

УДК 004.932

Оценка временной эффективности реализаций алгоритма водораздела в программных библиотеках с открытым исходным кодом

Корнилов А. С., Сафонов И. В.

Постановка задачи: преобразование водораздела – один из алгоритмов, применяемых для сегментации цифровых изображений. В настоящее время существует множество программных реализаций библиотек с открытым исходным кодом, позволяющих выполнять сегментацию с помощью водораздела двумерных и трехмерных изображений. Данные реализации применяют для решения различных задач и затрачиваемое время является одним из наиболее значимых показателей. Зачастую на практике используется не самая оптимальная реализация. Одной из причин является отсутствие глубокого понимания реализованных алгоритмов, а также отсутствие информации о существующих альтернативных решениях. **Целью работы** является анализ алгоритмов водораздела, которые реализуются в различных библиотеках с открытым исходным кодом, а также их сравнение по критерию оперативности выполнения в интересах обоснования оптимизированных реализаций. **Результат:** описаны основные алгоритмы, которые на данный момент применяются в реализациях преобразования водораздела, представлена сравнительная оценка оперативности выполнения реализаций в библиотеках с открытым исходным кодом, имеющих интерфейс вызова из языка Python, выявлены наиболее оперативные на данный момент реализации для обработки 2D и 3D изображений.

Ключевые слова: преобразование водораздела, сегментация, временная эффективность, библиотеки с открытым исходным кодом.

Введение

Сегментация цифрового изображения – это процесс его разделения на ряд непересекающихся регионов для упрощения или изменения представления изображения с целью его дальнейшего анализа [1]. При этом пиксели одного региона имеют сходные характеристики. Сегментация является одной из важнейших задач, решаемых в ходе процессов распознавания и классификации образов [2], визуализации [3], сжатия изображений на основе объектов интереса [4] и т.д. [5].

Одним из наиболее распространённых алгоритмов сегментации в области обработки медицинских [6, 7] и материаловедческих [8] изображений является алгоритм водораздела (watershed), основанный на представлении изображения в оттенках серого в качестве топографического рельефа, затопляемого водой, где водоразделами являются линии, разделяющие области воды из разных бассейнов. После первого описания в [9] данный подход получил значительное разви-

Библиографическая ссылка на статью:

Корнилов А. С., Сафонов И. В. Оценка временной эффективности реализаций алгоритма водораздела в программных библиотеках с открытым исходным кодом // Системы управления, связи и безопасности. 2019. № 1. С. 123-140. DOI: 10.24411/2410-9916-2019-10107

Reference for citation:

Kornilov A. S., Safonov I. V. Time efficiency estimation of watershed algorithm implementations in open source libraries. *Systems of Control, Communication and Security*, 2019, no. 1, pp. 123-140. DOI: 10.24411/2410-9916-2019-10107 (in Russian).

тие [10]. Часто с помощью алгоритма водораздела производится обработка больших 3D изображений [11]. Например, изображения размером $2048 \times 2048 \times 1000$ и $4000 \times 4000 \times 2000$, полученные с помощью рентгеновской микротомографии для анализа минеральных частиц [12, 13]. Естественно, что при работе с подобными огромными 3D изображениями затрачиваемое на их анализ время достаточно велико. По этой причине появилось множество предложений по параллелизации алгоритма водораздела, однако значительного результата оказалось сложно достичь по причине его последовательной природы [10]. Поскольку задача снижения временных затрат касается не только трехмерных, но и двумерных изображений большого размера, в данной статье используется термин «элемент изображения» (ЭИ) для объединения терминов «пиксел» и «воксел».

На практике часто применяются или модифицируются реализации алгоритма водораздела из различных библиотек с открытым исходным кодом [14–17]. Однако, зачастую исследователи используют функции программного обеспечения (ПО) с открытым исходным кодом без глубокого понимания деталей и ограничений реализуемых алгоритмов. Также иногда у разработчиков нет информации об альтернативных решениях. Это может привести к ухудшению производительности в исследованиях, а также в разработанных программных инструментах. Данная статья нацелена на обзор существующих решений и представление сравнительной оценки их производительности. В статье рассматриваются реализации алгоритма сегментации водоразделом из следующих программных библиотек с открытым исходным кодом: Insight Segmentation and Registration Toolkit (ITK) [18], Open Source Computer Vision Library (OpenCV) [19], Mahotas [20], Mathematical Morphology library Image (Mamba) [21], scikit-image (skimage) [22] и Simple Morphological Image Library (SMIL) [23]. Для анализа времени обработки изображений данными реализациями и выявления наиболее оптимальной по критерию минимума времени сегментации представлено описание реализуемых ими алгоритмов и выполняется оценка времени их работы на ряде тестовых изображений, производится сравнение результатов оценки. Также существуют и другие библиотеки с открытым исходным кодом, в которых реализованы алгоритмы водораздела, например, ImageJ [24]. Однако в данной статье рассматриваются только те, которые предоставляют интерфейс вызова своих функций на языке Python.

Данная статья состоит из следующих разделов. В разделе 1 описаны основные подходы сегментации водоразделом. В разделе 2 описываются алгоритмы, применяющиеся на данный момент в программных библиотеках с открытым исходным кодом. В разделе 3 представлены результаты исследования зависимости времени исполнения от размеров изображений тестовой выборки, также представлено среднее время исполнения, затрачиваемое каждой реализацией на один элемент изображения. Завершается статья заключением о современном положении дел в области решения задач сегментации изображений с помощью реализаций алгоритма водораздела, представленных в библиотеках с открытым исходным кодом.

1. Основные принципы сегментации водоразделом

Как указано ранее, алгоритмы преобразования водораздела основаны на представлении изображения в виде топографического рельефа, где значение яркости каждого ЭИ характеризует его возвышенность в данной точке. Иногда сегментации подвергают не само изображение, а карту расстояний (distance map), полученную из результата его бинаризации [25, 26]. Таким образом, данный рельеф состоит из «низинных долин» (минимумов), «высотных хребтов» (линий водоразделов) и склонов (водосборных бассейнов). Также используется понятие «плато» (область с одинаковой высотой элементов). Основной задачей в данном способе сегментации является определение расположения всех водосборных бассейнов и/или линий водоразделов (ВЛ), так как в данном случае каждый водосборный бассейн считается отдельным сегментом изображения. На основе такого представления существует два основных подхода к нахождению водоразделов: симуляция процесса наводнения (by flooding) и процесса дождя (by raining) [26, 27].

