

## **Параллельный метод реконструкции в ортогональных плоскостях**

М. М. Новожилов, М. П. Кочуков

Нижегородский госуниверситет им. Н. И. Лобачевского

Решается задача реконструкции внешних и внутренних границ органов, имеющих сложную пространственную структуру (мозг, сложные кости) по данным томографии. Для разрешения неоднозначностей реконструкция выполняется в ортогональных плоскостях. Рассматриваются 2 подхода. В первом, производится трехмерная сегментация методом разрастания регионов, и решается задача реконструкции двумерных контуров в двух ортогональных сечениях. Во втором – алгоритм реконструирует контуры в плоскостях, ортогональных осям координат, и объединяет их в граф поверхности. В обоих алгоритмах, полученный набор контуров триангулируется. За счет независимой обработки плоскостей, в параллельной реализации удалось добиться достаточно высокого ускорения.

*Ключевые слова:* реконструкция, томография, сегментация, контур, неоднозначность.

### **1. Введение**

Множество приложений в медицинской визуализации, компьютерной графике и моделировании поверхностей предполагают восстановление поверхности из набора параллельных контуров. Реконструкция таких поверхностей достаточно сложная проблема, у которой не существует однозначного решения. Различия в топологии контуров и разреженность слоев говорят о том, что существует бесконечное множество поверхностей, удовлетворяющих данным контурам. Первый подход основан на реконструкции поверхности с помощью прямого анализа контуров на соседних слоях. Подобный алгоритм контурного анализа описан в работах [1-4]. Например, в работе [1] каждая пара слоев проецируется на параметрическую плоскость, проводится триангуляция Делоне и далее производится большое количество расчетов для нахождения соответствующих точек. В работе [2] предлагается находить наилучшую триангуляцию соседних контуров путем вычисления поверхности-пояса с наименьшей площадью.

Второй подход тесно связан с алгоритмом Marching Cubes, описанным в [5]. Он позволяет реконструировать полигональную поверхность, не предъявляя к данным каких-либо особых требований. Алгоритм разбивает пространство на кубические ячейки и восстанавливает поверхность независимо в каждой из них. Благодаря этому алгоритм хорошо распараллеливается. Однако данный алгоритм имеет проблемы неоднозначности, которые могут привести к нарушению топологии. Например, для некоторых случаев существуют несколько вариантов примыкания поверхностей в соседних ячейках. Решение данной проблемы предложено в работе [6]. Также, существует проблема неоднозначности для расположения поверхности внутри ячейки, подробно рассмотренная в [8]. Изложенные в этой работе решения дополняются в работе [9], гарантирующей корректную топологию получаемой поверхности. Дальнейшая работа над алгоритмом Marching Cubes направлена на улучшение качества поверхности. Например, в работах [11, 12] уделено внимание интерполяции при вычислении вершины на ребре кубической ячейки.

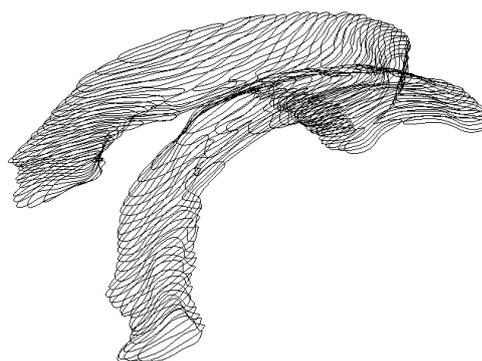
### **2. Последовательная сегментация и реконструкция в ортогональных направлениях**

На вход алгоритма поступают данные компьютерной томограммы, представляющие собой множество точек со значением поля. Точки размещаются в узлах трехмерной прямоугольной сет-

ки. С помощью этих точек, используя интерполяцию, можно вычислить значение поля в произвольной точке. Совокупность точек с одинаковым значением поля образуют изоповерхность. Необходимо выполнить ее полигональную реконструкцию с информацией о смежных вершинах.

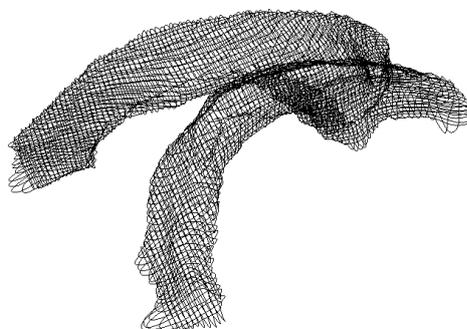
Идея первого подхода заключается в следующем. Сначала производится сегментация какого-либо органа. В данной работе применяется трехмерная сегментация с использованием простейшего метода разрастания регионов и метода активных контуров. В результате этого получаем набор вокселей, представляющий сегментированный объект. Затем, используя полученный набор вокселей, получаем контуры на каждом слое путем рассечения вокселей плоскостью, параллельной плоскости XY. Полученные контуры отдельно для каждого слоя аппроксимируются с помощью периодических B-сплайнов (Рис. 1).

