

1 Основные определения

Пусть v определяет трёхмерное изображение, (x, y, z) – воксели этого изображения, тогда $v(x, y, z)$ будет значением яркости вокселя (x, y, z) изображения v . К k -тому слою или $v(*, *, k)$ изображения v относятся все воксели имеющие координату $z = k$. Пусть w_k – это рабочий слой, соответствующий $v(*, *, k)$, и $w_k(x, y)$ хранит временные значения $v(x, y, k)$. Воксели фона имеют значение $v(x, y, z) = 0$.

Чтобы процесс сегментации был эффективным, необходимо учитывать не только распределение яркости в изображении, но также и пространственную взаимосвязь (соседство) вокселей. Представленный здесь алгоритм учитывает оба этих фактора и является членом класса симметричных алгоритмов выделения областей [1]. Основное свойство таких алгоритмов состоит в том, что они выделяют такую же область в независимости от того, из какой точки начнётся процесс сегментации. Это свойство даёт дополнительную устойчивость получаемого результата к различным начальным условиям сегментации.

2 Алгоритм роста областей для трехмерных изображений

Алгоритм включает два этапа: двумерное извлечение областей в каждом слое изображения и слияние этих областей, находящихся в соседних слоях, в трехмерные области. Такой подход позволяет осуществить сегментацию за один просмотр изображения. В каждом слое процесс начинается работу с поиска особых точек роста, удовлетворяющих определенным критериям. Затем из этих точек роста формируются области путём рекурсивного добавления соседних точек. Следует учитывать, что трёхмерные объекты могут иметь различные формы и расти в произвольных направлениях. Поэтому надо обратить внимание на следующее: первое – как следует объединять области, находящиеся в соседних слоях; второе – как построить объект, если он состоит из областей, находящихся более чем в двух последовательных слоях.

Для решения первого вопроса алгоритм использует *таблицу областей* и *таблицу эквивалентности*. Таблица областей хранит информацию о каждой области, появляющейся в процессе сегментации: её идентификатор ID , её границы, количество точек роста в области и общее количество точек. Таблица эквивалентности содержит интегральную информацию об областях после этапа их слияния. Элементами таблицы эквивалентности являются списки идентификаторов областей, форми-

рующих один объект (одну трёхмерную область). Элементы таблицы областей также содержат ссылку на элемент таблицы эквивалентности, к которой они относятся.

Решение второго вопроса является более сложным. Для этого после этапа выделения двумерных областей в текущем слое различают интересные и подозрительные области. А после этапа слияния в таблице эквивалентности различают активные, неактивные, приемлемые и неприемлемые объекты (трёхмерные области). Так, после работы алгоритма на первом слое, области, содержащие точки роста, помечаются как интересные, а области не содержащие точек роста, помечаются как подозрительные. Информация об этих областях сохраняется в таблице областей. Такие данные определяются для каждой области каждого слоя. Тогда на этапе слияния k -того и $(k-1)$ -ого слоев эта информация обрабатывается. Трёхмерная область в таблице эквивалентности помечается как активная, если к ней добавилась область из k -того слоя, и неактивная – в обратном случае. В свою очередь, неактивные объекты (трёхмерные области) сортируются в зависимости от количества точек роста и общего количества точек в них. Если область не удовлетворяет минимально необходимому количеству точек в ней, то объект помечается как неприемлемый, иначе – как приемлемый. После просмотра алгоритмом последнего слоя и выполнения слияния, все объекты в таблице эквивалентности будут либо приемлемыми, либо нет. Только приемлемые области формируют конечный результат.

В	А	В
А	С	А
В	А	В

Рис. 1: В плоскости у точки С может быть четыре соседа – точки А, – так и шесть – точки А и В.

Преимуществом этого метода является то, что на этапе выделения областей для каждого вокселя просматриваются его соседи, находящиеся в том же слое, что и текущий воксел, и не проверяются соседи в верхнем и нижнем слоях. Это значительно сокращает время работы алгоритма, особенно для больших изображений.

