

**Реализация иерархической модели данных в системе компьютерного планирования хирургических операций в ортопедии**

**© Гончаренко В.Г., Архипов В.И., Тузиков А.В.**

Объединенный институт проблем информатики

НАН Беларуси

ул. Сурганова, 6, г. Минск, 220012, Беларусь

E-MAIL: [vasily@mpen.bas-net.by](mailto:vasily@mpen.bas-net.by), [arkhipau@gmail.com](mailto:arkhipau@gmail.com), [tuzikov@newman.bas-net.by](mailto:tuzikov@newman.bas-net.by)

**Abstract.** Hierarchical model describing transformation results of large images is discussed. A procedure of binary image resection by a polygon ("scissor") selected by a user is described. All connected components, completely crossed by the polygon, are resected onto several parts.

## **ВВЕДЕНИЕ**

В 1972 году в связи с появлением компьютерной томографии появилась возможность создавать трехмерные изображения внутренних органов и тканей. Эти изображения строятся в виде последовательности двумерных слоев, составляющих трехмерное объемное изображение. Каждый слой томографического изображения независим от других, при этом яркость каждой точки на томографическом изображении характеризует плотность соответствующей анатомической области, что позволяет по яркости точки с определенной достоверностью классифицировать тип ткани (костная, мышечная, жировая и т.д.). Все это позволяет использовать трехмерные томографические изображения для построения объемных моделей внутренних органов и применять их для планирования хирургических операций при помощи компьютера [1, 2].

Планирование операций выполняется предварительно для вычисления параметров основных процедур хирургической операции в целях подготовки к реальной операции. Объектами операции являются органы и ткани пациента, имеющие выраженную патологию и требующие хирургического вмешательства. Обычно последовательность действий во время хирургической операции определяется хирургом самостоятельно на основании изучения состояния анатомических объектов по результатам предоперационного планирования [3].

Компьютерное планирование выполняется при помощи специальных программных средств путем воспроизведения на компьютере всей последовательности действий, составляющих хирургическую операцию. Компьютерное планирование операции можно назвать виртуальной операцией. Виртуальная операция не несет никакого риска для пациента и, в то же время, позволяет легко определить действия и их параметры, необходимые для проведения реальной операции. Для определения геометрических параметров костей и суставов анализируются цифровые изображения.

Процесс компьютерного планирования хирургических операций в ортопедии обычно состоит из 8 этапов:

- 1) сканирование пациента с помощью компьютерного томографа;
- 2) предварительная обработка томографического изображения и построение иерархической модели;
- 3) сегментация и классификация объектов планирования;
- 4) оценка визуальной информации;
- 5) измерение моделей костей;
- 6) оценка результатов измерений и принятие решений относительно дальнейших действий;
- 7) виртуальная остеотомия;
- 8) виртуальный остеосинтез.

Подробное описание этих этапов приводится в работе [3]. Здесь рассматривается понятие иерархической модели, а также задача разрезания объектов иерархической модели в процессе виртуальной остеотомии.

### 1. КОНЦЕПЦИЯ ИЕРАРХИЧЕСКОЙ МОДЕЛИ

В целях организации работы с различными типами преобразований отдельных частей изображений была спроектирована так называемая «иерархическая модель». В связи с тем, что различные части изображения могут потребовать различных методов их преобразования, предложено использование одновременно нескольких различных способов выделения объектов на изображении.

Иерархическая модель представляет собой дерево, вершинами которого являются объекты, получившиеся в результате преобразований различных участков исходного изображения. В корне дерева находится исходное изображение. В дереве выделяется один из объектов, который считается текущим – с ним пользователь работает в данный момент. Так, например, сегментация изображения всегда применяется к части текущего объекта, определяемой областью интереса в виде прямоугольного параллелепипеда. Объекты иерархической модели с точки зрения программной реализации представляют собой объекты класса `C++ KPOHierarchicalObject`. Это абстрактный класс, имеющий ряд чистых виртуальных функций, таких как `GetObjectVolume` и `GetObjectSize`. Такая концепция позволяет реализовать два типа операций: хорошо параметризуемые и плохо параметризуемые. Объекты, являющиеся результатом хорошо параметризуемой операции, можно реконструировать за приемлемое время, используя в качестве параметров процедуры реконструкции некоторые атрибуты, требующие мало памяти по сравнению с самим объектом. Таким образом, объекты, получившиеся в результате хорошо параметризуемой операции, не хранят в явном виде значения вокселей соответствующего им участка. Вместо этого они хранят атрибуты, которые позволяют строить массив вокселей, используя данные их родителей и исходного изображения. Так, например, пороговую сегментацию можно трактовать как хорошо параметризуемую операцию, атрибутами которой являются пороговые уровни. Значения вокселей (бинарного изображения) можно построить из исходного

