

Министерство образования и науки Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Институт прикладной математики и механики

Работа допущена к защите
Заведующий кафедрой
«Прикладная математика»
_____ М.Е. Фролов
«__» _____ 20__ г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

АЛГОРИТМ И ПРОГРАММНАЯ РЕАЛИЗАЦИЯ МЕТОДА БЫСТРОГО РАСПРОСТРАНЕНИЯ КОНТУРА ДЛЯ РЕШЕНИЯ ЗАДАЧИ ПОЛУАВТОМАТИЧЕСКОЙ СЕГМЕНТАЦИИ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

по направлению 01.03.02 «Прикладная математика и информатика»
по образовательной программе
01.03.02_02 «Системное программирование»

Выполнил

студент гр.43601/2

М.В. Позигун

Руководитель

ассистент кафедры

В.С. Чуканов

Санкт-Петербург

2018

РЕФЕРАТ

МЕДИЦИНСКИЕ ИЗОБРАЖЕНИЯ, СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ, ПОЛУАВТОМАТИЧЕСКАЯ СЕГМЕНТАЦИЯ, УРАВНЕНИЕ ЭЙКОНАЛА, ФИЛЬТРАЦИЯ ИЗОБРАЖЕНИЙ

На 29 с., 8 рисунков, 4 таблицы, 2 приложения.

В данной работе представлен алгоритм и программная реализация метода, основанного на алгоритме быстрого распространения контура (англ. Fast Marching), для решения задачи полуавтоматической сегментации медицинских изображений. Основной идеей алгоритма является выполнение обхода пикселей (вокселей) изображения в порядке, обусловленном значениями восстанавливаемой функции времени, представляющей собой решение дифференциального уравнения. В работе предложены новые условия остановки алгоритма и вычисления весов пикселей (вокселей), исследовано применение фильтрации в качестве предобработки, разработано решение для проблемы сегментации слабых границ.

СОДЕРЖАНИЕ

Введение.....	5
1. Постановка задачи.....	6
2. Обзор литературы.....	6
3. Алгоритм быстрого распространения контура.....	8
3.1. Двумерный случай.....	8
3.2. Трёхмерный случай.....	11
4. Модификации алгоритма.....	12
4.1. Критерий остановки.....	12
4.2. Изменение вычисления веса и критерия остановки.....	14
4.3. Предобработка изображений.....	16
4.3.1. Фильтр анизотропной диффузии.....	17
4.3.2. Подбор параметров фильтра.....	18
4.4. Введение дополнительного параметра.....	19
5. Реализация.....	21
5.1. Архитектура реализации алгоритмов.....	21
6. Результаты.....	22
6.1. Измерения времени работы алгоритмов.....	22
6.2. Оценка качества сегментации.....	24
7. Средства разработки.....	26
Заключение.....	27
Список использованных источников.....	28
Приложение 1. Изображения областей, эталонной сегментации и результатов работы алгоритма, двумерный случай.....	30
Приложение 2. Изображения областей, эталонной сегментации и результатов работы алгоритма, трёхмерный случай.....	31

Введение

Магнитно-резонансная томография (МРТ) и компьютерная томография (КТ) – современные методы неинвазивного медицинского исследования. Важными этапами предоперационного планирования являются анализ и сегментация изображений, полученных в ходе проведения этих процедур.

Сегментация изображений – процесс разбиения множества пикселей (вокселей) изображения на два подмножества: множества пикселей (вокселей), принадлежащих некоторой области интереса и множества пикселей (вокселей) вне области интереса.

Особенностью медицинских изображений является то, что их диапазон интенсивностей может быть значительно шире, чем $[0; 255]$. Помимо двумерных, также рассматриваются и трёхмерные изображения, полученные в результате объединения и упорядочивания множества двумерных изображений.

Разработка эффективных по качеству результата и времени работы инструментов сегментации является важной задачей, так как такие инструменты позволяют облегчить работу медицинских специалистов, в ином случае вынужденных проводить сегментацию областей вручную на каждом снимке попиксельно, что может потребовать особенно много времени для случая областей на трёхмерных изображениях.

Алгоритмы сегментации, использующиеся для решения этой задачи, разделяются на автоматические и полуавтоматические. Автоматические алгоритмы на данный момент не обладают идеальной точностью выделения, потому вопрос быстрой корректировки сегментированного региона особенно важен, равно как и возможность сегментировать некую область снимка с нуля.

Выполнить корректировку результатов сегментации или сегментировать область с нуля позволяют алгоритмы полуавтоматической сегментации. Исследованию и реализации одного из таких алгоритмов и посвящена данная работа. Ранее часть результатов данной работы была мной представлена в рамках научной конференции с международным участием XLVI "Неделя науки СПбПУ" [1].

1. Постановка задачи

Требуется разработать и реализовать алгоритм, основанный на методе быстрого распространения контура (англ. Fast Marching, далее – БРК), позволяющий выполнять полуавтоматическую сегментацию медицинских изображений. Алгоритм требуется разработать и реализовать в двух версиях: для двумерных и трёхмерных медицинских изображений, полученных методами КТ и МРТ диагностики. Версия алгоритма для двумерных изображений должна работать в реальном времени (время выполнения соответствует частоте 30 кадров в секунду, то есть составляет не более чем 33 мс).

Помимо сегментируемого изображения, на вход алгоритма подаются два параметра: радиус и некоторый пиксель (воксель) изображения – центр. Требуется сегментировать множество пикселей (вокселей) изображения, лежащих в круге (шаре) заданного радиуса с центром в данной позиции и относящихся к той же области интереса что и заданный пиксель (воксель) центра.

Для случая медицинских изображений областями интереса могут являться, например, органы или их части. В частности, такие области могут обладать свойством, что значения интенсивностей их пикселей (вокселей) отличаются несущественно. Однако, это справедливо не для всех областей, что усложняет задачу сегментации.

2. Обзор литературы

Можно выделить несколько основных групп методов сегментации изображений [10]:

- Методы, основанные на машинном обучении, позволяющие сегментировать область снимка, основываясь на выборке заранее сегментированных изображений
- Методы роста региона (англ. Region Growing), основанные на итерационном процессе отбора пикселей (вокселей) области снимка по некоторому критерию

- Методы, основанные на минимизации энергии, позволяющие минимизировать функцию стоимости, определяющую разность между начальным приближением сегментации и областями интереса
- Методы фиксации уровня (англ. Level Set) и БРК, выполняющие сегментацию путём решения уравнений определённого вида

Алгоритм БРК [8] был разработан Д. Сетианом. Данный алгоритм позволяет найти решение уравнение эйконала на некоторой сетке узлов, часто используется для построения карт расстояний. Метод применяется во множестве разных областей, например, симуляции поведения жидкости, робототехнике, компьютерном зрении, сейсмологии и др. Некоторое количество модификаций алгоритма направлены на уменьшение его временной сложности: к одним из последних из них можно отнести, например, алгоритм под названием Fast Iterative Method [5]. Основной его идеей является отказ от использования очереди узлов, вместо очереди используется список узлов, при этом пересчёт значений восстанавливаемой функции выполняется итерационно для каждого узла фронта распространения контура. Другим примером модификации, позволяющей уменьшить временную сложность алгоритма, являются алгоритм под названием Untidy Fast Marching Method [9]. В рамках данной модификации предлагается внести упрощение в процесс нахождения значений функции на сетке, позволяя использовать более эффективную по времени структуру данных, при этом считая, что возможная ошибка приближения функции оправдана ускорением алгоритма.