Для задания «соседей» в изображении, используются структурные элементы. Для 2D изображений, как правило, используется отношение восьмисвязности (связность «по квадрату»), при котором каждый ЭИ имеет восемь «соседей», или четырёхсвязности (связность «по кресту»), где каждый ЭИ имеет четыре «соседа». Аналогичным образом задаются связность и соседство для 3D изображений: шестисвязность (связность «по сторонам куба»), где ЭИ имеет шесть «соседей»; 26-связность («по кубу»); 18-связность («по шару»).

1.1 Водораздел наводнением

В изначально предложенном подходе Vincent и Soille [28] реализуется следующее: «продельваются отверстия» в каждом минимуме, после чего весь рельеф начинают «погружать в воду». При этом, начиная с минимумов наименьшей возвышенности, вода постепенно заполняет все водосборные бассейны. В местах же, где встречается вода из разных бассейнов, возводятся водоразделы. Когда вода достигает максимального пика рельефа, процесс завершается, и в результате каждый водосборный бассейн становится охваченным ВЛ.

Для симуляции данного процесса наиболее часто применяются алгоритмы, предложенные Veucher и Meyer [26], первый из которых состоит из следующих шагов:

- A1. Производится нахождение локальных минимумов на изображении. Каждому из них присваивается уникальная метка.
- A2. Симулируется процесс наводнения, использующий очередь с приоритетом, которая состоит из N очередей, где N – это диапазон возможных значений элементов изображения (например, уровня серого цвета). При последующем её заполнении ЭИ со значением h помещаются в соответствующую очередь с номером h .
 - a) Изначально очередь с приоритетом заполняется локальными минимумами ЭИ.

- б) Очередь с приоритетом сканируется от меньших значений h к большему. Выбирается ЭИ из первой непустой очереди. Если же все очереди из очереди с приоритетом оказываются пустыми, то алгоритм завершается.
- в) Выбранный ЭИ удаляется из очереди, при этом его метка распространяется на всех непомиченных «соседей» ЭИ.
- г) «Соседи» ЭИ, помеченные на предыдущем шаге, помещаются в очередь с приоритетом. Производится переход к шагу А2б.

Результат в данном случае не будет содержать ВЛ. Если же их наличие требуется, то применяется второй подход [29]:

Б1. Аналогично шагу А1 первого алгоритма.

Б2. Используется аналогичная очередь с приоритетом, описанная в шаге А2 первого алгоритма. Кроме того, каждый ЭИ необходимо дополнительно помечать в зависимости от того, был ли он уже помещён в данную очередь или ещё нет (дополнительная метка).

- а) ЭИ, которые имеют уникальную метку, помечаются дополнительной меткой.
- б) В очередь с приоритетом заносятся ЭИ, которые имеют помеченных «соседей». Также данные элементы отмечаются дополнительной меткой.
- в) Аналогично шагу А2б первого алгоритма.
- г) Выбранный ЭИ удаляется из очереди, если его соседние помеченные ЭИ отмечены одинаковой меткой, то и он сам отмечается ей. Если же он является «соседом» элементов с различными метками, то он получает специальную метку, означающую, что он является элементом водораздела.
- д) «Соседи» данного ЭИ, которые не были ещё отмечены дополнительной меткой, помещаются в очередь с приоритетом. Производится переход к шагу Б2в.

В результате работы данного алгоритма ЭИ, которые соответствуют ВЛ, будут иметь специальную метку, поставленную на шаге Б2г, метки остальных ЭИ указывают на то, к какому региону они относятся.

1.2 Водораздел симуляцией дождя

Другой подход, основанный на компонентах связности и симулирующий процесс дождя, был предложен Bieniek и Moga [27]. После падения на поверхность рельефа изображения капля дождя стекает вниз вдоль склона до соответствующей долины. Маршрут, по которому стекает капля, является компонентой связности, а также самым крутым путём между точкой падения и долиной. Все связные компоненты, которые ведут к одной и той же долине, образуют один водосборный бассейн. Для изображений, которые содержат только минимальные плато, данный алгоритм требует только трёх полных проходов изображения. Однако, так как обычно изображения содержат и неминимальные плато, то требуется дополнительный шаг с ещё одним полным проходом, а также использование очереди. Таким образом, алгоритм производит разметку всех водо-

сборных бассейнов при помощи только четырёх полных проходов по изображению.

Классический алгоритм состоит из следующих шагов:

- В1. Проход 1. Производится маркировка каждого элемента изображения расположением более низкого соседнего с ним элемента, имеющего наименьшее значение (обычно реализуемое при помощи указателей). Если же таких элементов по соседству нет, то текущий маркируется как элемент плато.
- В2. Устранение неминимальных плато.
 - а) Проход 2. У каждого элемента, который отмечен как плато, проверяются «соседи». Если среди них найден такой, который не отмечен как плато и имеет то же значение, то данный «сосед» помещается в очередь, а проверка «соседей» у данного элемента завершается.
 - б) Пока очередь «соседей» не пуста, из неё извлекаются элементы, у каждого из которых просматриваются «соседи», которые отмечены как плато. Данным «соседям» проставляются маркеры, указывающие расположение извлекаемого элемента, а затем они сами помещаются в очередь.
- В3. Проход 3. Маркировка элементов минимальных плато репрезентативными значениями.
 - а) Каждому элементу, который отмечен как плато, сначала проставляется метка, указывающая на самого себя.
 - б) Поочередно берутся соседи текущего элемента, которые уже были пройдены и имеют такое же значение.
 - в) Для текущего элемента и данного «соседа» производится процедура сокращения компонент связности. Она заключается в прохождении пути, используя метки в качестве указателей движения, от заданного элемента до того, который отмечен меткой, указывающей на него же, с последующим присваиванием всем меткам элементов пути значения, которое имеет метка конечного элемента на нём.
 - г) После этой процедуры конечные элементы данных путей маркируются минимальным значением из их меток.
- В4. Проход 4. Для каждого элемента изображения производится процедура сокращения компонент связности, описанная в пункте В3в. А сами элементы маркируются метками конечных элементов на пути, пройденном процедурой сокращения (своим репрезентативным значением). На этом алгоритм завершается.

Данный алгоритм не зависит от диапазона возможных значений элементов изображения, в нём используются простые структуры данных, при его применении затрачивается меньше время на работу по сравнению со случаями использования алгоритмов, основанных на симуляции затопления. Алгоритм не изменяет принцип сегментации водораздела, результат на выходе такой же, как и при использовании любого традиционного алгоритма, который не создает ВЛ [27].