изображения, используя атрибуты (значения пороговых уровней). Такая схема позволяет освободить память, выполняя построение массива вокселей только тогда, когда он необходим, после чего он удаляется из оперативной памяти. Объекты, являющиеся результатами плохо параметризуемой операции, используются, когда построение массива вокселей объекта нельзя произвести за приемлемое время, или когда атрибуты занимают больше памяти, чем сам результат операции. В качестве примера плохо параметризуемой операции может выступать ручная сегментация.

Каждый объект иерархической модели имеет свои размеры, а также координаты и матрицу ориентации относительно своего объекта-родителя. Это позволяет быстро выполнять перенос и вращения всей ветви дерева иерархической модели.

Объектами иерархической модели могут быть не только результаты сегментации изображения, но и результаты промежуточных преобразований (например, фильтрации). Предложенная концепция хорошо и плохо параметризуемых операций позволяет учесть множество различных преобразований участков изображения.

После загрузки исходного изображения оно помещается в корень иерархической модели. Как только пользователь выполнит сегментацию какой-либо части исходного изображения, отсегментированное изображение помещается в иерархическую модель как потомок корневого элемента (т.е. исходного изображения). Обычно после сегментации выполняется классификация отсегментированных объектов – они помещаются на третий уровень иерархической модели (рисунок 1). Количество классифицированных объектов соответствует количеству уникальных объектов, имеющих на классифицируемом участке текущего объекта. Пользователь в любое время может выбрать текущий объект и те объекты, которые необходимо отображать (текущий объект всегда является отображаемым объектом). Если отображается какой-либо объект, то в это же время его потомки и предки отображаться не могут. Все действия по выбору объектов пользователь выполняет при помощи менеджера объектов – специального диалогового окна, где отображена древовидная структура текущей иерархической модели.

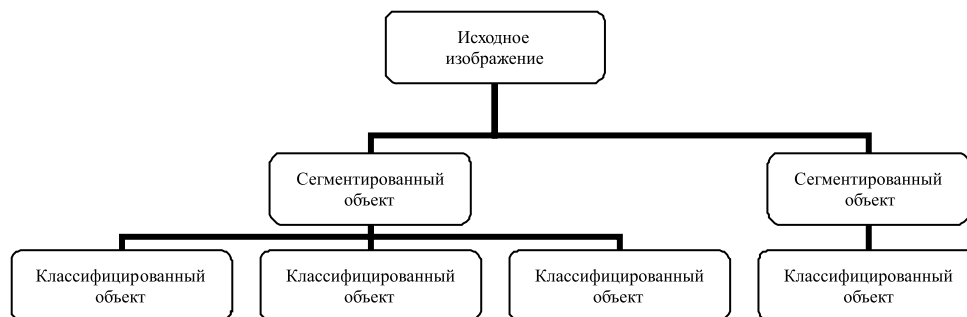


Рис. 1. Примерная структура обычной иерархической модели

## 2. ВЫДЕЛЕНИЕ СВЯЗНЫХ КОМПОНЕНТ И РАЗРЕЗАНИЕ ОБЪЕКТОВ

Одной из важных плохо параметризуемых операций над бинарными объектами иерархической модели в системе компьютерного планирования выступает разрезание объектов. Эта операция имеет большое значение при планировании остеотомии [2].

