

УДК 004.92

## ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ МЕТОДОВ СЕГМЕНТАЦИИ ПО МОРФОЛОГИЧЕСКИМ ВОДОРАЗДЕЛАМ

А. Р. КАРИМОВ<sup>1</sup>, М. А. ВЕРХОТУРОВ<sup>2</sup>

<sup>1</sup>xkarimov@hotmail.com, <sup>2</sup>verhotur@vmk.ugatu.ac.ru

ФГБОУ ВПО «Уфимский государственный авиационный технический университет» (УГАТУ)

Поступила в редакцию 22.12.2013

**Аннотация.** Рассматривается задача параллельной реализации методов сегментации по морфологическим водоразделам (*WS*, *Watershed Segmentation*). Приводятся известные подходы к ее решению и анализируются проблемы, с ними связанные. Предлагается подход к решению поставленной задачи на основе разбиения изображения на блоки для независимой параллельной обработки и объединения результатов согласно критерию сегментации. Используемые в нем процедуры описываются в виде псевдокода. Проводится эксперимент по оценке меры увеличения временной эффективности при использовании предложенного подхода.

**Ключевые слова:** распараллеливание; сегментация по морфологическим водоразделам.

Цель сегментации цифрового изображения – это разметка на нем областей, называемых сегментами и обладающих заданными свойствами.

Среди многочисленных методов сегментации существует семейство методов сегментации по морфологическим водоразделам (*Watershed Segmentation*, *WS*), основанных на поиске площадей водосбора на топографической поверхности. Найденные площади и представляют собой искомые сегменты. Методы *WS* часто используются в качестве начального этапа сегментации изображений. Большинство из них обладают линейной вычислительной сложностью от числа пикселей изображения.

Требования к временной эффективности алгоритмов сегментации всегда высоки. Так, например, средний размер типичного медицинского трехмерного изображения (снимки компьютерной томограммы), подвергаемого сегментации, составляет  $512 \times 512 \times 512$  вокселей. Данные промышленной томографии обычно имеют еще больший размер –  $2048 \times 2048 \times 2048$  вокселей. Данные топографической съемки могут иметь размеры  $6000 \times 6000$  пикселей и более на 5 градусов широты и долготы. Часто все изображение требуется обрабатывать в режиме реального времени. Существующее на сегодняшний день аппаратное обеспечение позволяет обрабатывать данные такого объема в режиме реального времени только путем параллельной реализации алгоритмов. В связи с тем, что было разработа-

но целое семейство методов *WS*, стоит задача распараллеливания методов *WS* для общего случая.

### БАЗОВЫЕ ОПРЕДЕЛЕНИЯ

Цифровое двумерное изображение – это функция  $I(x, y): \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ , определенная на дискретных значениях  $x, y$  и принимающая дискретные значения. Цифровое трехмерное изображение – это функция  $I(x, y, z): \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ , определенная на дискретных значениях  $x, y, z$  и принимающая дискретные значения. Точки  $(x, y, z)$  называют пикселями, а  $(x, y, z)$  – вокселями. Значение функции цифрового изображения называют также яркостью. Далее рассматриваются двумерные изображения, однако все рассуждения можно обобщить на случай трехмерных изображений.

Множество пикселей обозначим  $D$ . Множество значений яркости обозначим  $V$ . В статье рассматриваются только цифровые изображения вида  $(x, y) \in D, I(x, y) \in V$ . Множество соседних пикселей пикселя  $p$  обозначим  $N(p)$ . Если выполняется  $(q, I(q)) \in N(p, I(p))$ , то пиксели  $p, q$  называют смежными. Расстояние между двумя смежными пикселями  $d(p, q)$  зависит от выбранной связности изображения (для двумерных изображений: 4-связность, 8-связность, для трехмерных изображений: 6-связность, 26-связность) (т. е.  $d(p, q) > 0$ ). Отдельно огово-

ривается случай  $d(p, p) = 0$ . Для двух несмежных пикселей  $p, q$

$$d(p, q) = \min_{\substack{(x_1, \dots, x_n): x_1=p, x_n=q, \\ i=1, n-1: x_{i+1} \in N_G(x_i)}} \sum_{i=1, n-1} d(x_i, x_{i+1}).$$

