся на линзе микроскопа или стекле препарата, а не сам образец.

### 3.2. Алгоритм отбора сфокусированных кадров

Для улучшения качества итогового стека и уменьшения его размера было принято решение создать свою модификацию алгоритма отбора сфокусированных кадров, которая будет описана в данном разделе. Алгоритм состоит из применения оператора фокусной меры, а также постобработки полученного стека, позволяющей сократить количество кадров в финальном стеке и существенно повысить его общее качество, оцениваемое по описанной выше метрике.

Для отбора всех элементов образца в качестве основы алгоритма

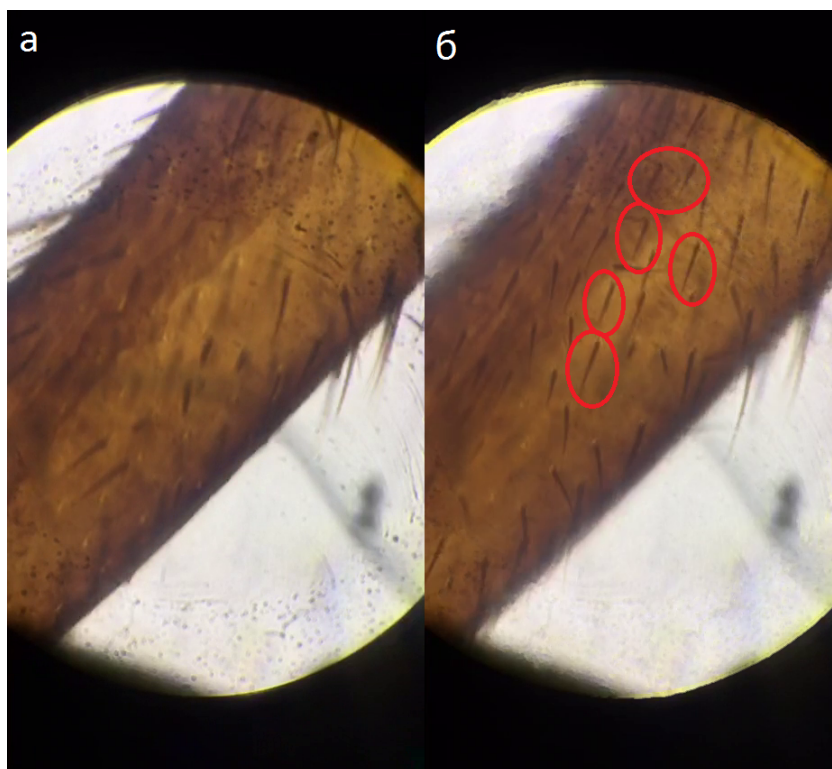


Рис. 9: Пример пропуска мелких частей образца алгоритмом отбора сфокусированных кадров. а. Результат фокус-стекинга отобранного стека. б. Один из кадров видеозаписи, из которой производился отбор кадров.

был выбран подход при котором оператор фокусной меры применяется к частям изображения. В качестве такого оператора решено было взять оператор Тененграда, т.к. он получил наивысшую оценку при сравнении фокусных операторов.

### 3.2.1. Описание

Шаги работы алгоритма представлены на рис. 10.

Алгоритм получает на вход набор кадров в порядке их следования в видеозаписи. Первым шагом алгоритма является применение оператора Тененграда к частям изображения. Каждое изображение делится на сетку из прямоугольников, размер сетки параметризуем. Так как часть выделенных секторов не содержит полезной информации (из-за эффекта затемнения по краям), такие сектора не должны учитываться при дальнейших вычислениях. Для выделения секторов с затемне-

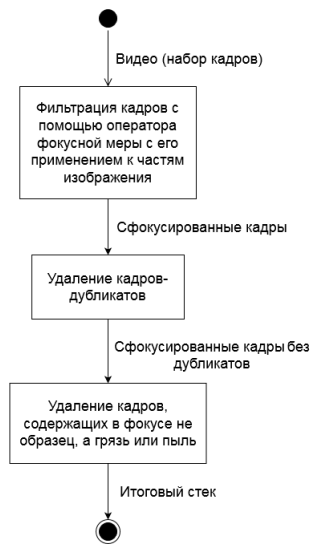


Рис. 10: Основные шаги работы алгоритма отбора сфокусированных кадров.