Также стоит отметить, что не только алгоритм БРК позволяет найти решение уравнения эйконала. Одним из последних алгоритмов, решающих аналогичную задачу, является алгоритм под названием Fast Sweeping Method [11]. Главными его отличиями по сравнению с алгоритмом БРК является то, что алгоритм итерационный, а также то, что используется иной способ обхода узлов сетки. Данный алгоритм не использовался для решения поставленной задачи, так как, согласно [4], его использование позволило бы получить прирост произ-

водительности только для случая сегментации однородных областей, однако не всегда области интереса являются однородными.

Для улучшения результата сегментации используются различные процедуры предобработки изображений. Одной из таких процедур является фильтрация. Например, сглаживающие фильтры позволяют избавиться от шумов на изображении. К таким фильтрам относятся свёрточные фильтры: например, фильтр Гаусса, медианный фильтр. Основным недостатком свёрточных фильтров является то, что помимо шумов, также сглаживаются границы областей, тем самым усложняется исходная задача сегментации.

Примером фильтра, позволяющего сгладить шумы, но в то же время сохранить границы областей изображения, является фильтр анизотропной диффузии [7]. Данный фильтр может быть использован как для двумерных, так и для трёхмерных изображений.

Сохранение границ, возможность обобщения фильтра на трёхмерный случай определили выбор именно фильтра анизотропной диффузии как возможного способа предобработки. Кроме того, хотя фильтр требует задания нескольких входных параметров, для них существуют оценки или способы автоматического подбора.

3. Алгоритм быстрого распространения контура

3.1. Двумерный случай

Рассмотрим сначала оригинальный алгоритм БРК для случая двумерного изображения.

В алгоритме БРК рассматривается замкнутый контур G на сетке. Контур расширяется, распространяется со скоростью $F > 0$ (рис. 3.1).

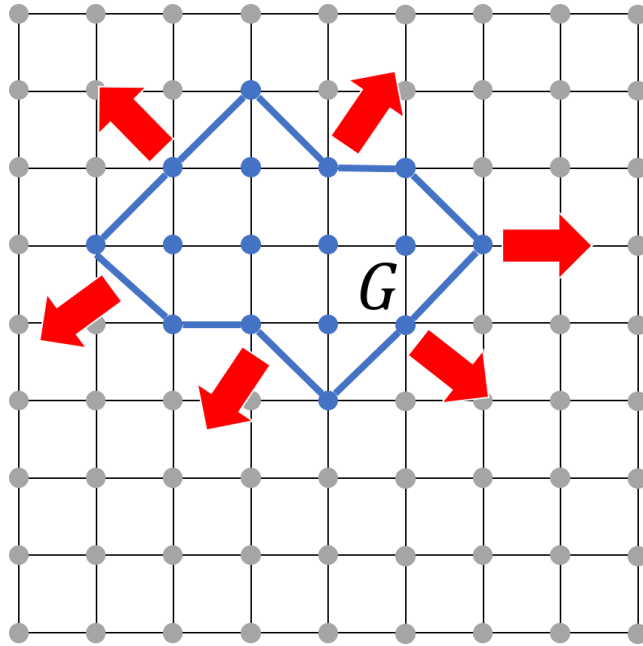


Рис. 3.1. Распространение контура G на сетке узлов

Положение контура можно охарактеризовать, используя время прибытия $T(x, y)$ контура в узел (x, y) . Тогда $T(x, y)$ – решение уравнения эйконала:

$$|\nabla T|F = 1, T = 0 \text{ на } G \quad (3.1)$$

Алгоритм БРК позволяет найти значения функции T в узлах сетки. Входом алгоритма является начальное множество узлов контура. Алгоритм совершает обход узлов сетки. За каждый шаг контур распространяется на один узел. На каждом шаге, множество узлов сетки представляет собой объединение трёх непересекающихся подмножеств:

1. Множества помеченных узлов (англ. “Active”)
2. Множества узлов, соседних к узлам контура (англ. “Trial”)
3. Множества узлов, до которых контур ещё не распространился (англ. “Far”).

Псевдокод алгоритма БРК:

Вход: сетка узлов, начальное множество узлов

1. Поместить в Trial начальное множество узлов, для них $T = 0$

2. Поместить в Far все узлы, которых нет в Trial, для них $T = +\infty$
3. Цикл: пока множество Trial не пусто:
 - 3.1. Пусть p – элемент Trial с минимальным весом T
 - 3.2. Положить p в Active, убрать из Trial
 - 3.3. Пересчитать значения T для соседних к p узлов
 - 3.4. Поместить в Trial соседние к p узлы

Соседними являются узлы, у которых координаты вдоль одной из осей совпадают.

Для двумерной сетки у узла, не лежащего на границе сетки, имеется 4 соседних узла, у граничных узлов соседей соответственно меньше.

Пересчёт значения $T_{i,j}$ для каждого узла (i, j) выполняется следующим образом:

Согласно [8], уравнение (3.1) можно аппроксимировать следующим образом:

$$\max(D_{ij}^{-x}T, -D_{ij}^{+x}T, 0)^2 + \max(D_{ij}^{-y}T, -D_{ij}^{+y}T, 0)^2 = \frac{1}{F_{ij}^2}, \quad (3.2)$$

$$\text{где } D_{ij}^{\pm x}T \triangleq \frac{T_{i\pm 1,j} - T_{i,j}}{\pm h}, D_{ij}^{\pm y}T \triangleq \frac{T_{i,j\pm 1} - T_{i,j}}{\pm h}, h - \text{ шаг сетки.}$$

Введём дополнительные обозначения:

$$T_V \triangleq \min(T_{i,j-1}, T_{i,j+1}), T_H \triangleq \min(T_{i-1,j}, T_{i+1,j}) \quad (3.3)$$

Тогда уравнение (3.2) может быть записано в следующем виде:

$$\left(\frac{T_{i,j} - T_H}{h}\right)^2 + \left(\frac{T_{i,j} - T_V}{h}\right)^2 = \frac{1}{F_{i,j}^2} \quad (3.4)$$

Если дискриминант этого квадратного уравнения неотрицателен, то искомое значение вычисляется как:

$$T_{i,j} = \frac{1}{2}(T_H + T_V) + \frac{1}{2}\sqrt{(T_H + T_V)^2 - 2\left(T_H^2 + T_V^2 - \frac{h^2}{F_{i,j}^2}\right)} \quad (3.5)$$

Иначе, согласно [3]:

$$T_{i,j} = \min(T_H, T_V) + \frac{h}{F_{i,j}} \quad (3.6)$$

Для поставленной задачи множество узлов на сетке – множество пикселей изображения, в каждом из которых определена функция интенсивности I , шаг h сетки равен 1. Исходное множество узлов контура включает только центр. Как утверждается в [6], функция F может быть определена следующим образом:

$$F(x, y) = \exp\left(-\alpha * \left(\max(D_{ij}^{-x}I, -D_{ij}^{+x}I, 0)^2 + \max(D_{ij}^{-y}I, -D_{ij}^{+y}I, 0)^2\right)^{\frac{1}{2}}\right), \quad (3.7)$$

где $\alpha \in R, \alpha \geq 0$.