Таким образом, получаем достаточно точное приближение для контура на каждом слое томограммы. B-сплайны параметризованы по длине, поэтому в процессе триангуляции генерируются треугольники из соответствующих точек B-сплайнов. В данном подходе приходится иметь дело с неоднозначностями. Обычно, неоднозначности в контурном анализе разделяют на 3 группы. Во-первых – это проблема нахождения когерентных контуров. Имея несколько контуров на предыдущем и последующем слое, необходимо определить какие, контуры соединять друг с другом. Во-вторых, необходимо найти соответственные точки на когерентных контурах. В-третьих, необходимо разрешить проблему ветвления в случае, если, число контуров меняется от слоя к слою. Решение для всех этих проблем предлагается, например, в статьях [1-3]. В данной статье проблема неоднозначности частично разрешаются за счет использования контуров в других плоскостях.



**Рис. 1.** Результат аппроксимации контуров для всех слоев

Аналогичные операции производятся в ортогональном направлении. Из-за использования аппроксимации для приближения контура, в общем случае, нельзя получить точек пересечения ортогональных контуров. Поэтому, для того, чтобы получить сетку, учитываются точки уже сгенерированных контуров для сплайн-аппроксимации в ортогональном направлении (Рис. 2).



**Рис. 2.** Результат аппроксимации контуров для всех слоев в двух ортогональных направлениях

### 3. Реконструкция методом объединения контуров в ортогональных плоскостях

Первым этапом реконструкции является получение набора точек. В данном подходе, будем рассматривать поверхность, интерполирующую данные точки, то есть каждая вычисленная точка будет являться вершиной поверхности. Рассматривая произвольный набор точек в пространстве, возникает ряд неоднозначностей, наиболее важной из которых является определение соседних вершин.

Для разрешения этих неоднозначностей алгоритм Marching Cubes предлагает принять следующие условия:

- В пространстве рассматривается прямоугольная сетка;
- Ребро ячейки сетки пересекает только одну изоповерхность;
- Вершины изоповерхности находятся на узлах и ребрах сетки;
- Вершины, находящиеся на грани ячейки, являются соседними (их можно соединить линией).

Данные свойства дают возможность применить несколько иной подход к реконструкции – вначале реконструировать кривые (или контуры) исходной поверхности. Данный подход является широко распространенным, и, например, хорошо описан в [13]. Однако если учесть упомянутые выше условия, появляется возможность извлечь из исходных данных немного больше информации. Алгоритмы, подобные приведенному в работе [13], рассматривают последовательность контуров в плоскостях, ортогональных только одной из осей. Описанный ниже подход предлагает рассмотреть объединение контуров, полученных на ортогональных трех осях плоскостях, и предлагает решение задачи эффективного поиска точек их пересечения.

Алгоритм реконструкции состоит из трех этапов. На каждом из них вычисляются вершины и выявляются связи между ними, основываясь на установленном критерии. В процессе работы алгоритма последовательно рассматриваются 2D сетки, ортогональные осям. Для оси X получаем последовательность контуров (Рис. 3, слева) и множество вершин на ребрах сеток. Затем рассматриваются 2D сетки, ортогональные оси Z. Множество вершин на сетках, полученное на предыдущем этапе, переносится на текущие сетки, и вычисляются новые вершины. После выявления связей получаем последовательность контуров (Рис. 3, в середине). В итоге, рассматриваются 2D сетки, ортогональные оси Y. Вычисление новых вершин не требуется, и остается перенести полученные на предыдущих этапах вершины на текущие сетки. Получаем последовательность контуров (Рис.3, справа).

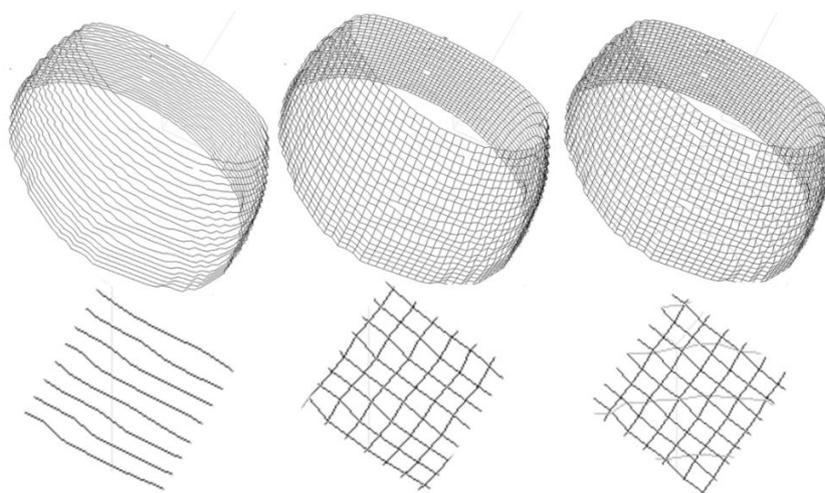


Рис. 3. Результат работы алгоритма

Последовательность направлений может быть любой. Но только выбор по трем направлениям гарантирует единственность получаемой модели. Сетки в пределах одного направления обрабатываются независимо, что дает возможность распараллелить вычисления.