В процессе выполнения виртуальной операции появляется необходимость разрезать текущий объект на две или несколько частей замкнутой ломаной линией  $\rho = (S_1, \dots, S_k)$ , образующей  $k$ -угольник, все вершины которого лежат в одной плоскости  $\Omega(\rho)$ . Эта ломаная образует активную часть плоскости разрезания, называемую «резцом» (рисунок 2). Бинарный объект состоит из некоторого множества связных компонент. При использовании резца разрезаются только те связные компоненты, которые полностью пересекаются активной частью плоскости разреза. Сама процедура разрезания использует операцию выделения связных компонент и является, по сути, её модификацией.

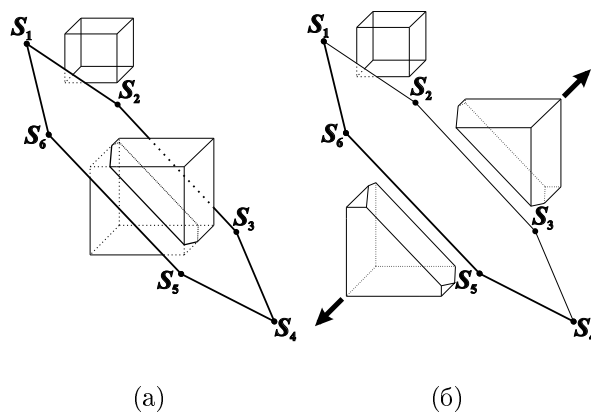
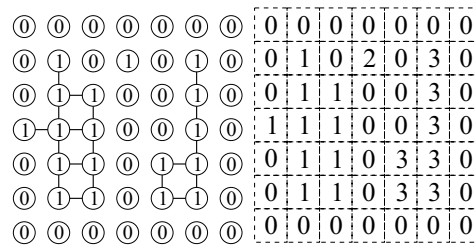


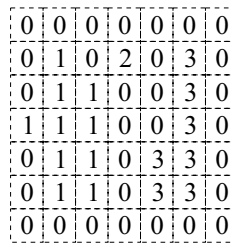
Рис. 2. Разрезание объекта инструментом «резец», в котором активная часть плоскости разрезания задаётся шестиугольником  $(S_1, \dots, S_6)$ , и разрезаются только объекты, попадающие в активную часть плоскости разреза: а) вид объектов до разрезания; б) вид объектов после разрезания

Рассмотрим, как определяются связи между соседними вокселями трехмерного изображения. Если инструмент «резец» не задан, либо он не пересекается с текущим объектом, то все вокселя, принадлежащие объекту и являющиеся соседями в используемой метрике, имеют связи между собой (рисунок 3(а)). Как только задается «резец», пересекающий текущий объект, вычисляются точки пересечения активной части плоскости резца  $\rho$  и связей между вокселями, образующими объект (см. рисунок 3(б)). «Резец» разрезает только ребра решетки, пересекаемые активной частью плоскости разрезания  $\rho$ .

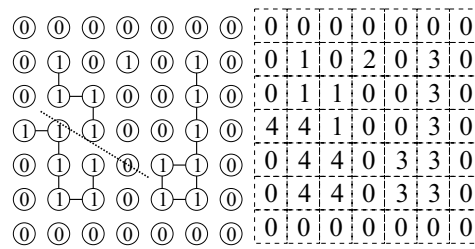
Опишем более подробно процесс построения связей элементов изображения при использовании инструмента «резец». Пусть задано бинарное изображение  $B$ , на котором объекты имеют значение 1, а фон – значение 0. Обозначим через  $\mathbf{n}$  вектор



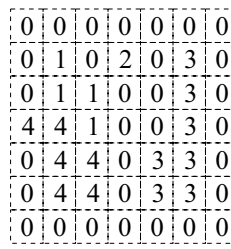
(а)



(б)



(в)



(г)

Рис. 3. Пример разрыва связей пикселей связной компоненты инструментом «резец» с использованием четырехсвязной метрики; а) построение связей между пикселями при выделении связных компонент; б) связные компоненты изображения, помеченные соответствующими номерами; в) разрыв связей между пикселями (пунктирная линия); г) связные компоненты, получившиеся при разрезании объекта.