Несмотря на ряд преимуществ реализаций алгоритма водораздела на основе симуляции дождя над реализациями на основе наводнения, описанных в [27], они также обладают рядом существенных недостатков. В основном они связаны с трудностями, вызванными неминимальными плато – невозможностью их разбиения на несколько частей при использовании подхода на основе симуляции дождя, что, возможно, лучше бы соответствовало физической природе дождя (например, высотный хребет, который должен быть разделён посередине). Также возникает неоднозначность при маркировке нескольких соседних регионов с равной высотой. Реализации, основанные на принципе наводнения, позволяют избежать этих затруднений [30].

1.3 Чрезмерная сегментация

При применении к реальным изображениям преобразование водораздела часто приводит к чрезмерной сегментации (*oversegmentation*). Это происходит по причине того, что в подходах, описанных выше, исходными бассейнами, на основе которых производится водораздел, является каждый локальный минимум изображения. Однако при этом каждый из маркированных элементов, а, следовательно, и каждый из локальных минимумов становится центром водосборного бассейна, но не все из данных минимумов являются одинаково значимыми. Так, например, некоторые из них формируются за счёт шума, незначительных колебаний яркости и второстепенных структур изображения [25]. Пример такого явления представлен на рис. 1 и рис. 2. Решение в данном случае заключается в том, что необходимо отметить только значимые части исходного изображения и ограничить процесс сегментации, чтобы вырастить только одну область вокруг каждого маркера. Данный метод называется маркерным водоразделом (*marker-controlled watershed*). Возможны два подхода: во-первых, преобразовать изображение таким образом, что региональные минимумы будут располагаться только в выбранных местах маркеров (обычно производится при помощи морфологической реконструкции на основе дилатации), а затем применить классический алгоритм водораздела [11, 26]; во-вторых, применить постобработку чрезмерно сегментированного изображения, в результате которой происходит объединение каждой немаркированной области с ближайшей маркированной согласно установленной метрике расстояния [31, 32].



Рис. 1. Пример сегментируемого изображения

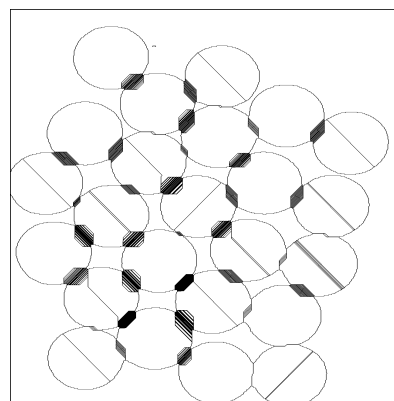


Рис. 2. Результат сегментации изображения с рис. 1

2. Реализации программных библиотек

В данной статье рассматриваются реализации алгоритмов, применяемых в библиотеках: ИТК, OpenCV, Mahotas, Mamba, Skimage и SMIL. Существенным фактором является активность разработки и обновления библиотек. Так, рассматриваемые библиотеки на данный момент можно ранжировать в следующем порядке по количеству участников разработки (по убыванию): OpenCV, Skimage, ИТК, Mahotas, SMIL, Mamba и по величине отношения времени существования библиотеки к количеству её релизов (по возрастанию): ИТК, OpenCV, Skimage, Mahotas, SMIL, Mamba. Данные количественные характеристики, а также сколько месяцев прошло со времени последнего обновления представлены в таблице 1.

Таблица 1 – Характеристики рассматриваемых библиотек

Библиотека	Участников разработки	Отношение времени существования (дней) к количеству релизов	Последнее обновление, месяцев назад
OpenCV	971	31	0
ИТК	209	26	0
Skimage	300	54	1
Mahotas	20	66	5
SMIL	5	154	9
Mamba	2	424	14

В данной статье производится сравнение времени сегментации реализациями данных библиотек без влияния шага получения начальных маркеров. В таблице 2 указана информация о рассматриваемых реализациях.

Таблица 2 – Исследуемые реализации алгоритма водораздела

Библиотека	Версия	Язык исходного кода реализации	Python объект
ИТК	4.13.0	C++	MorphologicalWatershedFromMarkersImageFilter
OpenCV	3.4.1	C++	watershed
Mahotas	1.4.4	C++	cwatershed
Mamba	2.0.2	C	basinSegment, watershedSegment
Skimage	0.14	Cython	morphology.watershed
SMIL	0.9	C++	basins, watershed

В библиотеках ИТК, Mahotas, Mamba, Skimage и SMIL реализованы алгоритмы водораздела с конструированием ВЛ и без их конструирования, описанные в [30, 33]. В библиотеке OpenCV реализован модифицированный для обработки цветных изображений алгоритм, описание которого представлено далее. Реализации всех библиотек, кроме OpenCV, позволяют помимо обработки 2D изображений также использовать трёхмерные типы связности и работать и с объёмными 3D изображениями.

Несмотря на то, что большинство реализаций основано на одних и тех же алгоритмах, имеющих асимптотическую оценку временной сложности $O(n)$ относительно числа элементов изображения n , их временные затраты также силь-

но зависят от языка реализации, применяемых программных оптимизаций и других параметров. В библиотеках ИТК, Mamba, SMIL оба алгоритма (как с построением ВЛ, так и без) реализованы отдельно, в то время как в Mahotas и Skimage реализации объединены в одну. Кроме того, во всех реализациях, кроме OpenCV, предоставляется возможность выбора структурного элемента для задания связности элементов изображения. В библиотеке Skimage, кроме того, присутствует реализация компактного водораздела [34] – алгоритма, который позволяет варьировать значение параметра компактности, влияющего на размеры результирующих сегментов, что позволяет уменьшить их размеры и сократить время вычислений.

В результате работы алгоритмов на выходе получают итоговую маркировку, а также маркировку ВЛ в случае их конструирования. Некоторые реализации объединяют маркировки в одну и хранят их в одном итоговом массиве (ИТК, OpenCV, Skimage), а некоторые – отдельно в двух (Mahotas, Mamba, SMIL).

В библиотеке OpenCV реализован алгоритм, предложенный Meyer [35]. Он является расширением алгоритма водораздела наводнением с конструированием ВЛ, описанного в разделе 1, для сегментации цветных RGB изображений и вносит следующие изменения.

1. Мерой разницы между элементом и его соседями, а, соответственно, и мерой приоритета в очереди, выбирается максимальная разница из всех цветовых каналов. Если имеются два элемента x и y , имеющие соответствующие цвета (r_x, g_x, b_x) и (r_y, g_y, b_y) , то разница рассчитывается по следующей формуле:
$$d(x, y) = \text{Max}(|r_x - r_y|, |g_x - g_y|, |b_x - b_y|). \quad (1)$$
2. В соответствии с этой мерой размер очереди с приоритетом равен максимально возможной разнице между цветовыми каналами.

