

Построение иерархического дерева детальности сегментации изображения через поиск минимальных разрезов графа

Непомнящий П.В., Юрин Д.В.

ЦОС и ВТ МФТИ

Москва, Россия

Аннотация

Разработан алгоритм снижения детальности отсегментированного изображения на основе построения иерархического дерева детальности сегментов. В основу алгоритма легло использование задачи о поиске минимального разреза на графе. Отсегментированное изображение представляется в виде графа, в котором сегменты соответствуют узлам, а границы ребрам. Веса ребер определяются исходя из степени заметности границы и различимости сегментов, которые она разделяет. Путем многократного решения задачи о минимальном разрезе граф разделяется и строится иерархическое дерево детальности. Получившееся дерево отображается в последовательность выделения границ на сегментированном изображении.

Ключевые слова: иерархическая сегментация, слияние сегментов, снижение детальности, векторизация изображений, минимальный разрез, графы.

1. ВВЕДЕНИЕ

Векторизация изображения является важным этапом в анализе изображений. Векторизованное изображение используется для задачи совмещения и взаимной привязки нескольких изображений одной сцены или объекта. Основные приложения этой задачи – привязка аэрокосмических изображений к контурной карте, восстановление трехмерных объектов [3,16] и сцен по серии снимков, создание круговой панорамы по набору изображений полученных путем неравномерного поворота камеры вокруг какой-либо оси (с переменным увеличением), восстановление трехмерных моделей помещений после съемки их на камеру.

Для каждого из контуров векторного изображения можно построить вектор параметров: длина, извилистость, фрактальные характеристики, точность аппроксимации типовыми формами (дугой, прямой, полиномом), точки высокой кривизны, углы [1,14,15]. По построенным векторам параметров возможно произвести совмещение контуров из двух и более разных изображений (задача о максимально-взвешенном соответствии [4]).

Для целей совмещения изображений, требуется учитывать наиболее характерные и заметные черты изображения, поэтому в векторном изображении должны быть только самые заметные контуры, составляющие “набросок” сцены. Отсюда возникает задача снижения детальности исходной сегментации [18,19], и выделения из множества обнаруженных границ наиболее заметных.

Существует множество алгоритмов для векторизации растрового изображения, которые подразделяются на алгоритмы сегментации [2,7,8,20] (выделение на изображении

областей с замкнутыми границами), и на детекторы границ [5,9] (поиск границ, не обязательно замкнутых). В настоящее время алгоритмы на графах, в частности основанные на нахождении минимального разреза часто применяются для решения задач о сегментации изображений [10,18].

Используемый в качестве векторизатора в данной работе алгоритм RSEG [2] имеет преимущество в том, что он выделяет даже малозаметные для человеческого глаза границы (рис 1,2). Однако это приводит к тому, что количество выделяемых им сегментов очень велико и в результате получившаяся сегментация очень детальна.

Задача снижения детальности сегментации состоит в снижении числа исходных сегментов путем их слияния. В результате на отсегментированном изображении должно остаться небольшое число крупных сегментов, разделенных хорошо заметными границами. Однако информация о более мелких сегментах также может понадобиться, поэтому целесообразно построить иерархическую сегментацию, которая образует пирамиду детальности – на самом верхнем уровне находятся крупные сегменты, а на более низких размер сегментов последовательно уменьшается, а их число увеличивается.