нормали плоскости разрезания. Введем функцию  $\delta_{\Omega}(\mathbf{p})$  положения вокселя  $\mathbf{p}$  относительно плоскости разрезания  $\Omega(\rho)$ :

$$\delta_{\Omega}(\mathbf{p}) = \mathbf{n} \cdot (\mathbf{S}_1 - \mathbf{p}), \tag{1}$$

где « $\cdot$ » обозначает скалярное произведение двух векторов.

Если воксел  $\mathbf{p}$  принадлежит объекту и у него имеется сосед  $\mathbf{q}$ , принадлежащий тому же объекту, то можно определить, пересекает ли плоскость  $\Omega(\rho)$  связь между  $\mathbf{p}$  и  $\mathbf{q}$ . Связь пересекается, если значение функции  $\delta_{\Omega}$  для вокселей  $\mathbf{p}$  и  $\mathbf{q}$  имеет разный знак:

$$\delta_{\Omega}(\mathbf{p})\delta_{\Omega}(\mathbf{q}) < 0. \tag{2}$$

Связь между вокселом  $\mathbf{p}$  и его соседом  $\mathbf{q}$  пересекается «резцом»  $\rho$  и разрывается, если она пересекается плоскостью  $\Omega(\rho)$ , и точка  $\mathbf{I}_{\mathbf{p},\mathbf{q}}$  пересечения прямой, проходящей через  $\mathbf{p}$  и  $\mathbf{q}$ , и плоскости разрезания  $\Omega(\rho)$  находится внутри многоугольника  $\rho$ .

Координаты точки  $\mathbf{I}_{\mathbf{p},\mathbf{q}}$  вычисляются по формуле:

$$\mathbf{I}_{\mathbf{p},\mathbf{q}} = \mathbf{p} + (\mathbf{q} - \mathbf{p}) \frac{\delta_{\Omega}(\mathbf{p})}{\mathbf{n} \cdot (\mathbf{q} - \mathbf{p})}. \quad (3)$$

В случае, если знаменатель в (3) равен 0 (прямая, соединяющая  $\mathbf{p}$  и  $\mathbf{q}$ , параллельна плоскости разрезания  $\Omega(\rho)$ ), связь между  $\mathbf{p}$  и  $\mathbf{q}$  считается неразорванной.

Перейдем к двумерным координатам и обозначим через  $\rho' = (\mathbf{S}'_1, \dots, \mathbf{S}'_k)$  двухмерный многоугольник, координаты вершин которого определяются следующим образом:

$$\begin{cases} \mathbf{S}'_i{}^x = \mathbf{S}_i \cdot \mathbf{i}, \\ \mathbf{S}'_i{}^y = \mathbf{S}_i \cdot \mathbf{j}, \end{cases} \quad (4)$$

где  $i$  изменяется от 1 до  $k$ , а  $\mathbf{i}, \mathbf{j}$  – базисные векторы, определяющиеся следующим образом:

$$\begin{cases} \mathbf{i} = \frac{\mathbf{S}_2 - \mathbf{S}_1}{\|\mathbf{S}_2 - \mathbf{S}_1\|}, \\ \mathbf{j} = \mathbf{i} \times \mathbf{n}. \end{cases}$$

Обозначим через  $\mathbf{I}'_{\mathbf{p},\mathbf{q}} = (\mathbf{I}'_{\mathbf{p},\mathbf{q}}{}^x, \mathbf{I}'_{\mathbf{p},\mathbf{q}}{}^y)$  точку  $\mathbf{I}_{\mathbf{p},\mathbf{q}}$  в базисе  $(\mathbf{i}, \mathbf{j})$ :

$$\begin{cases} \mathbf{I}'_{\mathbf{p},\mathbf{q}}{}^x = \mathbf{I}_{\mathbf{p},\mathbf{q}} \cdot \mathbf{i}, \\ \mathbf{I}'_{\mathbf{p},\mathbf{q}}{}^y = \mathbf{I}_{\mathbf{p},\mathbf{q}} \cdot \mathbf{j}. \end{cases}$$