Было выбрано значение $\alpha = 1$.

Для случая трёхмерной сетки алгоритм БРК выглядит аналогично, за исключением особенностей, которые будут отмечены в следующем разделе.

3.2. Трёхмерный случай

Для трёхмерной сетки у узла, не лежащего на границе сетки, имеется 6 соседних узлов.

В трёхмерном случае аппроксимация уравнения (3.1) для узла (i, j, k) выглядит как

$$\begin{aligned} & \max(D_{ijk}^{-x}T, -D_{ijk}^{+x}T, 0)^2 + \max(D_{ijk}^{-y}T, -D_{ijk}^{+y}T, 0)^2 + \\ & + \max(D_{ijk}^{-z}T, -D_{ijk}^{+z}T, 0)^2 = \frac{1}{F_{ijk}^2}, \end{aligned} \quad (3.8)$$

$$\text{где } D_{ijk}^{\pm x}T \triangleq \frac{T_{i\pm 1, j, k} - T_{i, j, k}}{\pm h}, D_{ijk}^{\pm y}T \triangleq \frac{T_{i, j\pm 1, k} - T_{i, j, k}}{\pm h}, D_{ijk}^{\pm z}T \triangleq \frac{T_{i, j, k\pm 1} - T_{i, j, k}}{\pm h}.$$

Пересчёт значений $T_{i, j, k}$:

Введём обозначения:

$$\begin{aligned} T_V & \triangleq \min(T_{i, j-1, k}, T_{i, j+1, k}), T_H \triangleq \min(T_{i-1, j, k}, T_{i+1, j, k}), \\ T_D & \triangleq \min(T_{i, j, k-1}, T_{i, j, k+1}) \end{aligned} \quad (3.9)$$

Тогда уравнение (3.8) может быть записано в следующем виде:

$$\left(\frac{T_{i, j, k} - T_H}{h}\right)^2 + \left(\frac{T_{i, j, k} - T_V}{h}\right)^2 + \left(\frac{T_{i, j, k} - T_D}{h}\right)^2 = \frac{1}{F_{i, j, k}^2} \quad (3.10)$$

При неотрицательном дискриминанте:

$$T_{i,j,k} = \frac{1}{3}(T_H + T_V + T_D) + \frac{1}{3} \sqrt{(T_H + T_V + T_D)^2 - 3 \left(T_H^2 + T_V^2 + T_D^2 - \frac{h^2}{F_{i,j,k}^2} \right)} \quad (3.11)$$

Иначе:

$$T_{i,j,k} = \min(T_H, T_V, T_D) + \frac{h}{F_{i,j,k}} \quad (3.12)$$

Функция F для данного случая может быть определена следующим образом:

$$F(x, y, z) = \exp \left(-\alpha * \left(\max(D_{ijk}^{-x}I, -D_{ijk}^{+x}I, 0)^2 + \max(D_{ijk}^{-y}I, -D_{ijk}^{+y}I, 0)^2 + \max(D_{ijk}^{-z}I, -D_{ijk}^{+z}I, 0)^2 \right)^{\frac{1}{2}} \right), \quad (3.13)$$

где $\alpha \in R, \alpha \geq 0$.

Для трёхмерного случая также было выбрано значение $\alpha = 1$.

Алгоритм метода БРК позволяет обойти в определённом порядке пиксели (воксели) изображения. Однако, согласно поставленной задаче сегментации области изображения, требуется, используя получаемый в ходе работы алгоритма порядок обхода, выделить некоторое подмножество пикселей (вокселей) изображения. В связи с этим алгоритм БРК потребовалось модифицировать для решения с помощью него поставленной задачи.

4. Модификации алгоритма

4.1. Критерий останова

Одной из возможных модификаций алгоритма БРК является добавление дополнительного критерия останова, который позволил бы обойти лишь некоторое подмножество пикселей (вокселей) изображения. Такое подмножество пикселей (вокселей) изображения и являлось бы искомым множеством, определённым в постановке задачи.

Первоначально был предложен следующий критерий остановки: установить минимальное и максимальное значения интенсивности для пикселей (вокселей), принадлежащих сегментируемой области интереса. Однако, так как в рамках поставленной задачи требовалось минимизировать число параметров алгоритма, а данный критерий требует задания разных значений границ интенсивностей для разных областей, то от него пришлось отказаться.

Впоследствии был предложен альтернативный критерий остановки: если расстояние от узла с минимальным весом до начальной позиции не меньше, чем заданный радиус, то цикл алгоритма прерывается. Чтобы узлы внутри круга (шара) имели наибольший приоритет при равных значениях функции T и тем самым контур распространялся вдоль окружностей (сфер) внутри области интереса, веса, которые в оригинальном алгоритме совпадают со значением T , вычислялись для каждого узла q отдельно, следующим образом:

$$d(q) * (1 + T_q), \quad (4.1)$$

$$\text{где } d(q) = 1 + \max(0, \|q - p\|_2 - d_{\max});$$

p – начальный узел (центр);

$$d_{\max} \triangleq \max_{v \in \text{Active}} \|v - p\|_2.$$

С учётом внесённых изменений алгоритм принимает следующий вид:

Вход: изображение, центр p_0 , значение радиуса r .

Выход: множество B пикселей (вокселей) области интереса.

1. Поместить в Trial начальный узел p_0 , для него $T = 0$
2. Поместить в Far все узлы, которых нет в Trial , для них $T = +\infty$
3. $d_{\max} := 0$
4. Цикл: пока множество Trial не пусто:
 - 4.1. Пусть p – элемент Trial с минимальным весом T
 - 4.2. Положить p в Active , убрать из Trial
 - 4.3. Если $\|p - p_0\|_2 \geq d_{\max}$, то $d_{\max} := \|p - p_0\|_2$

- 4.4. Если $d_{\max} \geq r$, то выход из цикла
- 4.5. Пересчитать значения T для соседних к p узлов
- 4.6. Поместить в $Trial$ соседние к p узлы с весами, посчитанными согласно формуле (4.1)
5. Поместить в V узлы из $Active$

4.2. Изменение вычисления веса и критерия остановки

Однако, как показали результаты использования инструмента на тестовых изображениях, алгоритм учитывал слишком много границ внутри области интереса, возникающих из-за шумов. В связи с этим было решено изменить способ вычисления веса узла для того, чтобы алгоритм игнорировал шумы внутри области интереса. На рис. 4.1 приведены примеры сегментации для двух значений центров, для которых изменение вычисления веса позволило улучшить результат сегментации.