2. АЛГОРИТМ СЛИЯНИЯ СЕГМЕНТОВ

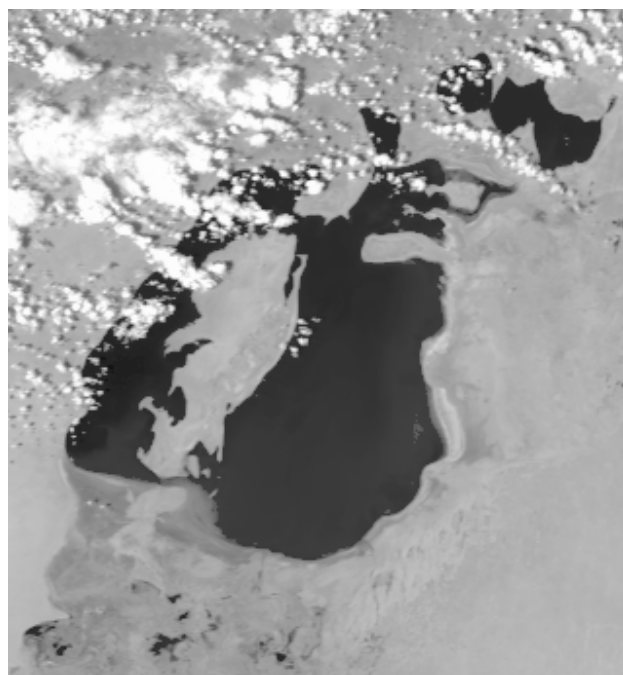


Рисунок 1. Исходное изображение



Рисунок 2. Отсегментированное изображение (RSEG)

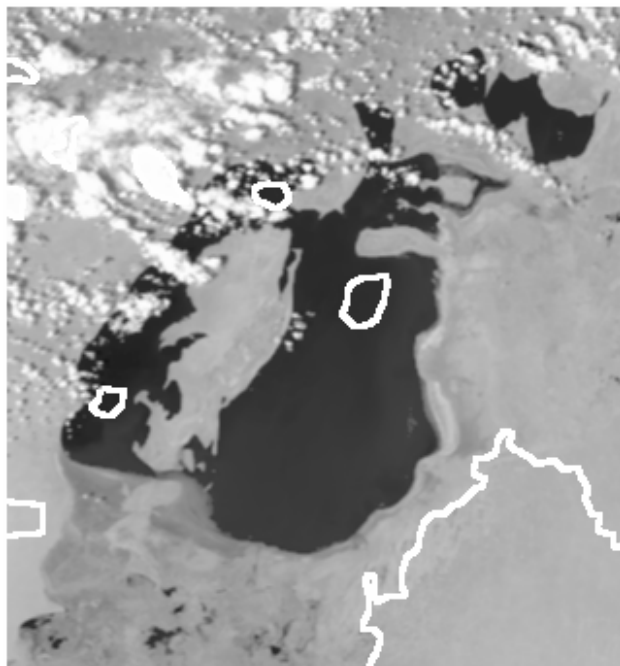


Рисунок 3. Уменьшение детальности до 10 сегментов (R-величина)

Ранее для слияния сегментов в сегментации, полученной с помощью алгоритма RSEG, применялись следующие алгоритмы: по R-величине соседних сегментов, введенной в [2] и путем минимизации функционала Мамфорда-Шаха [7], но

все они не давали хорошего результата, во многих случаях сегменты сливались не адекватно. (рис 3, 4).

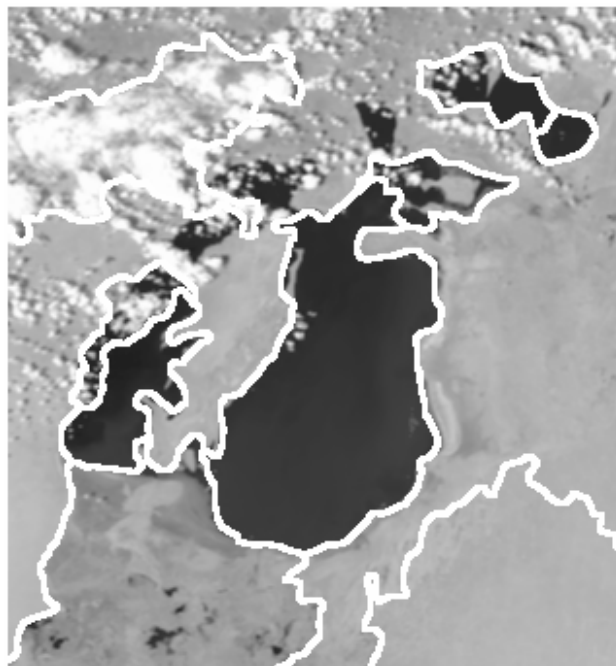


Рисунок 4. Уменьшение детальности до 10 сегментов (минимизация функционала)

На изображениях видно, что слияние по R-величине дает очень плохие результаты, есть тенденция к выделению мелких сегментов. Второй способ слияния дает существенно лучшие результаты, но наряду с четкими сегментами (водоемы), выделяет и плохие (часть суши в нижнем левом углу). Поэтому для осуществления слияния был разработан новый алгоритм, основанный на идее сегментации на графах через минимальные разрезы [10].