Рис. 11: Маска изображения для выделения сектора с полезной информацией об образце. Серая область - область в которой виден образец, черные области - затемненные области по краям изображения.

нием используется построение маски изображения с помощью порога (пример на рис. 11). Так как все снимки с использованием мобильного телефона характеризуются наличием в центре изображения окружности с образцом внутри, достаточно отбросить сектора, большая часть пикселей которых находится вне этой окружности.

Для каждого из оставшихся прямоугольников вычисляется значение оператора фокусной меры и ищется максимум среди всех обработанных изображений независимо по каждому прямоугольнику. Кадр, содержащий сегмент с максимальным значением оператора фокусной

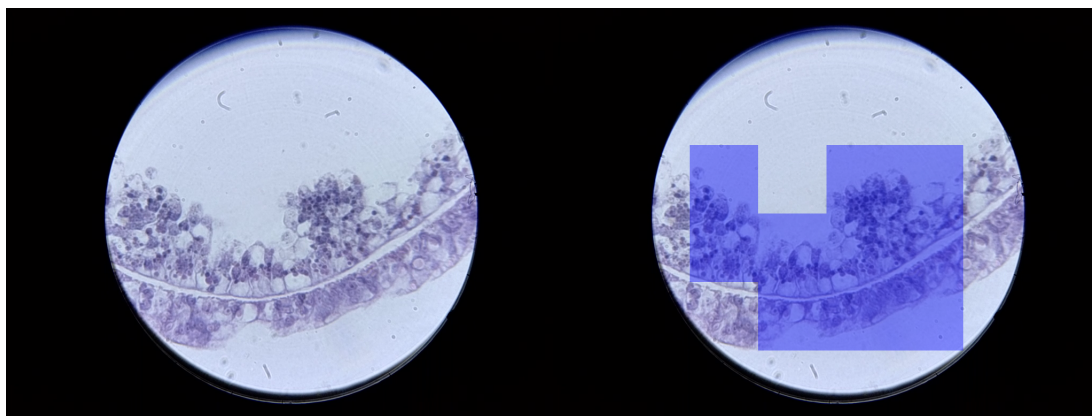


Рис. 12: Пример выделения сфокусированных секторов на изображении.

меры, попадает в финальный стек, так как имеет сфокусированный объект в данной области. Пример работы алгоритма можно увидеть на рис. 12.

Следующим шагом алгоритма является фильтрация кадров-дубликатов. Для ее проведения вычисляется попиксельная разность последовательно отобранных изображений, и отсеиваются те изображения, которые почти идентичны (их попиксельная разность меньше порогового параметризуемого значения). Данная фильтрация работает за линейное относительно количества кадров время, так как похожие кадры всегда идут последовательно в силу того, что мы обрабатываем видеозапись, записанную при движении микроскопа в одну сторону.

Финальным шагом алгоритма является фильтрация кадров, не содержащих сфокусированных элементов исследуемого образца. Такие кадры попадают в выборку из-за наличия в фокусе грязи на линзе микроскопа. В связи с тем что грязь и пыль находятся на приборном стекле или линзе микроскопа, кадры, на которых в фокусе оказываются не относящиеся к исследуемому образцу артефакты, находятся в видеофайле на удалении от кадров, содержащих в фокусе образец. Также грязь и пыль обычно занимают небольшую часть изображения. В связи с этим для их фильтрации вычисляется среднее значение оператора фокусной меры по всем секторам изображения и с помощью алгоритма поиска пиков с параметризуемой относительной высотой выбирается подпоследовательность кадров, в которой гарантированно находятся кадры,