Задача сегментации цифрового изображения (*Digital Image Segmentation*) формулируется так [1]: необходимо найти  $n$  сегментов  $S_i$  на  $I(D, V)$  с заданным критерием сегментации  $P$  так, чтобы одновременные выполнялись следующие условия:

$$\bigcup_{i=1}^n S_i = D$$

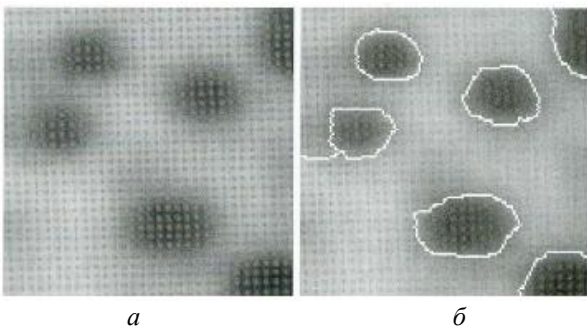
$i = \overline{1, n}: S_i$  – является связной областью

$$i = \overline{1, n}, j = \overline{1, n}, i \neq j: S_i \cap S_j = \emptyset$$

$$i = \overline{1, n}: P(S_i) = \text{TRUE}$$

$$i = \overline{1, n}, j = \overline{1, n}, i \neq j: P(S_i \cup S_j) = \text{FALSE}$$

Часто задача сегментации имеет дело с объектами, обладающими относительно слабым разбросом яркости. Подобное свойство искомым объектам часто встречается во многих приложениях цифровой обработки изображений: в сегментации медицинских изображений, в сегментации изображения лица в охранных системах, в оптическом распознавании символов и в системах слежения. Пример таких объектов приведен на рис. 1.



**Рис. 1.** Клетки крови под микроскопом: *a* – исходное изображение, *б* – результат сегментации методом *WS*

### МЕТОДЫ СЕГМЕНТАЦИИ ПО МОРФОЛОГИЧЕСКИМ ВОДОРАЗДЕЛАМ

Семейство методов сегментации по морфологическим водоразделам *WS* предназначено для распознавания объектов как на рис. 1.

Методы *WS* имеют ряд базовых определений:

- минимум изображения  $M_i$  – это связное множество пикселей, яркость которых принад-

лежит локальному минимуму яркости на изображении;

- путь наибольшего ската – это упорядоченный набор смежных пикселей, в котором яркость пикселей не возрастает, а последний пиксель находится в некотором минимуме изображения;

- сегмент  $S_i$  – это множество пикселей, содержащее один минимум изображения  $M_i$ , и в котором для любого пикселя есть путь наибольшего ската только в  $M_i$ ;

- пиксель водораздела – это пиксель, имеющий, по крайней мере, 2 пути наибольшего ската в два минимума изображения  $M_i, M_j: i \neq j$ ;

- плато (пикселей) – это связное множество пикселей одинаковой яркости, у которого есть внутренние пиксели;

- ниже-полное изображение (*Lower Complete Image*) – это особый вид изображения, в котором отсутствуют плато пикселей, за исключением минимумов изображения.

Обобщенный алгоритм для методов *WS* можно записать так:

- начальные точки сегментов ставятся в точках локального минимума модуля градиента яркости;

- сегменты постепенно увеличиваются путем добавления смежных к ним пикселей;

- на изображении проводятся линии из пикселей («водоразделы»), разграничивающие отдельные сегменты.

Методы *WS* обладают одинаковым критерием сегментации (1).

$$P(S_i) = \begin{cases} \text{TRUE} & | \text{ в } S_i \text{ есть только один} \\ & \text{локальный минимум модуля} \\ & \text{градиента яркости,} \\ \text{FALSE} & | \text{ в противном случае.} \end{cases} \quad (1)$$

Классификация методов *WS* приведена на рис. 2. В ее основе лежат базовые принципы различных методов *WS*, приведенные в табл. 1. Детальное описание методов *WS* есть в [2].

Методы *WS* используют в своей работе информацию о перепадах яркости на изображении. В связи с этим непосредственной сегментации подвергается не исходное изображение  $I(D, V)$ , а его модуль градиента  $G(D, V)$ , который количественно описывает перепады яркости. Поэтому далее вместо  $I(D, V)$  будет подразумеваться  $G(D, V)$ , если не сказано обратное.