3. Оценка времени сегментации

Для оценки затрачиваемого на сегментацию рассматриваемыми реализациями времени применялся язык Python, поскольку все рассматриваемые библиотеки предоставляют интерфейс для вызова своих функций на данном языке. Оценка времени работы производилась на персональном компьютере с центральным процессором Intel Core i7-2600, имеющим 8 ядер, 8 Гб оперативной памяти и работающим на тактовой частоте 3.40 ГГц под управлением операционной системой GNU/Linux версии 4.17.3 и разрядностью 64 бита. Библиотеки собраны при помощи GCC 8.1.1 и запускались с помощью Python 3.6.4.

Замеры времени осуществлялись без учёта влияния шагов генерации исходных данных алгоритмов. В качестве входных данных использовалась карта расстояний, полученная из результата бинаризации исходного изображения и сгенерированная на её основе начальная маркировка. Структурными элементами связности выбирались элементы с полной связностью: квадратный, связывающий 8 соседних элементов для 2D изображений и кубический, связывающий 26 соседних элементов для 3D соответственно. Следует отметить, что реализация OpenCV поддерживает только квадратный структурный элемент. Для

каждой библиотеки, кроме OpenCV, производилась оценка времени для случаев без генерации ВЛ и с ними.

3.1. Тестовые изображения

Оценка затрачиваемого на сегментацию времени в зависимости от размера изображения выполнялись для ряда 2D изображений (board, cells, coins, fruits и maze) и сгенерированного объемного 3D изображения (balls), представленных на рис. 3-8. Данная тестовая выборка содержит: изображения объектов со схожими размерами, как с относительно малым количеством объектов (board, coins, balls), так и с относительно большим (cells); стандартное изображение (fruits) и искусственное изображение со спиральной структурой (maze).

Каждое из этих изображений масштабировалось с использованием интерполяции нулевого порядка, а затем производилась оценка времени его сегментации. Для устранения влияния случайных факторов на оценку времени работы и получения более объективного результата для каждого изображения замеры производились 40 раз и вычислялось среднее значение.

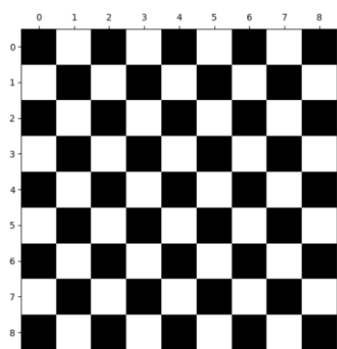


Рис. 3. Board

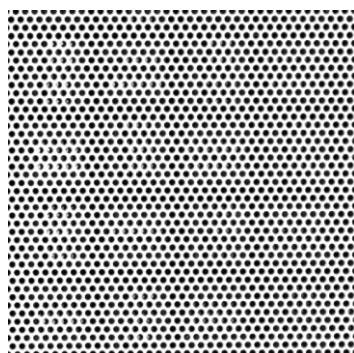


Рис. 4. Cells

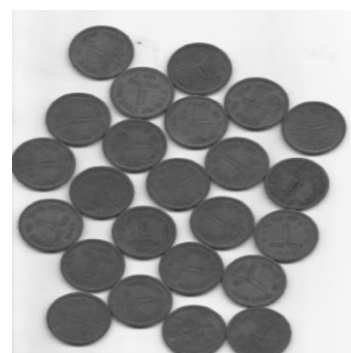


Рис. 5. Coins

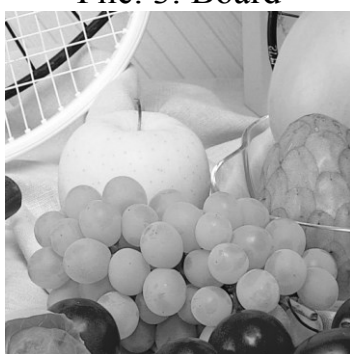


Рис. 6. Fruits

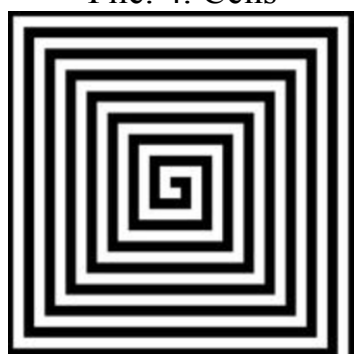


Рис. 7. Maze

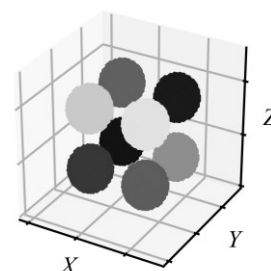


Рис. 8. Balls

3.2. Результаты оценки времени сегментации

В результате масштабирования получено 6 рядов, содержащих изображения с меняющимися размерами. Для каждого из этих изображений осуществлялась оценка времени, затрачиваемого на их сегментацию реализациями водораздела исследуемых библиотек. На основе результатов оценок получены зависимости временных затрат на сегментацию исследуемыми алгоритмами от размеров тестовых изображений и для различных видов связности. Так, график данной зависимости для 26-связности и 3D изображений с реализациями без конструирования ВЛ представлен на рис. 9, а с конструированием – на рис. 10.

Для 2D изображений зависимости имеют схожий характер, при уменьшении самих значений в 1,5–3 раза.

По зависимостям, представленным на рис. 9 и рис. 10, можно увидеть, что асимптотическая вычислительная сложность каждого представленного алгоритма действительно является $O(n)$ относительно числа ЭИ, поскольку в данном логарифмическом масштабе графики представляют логарифмическую зависимость, а размер изображений изменяется линейно.