Пусть $\text{NumOfSeeds}(R_i)$ возвращает число точек-роста области R_i ,

Так как алгоритм просматривает трёхмерное изображение слой за слоем и только один раз, то пройденные слои больше не используются и воксели в них могут хранить номера (идентификаторы) объектов, к которым они принадлежат. Поэтому данному алгоритму требуется только одна копия изображения и небольшой объём рабочего буфера для хранения таблицы областей.

$\text{NumOfVoxels}(R_i)$ возвращает количество вокселей R_i , а NumOfRegions определяет общее число областей. Перед началом работы алгоритма следующие значения должны быть определены: $G_{seedMin}$, $G_{seedMax}$, G_{min} , G_{max} , $G_{tolerance}$, $G_{leastSeeds}$, $G_{minSize}$. Интервал $[G_{seedMin}, G_{seedMax}]$ определяет диапазон яркости для точек роста. $[G_{min}, G_{max}]$ определяет интервал разрешённой яркости — воксели, не попавшие в заданный диапазон яркости, не будут рассматриваться. Поэтому $G_{min} \leq G_{seedMin} \leq G_{seedMax} \leq G_{max}$. Воксел, не являющийся точкой роста, добавляется в растущую область только если разница его яркости с любым его соседом из данной области не превосходит значения $G_{tolerance}$. Соседями точки в двумерной области могут считаться как четыре, так и восемь соседей (см. рис. 1). Значение $G_{leastSeeds}$ задаёт минимальное количество точек роста в трёхмерной области и минимальное количество вокселей $G_{minSize}$ самой области.

3 Псевдокод

1. Трёхмерное выделение областей:

Для каждого слоя $k = 0$ до N_z

Выделить двумерные области в слое k ;

Если $k \geq 0$ то Выполнять слияние областей слоёв k и $k - 1$;

Перекрасить все воксели $v(x, y, z)$, не входящие в приемлемые области в 0, остальные воксели перекрасить в 1.

2. Двумерное выделение областей k -того слоя:

(a) Инициализация. Начать с вокселя $(0, 0, k)$;

(b) Найти воксел (x, y, k) такой что $G_{min} \leq v(x, y, k) \leq G_{max}$;
 $w_k(x, y) = \text{NumOfRegions}$;
 $\text{NumOfRegions} = \text{NumOfRegions} + 1$;

(c) Проверить соседей текущей точки $v(x, y, z)$.
 Соседняя точка $v(x', y', k)$ присоединяется к области тогда и только тогда, когда данный предикат возвращает значение "истина":

$(P0 \text{ AND } ((P0 \text{ AND } (P2 \text{ OR } (P3 \text{ AND } P4))) \text{ OR } (\overline{P0} \text{ AND } (P3 \text{ AND } P4))))$, где

$P0 = \{v(x, y, k) \in [G_{seedMin}, G_{seedMax}]\} - (x, y, k)$ точка роста.

$\overline{P0} = \{v(x, y, k) \notin [G_{seedMin}, G_{seedMax}]\}$ — (x, y, k) не является точкой роста.

$P1 = \{w_k(x', y') - \text{значение неопределено}\}$ — точка $v(x', y', k)$ ещё не просматривалась.

$P2 = \{v(x', y', k) \in [G_{seedMin}, G_{seedMax}]\}$ — соседняя точка (x', y', k) является точкой роста.

$P3 = \{v(x', y', k) \in [G_{min}, G_{max}]\}$ — соседняя точка (x', y', k) находится в допустимом диапазоне яркости.

$P4 = \{ |v(x', y', k) - v(x, y, k)| \leq G_{tolerance} \}$ — яркость соседа (x', y', k) достаточна близка к яркости вокселя (x, y, k) , чтобы добавить соседнюю точку в область роста R_i .

- (d) Перейти на шаг 2b, если просмотрены ещё не все воксели. Иначе записать значения w_k в текущий слой, то есть $v(x, y, k) = w_k(x, y)$ для любых x, y .
- (e) Сгенерировать информацию о двумерных областях текущего слоя: идентификатор области ID , границы области, количество точек роста и общее количество точек в области. Поместить данные в таблицу областей. Если область не содержит точек роста, то пометить её как подозрительную, иначе — как интересующую.