Таким образом, основными этапами алгоритма являются реконструкция сетки поверхности и триангуляция полученной сетки.

#### 4. Используемые инструменты и реализация

Для реализации сегментации и выделения контуров использовались библиотеки ИТК и VTK. Данные библиотеки с открытым исходным кодом и распространяются под лицензией Apache 2.0. Лицензия Apache дает пользователю право использовать программное обеспечение для любых целей, свободно распространять, изменять, и распространять измененные копии, за исключением названия. Библиотека ИТК содержит большое количество реализаций для различных методов сегментации, причем сегментация может быть произведена в пространстве любой размерности. Библиотека работает по принципу конвейера, то есть данные для сегментации и любых других операций загружаются в память порциями, размер которых определяется пользователем. Таким образом, можно работать с данными очень больших объемов и производить параллельные вычисления. Например, для сегментации методом разрастания регионов достаточно выбрать начальную точку алгоритма и указать, что сегментация будет произведена в трехмерном пространстве. Для сегментации методом активных контуров необходимо задать начальный контур. Для анализа была взята томограмма мозга, которую можно бесплатно получить в Интернете в рамках проекта brainweb [14]. Объектом для сегментации и реконструкции являются желудочки мозга. Так как они представляют собой достаточно простой объект, можно применить метод разрастания регионов. Классы библиотеки `ITK::ConfidenceConnectedImageFilter` (Рис. 4, слева) и `itk::ContourExtractor2DImageFilter` (Рис. 4, справа) позволяют провести сегментацию и выделение контуров соответственно.

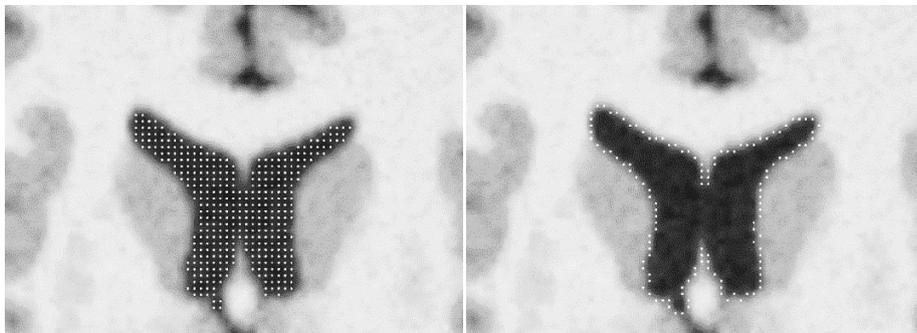


Рис. 4. Результат работы алгоритма.

После выделения точечных контуров, они аппроксимируются с использованием В-сплайнов. Для этой задачи используется библиотека Geometric Tools [15]. Данная библиотека также является открытой и содержит некоторые удобные геометрические алгоритмы, в частности алгоритм аппроксимации контура В-сплайнами. В библиотеке нет поддержки аппроксимации закрытых контуров (периодические В-сплайны), поэтому в класс `gte::BSplineCurveFit` была добавлена необходимая функциональность.

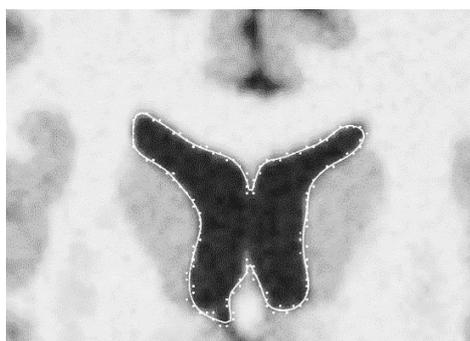


Рис. 5. Аппроксимация В-сплайном.

В-сплайны параметризованы по длине. Для простых структур, у которых форма и размеры контура не сильно отличаются от слоя к слою, этого оказывается достаточно для нахождения соответственных точек и триангуляции.

Библиотека VTK служит для визуализации результатов обработки изображений. К сожалению, на данный момент, она не содержит механизмов триангуляции параллельных контуров, и, тем более, разрешения неоднозначностей.

Реализацию второго подхода рассмотрим на примере трехмерной прямоугольной сетки 6x5x5 с известными значениями поля в узлах. Для построения модели сначала рассматриваются 2D сетки, ортогональные оси Z.

Для текущей 2D сетки, введем нумерацию узлов (совпадает с нумерацией вертикальных ребер) и горизонтальных ребер (Рис. 6).