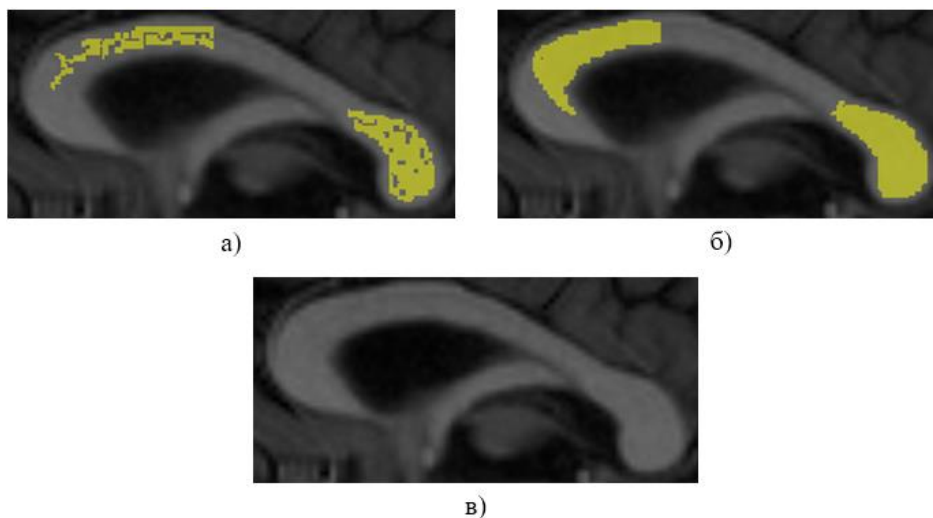


Рис. 4.1. Результаты сегментации при одних и тех же значениях параметров: с прежним вычислением веса (а) и с изменённым (б). Также приведено оригинальное изображение области (в)

Идеей улучшения было добавить экспоненциальную зависимость d от разности расстояний, тем самым сделав переход на новое значение радиуса d_{\max} менее приоритетным для точек с одинаковыми значениями функции T .

В финальной версии алгоритма веса вычисляются следующим образом:

$$d(q) * (1 + T_q), \quad (4.2)$$

где $d(q) = 1 + \exp(\alpha * \max(0, \|q - p\|_2 - d_{\max}))$;

$$\alpha \in R, \alpha > 0.$$

Кроме того, как показала проведённая тестовая сегментация, как правило некоторая часть пикселей (вокселей) на границах областей оставалась неотмеченной, то есть контур распространялся недостаточно близко к границам за время работы алгоритма. Частично решить данную проблему позволило следующее: внести изменения в шаги 4.2-4.4, оставив только критерий остановки оригинального алгоритма:

4.2. Убрать из Trial; если $\|p - p_0\|_2 < r$, то положить p в Active

4.3. Если $\|p - p_0\|_2 \geq d_{\max}$, то $d_{\max} := \|p - p_0\|_2$

4.4. Если $d_{\max} \geq r$, то перейти к шагу 4.1

Данные изменения алгоритма приводят к тому, что после первого выполнения неравенства $d_{\max} \geq r$ на дальнейших итерациях число элементов множества Trial только уменьшается, при этом в Active попадают пиксели (воксели) внутри круга (шара). Цикл завершается, когда Trial оказывается пустым. Так как дополнительные элементы Active являются соседними к узлам контура на момент выполнения условия $d_{\max} \geq r$, то контур за счёт дополнительных итераций приближается ближе к границам области. Изменённый алгоритм позволяет отметить часть пикселей (вокселей), неотмеченных алгоритмом с изменённым критерием.

На рис. 4.2 показаны примеры, для которых изменение алгоритма позволило сегментировать дополнительные пиксели области.

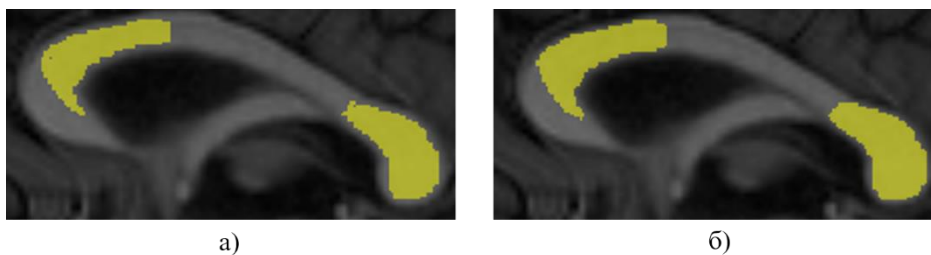


Рис. 4.2. Результат сегментации при одних и тех же значениях параметров: с прежним критерием остановки (а) и с внесёнными изменениями (б)

4.3. Предобработка изображений

Выбранный способ вычисления весов позволяет игнорировать часть шумов области интереса. Однако, как показало тестирование, этого оказалось недостаточно в некоторых случаях. В результате возникла идея фильтрации входного изображения в качестве предобработки. Для этого использовался фильтр анизотропной диффузии. На рис. 4.3 приведён пример случая, для которого фильтрация позволила улучшить результат сегментации: выходное множество алгоритма целиком относится к однородной области изображения, к которому применена предобработка.

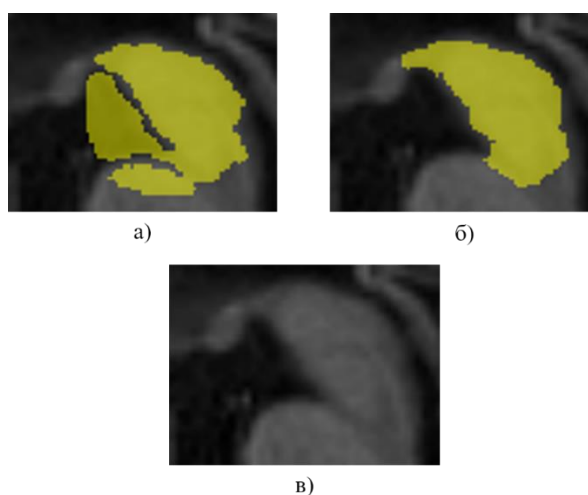


Рис. 4.3. Результат сегментации при одних и тех же значениях параметров: без фильтрации (а) и с фильтрацией (б). Также приведено оригинальное изображение области (в)

4.3.1. Фильтр анизотропной диффузии

Основная идея фильтра анизотропной диффузии – получение фильтрованного изображения как решения уравнения анизотропной диффузии. В случае двумерных изображений оно выглядит следующим образом:

$$I_t = \text{div}(c(x, y, t)\nabla I) \quad (4.3)$$

Начальное условие:

$$I(x, y, 0) = I_0(x, y), \quad (4.4)$$

где $I_0(x, y)$ – исходное изображение.