Рис. 2. Классификация методов WS

Таблица 1

Базовые принципы, характерные для различных методов WS

Метод	Особенности
WSI	Представление изображения в виде трехмерной модели рельефа
WSHC	Топологическое расстояние как мера расстояния между пикселями изображения
WSRF	Имитация падения и ската капель дождя по трехмерной поверхности, построенной по изображению
WSII	Представление в виде графа, где вершины – пиксели, а ребра – связи между пикселями. Ищутся кратчайшие пути в графе
WSGC	Представление в виде графа, где вершины – связные компоненты изображения. Строится остов графа
WSLC	Вычисление локальных свойств яркости

### СПОСОБЫ ПАРАЛЛЕЛЬНОЙ РЕАЛИЗАЦИИ АЛГОРИТМОВ WS

В [2] указывается, что существуют два основных способа параллельной реализации алгоритмов WS:

- распределение минимумов изображения между параллельными потоками;
- разделение изображения на блоки, обрабатываемые параллельными потоками отдельно.

Первый способ имеет существенный для временной эффективности недостаток, который принципиально невозможно устранить. Проблема заключается в том, что при этом способе временная эффективность параллельной реали-

зации зависит от числа локальных минимумов яркости на изображении и числа используемых параллельных потоков. Если эти два числа различны, будут наблюдаться либо простые потоки, либо пересылки больших объемов данных между потоками для обеспечения синхронности операций.

Второй способ также имеет существенный, но устранимый недостаток: вычисления параллельного потока зависят не только от выделенной ему части изображения, но и от всего изображения в целом. Из-за этого появляется необходимость в обмене данными между потоками, что снижает временную эффективность параллельной реализации.

Накладывая на сегментируемое изображение некоторые ограничения, можно устранить указанную зависимость полностью, в отличие от первого способа.

### Устранение зависимости вычислений параллельных потоков от всего изображения

Пиксели, находящиеся в минимуме изображения, могут находиться в плато. Однако их метки однозначно определяются меткой сегмента, присваиваемой всему минимуму изображения. Для пикселей, не входящих в плато, метки сегментов можно определить по меткам сегментов соседних пикселей. Таким же образом определяются метки сегментов внешних пикселей плато.

Метки сегментов внутренних пикселей плато определить непосредственно по минимуму изображения или соседним пикселям не удастся, т.к. на плато все пиксели имеют одинаковую яркость и, следовательно, для их разметки требуется учитывать минимальные расстояния до всех окружающих плато минимумов изображения. Это и обуславливает зависимость результата сегментации от всего изображения в целом.

В результате источник указанной выше зависимости – это плато пикселей, не являющихся минимумами изображения.

Нижне-полные изображения по определению не содержат плато пикселей, не являющихся минимумами изображения, следовательно, для них зависимость вычислений параллельных потоков от всего изображения отсутствует. Используя нижне-полные изображения и второй способ распараллеливания методов WS, можно достичь высокой временной эффективности параллельной реализации.

Существуют процедуры приведения произвольного изображения  $f$  к нижне-полному изо-

бражению  $f_{LC}$  [2]. По своему содержанию  $f_{LC}$  эквивалентно  $f$  с точки зрения методов  $WS$ .

Это означает, что любое изображение можно сегментировать параллельной реализацией методов  $WS$  согласно второму способу распараллеливания без обмена данными между потоками при условии предварительного приведения изображения к ниже-полному.

Требуется только указать процедуру объединения результатов вычислений параллельных потоков.

### Объединение результатов вычислений параллельных потоков

Разбиение изображения на блоки может привести:

- к появлению новых минимумов изображения, которых на исходном изображении не было,
- к разделению сегментов на части, если они (сегменты) располагались в двух и более блоках.

Обе ситуации не указывают на возврат к начальной ситуации.

Для иллюстрации рассмотрим рис. 3.

На рис. 3, *а* присутствует только один минимум изображения. Однако в каждом блоке на рис. 3, *б* есть свой минимум изображения. Для обработки новых минимумов изображения используется процедура разметки минимумов изображения, которая запускается отдельно для каждого блока. Все методы  $WS$  содержат эту процедуру в том или ином виде [2].

На рис. 3, *а* может быть только один сегмент, так как есть только один минимум изображения. Но блоки на рис. 3, *б* содержат 16 сегментов. Чтобы получить результат сегмента-

ции изображения, некоторые сегменты на рис. 3, *б* потребуется объединить.

Ход объединения сегментов приведен на рис. 3, *в*. Каждая стрелка указывает пару объединяемых сегментов. Порядок объединения сегментов выбирается произвольно.

В примере использовался следующий порядок объединения:

- сегмент с большей средней яркостью присоединяется к сегменту с меньшей средней яркостью;
- если средние яркости сегментов равны, сегмент с меньшим порядковым номером присоединяется к сегменту с большим порядковым номером.

В итоге остается сегмент № 11 на рис. 3, *г*. Все остальные сегменты были с ним объединены. Это и есть правильное решение задачи сегментации по водоразделам.