3. МИНИМАЛЬНЫЙ РАЗРЕЗ ГРАФА

Допустим A и B есть два подмножества из множества вершин V графа $G(V, E)$, такие что:

(2.1)

$$V = A \cup B$$

(1)

$$A \cap B = \emptyset$$

Тогда разрезом C называется множество ребер, соединяющих вершины из A с вершинами из B , а весом разреза называется сумма всех входящих в него ребер (рис.2.7):

$$c(C) = \sum_{p \in A, q \in B} w(p, q) \quad (2)$$

Минимальным разрезом называется разрез минимального веса. Одним из видов минимального разреза является (s,t)-разрез, когда находится такой минимальный разрез, при котором вершины s и t лежат по разные его стороны.

Задача о минимальном разрезе находится путем решения задачи о максимальном потоке между вершинами s и t . Существует много алгоритмов для решения этой задачи, из которых наиболее эффективным является метод проталкивания

предпотока, предложенный Голдбергом и Тарьяном [13] (в оригинальном варианте алгоритм имел сложность $O(V^3)$). В этой работе применялся алгоритм, описанный в [11] (сложность $O(VE \log(V^2/E))$), использована реализация библиотеки Boost Graph Library [17], самый быстрый из известных алгоритмов описан в [12] (сложность $O(\min(V^{2/3}, E^{1/2})E \log(n^2/m))$).

4. СХЕМА АЛГОРИТМА СЛИЯНИЯ СЕГМЕНТОВ

Основная идея алгоритма почерпнута из статьи [10], где путем нахождения минимального разреза производилась сегментация растрового изображения. В данном алгоритме узлами графа выступают не пиксели, а сегменты, что позволяет снизить размерность задачи примерно на два порядка (характерная площадь сегмента пропорциональна произведению ширины и высоты рамки, которой производилась сегментация [2]; типичный размер рамки при сегментации - 8 или 16; отсюда следует вывод, что сегментов на два порядка меньше чем пикселей). За счет того, что для сегментов можно посчитать гораздо больше характеристик, чем для пикселей, можно учитывать существенно большее число факторов в весовой функции ребер. Также была изменена стратегия выбора стока и истока и использован другой способ выделения сегментов, путем последовательного разреза графа на подграфы.

На входе алгоритма (рис. 6) слияния сегментов формируется граф, который соответствует отсегментированному изображению. В роли узлов в графе выступают сегменты, а в роли ребер – границы между сегментами. Каждому ребру приписывается вес, который отражает степень “похожести” соседних сегментов и “заметность” границы, причем, чем более сегменты непохожи и чем более заметна граница, тем меньше вес ребра. Все веса неотрицательны (рис. 5).

На этом графе выбираются две вершины, которые являются стоком и истоком и решается задача максимального потока, по которому находится минимальный разрез. Этот разрез соответствует самой заметной границе, разделяющей сегменты, выбранные как сток и исток. После этого ребра, составляющие минимальный разрез удаляются из графа, в результате чего образуется два подграфа. В этих подграфах, в свою очередь, выбираются (в каждом), сток и исток и снова находится минимальный разрез, после чего в получившихся подграфах этот алгоритм повторяется до тех пор, пока все вершины исходного графа не будут разъединены. По ходу алгоритма, постепенно все увеличивается детальность получающейся сегментации, до тех пор, пока она не достигнет вида исходной сегментации. Для отражения этой иерархии детальности можно построить дерево с выделенным корнем, узлами которого будут все подграфы, образовывавшиеся в процессе разрезания исходного графа. Корневой вершиной будет исходный граф, от него будут идти два ребра к узлам, соответствующим подграфам, получившимся в результате первого разреза, от этих узлов к следующим и так далее (рис. 7). Кончатся ветки дерева будут на тех узлах, которые соответствуют подграфам из одного узла. В каждый узел дерева (кроме корневого) входит одно ребро. Ему можно сопоставить вес того минимального разреза, который разрезает подграф, соответствующий узлу, в которое входит ребро.

Получившееся дерево детальности можно обойти начиная с корня так, что из возможных в каждый конкретный момент

путей продолжения обхода (то есть из всех подграфов, к которым можно применить разрез) выбирается ребро, которому соответствует разрез с минимальным весом, то есть более заметной границе.

