



Министерство образования Российской Федерации
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени Н.Э. БАУМАНА

Реферат по курсу **алгоритмов компьютерной графики**

СЕГМЕНТАЦИЯ МЕДИЦИНСКИХ ИЗОБРАЖЕНИЙ

Выполнил: студент группы ИУ9-42Б

Масягин М.М.

Проверил: старший преподаватель

Вишняков И.Э.

Москва, 2018 г.

СОДЕРЖАНИЕ

Введение	2
Раздел 1. Современное состояние сегментации изображений в медицине	3
1.1 Основные понятия	3
1.2 Краткий исторический экскурс	4
Раздел 2. Алгоритмы первого поколения	5
2.1 Методы, основанные на детектировании границ	5
2.2 Пороговая обработка	12
2.3 Алгоритмы выращивания регионов	17
Заключение	18
Список используемой литературы	19

ВВЕДЕНИЕ

С каждым днем все большее количество достижений в области информационных технологий находит свое применение в медицине. Так, за последние двадцать лет большое влияние на развитие методов диагностирования и отслеживания динамики заболеваний оказало компьютерное зрение, в частности один из его подразделов – сегментация изображений. Под сегментацией изображения принято понимать его разбиение на отдельные фрагменты, схожие по некоторому признаку, например, яркости, текстуре или цвету. Сегментация является одним из самых важных и быстро развивающихся направлений машинного зрения, потому что зачастую именно от ее точности зависит успех более высокоуровневых алгоритмов представления и описания.

Целью данного реферата является рассмотрение и сравнение базовых механизмов автоматической сегментации медицинских изображений, таких как пороговая и граничная обработка, метод выращивания регионов. Также будут упомянуты предобработка и постобработка, направленные на повышение качества входного и выходного изображений, но без какой-либо конкретики.

Планируется провести анализ асимптотической сложности алгоритмов, возможностей их оптимизации и реализации на современных аппаратных и программных средствах.

РАЗДЕЛ 1 СОВРЕМЕННОЕ СОСТОЯНИЕ СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ В МЕДИЦИНЕ

1.1 ОСНОВНЫЕ ПОНЯТИЯ

Прежде чем перейти к рассмотрению конкретных методов выделения деталей изображения, формализуем понятие сегментации и введем классификацию алгоритмов по основному принципу их действия.

Процесс сегментации изображений удобно рассматривать с точки зрения теории множеств. Пусть:

1) X - конечное множество всех пикселей изображения. Кол-во элементов в X $|X| = M * N$, где M – разрешение изображения по горизонтали, а N – по вертикали;

2) 2^X – конечное множество всех подмножеств множества X , его булеан.

Количество элементов в булеане – $|2^X| = 2^{|X|} = 2^{(M*N)}$;

3) Очевидно, что 2^X содержит в том числе и такие наборы элементов X_1, \dots, X_n , что $X = X_1 \cup X_2 \cup \dots \cup X_n$, $X_i \cap X_j = \emptyset \forall i \neq j$ и X_i связно $\forall i \in 1, \dots, n$ в том смысле, что все пиксели X_i являются связными (обычно подразумевается 8-связность). Среди всех подобных наборов найдется хотя бы один такой, который и будет соответствовать результату удачной сегментации;

4) Необходимо различать соседние выделенные части, иначе они сольются в одну область. Для этого введем логический предикат $P(X_i)$, определенный на подходящем наборе элементов булеана X_1, \dots, X_n (3) следующим образом: $P(X_i) = true \forall i \in 1, \dots, n$ и $P(X_i \cup X_j) = false \forall$ смежных сегментов X_i, X_j . Сегменты называются смежными, если результатом их объединения является связное множество пикселей.

Любой из рассматриваемых впоследствии алгоритмов будет удовлетворять всем четырем «условиям», потому что будет производить полное разбиение изображения, состоящего из конечного числа пикселей (1), на семейство непересекающихся связных множеств (2, 3) так, что все пиксели каждого множества будут удовлетворять некоторому условию, а все пиксели окружающих его множеств не будут (4).