Рассмотрим задачу определения принадлежности точки многоугольнику. Существует несколько различных способов решения этой задачи [4]. В данном случае, когда количество сторон многоугольника невелико, реализовано следующее решение. Вначале определяется некоторый базовый луч, исходящий из точки  $\mathbf{I}'_{\mathbf{p},\mathbf{q}}$ . Затем определяется количество ребер многоугольника  $\rho'$  пересекаемых этим лучом. Если количество пересечений нечетное, то точка  $\mathbf{I}'_{\mathbf{p},\mathbf{q}}$  лежит внутри многоугольника  $\rho'$  и, следовательно, связь между  $\mathbf{p}$  и  $\mathbf{q}$  разрывается. Иначе связь между  $\mathbf{p}$  и  $\mathbf{q}$  сохраняется.

В случае, если многоугольник  $\rho'$  является выпуклым, имеется более простое решение данной задачи. Вообще говоря, в большинстве случаев резец вполне можно сделать выпуклым многоугольником, не теряя при этом универсальности его применения и выигрывая в удобстве его использования. Достаточно удобно пользоваться резцом, представляющим собой выпуклый 4-угольник. Точка  $\mathbf{I}'_{\mathbf{p},\mathbf{q}}$  лежит внутри выпуклого многоугольника  $\rho'$ , если точка  $\mathbf{I}'_{\mathbf{p},\mathbf{q}}$  находится всегда с одной стороны по отношению ко всем сторонам многоугольника при обходе их в одном направлении. Другими словами, точка  $\mathbf{I}'_{\mathbf{p},\mathbf{q}}$  лежит внутри выпуклого многоугольника  $\rho'$ , если величина  $\mu_{\mathbf{p},\mathbf{q}}(i, i+1)$  имеет одинаковый знак для всех  $i = 1, 2, \dots, k$ :

$$\mu_{\mathbf{p},\mathbf{q}}(i, i+1) = (\mathbf{I}'_{\mathbf{p},\mathbf{q}}{}^y - \mathbf{S}'_i{}^y)(\mathbf{S}'_{i+1}{}^x - \mathbf{S}'_i{}^x) - (\mathbf{I}'_{\mathbf{p},\mathbf{q}}{}^x - \mathbf{S}'_i{}^x)(\mathbf{S}'_{i+1}{}^y - \mathbf{S}'_i{}^y), \quad (5)$$

здесь  $\mu_{\mathbf{p},\mathbf{q}}(k, k+1) = \mu_{\mathbf{p},\mathbf{q}}(k, 1)$ .

При реализации данной схемы вычислений необходимо учесть, что для каждого помещаемого в очередь воксела  $\mathbf{p}$  целесообразно заранее вычислять и сохранять значение  $\delta_{\Omega}(\mathbf{p})$ . Его можно использовать позже для проверки, пересекает ли плоскость  $\Omega(\rho)$  связь между выбранным из очереди вокселем  $\mathbf{p}'$  и его соседом  $\mathbf{q}$ . Вычисленное значение  $\delta_{\Omega}(\mathbf{q})$  также желательно сохранять вместе с помещаемым в очередь вокселем  $\mathbf{q}$ .

Для случая регулярной метрики необходимо принимать во внимание расположение соседних вокселей. Например, при использовании шестисвязной метрики для трехмерного изображения каждый воксел содержит по 6 соседей (кроме вокселей, расположенных на границе изображения). Расстояние между текущим вокселем и его соседями равно 1. Причем, положение каждого соседа отличается от положения текущего вокселя только одной координатой. Соответственно, выражения для  $\delta_{\Omega}(\mathbf{q})$  и  $\mathbf{I}_{\mathbf{p},\mathbf{q}}$  можно вычислять для каждого соседа в зависимости от его положения (таблица 1). Здесь  $\mathbf{p}^x, \mathbf{p}^y, \mathbf{p}^z$  – координаты вектора  $\mathbf{p}$  в трехмерном пространстве. Аналогично обозначаются координаты вектора  $\mathbf{n}$ .