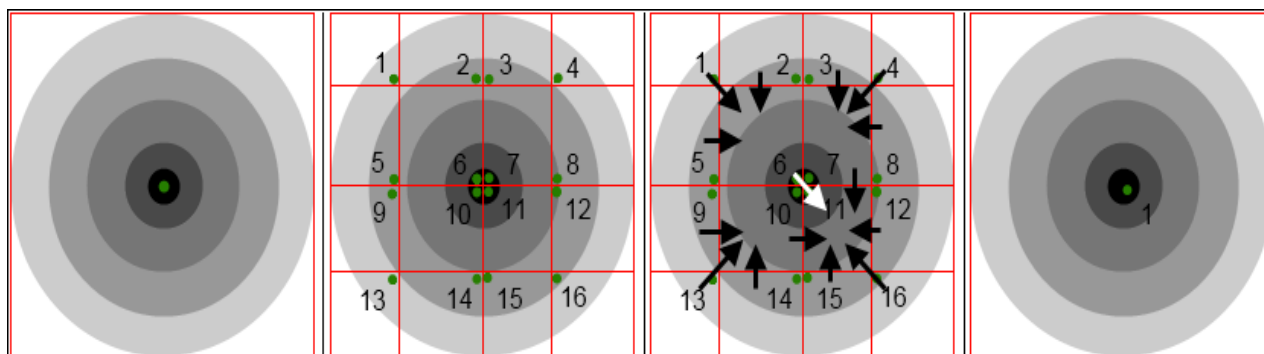
Объединение результатов сегментации блоков в результат сегментации изображения можно выполнять непосредственно в параллельных потоках или после окончания их работы. В первом случае потребуется синхронизация потоков, которой желательно избежать. Во втором случае потребуется некоторый критерий объединения сегментов, контролирующий процесс объединения.

Найти подобный критерий можно не для всех методов сегментации. В случае методов  $WS$  требуемый критерий был найден.

Алгоритм, реализующий объединение сегментов, приводится после раздела о разметке минимумов изображения.

### Разметка минимумов изображения

Процедура разметки минимумов изображения обычно имеет временную сложность  $O(n)$ , где  $n$  – это число пикселей на изображении.



**Рис. 3.** Появление новых минимумов изображения и объединение сегментов:

- а* – исходное изображение; *б* – результаты сегментации блоков до объединения; *в* – объединение сегментов, *г* – результат объединения, результат сегментации изображения; ● – минимум изображения; ●<sup>1</sup> – сегмент изображения с меткой; → – направление объединения двух сегментов

Процедура ищет пиксели, входящие в неразмеченные минимумы изображения. У искомого пикселя не должно быть метки сегмента, а также соседних пикселей с меньшей яркостью.

Когда такой пиксель будет найден, он получает уникальную метку нового сегмента. После этого используется одномерный цикл для разметки плато пикселей, попадающего в только что найденный минимум изображения.

Однако при параллельной реализации процедура разметки минимумов изображения должна обеспечивать уникальность присваиваемых меток.

Для этого можно использовать глобальную переменную, доступ к которой для всех параллельных потоков должен быть синхронизирован. Тогда операция получения уникальной метки сегмента будет считывать и возвращать текущее значение метки, а также создавать новое уникальное значение метки. Однако такое решение не является оптимальным с точки зрения временной эффективности.

Более рациональным решением является назначение всем параллельным потокам непересекающихся интервалов значений уникальных меток. Рассчитать эти интервалы можно заранее, так как одному потоку в худшем случае может понадобиться не более чем  $m$  значений уникальных меток, где  $m$  — это количество пикселей в сегментируемом блоке.

**Критерий объединения сегментов**

Пусть яркость изображения  $I$  в точке  $p$  обозначается как  $I(p)$ , а множество пикселей ската определяется как

$$\left\{ \begin{matrix} \text{---} \\ \text{---} \\ \text{---} \end{matrix} \right\}$$

Пусть также даны результаты сегментации двух блоков изображения  $I - R_1, R_2$ .

Тогда, два произвольных сегмента таких, что выполняется

$$), \tag{2}$$

$$), \tag{3}$$

$$), \tag{4}$$

$$), \tag{5}$$

должны быть объединены в новый сегмент  $S_2$ .