Существуют три основные группы алгоритмов сегментации, базирующиеся на различных свойствах изображения:

- 1) Граничные методы (boundary following), задачей которых является выделение контуров на медицинском снимке;
- 2) Методы, основанные на регионах (группах) пикселей (region-based). Они объединяют пиксели, удовлетворяющие некоторому условию, в сегменты изображения.
- 3) Методы, базирующиеся на классификации пикселей (pixel classification). Зачастую они сразу определяют в какой цвет нужно покрасить конкретный пиксель и работают быстрее, чем первые 2 вида.

1.2 КРАТКИЙ ИСТОРИЧЕСКИЙ ЭКСКУРС

Все существующие на данный момент методы сегментации медицинских изображений принято разделять на три поколения.

Первое поколение характеризуется низкоуровневыми алгоритмами, опирающимися на различные эвристики, которые дают грубое разбиение изображения. Как ни странно, эти методы по сей день остаются наиболее актуальными, так как, во-первых, на них базируются все более современные механизмы сегментации, а во-вторых, для решения задач, в которых требуется быстрая обработка большого количества изображений, они оказываются наиболее применимыми. Это связано с тем, что методы первого поколения обладают высокой производительностью и легко распараллеливаются. К ним относят алгоритмы пороговой обработки (threshold), выращивания регионов (region-growing) и различные детекторы границ (edge detection).

Алгоритмы второго поколения смогли практически полностью избавиться от эвристической составляющей своих предшественников, но все еще остаются сильно похожими на них. Они базируются на более совершенных вероятностных и оптимизационных методах разбиения изображений, зачастую сочетают в себе преимущества нескольких видов алгоритмов первого поколения. Методы сегментации второго

поколения являются эффективными и информативными, но все еще сильно зависят от качества исходного изображения.

Результаты работы алгоритмов первого и второго поколений не всегда совпадают с «ручным» разбиением высококвалифицированного эксперта-медика. Это связано с тем, что реальный человек пользуется собственным опытом и знаниями, которые, очевидно, недоступны обычным компьютерным программам. Алгоритмы третьего поколения обходят данную проблему за счет накопления и анализа информации об искомых элементах изображения. При этом для простоты можно считать, что $\forall i > j$ качество i -ой сегментации выше качества j -ой сегментации примерно одной сложности.

В следующей главе будет проведен анализ и сравнение различных алгоритмов сегментации первого поколения, как наиболее общих и фундаментальных. Мы не будем рассматривать алгоритмы последующих поколений, так как большинство из них легко выражаются в «терминах» первого поколения, а методы, базирующиеся на накоплении «знаний», в первую очередь, стоит рассматривать с точки зрения машинного обучения и нейронных сетей, а не сегментации.

2 АЛГОРИТМЫ ПЕРВОГО ПОКОЛЕНИЯ

2.1 МЕТОДЫ, ОСНОВАННЫЕ НА ДЕТЕКТИРОВАНИИ ГРАНИЦ

Одними из простейших алгоритмов сегментации медицинских изображений являются так называемые детекторы границ. Они основываются на том факте, что яркость множества пикселей представима в виде дискретной функции $f(x, y)$, где x – координата пикселя по вертикали, а y – по горизонтали (по сей день в сфере обработки изображений так и не решен вопрос о выборе направления координатных осей и точки начала отсчета; на протяжении всего реферата мы будем использовать систему координат, принятую в книгах Рафаэля Гонсалеса – пионера цифровой обработки: начало координат расположено в верхнем левом углу изображения, ось Ox направлена вниз, а ось Oy – вправо). Очевидно, что данная функция будет дифференцируема на всем наборе точек, и, как любая дифференцируемая функция,

может быть исследована с помощью 1-ой и 2-ой производных. 1-ая производная будет характеризовать наличие перепадов яркости, а 2-ая – их скорость. Так, для производных по x :

$$\text{sign}(f'(x, y)_x) = \begin{cases} -1, & \text{убывание яркости по } x \\ 0, & \text{постоянная яркость по } x \\ 1, & \text{возрастание яркости по } x \end{cases}$$

$$\text{sign}(f''(x, y)_{xx}) =$$

$$= \begin{cases} -1, & \text{нелинейное убывание яркости по } x \\ 0, & \text{линейное изменение яркости по } x \text{ или постоянная яркость} \\ 1, & \text{нелинейное возрастание яркости по } x \end{cases}$$

Для производных по y аналогично.