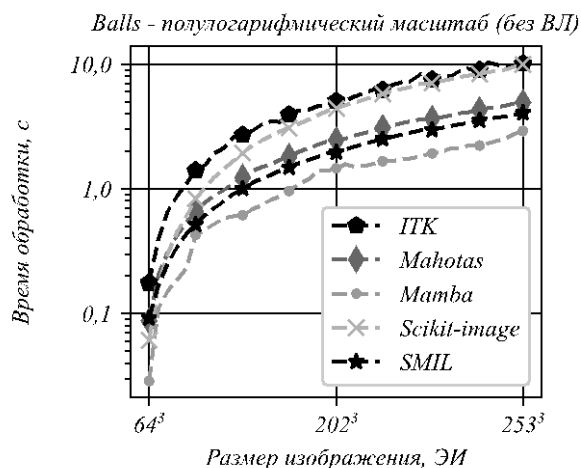


Рис. 9. Результаты оценки времени сегментации для 26-связности и 3D изображений без ВЛ

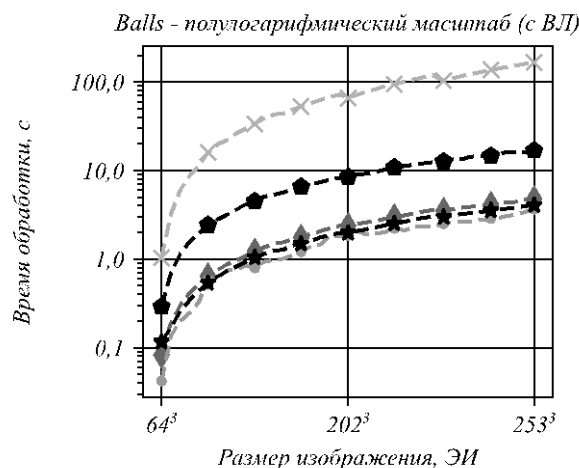


Рис. 10. Результаты оценки времени сегментации для 26-связности и 3D изображений с ВЛ

Для возможности численного сравнения рассчитано среднее по всем размерам изображений время обработки одного ЭИ каждой реализацией. Так, в таблице 3 представлены значения среднего времени сегментации в наносекундах на один ЭИ по всем рассматриваемым размерам изображений для реализаций без конструирования ВЛ, а в таблице 4 аналогично представлены с их конструированием, сегментированные области рассматривались как восьмисвязные для 2D изображений и как 26-связные для 3D.

Таблица 3 – Среднее время сегментации без ВЛ, 8-связность для 2D и 26-связность для 3D изображений, (нс/ЭИ)

Реализация	Время обработки изображений, нс/ЭИ						
	2D, 8-связные области						3D, 26-связные balls
	board	cells	coins	fruits	maze	Среднее	
ИТК	201	246	210	214	193	213	630
Mahotas	165	237	192	194	161	190	293
Mamba	49	115	75	75	40	71	160
Skimage	271	466	305	312	252	321	499
SMIL	102	163	140	141	116	132	245

Таблица 4 – Среднее время сегментации с ВЛ, 8-связность для 2D и 26-связность для 3D изображений, (нс/ЭИ)

Реализация	Время обработки изображений, нс/ЭИ						
	2D, 8-связные области						3D, 26-связные
	board	cells	coins	fruits	maze	Среднее	balls
ITK	306	380	328	332	292	328	1053
Mahotas	167	249	195	202	163	195	290
Mamba	64	141	94	94	57	90	207
Skimage	1225	2740	1251	1326	1439	1596	8392
SMIL	112	182	127	132	140	139	250
OpenCV	36	50	34	38	31	38	—

Из представленных графиков и таблиц можно увидеть, что реализации библиотек OpenCV и Mamba показывают наилучшие результаты по оперативности выполнения при обработке 2D и 3D изображений соответственно. Так, реализация OpenCV показывает наилучшие результаты в данном эксперименте при сегментации 2D изображений с конструированием водораздела, а реализация Mamba вторая – в тех случаях, когда первая неприменима. Время обработки с построением ВЛ всегда выше времени без их построения. Из данного факта можно сделать вывод о том, что в случаях, когда не требуется конструирование ВЛ, предпочтительнее использовать соответствующие реализации. В случае использования не самой оптимизированной реализации время обработки будет выше: для 2D изображений в 1,8–4,5 раза для случая без ВЛ и в 2,4–42 с ВЛ, для 3D соответственно в 1,5–3,9 и в 1,2–41, в зависимости от выбранной библиотеки.

Следует заметить, что время сегментации также зависит от выбора структурного элемента связности, количества исходных маркеров и размеров результирующих сегментов. Так, увеличение количества исходных маркеров или величины итоговых сегментов в 2 раза ведёт к увеличению затрачиваемого времени также приблизительно в 2 раза, т.е. оно возрастает линейно. Для демонстрации зависимости времени сегментации от выбора структурного элемента проведена аналогичная предыдущей оценка, но с использованием четырёхсвязности для 2D и 6-связности для 3D изображений. Использовались библиотеки, которые позволяют задать соответствующие структурные элементы: ITK, Mahotas, Skimage, SMIL. Так, в таблице 5 представлены значения среднего времени обработки в наносекундах на один ЭИ по всем рассматриваемым размерам изображений для реализаций без конструирования ВЛ, а в таблице 6 аналогично представлены с их конструированием.

Таблица 5 – Среднее время сегментации без ВЛ, 4-связность для 2D и 6-связность для 3D изображений, (нс/ЭИ)

Реализация	Время обработки изображений, нс/ЭИ						
	2D, 4-связные						3D, 6-связные
	board	cells	coins	fruits	maze	Среднее	balls
ITK	197	265	192	201	161	203	323
Mahotas	190	293	214	217	182	219	246
Skimage	298	672	348	359	299	395	537
SMIL	111	182	130	130	131	137	158

Таблица 6 – Среднее время сегментации с ВЛ, 4-связность для 2D и 6-связность для 3D изображений, (нс/ЭИ)

Реализация	Время обработки изображений, нс/ЭИ						
	2D, 4-связные						3D, 6-связные
	board	cells	coins	fruits	maze	Среднее	balls
ITK	291	378	283	288	226	293	448
Mahotas	195	301	218	223	181	224	252
Skimage	620	1389	675	704	659	808	1714
SMIL	121	198	132	134	142	145	182

Сравнение таблицы 3 с таблицей 5 и таблицы 4 с таблицей 6 позволяет заметить, что применение структурного элемента с меньшим числом соседей, не во всех случаях приводит к снижению времени сегментации. Так, для 2D и 3D изображений при сегментации без ВЛ реализацией библиотеки Skimage время обработки, наоборот, увеличилось. Также, при использовании реализации Mahotas для 2D изображений время увеличилось, как без ВЛ, так и с ВЛ. В остальных случаях время сегментации сократилось или осталось практически таким же. Особенно снижение заметно при обработке 3D изображений, а также для случая сегментации с генерацией ВЛ.

Заключение

В данной статье представлены результаты анализа реализаций алгоритмов сегментации водоразделом, применяемых в различных библиотеках с открытым исходным кодом. Так или иначе, во всех рассмотренных библиотеках представлены алгоритмы, предложенные Beucher и Meyer [26]. Кроме того, произведена оценка времени сегментации данными реализациями изображений из тестовой выборки. Исходя из полученных результатов, оптимальной по затрачиваемому времени реализацией на данный момент для сегментации 2D изображений с конструированием водораздельных линий является реализация из библиотеки OpenCV. Реализация из библиотеки Mamba является самой быстрой для сегментации 2D изображений без конструирования водораздельных линий и 3D изображений, как с конструированием, так и без. В случае применения реализаций других библиотек, затрачиваемое время повышалось в 1,2–42 раза в зависимости от необходимости построения водораздельных линий, количества исходных маркеров и величины итоговых сегментов изображения. Так, например, увеличение количества исходных маркеров или величины итоговых сегментов в 2 раза ведёт к увеличению затрачиваемого времени приблизительно в 2 раза.