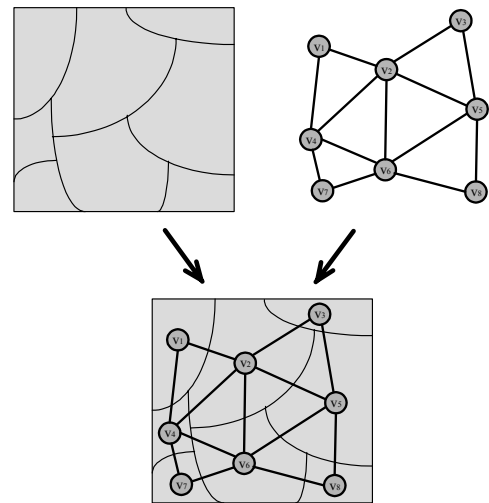


Рисунок 5. Построение графа на основе сегментации

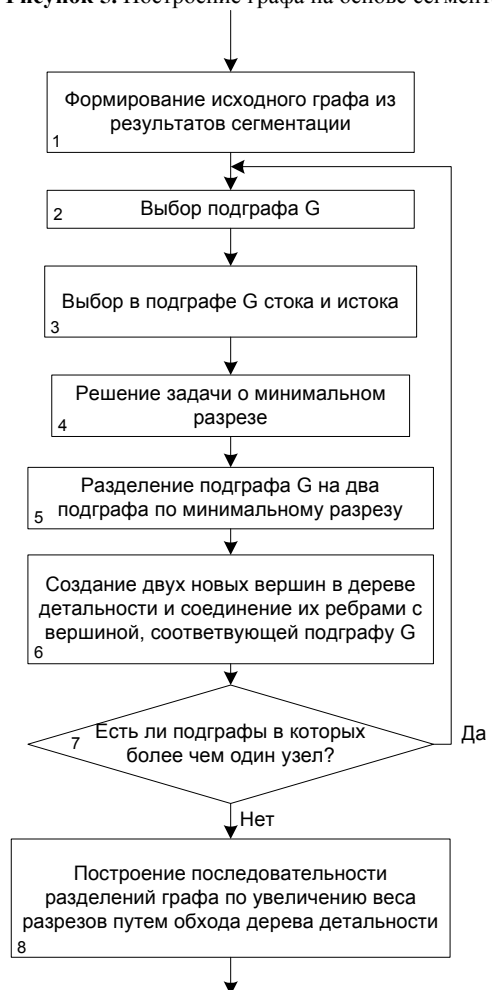


Рисунок 6. Общая схема алгоритма слияния сегментов

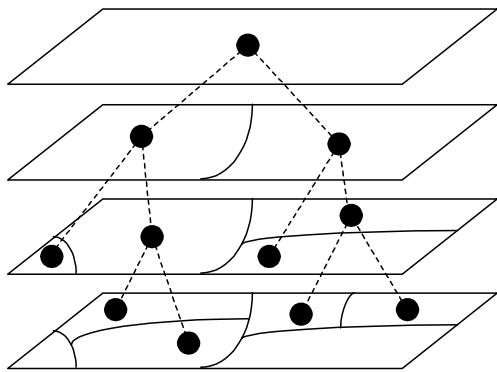


Рисунок 7. Дерево детальности

Последовательность, в которой ребра будут стоять при обходе дерева и является результатом работы алгоритма. Для того, чтобы получить результат слияния с N сегментами нужно применить первые N-1 разрез, которым соответствуют первые N-1 ребер, а после этого объединить сегменты, попавшие в один и те же подграфы.

5. ВЫБОР СТОКА И ИСТОКА

Результат работы алгоритма слияния сегментов существенно зависит от выбора сегментов, соответствующих стоку и истоку в задаче о максимальном потоке. К примеру, сток и исток расположенные на одном и том же однородном участке изображения приведут к тому, что разрез будет проходить где-то между ними, что приведет к возникновению неадекватной несильной границы. Следовательно, необходимо выбирать сток и исток таким образом, чтобы между ними были наиболее сильные и протяженные границы, а цветояростные характеристики сегментов максимально различались. В данной работе использовалось два подхода к выбору стока и истока.