Так как функция яркости дискретна и наименьший возможный шаг $\delta x = 1$, то формулы 1-ой и 2-ой производных по x принимают следующий вид:

$$f'(x, y)_x = f(x + 1, y) - f(x, y)$$

$$f''(x, y)_{xx} = f(x + 1, y) - 2 * f(x, y) + f(x - 1, y)$$

Производные по y вычисляются аналогично.

Учитывая, что границы элементов изображения характеризуются перепадами яркости, мы можем находить их посредством вычисления частных производных функции $f(x, y)$, а точнее, модуля ее градиента, так как при поиске границ нас в первую очередь интересует сам факт перепада.

$$\nabla f(x, y) = \begin{pmatrix} f'(x, y)_x \\ f'(x, y)_y \end{pmatrix}$$

$$|\nabla f(x, y)| \approx \sqrt{(f'(x, y)_x)^2 + (f'(x, y)_y)^2}$$

Операция извлечения корня в программировании считается “дорогой”, поэтому часто модуль градиента считают приближенно по следующей формуле:

$$|\nabla f(x, y)| \approx |f'(x, y)_x| + |f'(x, y)_y|$$

Сами производные удобно находить посредством пространственной фильтрации – последовательного перемещения по всему изображению матрицы-фильтра размера $m * n$, элементами которого являются коэффициенты при

значениях функции $f(x, y)$. Очевидно, что любая матрица, отличная от матрицы $1 * 1$, может «вылезти» за границы изображения. Это опасно в практических реализациях программ-сегментаторов, так как выход за границы массива данных приводит к падению программы. Чтобы избежать этого, изображение принято дополнять “рамкой” толщиной $m - 1$ пиксель по горизонтали и $n - 1$ по вертикали (яркость рамки при этом обычно используется либо нулевая, либо такая же, как и у соответствующих граничных пикселей).

Для вычисления частных производных $f(x, y)$ достаточно одномерных матриц: $\begin{pmatrix} -1 & 1 \end{pmatrix}$ для $f'(x, y)_x$ и $\begin{pmatrix} -1 & 1 \end{pmatrix}$ для $f'(x, y)_y$, однако на практике они малоприменимы, так как они не учитывают диагональные перепады яркости, что абсолютно неудовлетворительно при использовании 8-связности.

Первый оператор, который оказался употребимым в реальных системах сегментации изображений (в том числе и медицинских) был предложен Лоуренсом Робертсом в 1963 году. Он решал проблему потери диагональных перепадов яркости посредством использования матриц $2 * 2$: $\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ для $f'(x, y)_x$ и $\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ для $f'(x, y)_y$.

Главным недостатком перекрестного оператора Робертса была невозможность выделения центрального пикселя, обусловленная четностью размера матрицы (обычно в компьютерном зрении используются квадратные матрицы-фильтры, поэтому под четностью-нечетностью размера матрицы мы будем подразумевать сразу и четность-нечетность по вертикали, и четность-нечетность по горизонтали). Чтобы решить данную проблему, необходимо использовать операторы нечетного размера. К последним относится оператор, разработанный доктором Джудит Превитт для повышения качества сегментации медицинских изображений в 1970 году. Он использует приближенное вычисление производных, опирающееся на 6 контрольных

точек: $\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ для $f'(x, y)_x$ и $\begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$ для $f'(x, y)_y$.

Очевидно, что рассмотрение большего количества окрестных пикселей и наличие “центра” позитивно сказывается на качестве выделения границ.

Еще большей точности можно добиться, не просто учитывая контрольные точки, но и задавая им «веса». Самый большой по модулю коэффициент при этом оказывается у точек вида $|i - S \div 2| \neq |j - S \div 2|$ для соответствующей производной, где S - размер матрицы, а (i, j) пробегают все возможные индексы элементов матрицы от $(0,0)$ до $(S - 1, S - 1)$ (линейное направление). В качестве примера рассмотрим два оператора:

1) Оператор Ирвина Собеля и Гэри Фельдмана 1970 года:

$$\begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \text{ для } f'(x, y)_x \text{ и } \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \text{ для } f'(x, y)_y.$$