Таблица 1. Оптимизация вычислений в случае шестисвязной метрики на трехмерной регулярной решетке

Координаты соседа $\mathbf{q}$ вокселя $\mathbf{p}$	Значение $\delta_{\Omega}(\mathbf{q})$	Значение $\mathbf{I}_{\mathbf{p},\mathbf{q}}$
$(\mathbf{p}^x - 1, \mathbf{p}^y, \mathbf{p}^z)$	$\delta_{\Omega}(\mathbf{p}) + \mathbf{n}^x$	$(\mathbf{p}^x + \delta_{\Omega}(\mathbf{p})/\mathbf{n}^x, \mathbf{p}^y, \mathbf{p}^z)$
$(\mathbf{p}^x + 1, \mathbf{p}^y, \mathbf{p}^z)$	$\delta_{\Omega}(\mathbf{p}) - \mathbf{n}^x$	$(\mathbf{p}^x + \delta_{\Omega}(\mathbf{p})/\mathbf{n}^x, \mathbf{p}^y, \mathbf{p}^z)$
$(\mathbf{p}^x, \mathbf{p}^y - 1, \mathbf{p}^z)$	$\delta_{\Omega}(\mathbf{p}) + \mathbf{n}^y$	$(\mathbf{p}^x, \mathbf{p}^y + \delta_{\Omega}(\mathbf{p})/\mathbf{n}^y, \mathbf{p}^z)$
$(\mathbf{p}^x, \mathbf{p}^y + 1, \mathbf{p}^z)$	$\delta_{\Omega}(\mathbf{p}) - \mathbf{n}^y$	$(\mathbf{p}^x, \mathbf{p}^y + \delta_{\Omega}(\mathbf{p})/\mathbf{n}^y, \mathbf{p}^z)$
$(\mathbf{p}^x, \mathbf{p}^y, \mathbf{p}^z - 1)$	$\delta_{\Omega}(\mathbf{p}) + \mathbf{n}^z$	$(\mathbf{p}^x, \mathbf{p}^y, \mathbf{p}^z + \delta_{\Omega}(\mathbf{p})/\mathbf{n}^z)$
$(\mathbf{p}^x, \mathbf{p}^y, \mathbf{p}^z + 1)$	$\delta_{\Omega}(\mathbf{p}) - \mathbf{n}^z$	$(\mathbf{p}^x, \mathbf{p}^y, \mathbf{p}^z + \delta_{\Omega}(\mathbf{p})/\mathbf{n}^z)$

Рассмотрим подробнее алгоритм выделения связных компонент и разрезания объектов изображения (алгоритм 1). Он работает для изображения с любой связностью. Пусть каждый воксел имеет флаг состояния просмотра (*TEMP* – еще не просматривался, *DONE* – уже просматривался и включен в состав связной компоненты). Вначале инициализируется номер текущей связной компоненты  $c \leftarrow 1$ . Затем алгоритм переходит в стадию последовательного просмотра массива бинарных вокселей. Как только обнаружится непросмотренный (имеющий флаг *TEMP*) воксел  $\mathbf{p}$ , принадлежащий объекту ( $val(\mathbf{p}) = 1$ ), он помещается в хвост очереди при помощи функции *Enqueue*( $\mathbf{p}$ ), присоединяется к текущей связной компоненте *ConnectedComponent*( $\mathbf{p}$ )  $\leftarrow c$  и ему присваивается флаг *DONE*. После этого номер текущей связной компоненты увеличивается на 1 и алгоритм переходит в стадию наращивания связной компоненты, которой принадлежит только что просмотренный воксел  $\mathbf{p}$ . При этом, из головы очереди снимается один воксел  $\mathbf{p}' \leftarrow Dequeue()$ . Затем рассматриваются все его соседи, с которыми он имеет связи. Если воксел  $\mathbf{q}$  имеет