Для выявления влияния выбора структурного элемента на время сегментации одинаковые изображения обрабатывались с использованием 4-х и 8-и связности для 2D и 18-и и 26-и связности для 3D. В результате использования меньшего числа соседей снижение времени сегментации наблюдалось не для всех рассматриваемых реализаций. Так, для реализаций Mahotas и Skimage время, наоборот, увеличилось. Для остальных реализаций время сегментации снизилось или практически не изменилось. В случае 3D изображений разница

во времени обработки в зависимости от используемой связности достаточно существенна.

Следует также отметить, что одной из важных характеристик библиотек с открытым исходным кодом является активность их разработки и обновления под новейшие версии Python и операционные системы, так как это влияет на удобство, а иногда и возможность их использования с последними версиями ПО, оперативность исправления обнаруженных ошибок, добавления новых возможностей и тому подобное. Рассмотрены следующие характеристики каждой из библиотек: количество участников разработки, отношение времени существования библиотеки к количеству релизов и время с даты последнего обновления. На данный момент, исходя из перечисленных характеристик, наиболее активно ведётся разработка библиотек OpenCV, ИТК и Skimage.

Литература

1. Haralick R. M., Shapiro L. G. Image segmentation techniques // Computer Vision, Graphics, and Image Processing. 1985. Vol. 29. № 1. P. 100-132.
2. Suetens P., Fua P., Hanson A. J. Computational Strategies for Object Recognition // ACM Computing Surveys. 1992. Vol. 24. № 1. P. 5-62.
3. McAuliffe M. J., Lalonde F. M., McGarry D., Gandler W., Csaky K., Trus B. L. Medical Image Processing, Analysis and Visualization in clinical research // Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001. – Bethesda, Maryland, 2001. – P. 381-386.
4. Hsu W.-Y. Segmentation-based compression: New frontiers of telemedicine in telecommunication // Telematics and Informatics. 2015. Vol. 32. № 3. P. 475-485.
5. Myasnikov E. V. Hyperspectral image segmentation using dimensionality reduction and classical segmentation approaches. // Computer Optics. 2017. Vol. 41. № 4. P. 564-572.
6. Pham D. L., Xu C., Prince J. L. Current Methods in Medical Image Segmentation // Annual Review of Biomedical Engineering. 2000. Vol. 2. № 1. P. 315-337.
7. Садыков С. С., Буланова Ю. А., Захарова Е. А. Компьютерная диагностика новообразований на маммографических снимках // Компьютерная оптика. 2014. Т. 38. № 1. С. 131-138.
8. Waggoner J., Zhou Y., Simmons J., Graef M. D., Wang S. 3D Materials Image Segmentation by 2D Propagation: A Graph-Cut Approach Considering Homomorphism // IEEE Transactions on Image Processing. 2013. Vol. 22. № 12. P. 5282-5293.
9. Digabel H., Lantuéjoul C. Iterative algorithms // Actes du Second Symposium Européen d'Analyse Quantitative des Microstructures en Sciences des Matériaux, Biologie et Médecine. – Caen, 1977. – P. 85-99.
10. Roerdink J. B., Meijster A. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies // Fundamenta Informaticae. 2000. Vol. 41. № 1,2. P. 187-228.

11. Atta-Fosu T., Guo W., Jeter D., Mizutani C. M., Stopczynski N., Sousa-Neves R. 3D Clumped Cell Segmentation Using Curvature Based Seeded Watershed // *Journal of Imaging*. 2016. Vol. 2. № 4. P. 31.
12. Alvaro V., Lin C. L., Miller J. D. Watershed Functions Applied to a 3D Image Segmentation Problem for the Analysis of Packed Particle Beds // *Particle & Particle Systems Characterization*. 2006. Vol. 23. № 3-4. P. 237-245.
13. Yakimchuk I. V., Safonov I. V., Serkova E. P., Evstefeeva V. Y., Korobkov D. A. Ceramic Proppant Microstructure Characterization by X-Ray Microtomography // *Bruker Micro-CT User Meeting*. – Ghent, 2018. – P. 17-23.
14. Beare R., Chen J., Adamson C. L., Silk T., Thompson D. K., Yang J. Y. M., Anderson V. A., Seal M. L., Wood A. G. Brain extraction using the watershed transform from markers // *Frontiers in Neuroinformatics*. 2013. Vol. 7. P. 32.
15. Gostick J. T. Versatile and efficient pore network extraction method using marker-based watershed segmentation // *Physical Review E*. 2017. Vol. 96. P. 023307.
16. Gouillart E. Analyzing microtomography data with Python and the scikit-image library // *Advanced Structural and Chemical Imaging*. 2016. Vol. 2 № 1. P. 18.
17. Mashburn D. N., Lynch H. E., Ma X., Hutson M. S. Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues // *Cytometry Part A*. 2012. Vol. 81A № 5. P. 409-418.
18. Johnson H. J., McCormick M. M., Ibáñez L. *The ITK software guide Book 1: Introduction and Development Guidelines Fourth Edition Updated for ITK version 4.7*. – Kitware, Inc. 2015. – 248 p.
19. Bradski, G. *The OpenCV Library* // Dr. Dobb's journal of software tools. 2000. Vol. 25. № 11. P. 122-125.
20. Coelho L. P. Mahotas: Open source software for scriptable computer vision // *Journal of Open Research Software*. 2013. Vol. 1. P. e3.
21. Beucher N., Beucher S. *Mamba Image User Manual* // Mamba library website [Электронный ресурс]. 23.09.2017. – URL: <http://www.mamba-image.org/docs/2.0/mamba-um.pdf> (дата обращения 20.10.2018).
22. van der Walt S., Schönberger J. L., Nunez-Iglesias J., Boulogne F., Warner J. D., Yager N., Gouillart E., Yu T. scikit-image: image processing in Python // *PeerJ*. 2014. Vol. 2. P. e453.
23. *Simple Morphological Image Library* [Электронный ресурс]. 2011. – URL: <http://smil.cmm.mines-paristech.fr> (дата обращения 20.10.2018).
24. Rueden C. T., Schindelin J., Hiner M. C., DeZonia B. E., Walter A. E., Arena E. T., Eliceiri K. W. ImageJ2: ImageJ for the next generation of scientific image data. // *BMC Bioinform*. 2017. Vol. 18. № 1. P. 529.
25. Safonov I. V., Mavrin G. N., Kryzhanovsky K. A. Segmentation of Convex Cells with Partially Undefined Edges // *Pattern Recognition and Image Analysis*. 2008. Vol. 18. № 1. P. 112-117.
26. Beucher S., Meyer F. The morphological approach to segmentation: The watershed transformation // *Mathematical Morphology in Image Processing*. 1993. Vol. 34. P. 452-464.