2) Оператор Ханно Шарра 2000 года:

$$\begin{pmatrix} -3 & -9 & -3 \\ 0 & 0 & 0 \\ 3 & 9 & 3 \end{pmatrix} \text{ для } f'(x, y)_x \text{ и } \begin{pmatrix} -3 & 0 & 3 \\ -9 & 0 & 9 \\ -3 & 0 & 3 \end{pmatrix} \text{ для } f'(x, y)_y.$$

Также для нахождения ребер может быть использовано приближенное вычисление операторов Лапласа $\nabla^2(f(x, y)) = (f''(x, y)_{xx})^2 + (f''(x, y)_{yy})^2$:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{pmatrix} \text{ или } \begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Существуют и другие, более «экзотичные» операторы детектирования границ: направленные в первую очередь на нахождение диагональных перепадов, операторы с матрицами больших размеров, чем $3 * 3$, но в качестве основных, наиболее фундаментальных, выделяют именно данные пять.

Так как все ранее рассмотренные нами методы сегментации сводятся к пространственной фильтрации, использующей некоторую квадратную матрицу, мы можем утверждать, что асимптотическая сложность подобных алгоритмов — $O(M * N * S^2)$, а их практическая реализация легко поддается распараллеливанию на современных компьютерах. Для этого можно использовать простейший “worker pool”, во входной канал которого будут подаваться индексы текущего пикселя, а в

выходной – те же индексы и значение яркости пикселя после обработки. В языках с поддержкой ассемблерных вставок – D\C\C++ можно повысить производительность, явно используя SIMD-инструкции процессора и перегруппировку цветов изображения (вместо R1 G1 B1 R2 G2 ... – R1 ... Rn B1 ... Bn ... для соседних пикселей). Также очевидно, что можно ускорить обработку за счет использования вычислений на GPU, например, посредством NVIDIA CUDA.

Забегая вперед, заметим, что после получения граничного представления, мы можем повторно обработать все пиксели изображения согласно следующей формуле:

$$f_1(x, y) = \begin{cases} C_0, & f(x, y) \leq T \\ C_1, & f(x, y) > T \end{cases}$$

где T - некоторый порог яркости, а C_0 и C_1 два различных цвета (обычно черный и белый). Очевидно, что данная постобработка никак не сказывается на асимптотической сложности и также легко распараллеливается. О самом выборе T поговорим в части про пороговую обработку изображений.

Ниже приведены результаты применения рассмотренных выше операторов к МРТ мозга:

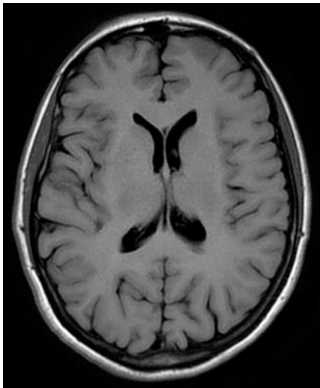


Рис 1.а. Исходное изображение.

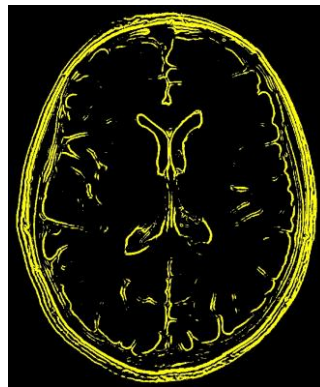


Рис 1.б. Робертс.

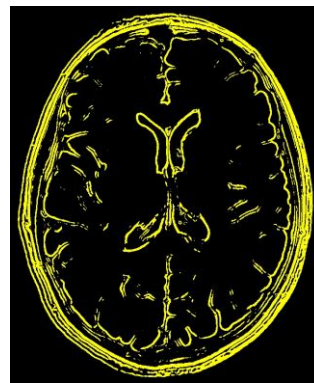


Рис. 1.в. Превитт.



Рис 1.г. Собель.

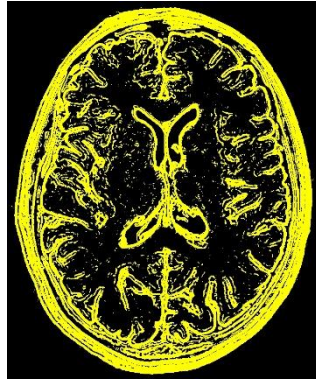


Рис. 1.д. Шарп.