флаг *TEMP*, принадлежит объекту и находится с обратной стороны от плоскости разрезания  $\Omega(\rho)$  относительно воксела  $\mathbf{p}'$ , то выполняется проверка на то, попадает ли точка пересечения связи вокселей  $\mathbf{p}'$  и  $\mathbf{q}$  с плоскостью  $\Omega(\rho)$  внутрь многоугольника  $\rho$ . Если связь между вокселями  $\mathbf{p}'$  и  $\mathbf{q}$  пересекается «резцом», то она удаляется. Иначе воксел  $\mathbf{q}$  добавляется в хвост очереди командой *Enqueue*( $\mathbf{q}$ ), присоединяется к связной компоненте *ConnectedComponent*( $\mathbf{p}'$ ) и ему присваивается флаг *DONE*. Такая проверка выполняется для всех соседей воксела  $\mathbf{p}'$ , с которыми у него есть связи. После того, как очередь опустеет, алгоритм опять переходит в стадию последовательного просмотра вокселей и переходит к следующему за  $\mathbf{p}$  вокселу. Процесс прекращается как только будут просмотрены все воксели объекта. В результате этой процедуры создается изображение, состоящее из вокселей со значениями индексов соответствующих им связных компонент (рисунок 3(б)). Каждая связная компонента с одинаковым значением вокселей будет образовывать отдельный объект.

---

### Алгоритм 1 Выделение связных компонент и разрезание бинарных объектов

---

**На входе:** Бинарное изображение  $B$  – массив вокселей  $\mathbf{p} \in B$  с множеством значений  $val(\mathbf{p}) \in \{0, 1\}$ ; множество  $R$  связей  $rel(\mathbf{p}, \mathbf{q}) \in R$  между вокселями  $\mathbf{p}, \mathbf{q} \in B$

**На выходе:** Бинарное изображение *ConnectedComponent*, состоящее из индексов связных компонент, получившихся после разрезания

```

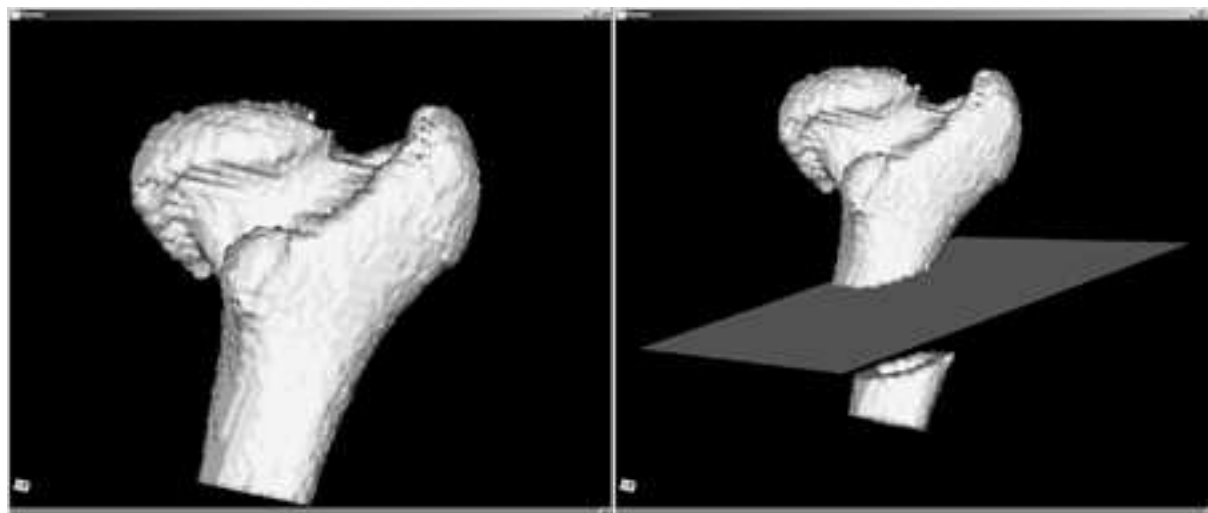
1: for all  $\mathbf{p} \in B$  do
2:    $flag(\mathbf{p}) \leftarrow TEMP$ ;
3:    $c \leftarrow 1$ ;
   {Просмотр вокселей}
4: for all  $\mathbf{p} \in B$  do
5:   if  $flag(\mathbf{p}) = TEMP$  and  $val(\mathbf{p}) = 1$  then
6:      $Enqueue(\mathbf{p})$ ;
7:      $ConnectedComponent(\mathbf{p}) \leftarrow c$ ;
8:      $flag(\mathbf{p}) \leftarrow DONE$ ;
9:      $c \leftarrow c + 1$ ;
   {Наращивание связной компоненты}
10:  while Очередь не пуста do
11:     $\mathbf{p}' \leftarrow Dequeue()$ ;
12:    for all сосед  $\mathbf{q}$  воксела  $\mathbf{p}'$  do
13:      if  $flag(\mathbf{q}) = TEMP$  and  $val(\mathbf{q}) = 1$  then
14:         $connection \leftarrow TRUE$ ;
        {Проверка на пересечение плоскостью  $\Omega(\rho)$  связи  $\mathbf{p}'$  с  $\mathbf{q}$ }
15:        if  $\delta_{\Omega}(\mathbf{p}')\delta_{\Omega}(\mathbf{q}) < 0$  then
16:          if знак  $\mu_{\mathbf{p}',\mathbf{q}}(i, i + 1)$  одинаков для всех  $i = 1, \dots, k$  then
17:             $connection \leftarrow FALSE$ ;
18:          if  $connection$  then
19:             $Enqueue(\mathbf{q})$ ;
20:             $ConnectedComponent(\mathbf{q}) \leftarrow ConnectedComponent(\mathbf{p}')$ ;
21:             $flag(\mathbf{q}) \leftarrow DONE$ ;