Смысл требований (2)–(5) следующий:

- требование (2) позволяет упорядочить пары объединяемых сегментов,

- требования (3), (4) относятся к выбору пикселя  $p$ , который проверяется при попытке объединения сегментов, в частности:

- по требованию (3) пиксель должен принадлежать минимуму сегмента  $S_2$ ,

- по требованию (4) пиксель должен находиться на границе  $S_1$  и на границе  $S_2$ ,

- требование (5) определяет, допустимо ли объединять  $S_1, S_2$  в один сегмент  $S$  на изображении  $I$ .

Условие (5) можно переформулировать следующим образом:

$$(5')$$

С учетом введенных требований можно сделать вывод о том, что проверять следует только те пиксели изображения, которые находятся на границах блоков. Внутренние же пиксели блоков проверять нет необходимости.

**Алгоритм объединения сегментов**

Критерий объединения сегментов (2)–(5) (или (2)–(4), (5')) таков, что его алгоритм обладает временной сложностью  $O(n^2)$  для трехмерного изображения и  $O(n)$  для двумерного изображения, где  $n$  — число пикселей на изображении.

Таблица 2

**Обозначения в процедуре объединения сегментов**

Обозначение	Пояснение
	Цифровое изображение
	Яркость пикселя
	$i$ -й сегмент
	Метка сегмента пикселя $p$
	Наименьшее значение яркости в сегменте $S$
	Множество внешних пикселей сегментируемых блоков
	Процедура объединения сегментов $S_i, S_j$ в сегмент $S$

Процедуру объединения сегментов можно представить следующим образом (используются обозначения из табл. 2):

1. Цикл по всем
2. Цикл по всем
3. Если
4. Если
5. Если
- 6.

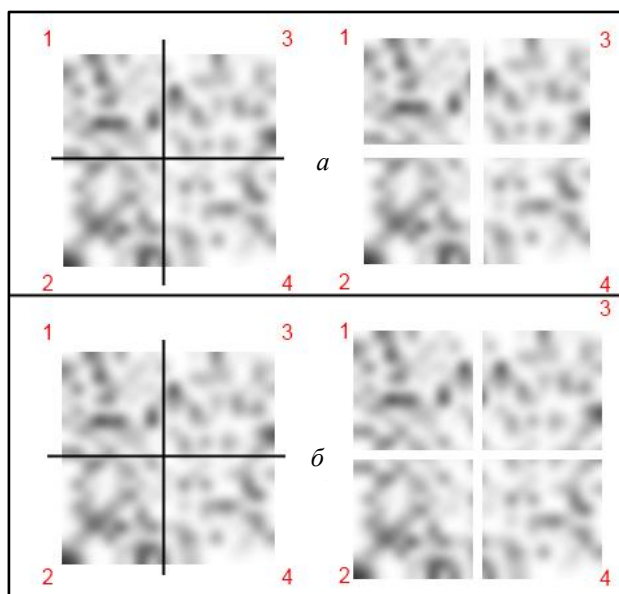
### Обработка пикселей водоразделов

Критерий объединения сегментов (2)–(5) (или (2)–(4), (5')) предполагает, что распараллеливаемый метод *WS* не использует метки водоразделов, то есть все пиксели изображения получают метки сегментов.

Определить пиксель водораздела можно по меткам сегментов соседних пикселей. Однако при обработке внешних пикселей блока  $R_i$  потоком  $T_k$  может получиться так, что некоторые соседние пиксели оказываются в блоке  $R_j$  ( $i \neq j$ ), который  $T_k$  не обрабатывает. В результате пиксели могут ошибочно не получить метку водораздела.

Если пиксели водоразделов все же требуются, то необходимо изменить процедуру разбиения изображения. Изменения заключаются в том, что для каждого внешнего пикселя блока должны быть доступны все его соседние пиксели. Это приводит к увеличению каждого блока по каждому измерению на два пикселя. Снижение временной эффективности параллельной реализации и повышение требований к объему памяти при данном изменении пренебрежимо мало.

Иллюстрация измененной процедуры разбиения изображения приведена на рис. 4. В случае на рис. 4, б блоки имеют большие размеры по вертикали и горизонтали, чем в случае на рис. 4, а. Это объясняется включением в них соседних пикселей. Эти пиксели дублируются в смежных блоках.



**Рис. 4.** Разбиение изображения на блоки:  
а – без учета водоразделов;  
б – с учетом водоразделов

### СРАВНЕНИЕ С ИЗВЕСТНЫМИ ПОДХОДАМИ

В [2] рассматривается подход к распараллеливанию метода *WSHC* без пикселей водоразделов с обменом сообщениями между потоками.

В [3] рассматривается подход к распараллеливанию методов *WSHC*, *WSRF* без пикселей водоразделов с обменом сообщениями между потоками.