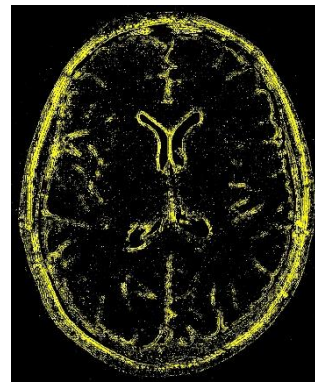


Рис 1.е. Лапласиан-4.

Долгое время единственным методом выделения границ было применение ко всему изображению малых фильтров, приближенно вычисляющих первые и вторые производные дискретной функции яркости. Ситуация изменилась в 1980 году, когда Дэвид Марр и Эллен Хилдрет предложили предварительно обрабатывать изображение масштабируемыми операторами.

Через шесть лет этой идеей воспользовался Джон Кэнни, разработав одноименный детектор, который по сей день является одним из самых точных и активно используется в медицинской сегментации. Рассмотрим принцип его действия.

На начальном этапе работы алгоритма мы строим две квадратные матрицы, являющиеся дискретизациями первых производных двумерной функции Гаусса.

$G(x, y) \approx e^{-(x^2+y^2)/(2\sigma^2)}$ – константа $1/(2 * \pi * \sigma^2)$, на которую домножается экспонента, во многих вычислительных методах игнорируется.

$$G'(x, y)_x = -x/\sigma^2 * e^{-(x^2+y^2)/(2\sigma^2)}$$

$$G'(x, y)_y = -y/\sigma^2 * e^{-(x^2+y^2)/(2\sigma^2)}$$

Размер матриц рекомендуется выбирать по следующему правилу: $S \geq 6 * \sigma$, S - нечетное.

Далее строим изображения модулей градиентов ($|\nabla G(x, y)| \approx |G'(x, y)_x| + |G'(x, y)_y|$) и углов их направления ($\alpha(x, y) = \arctg(G'(x, y)_y / G'(x, y)_x)$), у которых помимо всего прочего подавлен белый гауссов шум (сильная сторона алгоритма



Рис. 2.

Кэнни заключается в использовании общей матриц и для подавления шумов, и для вычисления производных). После этого «подавляем немаксимумы» яркости, определяя в какой из четырех участков «смотрят» вектора градиентов (см. рис.) и сравнивая их модули с модулями соседей, которые оказались в одном с ними классе. Если модуль градиента хотя бы одного из «соседей-одноклассников» превышает модуль градиента текущего пикселя, то яркость текущего пикселя зануляется, в противном случае — остается такой же. После данных действий к полученному изображению применяется трехуровневая пороговая обработка.

$$f_1(x, y) = \begin{cases} C_0, & f(x, y) \leq T_0 \\ C_1, & T_0 < f(x, y) \leq T_1 \\ C_2, & T_1 < f(x, y) \end{cases}$$

Работу алгоритма можно считать оконченной.

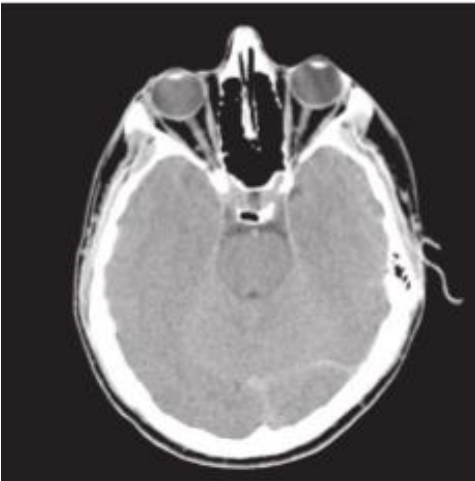


Рис 3.а. Томограмма мозга.

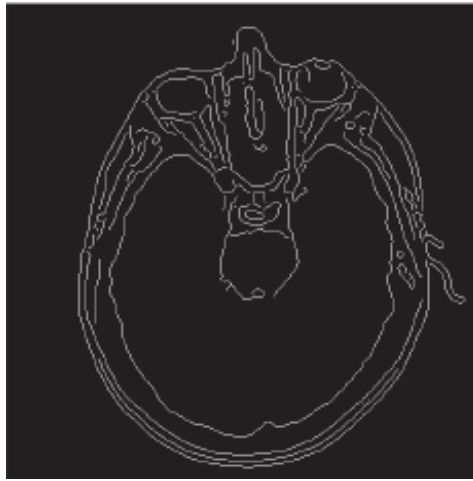


Рис 3.б. Кэнни.