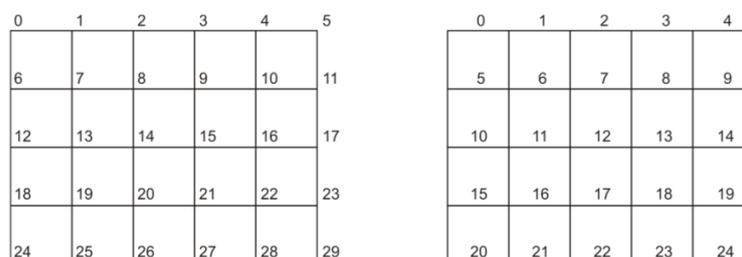


Рис. 6. Нумерация узлов

Для каждого ребра значения поля в инцидентных узлах сравниваются с изозначением. Если в одном узле значение больше, а в другом – меньше изозначения, то через это ребро проходит вершина поверхности. В этом случае интерполируется положение вершины (Рис. 7). Возможно, изозначение совпадет со значением поля в узле сетки. В этом случае положение вершины будет точно на узле.

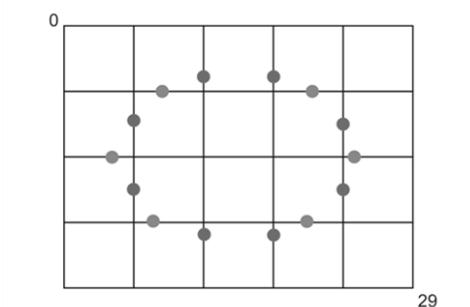
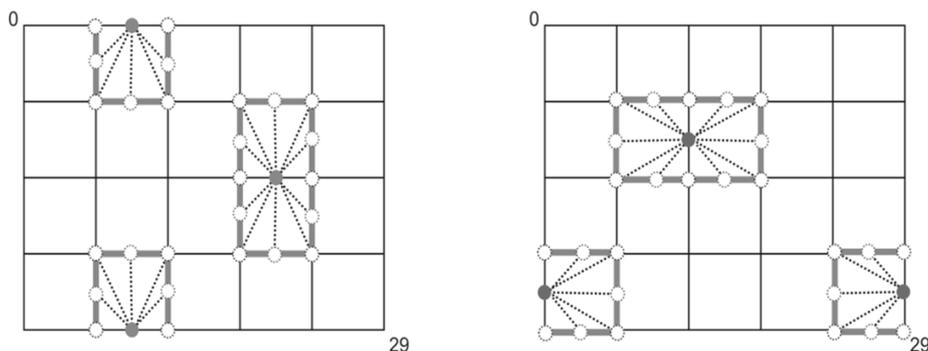


Рис. 7. Интерполированные положения вершин.

Таким образом, алгоритм работает с тремя типами вершин: 1) Вершины, лежащие на горизонтальных линиях сетки; 2) Вершины, лежащие на вертикальных линиях сетки; 3) Вершины, лежащие в узлах сетки. Вместе с типом вершины, соответствующий номер ребра определяет ее положение на сетке. В дальнейшем, номер будет использоваться при вычислении соседних вершин.

Для определения соседних вершин предлагается использовать критерий: связывать вершины, находящиеся на ближайших ребрах (Рис. 7). Для горизонтальных и вертикальных вершин можно выделить три возможных комбинации размещения соседей (Рис. 8).



**Рис. 8.** Вычисление связей для вершин. Для каждой вершины обозначена окрестность, на границе которой возможные вершины являются соседними.

Обращение к соответствующим ребрам для проверки наличия на них соседних вершин осуществляется через введенную нумерацию вертикальных (узловых) и горизонтальных вершин. Например, на рисунках 9 и 10 изображен способ вычисления номеров ребер для вершины на ребре J.

$J - W - 1$	$J - W$	$J - W + 1$
$J - 1$	$J$	$J + 1$
$J + W - 1$	$J + W$	$J + W + 1$

**Рис. 9.** Вычисление номеров нижнего и верхнего ребра, W – ширина сетки (в ребрах)

$J + R - W - 1$	$J$	$J + R - W$
$J + R$		$J + R + 1$

**Рис. 10.** Вычисление номеров для левых и правых ребер,  $R = J / W$

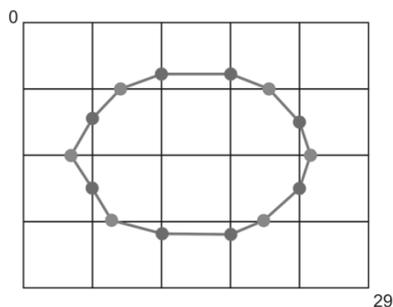


Рис. 11. Построенный контур

Таким образом, строим последовательность контуров, ортогональных Z (Рис. 12).