В [4] рассматривается подход к распараллеливанию метода *WSI* с пикселями водоразделов. Однако он требует операции редукции параллельных вычислений.

В [5] рассматривается подход к распараллеливанию метода *WSHC* без пикселей водоразделов с обменом сообщениями между потоками, близкий к предложенному в статье подходу. Обмен сообщениями в нем используется для обработки плато пикселей. Однако в [5] не рассматривается возможность использования схемы распараллеливания для других методов *WS* и работа схемы распараллеливания с пикселями водоразделов. Кроме того, в [2, 5] указывается, что в некоторых ситуациях возможны некорректные результаты сегментации.

Предложенный же в данной статье подход гарантирует корректность результатов сегментации и работает для любых методов *WS*.

### ЭКСПЕРИМЕНТ ПО ОЦЕНКЕ МЕРЫ УВЕЛИЧЕНИЯ ВРЕМЕННОЙ ЭФФЕКТИВНОСТИ

Эксперимент заключался в сравнении времени сегментации параллельной реализации метода *WSHC* (далее *PWS*) со временем сегментации обычной реализации метода *WSHC* (далее *LWS*).

Мера увеличения временной эффективности  $Q$  определяется как отношение времени сегментации обычной реализации алгоритма к времени сегментации параллельной реализации алгоритма. Чем больше  $Q$ , тем быстрее работает параллельная реализация алгоритма.

Использовался ПК с четырехядерным ЦП Intel® Core™ i7 Q720.

### Случай двумерных изображений

Сегментировались 10 тестовых двумерных изображений – снимков компьютерной томограммы с разрешением 768\*768 пикселей.

Использовались блоки со следующими размерами: 384\*768 (*PWS1*), 384\*384 (*PWS2*), 192\*192 (*PWS3*), 96\*96 (*PWS4*). Использовалась 4-связность между пикселями изображения.

Результаты измерения времени сегментации приведены в табл. 5. Зависимость времени сегментации от сегментируемого изображения и реализации алгоритма приведена на рис. 5. Среднее значение  $Q$  для блоков различных размеров приведено в табл. 3.

**Случай трехмерных изображений**

Сегментировались 10 тестовых трехмерных изображений – наборов снимков компьютерной томограммы с разрешением 512\*512\*256 вокселей.

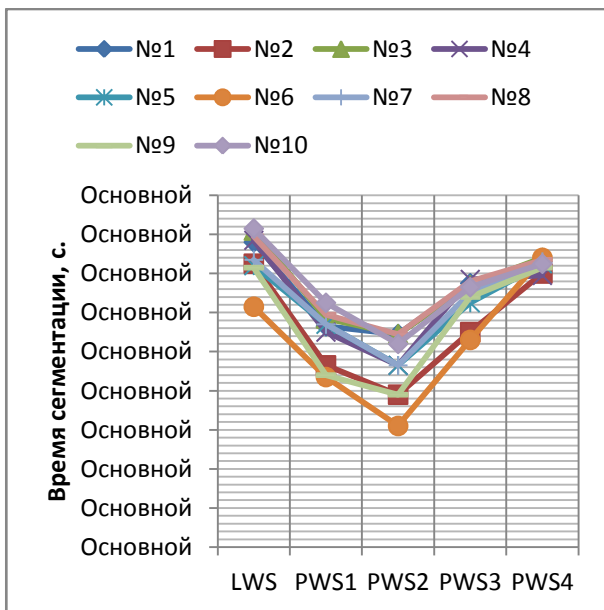
Использовались блоки со следующими размерами: 256\*512\*64 (*PWS1*), 256\*256\*64 (*PWS2*), 128\*128\*64 (*PWS3*), 128\*128\*128 (*PWS4*). Использовалась 6-связность между вокселями изображения.

Результаты измерения времени сегментации приведены в табл. 6. Зависимость времени сегментации от сегментируемого изображения и реализации алгоритма приведена на рис. 6. Среднее значение  $Q$  для блоков различных размеров приведено в табл. 4.