Существуют различные модификации алгоритма Кэнни. Так, например, в 1995 году на Симпозиуме IEEE Чарлз Ротуэлл предложил несколько улучшенный алгоритм Кэнни: вместо «подавления немаксимумов» использовалось «источение краев», а «петля гистерезиса» была заменена на адаптивную пороговую обработку.

Для улучшения результатов сегментации возможно использование различных фильтров для удаления шума, а также постобработки в виде связывания или аппроксимации ломаными линиями контуров.

Детекторы границ повсеместно используются в медицине. Наиболее эффективными они показали себя в задачах исследования (в том числе нахождения опухолей) сложных структур, таких как мозг, легкие, сердце.

2.2 ПОРОГОВАЯ ОБРАБОТКА

В прошлом подразделе мы уже касались методов пороговой обработки медицинских изображений. Поговорим о них подробнее.

Наиболее общо все методы пороговой обработки могут быть представлены в виде следующей формулы:

$$f_1(x, y) = \begin{cases} C_0, & 0 \leq f(x, y) \leq T_0 \\ C_1, & T_0 < f(x, y) \leq T_1 \\ \dots, & \dots \\ C_{n-1}, & T_{n-2} < f(x, y) \leq T_{n-1} \\ C_n, & T_{n-1} < f(x, y) \leq T_n \end{cases}, \text{ где } T_{0,\dots,n} - \text{некоторые пороги яркости, а}$$

$C_{0,\dots,n}$ - различные цвета.

При $n = 2$ мы получим частный случай, называемый «бинаризацией изображения».

Стоит различать пороговую обработку изображений, предварительно обработанных фильтрами-детекторами границ (она рассматривалась ранее), и бинаризацию без предобработки, так как в первом случае мы выделяем лишь границы (области постоянной яркости, даже если они не относятся к фону, дают нулевой отклик по модулю градиента и закрашиваются цветом фона), а во втором – можем выделить сами объекты (но при этом потерять их границы).

Самой сложной частью в алгоритмах пороговой обработки является собственно поиск порога: при выборе слишком высокого порога, мы теряем информацию, при выборе слишком низкого – получаем ложные данные.

Рассмотрим некоторые способы выбора порога и оценим их асимптотическую сложность (очевидно, что сложность преобразования изображения после нахождения граничных значений яркости – $O(M * N)$, и обычно либо такая же, либо ниже чем сложность нахождения порогов).

Самым простым и универсальным методом выбора порогов является использование гистограммы изображения (таблицы вида: яркость – количество пикселей данной яркости). После построения мы можем модифицировать ее, «выкинув» все яркости, которым соответствует нулевое количество пикселей. Затем в «обновленной» гистограмме мы ищем все «пики» и во «впадинах» между ними выбираем пороги. Таким образом, при наличии ярко выраженных мод распределения яркости и больших расстояний между ними мы можем с легкостью находить пороги. Асимптотическая сложность алгоритма – $O(M * N)$, так как нам необходимо проанализировать все пиксели изображения. А несколькими проходами

по массиву из не слишком большого числа элементов можно пренебречь. Также возможно использование этого алгоритма для градиента исходного изображения, что позволяет избежать ошибок, связанных с размерами объектов.



Рис. 4.а. Рентгенограмма руки.

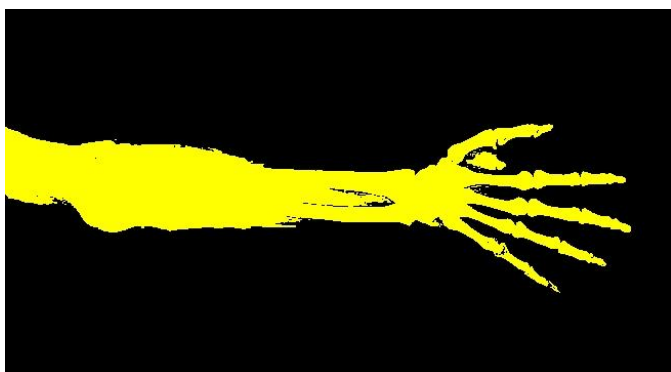


Рис. 4.б. Пороговая обработка по всему изображению (неудачная).