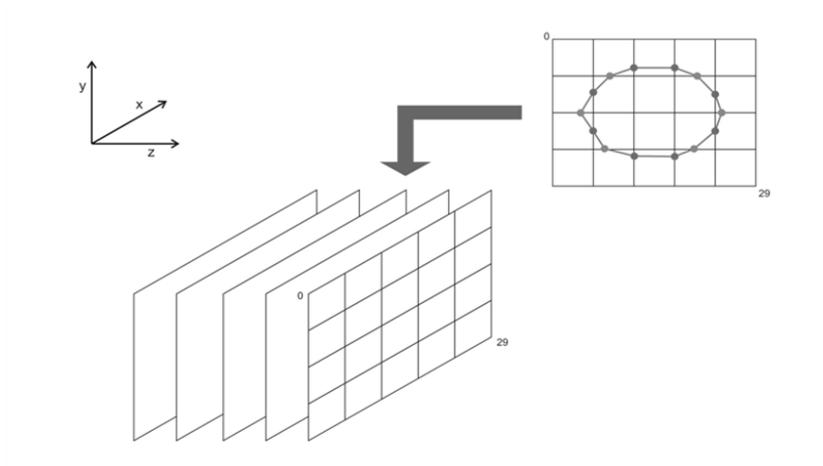
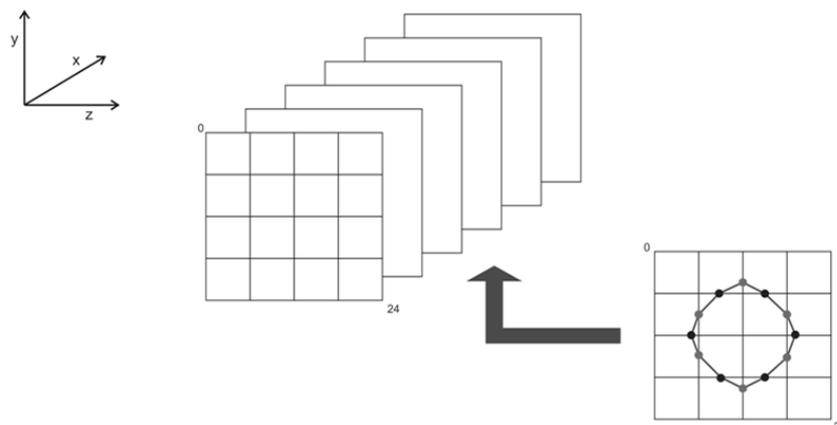


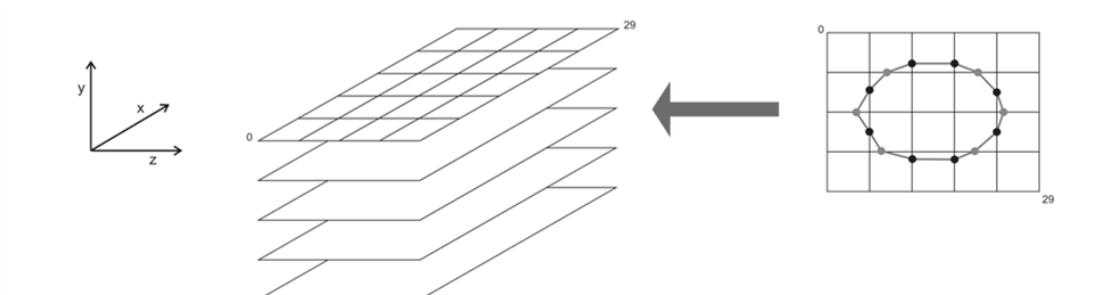
Рис. 12. Последовательность контуров, ортогональных оси Z

Можно заметить, что вертикальные и узловые вершины, принадлежащие сеткам, ортогональным Z, будут принадлежать и сеткам, ортогональным X. Эти вершины (вместе с информацией об их связях) переносятся на сетки, ортогональные X (предполагается использование индекса вершины или указателя), и там, аналогично, будут являться вертикальными или узловыми. Для переноса, выполняется обращение к соответствующим ребрам с помощью нумерации. Также, вычисляются новые вершины (На рисунке 13 – горизонтальные). После этого применяется описанный ранее алгоритм вычисления соседних вершин к каждой сетке.



**Рис. 13.** Последовательность контуров, ортогональных оси X

На последнем этапе вычислять новые вершины не требуется – они вычислены на предыдущих этапах. Их необходимо правильно перенести – вершины, вычисленные на первом этапе – останутся горизонтальными, а вершины со второго этапа – были горизонтальными, но станут вертикальными. Аналогично, к каждой сетке применяется алгоритм вычисления связей (Рис. 14)



**Рис. 14.** Последовательность контуров, ортогональных оси Y

В полученном графе, степень вершины и количество ребер в каждой внутренней грани, как правило, ограничены константой. Это дает возможность триангулировать данный граф, например, посредством перебора инцидентных граней для каждой вершины.

## 5. Результаты

Для первого подхода, на четырехъядерном процессоре AMD A-10 5745M 2.10GHz с 12 Гб оперативной памяти были получены следующие результаты.