Первый подход заключался в том, что в качестве предварительной обработки для каждого сегмента на изображении вычислялась средняя яркость (для RGB изображений – средняя яркость по каждому из трех каналов). Для выбора стока и истока ищались сегменты с максимально разницей между средними яркостями:

$$s, t : |\bar{I}_s - \bar{I}_t| = \max_{q, p \in V} (|\bar{I}_q - \bar{I}_p|) \quad (3)$$

Здесь $\bar{I}_q, \bar{I}_p$ - средние интенсивности сегментов, которым соответствуют на графе узлы q и p .

Недостатком этой стратегии является то, что в случае, когда на изображении присутствуют небольшие сегменты большой или маленькой яркости, минимальный разрез проходит по границам этих сегментов, вследствие малого веса ребер, соответствующим этим границам. В итоге может происходить выделение мелких сегментов, вместо обнаружения протяженных границ.

Второй стратегией выбора стока и истока было нахождение на изображении самой заметной границы (ребра с самым маленьким весом) и выбор стока и истока по разные стороны от этой границы. Этот способ приводит к выделению наиболее заметных и протяженных границ на начальном этапе разделения графа. Но в случае, если сток и исток были выбраны

около заметной, но короткой границы, не имеющей продолжения, найденный разрез не оптимален.

Способ выбора стока и истока задавался на входе алгоритма и не изменялся по ходу его выполнения. В дальнейшем планируется делать этот выбор адаптивным, в зависимости от характера участка изображения которому соответствует рассматриваемый подграф. Также планируется усовершенствовать второй способ, пытаясь продлить (т.е. составить одну сильную границу из нескольких коротких сильных границ сегментов) наиболее сильные границы сегментов и выбирать сток и исток по разные стороны от них. Возможно добавление других способов выбора стока и истока, например, путем выделения двух пиков на гистограмме.

6. ВЫБОР ВЕСОВОЙ ФУНКЦИИ РЕБЕР ГРАФА.

Вес ребер графа представляет собой пропускную способность ребра при решении задачи о максимальном потоке. Следовательно, чем меньше его вес, тем более вероятно, что минимальный разрез пройдет именно через это ребро.

Можно сформулировать критерии, которые будут определять, является ли граница между двумя сегментами достоверной. Например, различие яркостных характеристик сегментов, находящихся по сторонам границы. Таких критериев должно быть несколько, потому что могут существовать ситуации, когда один или несколько критериев не применимы и решение требуется принимать по каким-то другим критериям. Например, сегменты могут иметь близкую среднюю яркость, но иметь разные текстурные характеристики.

Исходя из этого, вес ребра должен определяться по совокупности критериев. В данной реализации алгоритма вес ребра рассчитывался исходя из двух критериев:

- различие средних яркостей сегментов;
- среднее значение R-величины вдоль границы, чем больше R-величина, тем достовернее граница.

Весовая функция ребра выглядела следующим образом:

$$w_{u,v} = l_{u,v} \left(\alpha e^{-\frac{|\bar{I}_u - \bar{I}_v|}{D_I}} + (1 - \alpha) e^{-\frac{\bar{R}_{u,v}}{D_R}} \right) \quad (4)$$

В этой формуле $l_{u,v}$ - длина границы между сегментами, которым соответствуют узлы u и v графа. $\bar{I}_u, \bar{I}_v$ - средние интенсивности этих сегментов, $\bar{R}_{u,v}$ - среднее значение R-величины вдоль границы сегментов. D_I, D_R, α - задаваемые коэффициенты. Коэффициент D_I можно оценить из дисперсии средних яркостей сегментов, путем домножения на некоторый коэффициент. D_R имеет смысл порога, определяющего, какие кривые имеют достаточный вес, чтобы повлиять на вес ребра.

Первое слагаемое в формуле отвечает за критерий о различии яркостей двух сегментов, второе слагаемое – за критерий о R-величине. По формуле видно, что чем лучше выполняются эти критерии, тем меньше итоговый вес, что соответ-

стствует меньшей пропускной способности в задаче о максимальном потоке. Чем меньше пропускная способность ребра, тем больше вероятность, что минимальный разрез пройдет именно через него.