Рис. 4.в. Пороговая обработка по контурам (удачная).

Если для обработки с несколькими порогами не удастся получить удовлетворительные результаты сегментации прошлым методом, то рекомендуется

выбрать другой тип алгоритма (например, метод выращивания регионов, метод за-
 краски, или же соответствующий алгоритм 2 поколения). При бинаризации возмож-
 но использование различных вероятностных и статистических методов, повышаю-
 щих точность задания глобального порога.

Наиболее известным среди подобных методов является алгоритм, разработан-
 ный Нобуюки Отсу в 1978 году. На начальном этапе мы должны построить и норма-
 лизовать гистограмму изображения (нормализация – деление каждого количества
 пикселей соответствующей яркости на $M * N - \|p_i\| = p_i / (M * N)$). Затем для всех
 возможных порогов яркости (то есть всех яркостей min, \dots, max) рассчитываются
 вероятность попадания и средняя яркость пикселей, оказавшихся по обе стороны от
 границы: $P_1(k) = \sum_{i=min}^k \|p_i\|$, $m_1(k) = P_1^{-1} * \sum_{i=min}^k i * \|p_i\|$ и $P_2(k) = 1 - P_1(k)$,
 $m_2(k) = P_2(k)^{-1} * \sum_{i=k+1}^{max} i * \|p_i\|$, а также вычисляется средняя яркость всего
 изображения: $m_G(k) = \sum_{i=min}^{max} i * \|p_i\|$. После этого, опираясь на полученные
 значения, мы можем оценить межклассовую дисперсию при конкретном пороге.
 Граница, соответствующая максимальной дисперсии, является искомой. Оценка
 дисперсии происходит по формуле $\sigma^2 = P_1 * (m_1 - m_G)^2 + P_2 * (m_2 - m_G)^2 = \dots =$
 $(m_G * P_1 - m) / (P_1 * (1 - P_1))$, где $m = \sum_{i=min}^k i * \|p_i\|$. Очевидно, что
 асимптотическая сложность алгоритма – $O(M * N)$, так как проходы по гистограмме
 можно считать константой. Тем не менее, эта константа имеет большую вычисли-
 тельную сложность, что необходимо учитывать при практических реализациях.

Метод Отсу использует глобальную пороговую обработку, то есть выбирает
 общий порог для всего изображения. Возможна и иная стратегия, при которой
 изображения разбивается на несколько областей, для каждой из которых, как для
 отдельного снимка, находится собственная граница яркости.

Классическими, наиболее выгодными с точки зрения скорости и качества, при-
 мерами такого подхода являются алгоритмы Ниблэка, Савуола и Кристиана. Рас-
 смотрим их.

Суть алгоритма Ниблэка заключается в вычислении локального порога для каждого пикселя по его соседям с использованием среднего значения и стандартного отклонению яркости.

$T(x, y) = m(x, y) + k * \sqrt{\sigma^2(x, y)}$, где T – значение локального порога, m – среднее значение, σ^2 – дисперсия, а $k \in R$. Обычно $k = -0.2$ при поиске темных объектов на светлом фоне и $k = 0.2$, наоборот, при поиске светлых объектов на темном фоне.



Рис 5.а. Применение алгоритма Ниблэка к рисунку 4.а.

Недостатком алгоритма Ниблэка является неудовлетворительная точность сегментации при низкоконтрастном входном изображении, что достаточно часто встречается в медицине. Сравнимо лучшее качество разбиения могут дать его модификации — методы Савуола и Кристиана. В них используются следующие формулы выбора локального порога:

$$T(x, y) = m(x, y) * \left(1 + k * (\sqrt{\sigma^2}/128 - 1)\right), \text{ где } k \in [0.2, 0.5].$$

$$T(x, y) = (1 - k) * m(x, y) + 0.5 * M + 0.5 * (\sqrt{\sigma^2}/R) * (m(x, y) - M), \text{ где } M - \text{минимальная яркость, } R - \text{максимальное среднеквадратичное отклонение яркости из локального окна.}$$

Зачастую пороговая обработка применяется совместно с детектированием границ при выделении костей на рентгенограммах. Она позволяет эффективно детектировать их повреждения и смещения, отсекая с изображений все лишнее, например, внешние контуры тела и связанные с ними засветы.