3. Слияние областей слоёв k и $k - 1$:

- (a) Среди всех активных областей таблицы эквивалентности найти те области $(k-1)$ -ого слоя, которые перекрываются областями k -того слоя. Для этого сравниваются границы областей, которые хранятся в таблице областей.
- (b) Проверить какие перекрывающиеся области следует объединить. Для этого каждая точка (x, y, k) k -того слоя сравнивается с пятью (точки А и С рисунка 1) либо с девятью (точки А, В, С рисунка 1) вокселями $(x', y', k - 1)$ $(k-1)$ -ого слоя. Область k -того слоя добавляется к объекту (трёхмерной области) таблицы эквивалентности только в случае, если данный предикат возвращает значение "истина":

$((P0 \text{ AND } (P1 \text{ OR } P2)) \text{ OR } (\overline{P0} \text{ AND } P2))$, где

$P0 = \{v(x, y, k) \in [G_{seedMin}, G_{seedMax}]\}$ — (x, y, k) точка роста.

$\overline{P0} = \{v(x, y, k) \notin [G_{seedMin}, G_{seedMax}]\}$ — (x, y, k) не является точкой роста.

$P1 = \{v(x', y', k-1) \in [G_{seedMin}, G_{seedMax}]\}$ — соседняя точка $(x', y', k-1)$ является точкой роста.

$P4 = \{|v(x', y', k-1) - v(x, y, k)| \leq G_{tolerance}\}$ — яркость соседа $(x', y', k-1)$ достаточна близка к яркости вокселя (x, y, k) .

- (с) Обновить таблицу эквивалентности: добавить идентификатор области ID из k-того слоя в список областей данного объекта в случае их объединения.

Перейти к шагу 3b, пока все области k-того слоя не будут проверены.

- (d) Трёхмерные объекты из таблицы эквивалентности пометить как активные, если к ним были добавлены области k-того слоя. Остальные трёхмерные области (объекты) помечаются как неактивные.

- (е) Проверить все неактивные объекты. Трёхмерная область R_i помечается как неприемлемая, если $\text{NumOfSeeds}(R_i) < G_{leastSeeds}$ или $\text{NumOfVoxels}(R_i) < G_{minSize}$. Все остальные неактивные объекты помечаются как приемлемые.

Данный алгоритм позволяет выделять достаточно большие по размеру кластеры, удаляя слишком малые при помощи ограничительных параметров $G_{leastSeeds}$ и $G_{minSize}$. Кроме того, параметр $G_{tolerance}$ позволяет добавлять к области только те воксели, которые достаточно близки по яркости к точкам самой области. Результатом алгоритма будет изображение, каждый воксел которого хранит идентификатор области, к которой он принадлежит, либо 0 для точек фона. Используя таблицу областей и таблицу эквивалентности, просматриваются все точки изображения и перекрашиваются либо в 0 — для точек фона и точек неприемлемых объектов, либо в 1 — для точек приемлемых объектов.

Таким образом, представленный алгоритм эффективно использует память, так как нет необходимости загружать томографическое изображение целиком — используется послойная обработка. Кроме того, алгоритм является быстрым — для полной сегментации трёхмерного изображения достаточно два прохода. Большим преимуществом данного алгоритма является то, что извлекаться могут объекты сложной формы. Пример использования указанного алгоритма роста областей для сегментации сосудистой системы головного мозга из трёхмерного магнитно-резонансного ангиографического изображения приводится в главе ??.

3.1 Библиографическая справка

Алгоритмы сегментации нашли широкое применение в области обработки трёхмерных медицинских изображений, в таких задачах как выделение гортанно-трахеального тракта [3, 4], выделение артериальных сосудов [6] и выделение кровеносной системы печени [5]. Алгоритмы сегментации используются для извлечения интересующих объектов с целью последующего их измерения, вычисления характеристик и визуализации. Благодаря своей простоте и эффективности, алгоритмы на основе областей роста являются одними из самых популярных.