```

---



На рисунке 4 показан пример разрезания бедренной кости.



(a)

(б)

*Рис. 4.* Пример работы алгоритма разрезания: а) объект до разрезания; б) объект после разрезания.

### ЗАКЛЮЧЕНИЕ

В статье разработана методика иерархического моделирования, позволяющая эффективно сохранять результаты различных преобразований частей изображения. Все преобразования делятся на два типа: хорошо и плохо параметризуемые. Результаты хорошо параметризуемых операций упаковываются таким образом, что в явном виде хранятся только их параметры, а результат строится «на ходу». Эффективность такой методики основывается на том, что параметры хорошо параметризуемой операции занимают значительно меньший объем памяти, чем ее результаты. Иерархическая модель представляет собой дерево, элементами которого выступают результаты преобразований частей одного изображения. В корне этого дерева находится исходное изображение. Каждое последующее преобразование части изображения создает ветвь дерева. Это позволяет связывать все операции, выполняемые над данными одного изображения.

Разработан алгоритм выделения связных компонент и последующего разрезания бинарных объектов специальным инструментом «резец». Этот инструмент представляет собой многоугольник, все точки которого лежат в одной плоскости, называемый активной частью плоскости разрезания. Разрезаются только те связные компоненты

объекта, которые пересекаются активной частью плоскости разрезания. Предложенный инструмент «резец» позволяет выполнять разрезание бинарных объектов любой формы, включая отдельные кости таза.

#### СПИСОК ЛИТЕРАТУРЫ

1. *Handels H., Ehrhardt J., Plötz W., Pöpl S.J.* Three-dimensional planning and simulation of hip operations and computer-assisted construction of endoprostheses in bone tumor surgery. *Computer Aided Surgery*, vol. 6, Nu. 2, p. 65–76, 2007.
2. *Krivonos O. Hesser J. Maenner R. Keppler P. Gebhard F. Kinzl L. Sakalouski A.A. Sakalouski A.M.* Precise computer aided correction of bone deformities. *Proceedings of the World Congress on Medical Physics and Biomedical Engineering WC2003*, Sydney, Australia, August 2003.
3. *Hancharenka V.G., Tuzikov A.V. Arkhipau V.I., Kryvanos A.* Preoperative planning of pelvic and lower limbs surgery by CT image processing. *Proceedings of 8th International Conference on Pattern Recognition and Image Analysis PRIA'2007*, 7-13 Oct., Yoshkar-Ola, Russia, vol. 1, p. 270–274, 2007.
4. *Haines E.* Point in Polygon Strategies. *Graphics Gems IV*, ed. Paul Heckbert, Academic Press, p. 24–46, 1994.

*Статья поступила в редакцию 19.04.2008*