27. Bieniek A., Moga A. An efficient watershed algorithm based on connected components // *Pattern Recognition*. 2000. Vol. 33. № 6. P. 907-916.
28. Vincent L., Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1991. Vol. 13. № 6. P. 583-598.
29. Meyer F. Un algorithme optimal de ligne de partage des eaux // *Proceedings 8th Congress AFCET*. Vol. 2. – Lyon-Villeurbanne, France, 1992. – P. 847-859.
30. Beare R., Lehmann G. The watershed transform in ITK - discussion and new developments // *The Insight Journal*. 2006. P. 24.
31. Meyer F. Minimum Spanning Forests for Morphological Segmentation // *Mathematical Morphology and Its Applications to Image Processing*. 1994. P. 77-84.
32. Moga A. N., Gabbouj M. Parallel Marker-Based Image Segmentation with Watershed Transformation // *Journal of Parallel and Distributed Computing*. 1998. Vol. 51. № 1. P. 27-45.
33. Beucher N., Beucher S. Hierarchical Queues: general description and implementation in MAMBA Image library. // *The open archive HAL [Электронный ресурс]*. 2011. – URL: <https://hal-mines-paristech.archives-ouvertes.fr/hal-00835024/document> (дата обращения 20.10.2018).
34. Neubert P., Protzel P. Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms // *2014 22nd International Conference on Pattern Recognition*. – Stockholm, 2014. – P. 996-1001.
35. Meyer F. Color image segmentation // *1992 International Conference on Image Processing and its Applications*. – Maastricht, Netherlands, 1992. – P. 303-306.

References

1. Haralick R. M., Shapiro L. G. Image segmentation techniques. *Computer Vision, Graphics, and Image Processing*, 1985, vol. 29, no. 1, pp. 100-132. doi: 10.1016/S0734-189X(85)90153-7
2. Suetens P, Fua P, Hanson A. J. Computational Strategies for Object Recognition. *ACM Computing Surveys*, 1992, vol. 24, no. 1, pp. 5-62. doi: 10.1145/128762.128763
3. McAuliffe M. J., Lalonde F. M., McGarry D., Gandler W., Csaky K., Trus B. L. Medical Image Processing, Analysis and Visualization in clinical research. *Proceedings 14th IEEE Symposium on Computer-Based Medical Systems. CBMS 2001*. Bethesda, Maryland, 2001, pp. 381-386. doi: 10.1109/CBMS.2001.941749
4. Hsu W. Y. Segmentation-based compression: New frontiers of telemedicine in telecommunication. *Telematics and Informatics*, 2015, vol. 32, no. 3, pp. 475-485. doi: 10.1016/j.tele.2014.11.003
5. Myasnikov E. V. Hyperspectral image segmentation using dimensionality reduction and classical segmentation approaches. *Computer Optics*, 2017, vol. 41, no. 4, pp. 564-572. doi: 10.18287/2412-6179-2017-41-4-564-572

6. Pham D. L., Xu C., Prince J. L. Current Methods in Medical Image Segmentation. *Annual Review of Biomedical Engineering*, 2000, vol. 2, no. 1, pp. 315-337. doi: 10.1146/annurev.bioeng.2.1.315

7. Sadykov S. S., Bulanova Yu. A., Zakharova E. A. Komp'yuternaya diagnostika novoobrazovaniy na mammograficheskikh snimkah [Computer diagnosis of tumors in mammograms]. *Computer Optics*, 2014, vol. 38, no. 1, pp. 131-138 (in Russian).

8. Waggoner J., Zhou Y., Simmons J., Graef M. D., Wang S. 3D Materials Image Segmentation by 2D Propagation: A Graph-Cut Approach Considering Homomorphism. *IEEE Transactions on Image Processing*, 2013, vol. 22, no. 12, pp. 5282-5293. doi: 10.1109/TIP.2013.2284071

9. Digabel H., Lantuéjoul C. Iterative algorithms. *Actes du Second Symposium Européen d'Analyse Quantitative des Microstructures en Sciences des Matériaux, Biologie et Médecine*. Caen, 1977, pp. 85-99.

10. Roerdink J. B., Meijster A. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies. *Fundamenta Informaticae*, 2000, vol. 41, no. 1,2, pp. 187-228. doi: 10.3233/FI-2000-411207

11. Atta-Fosu T., Guo W., Jeter D., Mizutani C. M., Stopczynski N., Sousa-Neves R. 3D Clumped Cell Segmentation Using Curvature Based Seeded Watershed. *Journal of Imaging*, 2016, vol. 2, no. 4, pp 31. doi: 10.3390/jimaging2040031

12. Alvaro V., Chen-Luh L., Miller J. D. Watershed Functions Applied to a 3D Image Segmentation Problem for the Analysis of Packed Particle Beds. *Particle & Particle Systems Characterization*, 2006, vol. 23, no. 3-4, pp. 237-245. doi: 10.1002/ppsc.200601055

13. Yakimchuk I. V., Safonov I. V., Serkova E. P., Evstefeeva V. Y., Korobkov D. A. Ceramic Proppant Microstructure Characterization by X-Ray Microtomography. *Bruker Micro-CT User Meeting*. Ghent, 2018, pp. 17-23.

14. Beare R., Chen J., Adamson C. L., Silk T., Thompson D. K., Yang J. Y. M., Anderson V. A., Seal M. L., Wood A. G. Brain extraction using the watershed transform from markers. *Frontiers in Neuroinformatics*, 2013, vol. 7, pp. 32. doi: 10.3389/fninf.2013.00032

15. Gostick J. T. Versatile and efficient pore network extraction method using marker-based watershed segmentation. *Physical Review E*, 2017, vol. 96, no.2-1, pp. 023307. doi: 10.1103/PhysRevE.96.023307

16. Guillard E., Nunez-Iglesias J., van der Walt S. Analyzing microtomography data with Python and the scikit-image library. *Advanced Structural and Chemical Imaging*, 2016, vol. 2, no. 1, pp. 18. doi: 10.1186/s40679-016-0031-0

17. Mashburn D. N., Lynch H. E., Ma X., Hutson M. S. Enabling user-guided segmentation and tracking of surface-labeled cells in time-lapse image sets of living tissues. *Cytometry Part A*, 2012, vol. 81A, no. 5, pp. 409-418. doi: 10.1002/cyto.a.22034

18. Johnson H. J., McCormick M., Ibáñez L. *The ITK Software Guide Book 1: Introduction and Development Guidelines Fourth Edition Updated for ITK version 4.7*. Kitware, Inc. 2015. 278 p.