Как отмечают авторы, в случае $c(x, y, t) = \text{const}$, применение фильтра эквивалентно применению сглаживающего фильтра Гаусса. Для размытия внутри областей значение функции c должно быть равно 1, на границах $c(x, y, t) = 0$. Для этого функция $c(x, y, t)$ рассматривается как некоторая монотонно-убывающая функция нормы градиента изображения $g(\|\nabla I(x, y, t)\|)$. Выбор нормы градиента в качестве аргумента позволяет не только сохранить, но и усилить границы.

После дискретизации уравнение выглядит следующим образом:

$$I_{i,j}^{t+1} = I_{i,j}^t + \lambda * \left(c_{N_{i,j}}^t \nabla_N I_{i,j}^t + c_{S_{i,j}}^t \nabla_S I_{i,j}^t + c_{W_{i,j}}^t \nabla_W I_{i,j}^t + c_{E_{i,j}}^t \nabla_E I_{i,j}^t \right), \quad (4.5)$$

$$\text{где } c_{N_{i,j}}^t = g(|\nabla_N I_{i,j}^t|), \nabla_N I_{i,j}^t = I_{i-1,j}^t - I_{i,j}^t;$$

$$c_{S_{i,j}}^t = g(|\nabla_S I_{i,j}^t|), \nabla_S I_{i,j}^t = I_{i+1,j}^t - I_{i,j}^t;$$

$$c_{W_{i,j}}^t = g(|\nabla_W I_{i,j}^t|), \nabla_W I_{i,j}^t = I_{i,j-1}^t - I_{i,j}^t;$$

$$c_{E_{i,j}}^t = g(|\nabla_E I_{i,j}^t|), \nabla_E I_{i,j}^t = I_{i,j+1}^t - I_{i,j}^t.$$

Для граничных пикселей вместо интенсивностей отсутствующих соседних пикселей подставляется значение 0.

При значениях параметра $0 \leq \lambda \leq 1/4$ вычислительная схема является устойчивой. В программной реализации использовалось значение $\lambda = 1/4$.

Аналогично выглядит уравнение анизотропной диффузии и вычислительная схема алгоритма в случае трёхмерных изображений, за исключением

добавлением аргумента по оси z в функциях, а также дополнительного слагаемого в схеме по оси z .

В качестве определения функции g авторами в оригинальной статье было предложено выбрать одно из следующих:

$$g(\|\nabla I\|) = e^{-(\|\nabla I\|/K)^2} \quad (4.6)$$

$$g(\|\nabla I\|) = \frac{1}{1+(\|\nabla I\|/K)^2} \quad (4.7)$$

Однако для таких способов требуется задание параметра K , подбираемого в зависимости от конкретного исходного изображения. В связи с этим было решено исследовать альтернативные способы задания функции g и возможные способы подбора параметра.

4.3.2. Подбор параметров фильтра

Изначально для фильтрации использовалась функция g вида (4.6) и константное значение K , подобранное по одному из изображений, однако в дальнейшем стало ясно, что для некоторых случаев можно добиться значительно лучшего качества фильтрации при других подбираемых значениях. Вследствие этого, было решено найти способ автоматического подбора значения параметра. В итоговой реализации алгоритма оценка в некотором смысле наиболее подходящего значения параметра K выполняется согласно тому, как предлагается в [2].

В данной статье фильтр анизотропной диффузии и задача фильтрации рассматриваются с точки зрения статистики: рассматривается выборка, состоящей из разностей значений интенсивностей соседних пикселей. Тогда границы областей изображения определяются как пиксели (воксели), для которых разность с соседними пикселями (вокселями) – выбросы такой выборки. С этой точки зрения, как утверждают авторы, наиболее подходящим является выбор $g(\|\nabla I\|)$ вида:

$$g(\|\nabla I\|) = \begin{cases} \frac{1}{2}(1 - (\|\nabla I\|/K)^2)^2, & \|\nabla I\| \leq K \\ 0, & \|\nabla I\| > K \end{cases} \quad (4.8)$$

Тогда робастная оценка значения K выглядит следующим образом [2]:

$$K = \sqrt{5}[1.4826 \text{MAD}(\nabla I)] = \sqrt{5}[1.4826 \text{med}(\|\nabla I - \text{med}(\|\nabla I\|)\|)] \quad (4.9)$$

4.4. Введение дополнительного параметра

Как было отмечено ранее, не всегда область интереса можно охарактеризовать как область, значения интенсивности пикселей (вокселей) которой отличаются несущественно. Примером области интереса, для которой это неверно, является мозжечок на МРТ-изображениях мозга (рис. 4.4).



Рис. 4.4. Мозжечок на МРТ-изображении

В пределах таких областей могут содержаться пиксели (воксели), имеющие интенсивности, значительно отличающиеся от интенсивности пикселя (вокселя) начальной позиции. Очевидно, такие пиксели (воксели) при этом не являются зашумлёнными, поэтому предобработки фильтрацией недостаточно для получения приемлемой сегментации области.

В связи с этим было предложено ввести дополнительный параметр алгоритма – некий параметр, позволяющий контролировать «чувствительность» алгоритма к распространению контура по таким пикселям (вокселям) области интереса.

Новый параметр был введён в процедуру обновления значения функции F . Как было описано выше, согласно [6], F содержит параметр α . Первоначально использовалось значение $\alpha = 1$. Было решено сделать α параметром «чувствительности» алгоритма.

При значении $\alpha = 0$ функция F не зависит от градиента интенсивности, поэтому независимо от области интереса на выходе алгоритма получается множество, образующее круг (шар) заданного радиуса. Чем больше значение α , тем меньше значение функции F , соответственно для одного и того же пикселя (вокселя) возрастает зависимость времени прибытия T от значения модуля градиента, выбор направления распространения контура становится более «чувствительным» к значениям интенсивности пикселя (вокселя) и его соседей.

В результате введения нового параметра алгоритма, из-за появившейся более тонкой настройки алгоритма, было решено отказаться от выполнения предобработки изображения с фильтром анизотропной диффузии. Как будет указано ниже, это также позволило существенно ускорить процесс сегментации в целом, особенно для случая сегментации трёхмерных изображений.

На рис. 4.5 показано, как, используя разные значения параметров, можно выполнить сегментацию области мозжечка на МРТ-изображении.