Так, в статье [6] описывается система обработки трёхмерных изображений и анализа артериального дерева. Предлагаются семь этапов для построения скелетного представления и вычисления характеристик кровеносной системы: (1) фильтрация для уменьшения шума в изображении; (2) сегментация для выделения кровеносной системы при помощи алгоритма роста областей; (3) удаление пустот; (4) утоньшение для извлечения скелета из дерева кровеносной системы; (5) преобразование скелета в компактную графовую модель; (6) определение корня кровеносной системы; (7) отсечение ненужных ветвей, где ветвью дерева кровеносной системы считается участок либо между двумя последовательными разветвлениями, либо между разветвлением и концом дерева. Для каждой ветви вычисляются различные характеристики, например, глубина, высота, среднее значение диаметра сечений, площадь поверхности, объём ветви, расстояние до корня и другие. Система имеет семь режимов визуализации: три режима для просмотра двумерных слоёв, три режима для просмотра проекций двумерных слоёв и режим просмотра трёхмерного дерева. Программа позволяет просматривать вычисленные значения для каждой ветви дерева, а так же редактировать структуру самого дерева.

Существуют также и другие способы сегментации объектов. Например, в статье [2] описывается алгоритм сегментации объектов кровеносной системы в двумерных изображениях. Для этого в четырёх направлениях: сверху вниз, справа налево и по диагоналям, – проводят скан-линии. Для каждого направления формируется своя карта, в которой отмечаются точки локальных минимумов для каждой скан-линии. Затем все четыре карты объединяются в одну большую. Для избавления от побочных шумов выполняется морфологическое замыкание. Полученное изображение скелетизируется, чтобы получить скелет шириной в один пиксел. Затем в скелете удаляются ветви короче заданной величины: находится конечная точка, находится самая длинная ветвь. Эта ветвь переносится в другое изображение и удаляется в исходном. Процесс продолжается пока все ветви, длиннее заданной величины, не будут

перенесены в новое изображение. После удаления мелких ветвей подсчитываются диаметры сосудов в каждой точке скелета. Диаметры строятся ортогонально линии скелета. Правые и левые границы сосудов определяются раздельно с помощью специальной логарифмической функции чувствительной к изменению фона. Для сглаживания построенных границ сосудов и более корректного вычисления диаметров используется медианный оператор. Для построения более гладких границ сосудов также могут быть применены сплайны, морфологические операции или активные контуры. Преимуществом описанного подхода является то, что скелет, полученный с помощью скан-линий, устойчив к шумам в исходном изображении, так как такой скелет строится на основе локальных минимумов, а не как средняя линия (она зависит от границ сосудов).

Список литературы

- [1] S.-V. Wan and W. Higgins, Symmetric region growing, *Image Processing*, pages II-439–II-442, September 10-13 2000.
- [2] Z. Chen and S. Molloy, Vascular tree object segmentation by deskeletonization of valley courses, *Computerized Medical Imaging and Graphics*, Vol. 26, pages 419-428, April 2002.
- [3] E. Sorantin and C. Halmai and B. Erdöhelyi and K. Palágyi and L.G. Nyúl and K. Ollé and B. Geiger and F. Lindbichler and G. Friedrich and K. Kiesler, Spiral-CT-Based Assessment of Tracheal Stenoses Using 3-D-Skeletonization, *IEEE Transactions On Medical Imaging*, Vol. 21, Num. 3, pages 263-273, March 2002
- [4] J. Udupa and S. Samarasekera, Fuzzy connectedness and object definition: Theory, algorithms, and applications in image segmentation, *Graphical Models Image Processing*, Vol. 58, Num. 3, pages 246-261, 1996
- [5] S.-Y. Wan and A.P. Kiraly and E.L. Ritman and W.E. Higgins, Extraction of the Hepatic Vasculature in Rats Using 3-D Micro-CT Images, *IEEE Transactions On Medical Imaging*, Vol. 19, Num. 9, pages 965-971, September 2000
- [6] S.-Y. Wan and E.L. Ritman and W.E. Higgins, Multi-generational analysis and visualization of the vascular tree in 3D micro-CT images, *Computers in Biology and Medicine*, Vol. 32, pages 55-71, 2002