19. Bradski G. The OpenCV Library. *Dr Dobb's Journal of Software Tools*, 2000, vol. 25, no. 11, pp. 122-125.
20. Coelho L. P. Mahotas: Open source software for scriptable computer vision. *Journal of Open Research Software*, 2013, vol. 1, pp. e3. doi: 10.5334/jors.ac
21. Beucher N., Beucher S. Mamba Image User Manual. 2017. Available at: <http://www.mamba-image.org/docs/2.0/mamba-um.pdf> (accessed 20 October 2018).
22. van der Walt S., Schönberger J. L., Nunez-Iglesias J., Boulogne F., Warner J. D., Yager N., Gouillart E., Yu T. scikit-image: image processing in Python. *PeerJ*, 2014, vol. 2, pp. e453. doi: 10.7717/peerj.453
23. Simple Morphological Image Library. 2011. Available at: <http://smil.cmm.mines-paristech.fr/> (accessed 20 October 2018).
24. Rueden C. T., Schindelin J., Hiner M. C., DeZonia B. E., Walter A. E., Arena E. T., Eliceiri K. W. ImageJ2: ImageJ for the next generation of scientific image data. *BMC Bioinform*, 2017, vol. 18, no. 1, pp. 529.
25. Safonov I. V., Mavrin G. N., Kryzhanovsky K. A. Segmentation of Convex Cells with Partially Undefined Edges. *Pattern Recognition and Image Analysis*, 2008, vol. 18, no. 1, pp. 112-117. doi: 10.1134/S1054661808010136
26. Beucher S., Meyer F. The morphological approach to segmentation: The watershed transformation. *Mathematical Morphology in Image Processing*, 1993, vol. 34, pp. 452-464.
27. Bieniek A., Moga A. An efficient watershed algorithm based on connected components. *Pattern Recognition*, 2000, vol. 33, no. 6, pp. 907-916. doi: 10.1016/S0031-3203(99)00154-5
28. Vincent L., Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1991, vol. 13, no. 6, pp. 583-598. doi: 10.1109/34.87344
29. Meyer F. Un algorithme optimal de ligne de partage des eaux. *Proceedings 8th Congress AFCET*. Lyon-Villeurbanne, France, 1992, vol. 2, pp. 847-859.
30. Beare R., Lehmann G. The watershed transform in ITK - discussion and new developments. *The Insight Journal*, 2006, pp. 24
31. Meyer F. Minimum Spanning Forests for Morphological Segmentation. *Mathematical Morphology and Its Applications to Image Processing*, 1994, pp. 77-84 doi: 10.1007/978-94-011-1040-2_11
32. Moga A. N., Gabbouj M. Parallel Marker-Based Image Segmentation with Watershed Transformation. *Journal of Parallel and Distributed Computing*, 1998, vol. 51, no. 1, pp. 27-45. doi: 10.1006/jpdc.1998.1448
33. Beucher N., Beucher S. Hierarchical Queues: general description and implementation in MAMBA Image library. 2011. Available at: <https://hal-mines-paristech.archives-ouvertes.fr/hal-00835024/document> (accessed 20 October 2018).
34. Neubert P., Protzel P. Compact Watershed and Preemptive SLIC: On Improving Tradeoffs of Superpixel Segmentation Algorithms. *2014 22nd 5 International Conference on Pattern Recognition*. Stockholm, 2014, pp. 996-1001. doi: 10.1109/ICPR.2014.181
35. Meyer F. Color image segmentation. *1992 International Conference on Image Processing and its Applications*. Maastricht, Netherlands, 1992, pp. 303-306.

Статья поступила 31 января 2019 г.

Информация об авторах

Корнилов Антон Сергеевич – соискатель ученой степени кандидата технических наук. Аспирант кафедры информатики и процессов управления. Национальный исследовательский ядерный университет «МИФИ» в г. Москва. Область научных интересов: обработка и анализ изображений, цифровая обработка сигналов, программирование. E-mail: kranton94@mail.ru

Сафонов Илья Владимирович – кандидат технических наук. Доцент кафедры информатики и процессов управления. Национальный исследовательский ядерный университет «МИФИ» в г. Москва. Область научных интересов: обработка сигналов и изображений, компьютерное зрение. E-mail: ilia.safonov@gmail.com

Адрес: 115409, Россия, г. Москва, Каширское шоссе, д. 31.

Time Efficiency Estimation of Watershed Algorithm Implementations in Open Source Libraries

A. S. Kornilov, I. V. Safonov

Problem definition. *The watershed transformation is one of the algorithms used for the segmentation of digital images. At present, there are many software implementations in open source software libraries that allow segmenting using the watershed for 2D and 3D images. These implementations are used to solve various tasks, and spent time is one of the most significant indicators. Often, researchers use in practice not the most optimal implementation. One of the reasons is the lack of a deep understanding of the implemented algorithms, as well as the lack of information about existing alternative solutions. Purpose.* *In this article, we analyze the watershed algorithms that are implemented in various open source libraries, and also we compare implementations by the criterion of minimizing of the processing time to identify the most optimized versions. Results.* *We described the main algorithms that are currently used in watershed transformation implementations, presented a comparison of the implementations in open source libraries with a Python call interface, determined the most optimized implementation for cases of processing 2D and 3D images at the moment.*

Key words: *watershed transformation, segmentation, time efficiency, open source libraries.*

Information about Authors

Anton Sergeevich Kornilov – Doctoral Student. The postgraduate student of the Computer Science and Control Systems Department. National Research Nuclear University MEPhI. Field of research: image processing, digital signal processing, programming. E-mail: kranton94@mail.ru

Ilia Vladimirovich Safonov – Ph.D. of Engineering Sciences. Associate Professor at the Computer Science and Control Systems Department. National Research Nuclear University MEPhI. Field of research: signal and image processing, computer vision. E-mail: ilia.safonov@gmail.com

Address: Russia, 115409, Moscow, Kashirskoe shosse, 31.