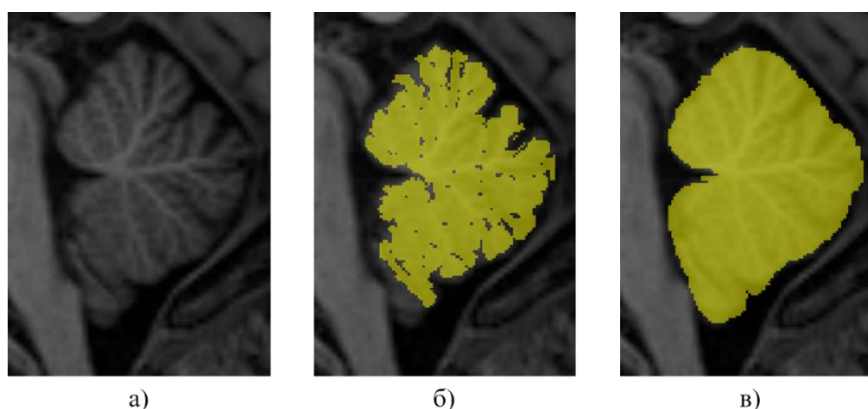


Рис. 4.5. Изображение мозжечка (а); результат множественного применения алгоритма для различных центров, расположенных внутри области, с радиусом 20 и параметром «чувствительности» 6.4 (б); последующее, после (б), аналогичное применение алгоритма для разных центров, но с радиусом 10 и «чувствительностью» 0.4

5. Реализация

Финальная версия алгоритма была реализована на языке C++. Инструмент сегментации, основанный на описанном алгоритме, был реализован в качестве части приложения, включающего в себя и другие инструменты сегментации. В этом же приложении уже существовал интерфейс, позволяющий загружать и просматривать медицинские изображения, вводить все необходимые входные параметры конкретного инструмента. Для обеих версий алгоритма графический интерфейс состоит из следующих элементов:

- Ползунок выбора значения радиуса, диапазон от 4 до 40, с шагом в 1
- Ползунок выбора значения параметра «чувствительности», диапазон от 0 до 10, с шагом в 0.1

Позиция центра выбирается нажатием кнопки мыши по одной из трёх типов проекций трёхмерного изображения. Инструмент двумерной сегментации выполняет сегментацию в той же проекции, трёхмерной – соответственно в трёхмерной области изображения.

Несмотря на то, что в финальной версии фильтр анизотропной диффузии не используется, он также был реализован на C++ в рамках того же приложения, и для двумерного, и для трёхмерного случая.

Изначально промежуточные версии алгоритма для случая двумерных изображений были написаны на языке Python, на нём же проводились исследования по внедрению модификаций в алгоритм, а также подбирались способы предобработки изображений.

5.1. Архитектура реализации алгоритмов

Переход от двумерного случая алгоритма к трёхмерному не меняет существенно алгоритм, с добавлением ещё одной оси изменяются только вид окрестности узла, формулы вычисления функций F и T . Поэтому в программной реализации класс алгоритма сегментации двумерных изображений наследуется от класса сегментации трёхмерных изображений, сам алгоритм при этом

реализован только в родительском классе. На рис. 5.1 отображена UML-диаграмма классов инструментов, основанных на рассмотренных алгоритмах.

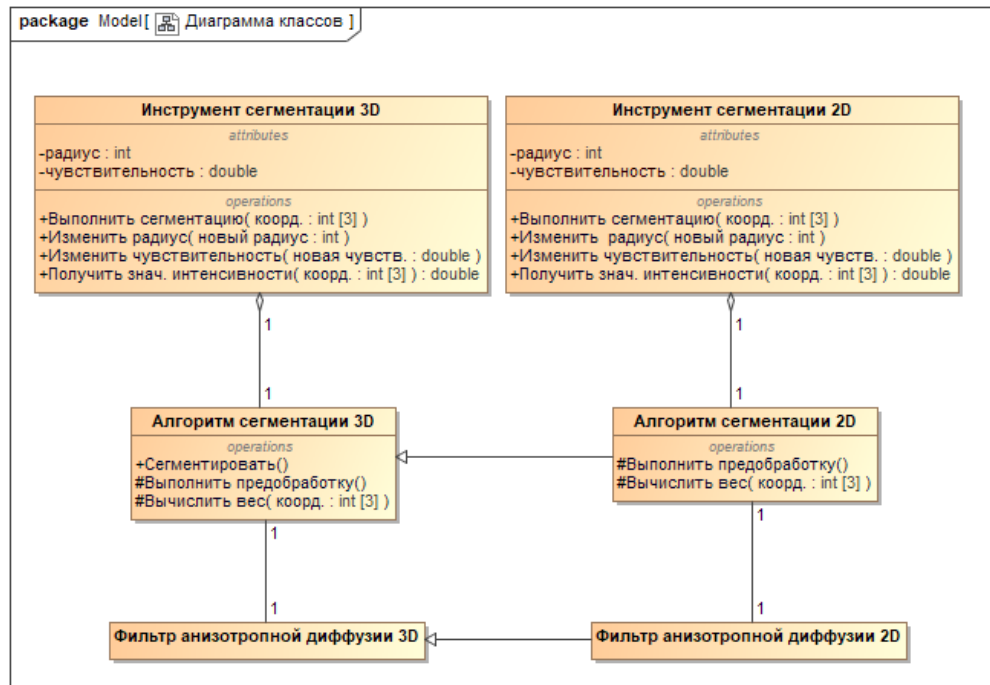


Рис. 5.1. Диаграмма классов

6. Результаты

6.1. Измерения времени работы алгоритмов

Для проверки того, что алгоритм работает в реальном времени для указанных диапазонов значений параметров, были выполнены измерения времени работы алгоритмов. Измерения проводились следующим образом: в разных позициях изображения алгоритмы запускались с одним и тем же определённым значением радиуса, результаты измерений усреднялись.

В табл. 6.1 показаны результаты измерений времени для алгоритма с фильтрацией и финальной версии алгоритма в случае двумерных изображений. В табл. 6.2 показаны аналогичные результаты для алгоритмов сегментации трёхмерных изображений.

Таблица 6.1

Среднее время работы двух версий алгоритма при разных радиусах,
двумерный случай

Реализация	Радиус			
	10	20	30	40
Время работы алгоритма с фильтрацией, мс	5.54	6.88	9.11	12.61
Время работы алгоритма с доп. параметром, мс	2.12	2.65	3.87	6.56

Таблица 6.2

Среднее время работы двух версий алгоритма при разных радиусах,
трёхмерный случай

Реализация	Радиус			
	10	20	30	40
Время работы алгоритма с фильтрацией, мс	14.89	81.04	319.19	570.45
Время работы алгоритма с доп. параметром, мс	3.87	22.22	73.68	166.20

По результатам измерений можно сделать вывод, что двумерная версия алгоритма удовлетворяет условию работы в реальном времени при любых значениях параметра радиуса, трёхмерная версия – только для значений радиуса до 20 - 30. Кроме этого, из табл. 6.1 и табл. 6.2 видно, что время работы финальной версии алгоритма, то есть алгоритма с дополнительным параметром, существенно ниже времени работы алгоритма с фильтрацией, особенно для трёхмерной версии.