**Таблица 1.** Результаты сегментации

Число итераций	Время, 1 ядро	Время, 2 ядра	Время, 4 ядра	Ускорение, 2 ядра	Ускорение, 4 ядра
5	2,005	1,753	1,333	1,315	1,504
10	4,367	3,608	3,002	1,202	1,455
15	6,713	5,840	4,320	1,352	1,554

Как видно из таблицы 1, распараллеливание сегментации дает небольшое ускорение. Это связано в первую очередь с тем, что для сегментации используется библиотека ГТК и распараллеливание производилось встроенными средствами самой библиотеки. Несмотря на то, что доля паралл-

лельных вычислений высока, было получено небольшое ускорение. В дальнейшем, планируется распараллелить этот процесс с помощью OpenMP, что должно повысить производительность.

**Таблица 2.** Результаты: выделение контуров, аппроксимация и триангуляция (для всех слоев)

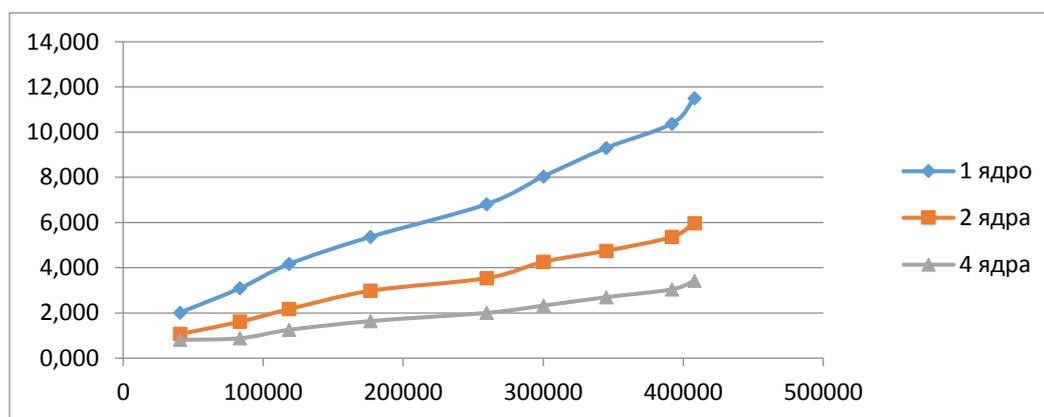
Число точек сплайна	Время, сек (1 ядро)	Время, сек (2 ядра)	Время, сек (3 ядра)	Время, сек (4 ядра)	Ускорение, 2 ядра	Ускорение, 3 ядра	Ускорение, 4 ядра
100	3,960	2,116	1,658	1,335	1,871	2,388	2,966
300	7,121	3,969	3,415	3,063	1,794	2,085	2,325
500	10,975	6,894	5,900	4,964	1,592	1,860	2,211
1000	30,003	23,367	19,713	15,338	1,284	1,522	1,956

Ускорение, полученное для данной фазы, согласуется с теоретическими оценками для доли параллельного кода около 75% (закон Амдала). В дальнейшем, планируется оптимизировать алгоритм для повышения доли параллельных вычислений.

Для второго подхода, на двухъядерном процессоре Intel®Core™ i5 750 2.66 GHz с 4 Гб оперативной памяти были получены следующие результаты.

**Таблица 3.** Таблица производительности для алгоритма вычисления сетки контуров

Размер томограммы	Количество вершин	Время, 1 ядро	Время, 2 ядра	Время, 4 ядра	Ускорение, 2 ядра	Ускорение, 4 ядра
100x512x554	40617	2,010	1,073	0,810	1,874	2,480
150x512x554	83458	3,088	1,616	0,884	1,910	3,492
200x512x554	118574	4,161	2,171	1,255	1,917	3,316
250x512x554	176668	5,360	2,983	1,641	1,797	3,267
300x512x554	259546	6,806	3,540	2,005	1,922	3,395
350x512x554	300297	8,036	4,263	2,327	1,885	3,454
400x512x554	345159	9,299	4,750	2,701	1,958	3,443
450x512x554	391994	10,373	5,362	3,044	1,935	3,408
500x512x554	408058	11,498	5,972	3,420	1,925	3,362



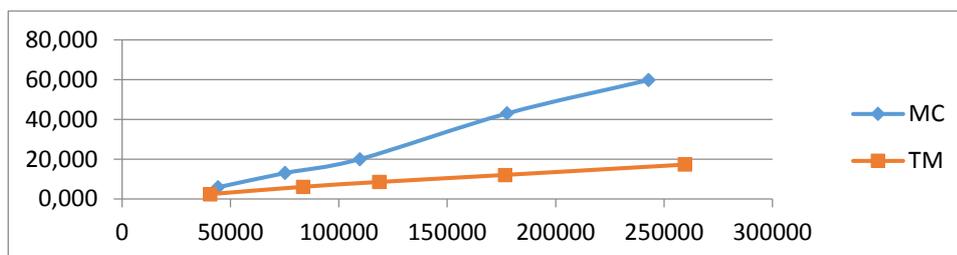
**Рис. 15.** График производительности, ось X – количество вершин, Y – время

Долю последовательного кода в данном алгоритме можно оценить отношением объема томограммы к числу полученных вершин. Таким образом, из-за небольшого количества вершин в изоповерхности по сравнению с количеством обрабатываемых ребер, и как следствие, малой долей последовательного кода, теоретическое ускорение, по закону Амдала, почти равно числу процессоров. Однако, на практике, из-за неравномерной загрузки и наличия кэш-промахов в функции вычисления связей, полученное ускорение оказалось меньше теоретического.