2.2 АЛГОРИТМЫ ВЫРАЩИВАНИЯ РЕГИОНОВ

В прошлых двух подразделах мы рассматривали методы, принцип действия которых заключался в построчно-попиксельной обработке всего изображения некоторой матрицей (числа – матрицы размера $1 * 1$). Алгоритмы выращивания регионов работают несколько иначе. Сначала на изображении находятся «точки кристаллизации», которые наверняка удовлетворяют заданным свойствам. Затем от них начинается разрастание пикселей, также удовлетворяющих заданным свойствам. Работа завершается, когда все пиксели изображения оказываются обработанными. Для фиксирования этого в реальной реализации можно завести переменную-счетчик количества необработанных пикселей и матрицу, элементы которой показывают, был ли элемент обработан или нет. Данный алгоритм способен показывать превосходные результаты при грамотном подборе предикатов. При сегментации медицинских изображений, которые обычно представлены градациями серого, в предикат включают связность, яркость, текстуру, а порой и некоторые статистические данные.

Другим вариантом алгоритмов выращивания регионов является предварительное разбиение всего изображения на отдельные области (для этого можно использовать слегка модифицированные детекторы границ) и последующее слияние/разбиение полученных областей согласно заданным предикатам.

Оценить асимптотику данных алгоритмов без рассмотрения конкретных предикатов довольно сложно. Если основываться на том, что каждый пиксель может быть обработан один раз и всегда рассматривается 8-связная окрестность, то получаем «классическое» $O(M * N)$, но стоит понимать, что реальные реализации данных алгоритмов не слишком быстрые. Параллельная работа таких алгоритмов более слож-

на, так как выходные пиксели могут зависеть друг от друга. Одним из подходов является выделение отдельных потоков под каждую «точку кристаллизации» (первый подтип алгоритма), под некоторые группы регионов, полученных предобработкой (второй подтип алгоритма).

Алгоритм выращивания регионов нашел широчайшее применение в медицине. Существует множество его модификаций, которые используют самые разные предикаты (а порой и методы искусственного интеллекта) и могут быть отнесены как к первому, так и второму и третьему поколениям алгоритмов сегментации.

ЗАКЛЮЧЕНИЕ

Возможность автоматически разбивать медицинские изображения на отдельные участки, а затем и выделять среди них интересующие участки произвела революцию в современной медицине. Даже простейшие методы, которые способен программно реализовать любой студент инженерной или физико-математической специальностей оказываются вполне эффективными, способными облегчить труд врача.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Гонсалес Р.С., Вудс Р.Е. Цифровая обработка изображений / науч. ред. П.А. Чочиа; пер. с англ. Л.И. Рубанова, П.А. Чочиа. 3-е изд. доп. и испр. М.: Техносфера, 2012. 1104 с.
2. Шапиро Л.Г., Стокман Дж.К. Компьютерное зрение / под ред. С.М. Соколова; пер. с англ. А.А. Богуславский. М.: Бином. Лаборатория знаний, 2006. 752 с.
3. Тропченко А.Ю., Тропченко А.А. Методы вторичной обработки и распознавания изображений. / Спб.: Университет ИТМО, 2015. 215 с.
4. Ватутин Э.И., Мирошниченко С.Ю., Титов В.С. Программная оптимизация оператора Собела с использованием SIMD-расширений процессоров семейства x86 / Телекоммуникации. 2006. N 6. С. 12-16.
5. Исрафилов Х.С. Исследование методов бинаризации изображений / М.: МГТУ им. Н.Э. Баумана, 2017.
6. Canny Jh. A Computational Approach to Edge Detection. / IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. PAMI-8. 1986. N. 6. С. 679-698.
7. Withey D.J., Koles Z.J. Medical Image Segmentation: Methods and Software / Proceedings of NFSI & ICFBI. 2007. С. 141-143.
8. Kumar S.N., Lenin Fred A., Muthukumar S., Ajay Kumar H., Sebastian Varghese P. / A voyage on medical image segmentation algorithms. 2018. Biomedical Research 2018. С. 75-87.
9. Pham D.L., Xu Ch., Prince J.L. Current methods in medical segmentation / Annu. Rev. Biomed. Eng. 2000. N 2. С. 315–337