6.2. Оценка качества сегментации

Для оценки результатов сегментации алгоритма были выбраны три области интереса МРТ-изображения мозга (рис. 6.1):

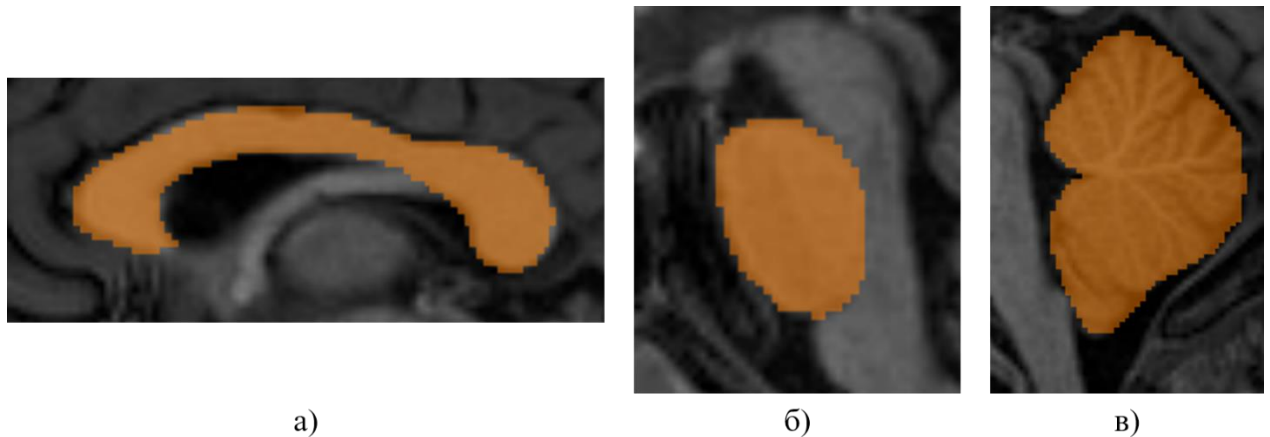


Рис. 6.1. Мозолистое тело (а), варолиев мост (б), мозжечок (в). Цветом показана эталонная сегментация

Мозолистое тело было выбрано, так как имеет преимущественно сильные границы, отделяющие его от смежных областей мозга. Варолиев мост имеет слабые границы. Мозжечок, как уже отмечалось ранее, является неоднородной областью интереса, что представляет собой дополнительную сложность для его сегментации.

Для сегментации областей алгоритм запускался многократно, для разных положений центра внутри области, как правило с несколькими выбранными значениями радиуса и «чувствительности».

Результаты оценивались относительно эталонной сегментации. Для этого использовались оценки *precision*, *recall* и F_1 , которые вычисляются по следующим формулам:

$$precision = \frac{TP}{TP+FP}, \quad (6.1)$$

где TP – число правильно отмеченных пикселей (вокселей);

FP – число ошибочно отмеченных пикселей (вокселей).

$$recall = \frac{TP}{TP+FN}, \quad (6.2)$$

где FN – число ошибочно неотмеченных пикселей (вокселей).

$$F_1 = 2 * \frac{precision*recall}{precision+recall} \quad (6.3)$$

В табл. 6.3 показаны значения оценок для трёх областей, для двумерного случая. В табл. 6.4 показаны результаты для трёхмерного случая.

Таблица 6.3

Значения оценок для тестовых областей, двумерный случай

Название области	Оценка		
	precision	recall	F_1
Мозжечок	0.99	0.91	0.95
Варолиев мост	0.98	0.84	0.91
Мозолистое тело	1.0	0.74	0.85

Таблица 6.4

Значения оценок для тестовых областей, трёхмерный случай

Название области	Оценка		
	precision	recall	F_1
Мозжечок	0.91	0.69	0.78
Варолиев мост	0.96	0.69	0.81
Мозолистое тело	0.87	0.64	0.74

Полученные значения оценки *recall* указывают на то, что обе версии алгоритма выделяют недостаточно пикселей (вокселей) вблизи границ областей несмотря на то, что алгоритм был модифицирован для частичного решения данной проблемы. Также не всегда хорошо сегментируются неоднородные области. Результаты для трёхмерной версии алгоритма оказались хуже, согласно значениям F_1 . Объяснить это можно тем, что у трёхмерных областей имеется значительно больше слабых границ. Кроме того, на результат повлияло то, что из-за сложности форм областей интереса, особенно в случае мозолистого тела, применить алгоритм оказывается довольно затруднительно, не выйдя за границы области.

Качество сегментации можно считать удовлетворительным для обеих версий алгоритма, в особенности для двумерной версии. Сравнение изображений эталонной сегментации и результатов работы алгоритма показано в Приложении 1 для двумерного случая и в Приложении 2 для трёхмерного.

7. Средства разработки

- Visual Studio 2015
- Python 3.6

Заключение

В рамках данной работы был успешно исследован и реализован алгоритм полуавтоматической сегментации медицинских изображений, основанный на методе БРК.

В ходе проведенного исследования было выполнено следующее:

1. На основе алгоритма БРК разработан метод сегментации, позволяющий обходить лишь подмножество пикселей (вокселей) изображения. Разработан оптимальный способ вычисления веса узла для сегментации внутри ограниченной радиусом окрестности
2. Исследовано применение фильтра анизотропной диффузии для предобработки изображений
3. Разработан параметр «чувствительности», подбор которого позволяет обеспечить более точную сегментацию областей в зависимости от уровня шума и необходимой точности выделения границ

Все исследования были выполнены как для двумерной, так и для трёхмерной версий алгоритма.

Обе версии алгоритма были успешно реализованы на языке C++, обеспечивают достаточно высокое качество сегментации.

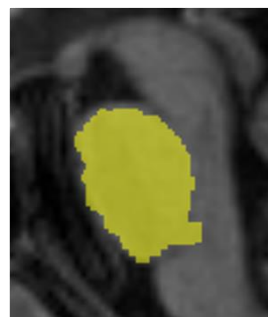
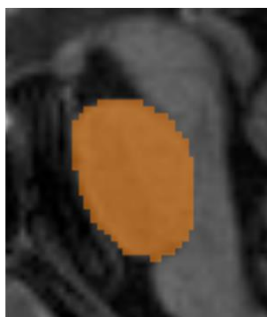
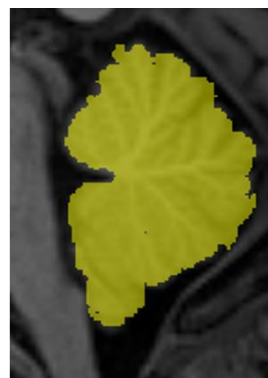
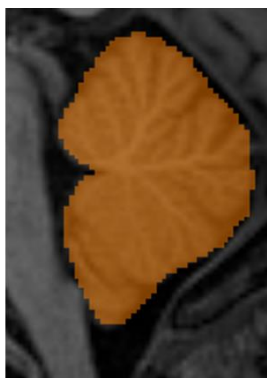
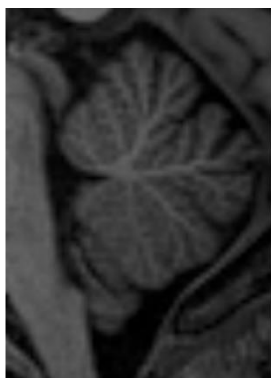
Двумерная версия алгоритма удовлетворяет условию работы в реальном времени для заданных диапазонов значений параметров. Трёхмерная версия также удовлетворяет этому условию, но не для всех значений параметра радиуса.