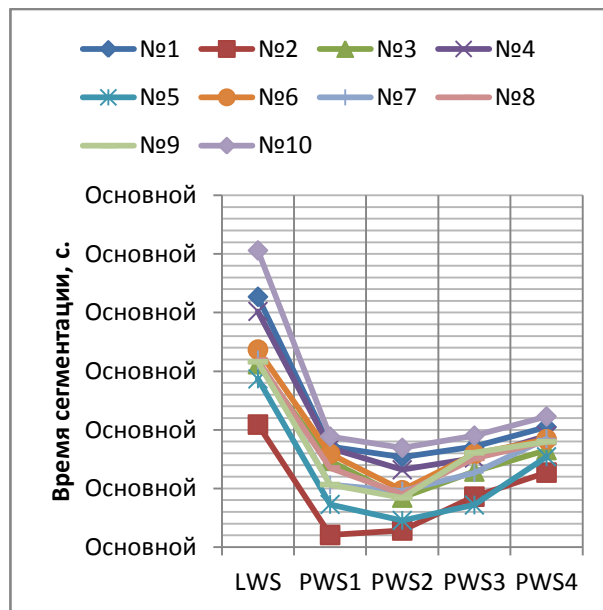
**Выводы по результатам эксперимента**

Из полученных экспериментально зависимостей  $Q_{ср}$  от размера блока можно сделать следующие выводы:

- Оптимальным с точки зрения увеличения временной эффективности размером блока для случая работы с  $N$  физическими ядрами ЦП является – часть изображения;
  - Оценка меры увеличения временной эффективности при оптимальном размере блока составила 1,64 для двумерного случая и 4,02 для трехмерного случая.
- При отклонении размера блока от оптимального значения наблюдается уменьшение  $Q$ :
- При уменьшении размера блока оно связано с дополнительными операциями на стадии объединения сегментов из-за увеличения числа проверяемых пикселей;
  - При увеличении размера блока оно связано с простым некоторых ядер ЦП.



**Рис. 5.** Зависимость времени сегментации от сегментируемого изображения и реализации алгоритма (двухмерный случай)



**Рис. 6.** Зависимость времени сегментации от сегментируемого изображения и реализации алгоритма (трехмерный случай)

Таблица 3

Среднее значение  $Q$  при работе с блоками различного размера (двухмерный случай)

<i>PWS1</i>	<i>PWS2</i>	<i>PWS3</i>	<i>PWS4</i>
1,40	1,64	1,18	1,04

Таблица 4

Среднее значение  $Q$  при работе с блоками различного размера (трехмерный случай)

<i>PWS1</i>	<i>PWS2</i>	<i>PWS3</i>	<i>PWS4</i>
3,42	4,02	2,53	1,89

Таблица 5

## Результаты измерения времени сегментации (двумерный случай)

Изображение	Число сегментов, шт.	Время <i>LWS</i> , с.	Время <i>PWS1</i> , с.	Время <i>PWS2</i> , с.	Время <i>PWS3</i> , с.	Время <i>PWS4</i> , с.
№ 1	7069	0,156	0,113	0,109	0,135	0,146
№ 2	8090	0,145	0,093	0,078	0,110	0,140
№ 3	14685	0,162	0,117	0,109	0,134	0,148
№ 4	20625	0,157	0,110	0,093	0,137	0,139
№ 5	21832	0,144	0,114	0,093	0,125	0,145
№ 6	17582	0,123	0,087	0,062	0,106	0,148
№ 7	16239	0,147	0,114	0,093	0,130	0,146
№ 8	14833	0,159	0,119	0,109	0,136	0,147
№ 9	15293	0,143	0,088	0,078	0,128	0,143
№ 10	15479	0,163	0,125	0,104	0,133	0,145

Таблица 6

## Результаты измерения времени сегментации (трехмерный случай)

Изображение	Число снимков, шт.	Число сегментов, шт.	Время <i>LWS</i> , с.	Время <i>PWS1</i> , с.	Время <i>PWS2</i> , с.	Время <i>PWS3</i> , с.	Время <i>PWS4</i> , с.
№ 1	183	1715725	21,347	8,574	7,675	8,565	10,243
№ 2	64	651390	10,452	1,034	1,419	4,302	6,356
№ 3	212	3089783	15,602	7,357	4,227	6,446	8,283
№ 4	319	3616802	20,105	8,425	6,598	7,603	9,506
№ 5	111	700009	14,347	3,641	2,277	3,593	7,705
№ 6	251	3855664	16,843	7,954	4,851	7,943	9,201
№ 7	228	3105224	15,956	5,314	4,789	6,346	9,240
№ 8	210	2880645	15,950	6,755	4,539	7,566	9,013
№ 9	191	2120845	15,803	5,335	4,212	8,045	8,983
№ 10	388	2918389	25,302	9,431	8,470	9,508	11,142

## ЗАКЛЮЧЕНИЕ

В заключение можно сделать следующие выводы:

- предложен подход к параллельной реализации методов *WS* на основе разбиения изображения на блоки для независимой обработки и объединения результатов согласно критерию сегментации;

- приведены псевдокоды основных процедур для параллельной реализации методов *WS*;

- проведен эксперимент по анализу временной эффективности параллельной реализации алгоритма *WSHC* и получено ее увеличение в 1,6 раза при работе с *2D*-изображениями и в 4 раза при работе с *3D*-изображениями по сравнению с обычной реализацией алгоритма.