Далее показаны таблица и график для сравнения с реализацией Marching Cubes [7], дополненной оптимизированной процедурой поиска общих вершин для соседних кубических ячеек.

**Таблица 4.** Сравнение с реализацией Marching Cubes

Marching Cubes		Построение модели с триангуляцией	
Количество вершин	Время	Количество вершин	Время
44188	5,940	40617	2,423
75174	13,100	83458	6,117
109627	20,083	118574	8,560
177577	43,046	176668	12,112
242750	59,804	259546	17,301



**Рис. 16.** Сравнение производительности, ось X – количество вершин, Y – время

Таким образом, предлагаемый подход является наиболее подходящим для случаев, когда на выходе необходимо получить не только набор треугольников, но и информацию о вершинах и их связях.

## 6. Заключение

В данной статье представлены два подхода, применяющиеся для реконструкции поверхностей произвольного топологического типа. Вначале реконструировались контуры поверхности. Показано, как наличие ортогональных плоскостей позволяет уточнять текущие и строить новые контуры. В рамках первого подхода, трехмерная сегментация органов и автоматическое выделение контуров производились с помощью библиотеки ИТК. С целью получения гладкого параметризованного представления выполнена аппроксимация контуров на каждом слое периодическими В-сплайнами третьей степени. Во втором подходе, реконструкция поверхности выполнялась за счет построения ее графа. Вычисление координат вершин поверхности производилось аналогично алгоритму Marching Cubes [5], развитого новой процедурой вычисления связей. Результат триангуляции контуров для первого подхода показан на рисунке 17 (слева), второго – 17 (справа).



**Рис. 17.** Результат триангуляции: желудочков мозга (300 точек в контуре на каждом слое) – для первого подхода (слева); височно-нижнечелюстного сустава – для 2-го подхода на основе контурной сетки (справа).

В процессе реконструкции, плоскости обрабатываются независимо. Это позволило получить достаточно высокие ускорения в параллельных версиях алгоритмов.

В дальнейшем, после получения замкнутой поверхности в рамках первого подхода, будет произведена оценка нормалей во всех точках, участвовавших в триангуляции. Затем будет применена технология криволинейных PN-треугольников для уточнения поверхности. Используя средства библиотеки ИТК, можно рассчитывать площади построенных поверхностей и их объемы. Также планируется использовать аппроксимацию всего набора контуров с помощью B-сплайновых поверхностей.

Для структуры, полученной во втором подходе, будет разработан более эффективный алгоритм триангуляции, позволяющий получить приближение поверхности криволинейными PN-треугольниками [10].

## Литература

1. Wang D., Hassan O., Morgan K., Weatherill N. P. Efficient Surface Reconstruction from Contours Based on Two Dimensional Delaunay Triangulation // *Journal for Numerical Methods in Engineering*. 2006. Vol. 65, No. 5. P. 734-751.
2. Sederberg T., Hong M., Kaneda K., Klimaszewski K. Triangulation of Branching Contours using Area Minimization // *Internat. J. Comput. Geom. Appl.* 1998. Vol. 8, No. 4, P. 389–406.
3. Meyes D., Skinner S., Sloan K. Surface from Contours // *ACM Transactions on Graphics* 1992. Vol. 11, No. 3. P. 228 –258.
4. Cong G., Parvin B. Robust and Efficient Surface Reconstruction from Contours // *The Visual Computer* 2001. Vol. 17, No. 4. P. 199–208.
5. Lorensen W. E., Cline H. E. Marching Cubes: A high Resolution 3D Surface Construction Algorithm // *Computer Graphics*. 1987. Vol. 21, No. 4. P. 163–169
6. Nielson G. M., Hamann B. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. *Proc. 2nd conference on Visualization (VIS' 91)*. 1991. P. 83–91.
7. Bourke Paul. Polygonising a scalar field: 1994. URL: <http://paulbourke.net/geometry/polygonise/>, (дата обращения 14.02.2015).
8. Chernyaev E.V. Marching Cubes 33: Construction of Topologically Correct Isosurfaces. Technical Report CERN CN 95-17, CERN, 1995.
9. Lewiner T., Lopes H., Vieira A.W., Tavares G. Efficient Implementation of Marching Cubes' Cases with Topological Guarantees // *Graphics Tools*. 2003. Vol. 8, No. 2. P. 1-15.
10. Vlachos A., Peters J., Boyd C., Mitchell J. L. Curved PN Triangles. *ACM Symposium on Interactive 3D Graphics* 2001. P. 159-166.
11. Manson J., Smith J., Schaefer S. Contouring Discrete Indicator Functions // *Computer Graphics Forum*. 2011. Vol. 30, No. 2. P. 385-39.
12. Dietrich C. A., Scheidegger C. E., Schreiner J., Comba J. L., Nedel L. P., Silva C. T. Edge Transformations for Improving Mesh Quality of Marching Cubes // *IEEE TVCG*. 2009. Vol. 15, No. 1. P. 150–159.
13. Лукашевич П. В., Залесский Б. А., Недзьведь А. М. Восстановление поверхности трехмерного объекта по обводкам его сечений // *Искусственный интеллект*. 2010. № 2. С. 1-8.
14. BrainWeb: Simulated Brain Database. URL: <http://brainweb.bic.mni.mcgill.ca/brainweb/> (дата обращения 12.02.2016).
15. GeometricTools. URL: <http://www.geometrictools.com/> (дата обращения 12.02.2016)