Список использованных источников

- [1] М.К. Вердина, А.В. Константинов, М.В. Позигун, В.С. Чуканов. Алгоритмы фиксации уровня и быстрого распространения контура для полуавтоматической сегментации медицинских изображений. Материалы научной конференции с международным участием "Неделя науки СПбПУ", 13-19 ноября 2017 года. Институт прикладной математики и механики / [редкол.: М. Е. Фролов (директор ин-та), Я. А. Гатаулин (отв. ред.) [и др.] Санкт-Петербург : Изд-во Политехн. ун-та, 2017, 310 с.
- [2] Black M. J. et al. Robust anisotropic diffusion //IEEE Transactions on image processing. – 1998. – Т. 7. – №. 3. – С. 421-432.
- [3] Chacon A., Vladimirsky A. Fast two-scale methods for eikonal equations //SIAM Journal on Scientific Computing. – 2012. – Т. 34. – №. 2. – С. A547-A578.
- [4] Gomez J. V. et al. Fast methods for eikonal equations: an experimental survey //arXiv preprint arXiv:1506.03771. – 2015.
- [5] Jeong W. K., Whitaker R. T. A fast iterative method for eikonal equations //SIAM Journal on Scientific Computing. – 2008. – Т. 30. – №. 5. – С. 2512-2534.
- [6] Malladi R., Sethian J. A. A real-time algorithm for medical shape recovery //Computer Vision, 1998. Sixth International Conference on. – IEEE, 1998. – С. 304-310.
- [7] Perona P., Malik J. Scale-space and edge detection using anisotropic diffusion //IEEE Transactions on pattern analysis and machine intelligence. – 1990. – Т. 12. – №. 7. – С. 629-639.
- [8] Sethian J. A. Level set methods and fast marching methods: evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials science. – Cambridge university press, 1999. – Т. 3.
- [9] Yatziv L., Bartesaghi A., Sapiro G. O (N) implementation of the fast marching algorithm //Journal of computational physics. – 2006. – Т. 212. – №. 2. – С. 393-399.
- [10] Zhao F., Xie X. An overview of interactive medical image segmentation //Annals of the BMVA. – 2013. – Т. 2013. – №. 7. – С. 1-22.

- [11] Zhao H. A fast sweeping method for eikonal equations //Mathematics of computation. – 2005. – Т. 74. – №. 250. – С. 603-627.

Изображения областей, эталонной сегментации и результатов работы алгоритма, двумерный случай

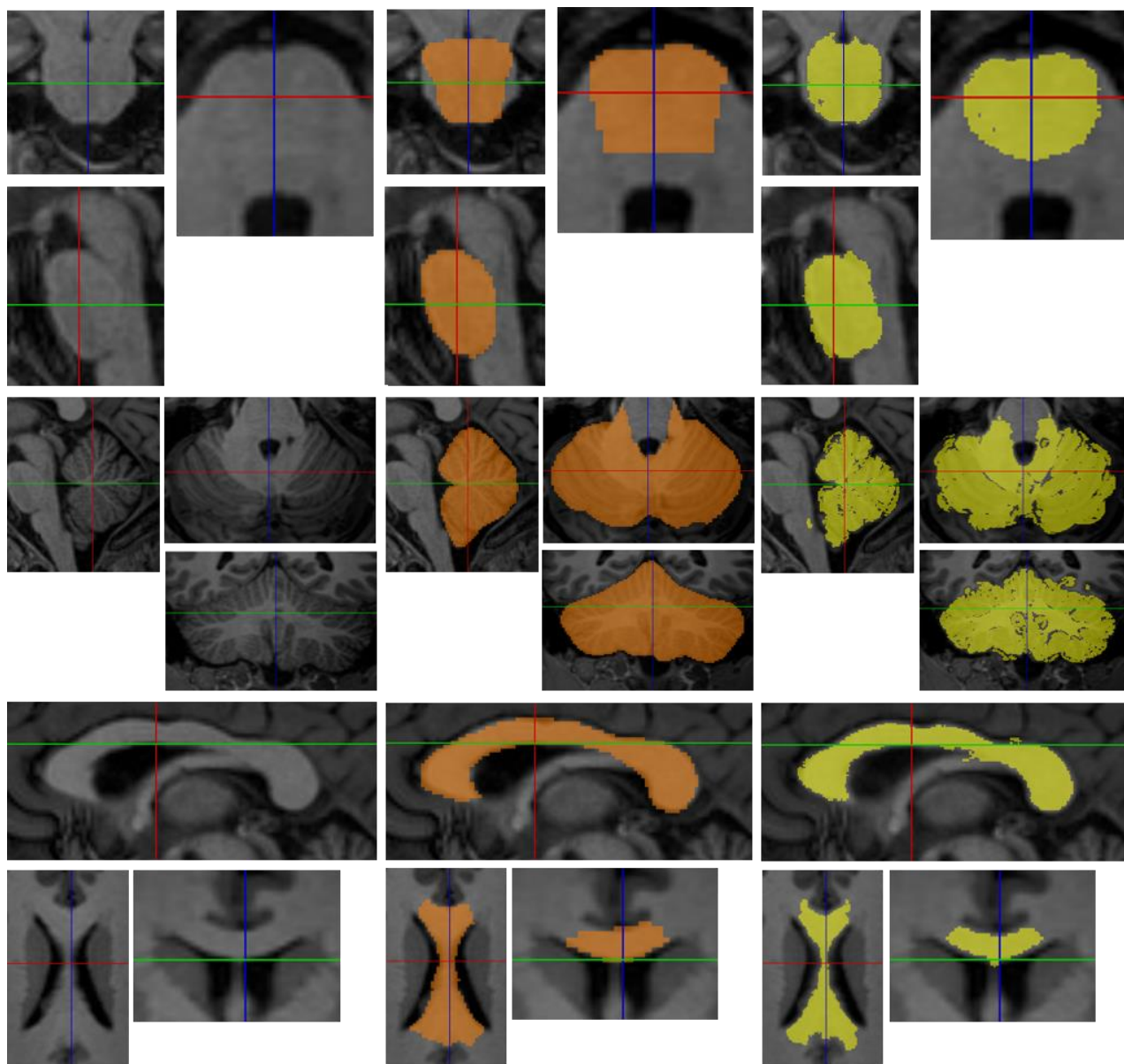


Изображения областей

Эталонная сегментация

Результаты работы алгоритма

Изображения областей, эталонной сегментации и результатов работы алгоритма, трёхмерный случай



*Изображения
областей*

*Эталонная
сегментация*

*Результаты работы
алгоритма*