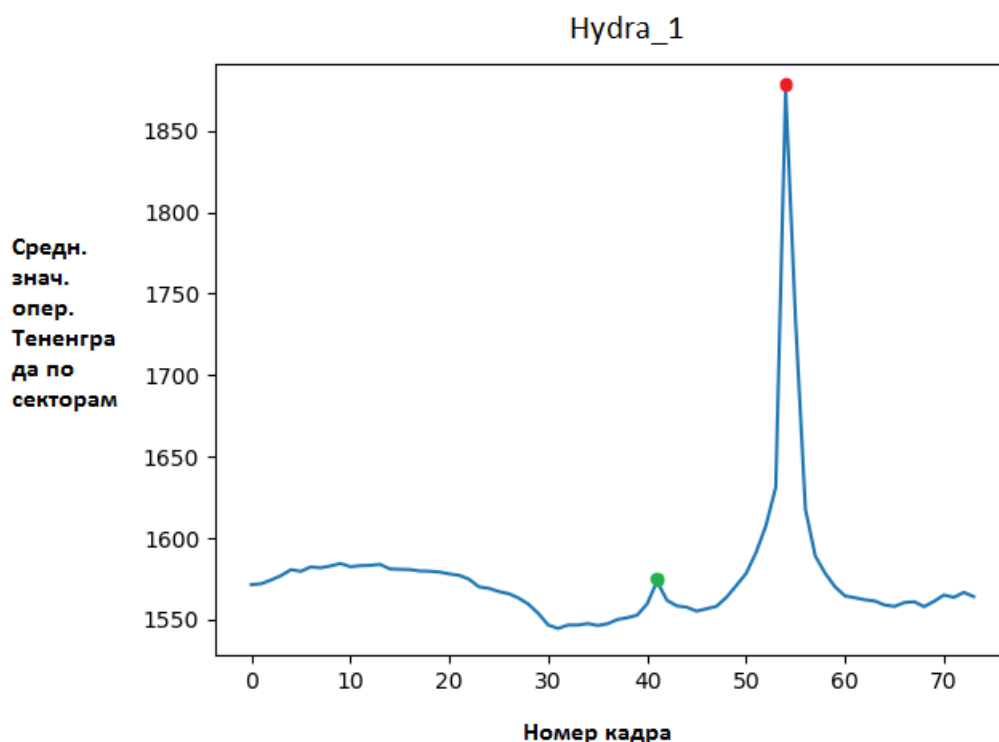


Рис. 13: График величины среднего значения оператора Тененграда для образца Hydra\_1. Красная точка показывает величину среднего значения для сфокусированного кадра, зеленая - для расфокусированного.

содержащие в фокусе образец и не содержащие грязь или пыль. График величины среднего значения оператора фокусной меры для одной видеозаписи из датасета можно увидеть на рис. 13.

Результатом алгоритма является стек небольшого размера (3-5 кадров), т.к. исследуемые образцы обладают малой толщиной и в стеке отсутствуют кадры-дубликаты.

### 3.2.2. Тестирование

Для тестирования разработанного алгоритма был использован датасет, описанный в разделе выше. Алгоритм достойно показал себя на всех образцах. Средний размер стека для плоских образцов составил 1 кадр, для многоуровневых - 3 кадра. Полные результаты тестирования можно найти по ссылке [15]. На отобранных с помощью разработанного алгоритма кадрах были запущены алгоритмы фокус-стекинга (в



случае если кадров было больше чем один), реализованные в прототипе системы отбора сфокусированных кадров. В ходе тестирования было выявлено, что на образцах с многоуровневой структурой, которые содержат разные детали в фокусе на нескольких кадрах, алгоритмы, основанные на одиночных пикселях и соседях пикселя, вносят в финальное изображения много шума и белых артефактов. В связи с этим для использования в финальном решении был выбран алгоритм фокус-стекинга с использованием комплексных вейвлетов Добеши.

### 3.2.3. Оптимальные параметры алгоритма

Разработанный алгоритм отбора сфокусированных изображений имеет несколько параметров, которые были подобраны в ходе тестирования.

- Алгоритм фокус-стекинга - алгоритм с использованием комплексных вейвлетов Добеши.
- Оператор фокусной меры - оператор Тененграда.
- Количество сегментов на которые делится изображение - 64 (8x8). Данный параметр был подобран исходя из размера объектов, которые могут быть частями образца - секторов должно быть достаточно много, чтобы внутри одного сектора не мог проявиться такой же эффект как для целого изображения, но при этом сектора не должны быть слишком мелкими - чтобы в них могли помещаться небольшие части образца целиком).
- Значение порога при построении маски изображения для удаления секторов, не содержащих полезной информации - 20. Данное значение было подобрано эмпирически таким образом, чтобы маска максимально покрывала область изображения с образцом и при этом не давала ложных срабатываний в зонах затемнения.
- Пороговое значение попиксельной разности кадров в фильтрации дубликатов - 2. Данный параметр был подобран таким образом,

чтобы большая часть дубликатов, возникающих при обработке всех образцов в датасете, удалялась, но ни один кадр, не являющийся кадром-дубликатом, не был удален ошибочно.