Для дальнейшего повышения временной эффективности ведется работа по программированию параллельной реализации алгоритма *WSHC* на основе технологии параллельных вычислений OpenCL.

## СПИСОК ЛИТЕРАТУРЫ

- Гонсалес Р., Вудс Р. Цифровая обработка изображений. М.: Техносфера, 2005. [ R. Gonzalez, R. Woods, *Digital Image Processing*, (in Russian). Moscow: Tehnosfera, 2005. ]
- Roerdink J., Meijster A. The Watershed Transform: Definitions, Algorithms and Parallelization Strategies // *Fundamenta Informaticae*. 2001. Vol. 41. P. 187–228. [ J. Roerdink, A. Meijster, "The Watershed Transform: Definitions, Algorithms and Parallelization Strategies," *Fundamenta Informaticae*, vol. 41, pp. 187-228, 2001. ]
- Moga A., Cramariuc B., Gabbouj M. Parallel watershed transformation algorithms for image segmentation // *Parallel Computing*. 1998. Vol. 24. P. 1981–2001. [ A. Moga, B. Cramariuc, M. Gabbouj, "Parallel watershed transformation algorithms for image segmentation", *Parallel Computing*, vol. 24, pp. 1981-2001, 1998. ]
- Sun H., Yang J., Ren M. A fast watershed algorithm based on chain code and its application in image segmentation // *Pattern Recognition*. 2005. Vol. 26. P. 1266–1274. [ H. Sun, J. Yang, M. Ren, "A fast watershed algorithm based on chain code and its application in image segmentation," *Pattern Recognition*, vol. 26, pp. 1266-1274, 2005. ]



5. **Moga A., Gabbouj M.** Parallel image component labeling with watershed transformation // IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997. Vol. 19, no. 5. P. 441–450. [ A. Moga, M. Gabbouj, "Parallel image component labeling with watershed transformation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 5, pp. 441-450, 1997. ]

#### ОБ АВТОРАХ

**КАРИМОВ Алексей Ришатович**, асп. каф. выч. матем. и кибернетики. Дипл. матем.-программист (УГАТУ, 2010). Готовит дис. о сегментации цифровых изображений.

**ВЕРХОТУРОВ Михаил Александрович**, проф. каф. выч. матем. и кибернетики. Дипл. инж.-системотехник (УАИ, 1983). Д-р техн. наук по АСУ (УГАТУ, 2001). Иссл. в обл. опт. геометр. проектирования и моделирования.

#### METADATA

**Title:** Parallel implementation of watershed segmentation.

**Authors:** A. R. Karimov, M. A. Verkhoturov.

**Affiliation:** Ufa State Aviation Technical University (UGATU), Russia.

**Email:** xkarimov@hotmail.com, verhotur@vmk.ugatu.ac.ru.

**Language:** Russian.

**Source:** Vestnik UGATU (scientific journal of Ufa State Aviation Technical University), vol. 18, no. 2 (63), pp. 172-180, 2014. ISSN 2225-2789 (Online), ISSN 1992-6502 (Print).

**Abstract:** The problem of parallel implementation of watershed segmentation (WS) is considered in this paper. Known approaches to solve the task are given and their problems are discussed. New approach is proposed, based on tiling of image and independent parallel processing of resulting tiles. Merging procedure for found segments is applied to meet segmentation criterion. All necessary procedures are described as pseudocode. Results of experiment are given, providing information on speedup gained using proposed approach.

**Key words:** parallel implementation; watershed segmentation.

#### About authors:

**KARIMOV, Aleksey Rashitovich**, Postgrad. (PhD) Student, Dept. of Computer Mathematics and Cybernetics. Dipl. Mathematic-Programmer (UGATU, 2010).

**VERKHOTUROV, Mikhail Aleksandrovich**, Prof., Dept. of Computer Mathematics and Cybernetics. Dipl. of Systems Engineering (Ufa Aviation Institute, 1983). Cand. of Tech. Sci. (UGATU, 1992), Dr. of Tech. Sci. (UGATU, 2001).