## Parallel reconstruction method in orthogonal planes

M. M. Novozhilov, M.P. Kochukov

Lobachevsky State University of Nizhni Novgorod

In this paper we solve the reconstruction problem of external and internal borders of organs with a complex spatial structure (brain, complex fracture). Reconstruction performed in orthogonal planes to resolve ambiguities. First approach is the following: at the beginning we perform the 3-dimensional segmentation using region growing algorithm. Then we solve the 2-dimensional contour reconstruction problem for the two orthogonal cuts. In the second case, the algorithm reconstructs contours in planes which are orthogonal to the coordinate axis, and combines them into graph of surface. Further, in the both approaches, resulting set of contours are triangulating. Due to the independent processing of the planes in the parallel implementation achieved a sufficiently high acceleration.

*Keywords:* reconstruction, computed tomography, segmentation, contour, ambiguities.

### References

1. Wang D., Hassan O., Morgan K., Weatherill N. P. Efficient Surface Reconstruction from Contours Based on Two Dimensional Delaunay Triangulation // Journal for Numerical Methods in Engineering. 2006. Vol. 65, No. 5. P. 734-751.
2. Sederberg T., Hong M., Kaneda K., Klimaszewski K. Triangulation of Branching Contours using Area Minimization // Internat. J. Comput. Geom. Appl. 1998. Vol. 8, No. 4, P. 389-406.
3. Meyes D., Skinner S., Sloan K. Surface from Contours // ACM Transactions on Graphics 1992. Vol. 11, No. 3. P. 228 -258.
4. Cong G., Parvin B. Robust and Efficient Surface Reconstruction from Contours // The Visual Computer 2001. Vol. 17., No. 4. P. 199-208.
5. Lorensen W. E., Cline H. E. Marching Cubes: A high Resolution 3D Surface Construction Algorithm // Computer Graphics. 1987. Vol. 21, No. 4. P. 163-169
6. Nielson G. M., Hamann B. The Asymptotic Decider: Resolving the Ambiguity in Marching Cubes. Proc. 2nd conference on Visualization (VIS' 91). 1991. P. 83-91.
7. Bourke Paul. Polygonising a scalar field: 1994. URL: <http://paulbourke.net/geometry/polygonise/>, (accessed: 14.02.2015).
8. Chernyaev E.V. Marching Cubes 33: Construction of Topologically Correct Isosurfaces. Technical Report CERN CN 95-17, CERN, 1995.
9. Lewiner T., Lopes H., Vieira A.W., Tavares G. Efficient Implementation of Marching Cubes' Cases with Topological Guarantees // Graphics Tools. 2003. Vol. 8, No. 2. P. 1-15.
10. Vlachos A., Peters J., Boyd C., Mitchell J. L. Curved PN Triangles. ACM Symposium on Interactive 3D Graphics 2001. P. 159-166.
11. Manson J., Smith J., Schaefer S. Contouring Discrete Indicator Functions // Computer Graphics Forum. 2011. Vol. 30, No. 2. P. 385-39.
12. Dietrich C. A., Scheidegger C. E., Schreiner J., Comba J. L., Nedel L. P., Silva C. T. Edge Transformations for Improving Mesh Quality of Marching Cubes // IEEE TVCG. 2009. Vol. 15, No. 1. P. 150-159.
13. Lukashovich P. V., Zalesskiy B. A., Nedz'ved' A. M. Vosstanovlenie poverkhnosti trekhmernogo ob"ekta po obvodkam ego secheniy [Surface Reconstruction of Three-Dimensional Object by Strokes of its Cross-Sections] // Iskusstvennyy intellekt [Artificial intelligence]. 2010. No. 2. P.1-8.
14. BrainWeb: Simulated Brain Database. URL: <http://brainweb.bic.mni.mcgill.ca/brainweb/> (accessed: 12.02.2016).

15. GeometricTools. URL: <http://www.geometrictools.com/> (accessed: 12.02.2016).