- Относительная высота пиков в фильтрации изображений, не содержащих в фокусе исследуемый образец - 0.8, относительной высотой вершины - 0.5 (все значения нормализуются перед поиском пика). Данный параметр был подобран таким образом, чтобы большая часть кадров, возникающих при обработке всех образцов в датасете и не содержащих в фокусе образец, удалялась, но ни один кадр, содержащий образец в фокусе, не был удален ошибочно.

## 4. Применение

В данной главе описано применение разработанного алгоритма на практике - его реализация для мобильных устройств.

### 4.1. Кроссплатформенная мобильная библиотека

Для использования разработанного алгоритма автоматического отбора сфокусированных кадров на мобильных устройствах он был реализован на языке C++ вместе с алгоритмом фокус-стекинга с использованием комплексных вейвлет-преобразований и внедрен в мобильную библиотеку, разрабатываемую в рамках дипломной работы Владимира Кутуева. Данная библиотека была разработана с использованием инструмента Djinni [7], благодаря чему ее можно использовать как на iOS, так и на Android. Для работы с изображениями в библиотеке используется OpenCV, что позволяет получить высокую производительность на большинстве смартфонов за счет присутствующих в OpenCV архитектурных оптимизаций для методов по обработке изображений. Архитектура библиотеки - процесс из четырех этапов обработки видеозаписи: отбор сфокусированных кадров, удаление грязи и фокус-стекинг, управляемый классом VideoProcessor. Для реализации алгоритмов отбора сфокусированных кадров и фокус-стекинга библиотека предоставляет интерфейсы FrameSelector и FocusStacker соответственно.

Реализованные в библиотеке алгоритмы были апробированы на мобильном приложении, написанном Владимиром Кутуевым. Для них была проведена верификация результата. Получившиеся в результате работы мобильного приложения изображения визуально-эквивалентны результатам полученным с помощью прототипа системы автоматического отбора сфокусированных кадров.

## Заключение

В ходе данной работы было проведено исследование существующих инструментов для получения полного изображения образца, исследуемого с помощью оптического микроскопа. В ходе изучения данной темы выявлено, что современные смартфоны способны захватывать изображение с микроскопа без потери данных и проводить его цифровую обработку. Также изучены различные алгоритмы фокус-стекинга и отбора сфокусированных кадров. Выявлено, что не существует полностью автоматического инструмента, реализующего все подходы и позволяющего работать с большим по размеру входному набору кадров.

Создан прототип, реализующий основные алгоритмы фокус-стекинга и операторы сфокусированности, которые подходят для использования в области микроскопии. В рамках прототипа был предложен и реализован алгоритм отбора сфокусированных кадров, основанный на анализе частей изображения. Затем, на базе результатов данного алгоритма отбора стека был подобран наиболее подходящий алгоритм фокус-стекинга. Данный алгоритм был внедрен в мобильную кроссплатформенную библиотеку для цифровой обработки в микроскопии. Его применение было продемонстрировано с помощью мобильного приложения на базе ОС Android.

## Список литературы

- [1] Adobe. Adobe Photoshop // официальный сайт. — Режим доступа: <https://www.adobe.com/ru/products/photoshop.html> (дата обращения: 20.11.2019).
- [2] Automated focusing in bright-field microscopy for tuberculosis detection / O.A. OSIBOTE, R. DENDERE, S. KRISHNAN, T.S. DOUGLAS // Journal of Microscopy. — 2010. — Vol. 240, no. 2. — P. 155–163. — Access mode: <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2818.2010.03389.x>.
- [3] Blurring-Effect-Free CNN Network of Structural Edge for Focus Stacking / G. Wang, W. Li, X. Chen et al. // IEEE Access. — 2019. — Vol. 7. — P. 167672–167682.
- [4] Charles McGuinness. focusstack // официальный репозиторий. — Access mode: <https://github.com/cmguinness/focusstack> (online; accessed: 20.11.2019).
- [5] Complex wavelets for extended depth-of-field: A new method for the fusion of multichannel microscopy images / Brigitte Forster-Heinlein, Dimitri Van De Ville, Jesse Berent et al. // Microscopy research and technique. — 2004. — 09. — Vol. 65. — P. 33–42.
- [6] Dr. Robert Berdan. Focus Stacking comparing Photoshop, Helicon Focus and Zerene // The Canadian Nature Photographer. — Access mode: [https://www.canadiannaturephotographer.com/rberdan\\_focus\\_stacking.html](https://www.canadiannaturephotographer.com/rberdan_focus_stacking.html) (online; accessed: 20.11.2019).
- [7] Dropbox. Djinni // Github repository. — Access mode: <https://github.com/dropbox/djinni> (online; accessed: 20.11.2019).
- [8] EPFL. Extended Depth Of Field plugin for ImageJ // официальный сайт. — Access mode: <http://bigwww.epfl.ch/demo/edf> (online; accessed: 20.11.2019).

- [9] Extended depth-of-field via focus stacking and graph cuts / C. Zhang, J. Bastian, C. Shen et al. // 2013 IEEE International Conference on Image Processing. — 2013. — Sep. — P. 1272–1276.
- [10] FocusALL: Focal Stacking of Microscopic Images Using Modified Harris Corner Response Measure / M. S. Sigdel, M. Sigdel, S. Ding et al. // IEEE/ACM Transactions on Computational Biology and Bioinformatics. — 2016. — March. — Vol. 13, no. 2. — P. 326–340.
- [11] Helicon Soft Ltd. Helicon Focus // официальный сайт. — Access mode: <https://www.heliconsoft.com/heliconsoft-products/helicon-focus> (online; accessed: 20.11.2019).
- [12] Hilsenstein V. Robust Autofocusing for Automated Microscopy Imaging of Fluorescently Labelled Bacteria. — 2005. — Dec. — P. 15–15.
- [13] Hunter J. D. Matplotlib: A 2D graphics environment // Computing in Science & Engineering. — 2007. — Vol. 9, no. 3. — P. 90–95.
- [14] Iarosh Dmitrii. Датасет для тестирования разработанного алгоритма // Google Drive. — Режим доступа: <https://drive.google.com/drive/folders/1kNgog1TJw1F8HJY54aw7u3fnmt8ICpnD?usp=sharing> (дата обращения: 20.11.2019).
- [15] Iarosh Dmitrii. Результаты тестирования алгоритма отбора сфокусированных кадров на датасете // Google Table. — Режим доступа: <https://docs.google.com/spreadsheets/d/1kAx-YtBFEWz7pZVTvdlftwEYiDJbjVcyR-LEdcuSto0/edit?usp=sharing> (дата обращения: 20.11.2019).
- [16] Lina jean-marc. Complex Daubechies Wavelets: Filters Design and Applications. — 1997. — 11.
- [17] NIH. ImageJ // официальный сайт. — Access mode: <https://imagej.nih.gov/ij/index.html> (online; accessed: 20.11.2019).

- [18] NIH. Public datasets in microscopy // веб-сайт. — Access mode: [https://imagej.net/Public\\_data\\_sets](https://imagej.net/Public_data_sets) (online; accessed: 17.04.2020).
- [19] NumPy developers. NumPy // официальный сайт. — Access mode: <https://numpy.org> (online; accessed: 17.04.2020).
- [20] OpenCV team. OpenCV // официальный сайт. — Access mode: <https://opencv.org> (online; accessed: 17.04.2020).
- [21] Pertuz Said, Puig Domenec, García Miguel. Analysis of focus measure operators in shape-from-focus // Pattern Recognition. — 2012. — 11. — Vol. 46.
- [22] Piotr Gródek. FocusStacking // официальный репозиторий. — Access mode: <https://github.com/rAum/FocusStacking> (online; accessed: 20.11.2019).
- [23] Quantitative Imaging with a Mobile Phone Microscope / Arunan Skandarajah, Clay D. Reber, Neil A. Switz, Daniel A. Fletcher // PLOS ONE. — 2014. — 05. — Vol. 9, no. 5. — P. 1–12. — Access mode: <https://doi.org/10.1371/journal.pone.0096906>.
- [24] SciPy developers. SciPy // официальный сайт. — Access mode: <https://www.scipy.org> (online; accessed: 17.04.2020).
- [25] Serif (Europe) Ltd. Affinity Photo // официальный сайт. — Access mode: <https://affinity.serif.com/ru/photo> (online; accessed: 20.11.2019).
- [26] Zerene Systems LLC. Zerene Stacker // официальный сайт. — Access mode: <https://zerenesystems.com/cms/stacker> (online; accessed: 20.11.2019).