R-величина строится следующим образом. Изображение пробегается рамкой, которая в каждой из позиций делится пополам под разными углами. По обеим половинкам считаются статистики – средняя яркость $B_{i\theta}$ и дисперсия $D_{i\theta}$. R-величина вычисляется как:

$$R = \begin{cases} \max_{\theta} (1/(1+Q_{\theta})), & \text{при } B_{1\theta} \neq B_{2\theta}, \\ 0, & \text{при } B_{1\theta} = B_{2\theta} \end{cases} \quad (5)$$

$$\text{где } Q_{\theta} = \frac{\max(\min(\sigma_{1\theta}, \sigma_{2\theta}), \delta)}{|B_{1\theta} - B_{2\theta}|}, \sigma_{i\theta} = \sqrt{D_{i\theta}}$$

Более подробно понятие R-величины изложено в [2].

7. УПОРЯДОЧИВАНИЕ ПОСЛЕДОВАТЕЛЬНОСТИ СЛИЯНИЙ

В процессе работы алгоритма изначальный граф разделяется иерархически. Сначала на два подграфа, затем получившиеся подграфы разделяются в свою очередь, до тех пор, пока все вершины не окажутся изолированными. Для минимальных разрезов, которыми разделяются графы, можно подчитать их вес.

Сопоставив все образовывавшиеся на всех уровнях иерархии подграфы узлам, можно построить взвешенное направленное дерево, корнем которого является изначальный граф. Вес ребра, входящего в узел, соответствует весу минимального разреза на том подграфе, который соответствует узлу.

Для получения конечного результата дерево разрезов нужно отобразить в линейную последовательность разрезов в таком порядке, чтобы в первую очередь выделялись наиболее заметные и длинные границы. Разрезы нельзя просто отсортировать по весу, так как важна последовательность – разрез не может быть применен, если подграф, который он разрезает, еще не сформирован. Чтобы соблюсти правильную последовательность, дерево обходится, начиная в корня, таким образом, что в каждый данный момент выбирается наилучший (в смысле веса) вариант разреза из возможных.

Ниже приведен алгоритм обхода дерева в виде псевдокода. В нем Q – приоритетная очередь, то есть очередь, в которой элементы сортируются по какому-то принципу. В данном случае Q содержит вершины дерева, отсортированные по возрастанию веса ребер в них входящих (т.е по весу разрезов подграфов, которым соответствуют эти вершины). r – корневая вершина дерева. L – последовательность, в которую заносятся вершины по мере обхода.

```
Q.push(r)
while( Q не пусто ) do
    v = Q.pop()
    L.push(v)
    for( вершины u в которую выходят дуги из v ) do
        Q.push(u)
```

Результатом работы алгоритма является последовательность L в которой находится последовательность вершин дерева, соответствующих подграфам. Чтобы получить слияние, при котором выделяются “наилучшие” в смысле веса разреза границы, нужно применить разрезы к подграфам в том порядке, в котором они представлены в последовательности L .

Однако, для преобразования дерева в последовательность можно использовать другой способ, построив эту последовательность во время работы алгоритма. Есть два условия, которые должны выполняться для последовательности:

1. Последовательность должна быть отсортирована по возрастанию весов разрезов.
2. Узел дерева не может находиться в последовательности раньше, чем узел, от которого к этому узлу приходит входящее ребро (т.е. узел, находящийся ближе к корню дерева).

Условие 2 сильнее, чем условие 1, то есть должно выполняться всегда. Условие 1 должно выполняться всегда, когда это не противоречит условию 2.

Теперь можно сформулировать способ построения последовательности по ходу выполнения алгоритма:

1. На первом этапе в последовательность заносится корневой узел дерева (весь граф).
2. По мере образования новых подграфов, каждый новый подграф заносится в последовательность, таким образом, чтобы он был после подграфа, из которого образовался (условие 1) и перед всеми подграфами, разрезы которых имеют больший вес, чем у данного подграфа (условие 2).

Таким образом, после завершения работы алгоритма, последовательность уже построена, поскольку на всем протяжении работы алгоритмы сохранялось выполнение необходимых условий.

8. ОЦЕНКА СЛОЖНОСТИ АЛГОРИТМА

Анализ алгоритма показывает, что основной цикл алгоритма (рис. 6, блоки 2-7) выполняется число раз равное количеству узлов в дереве слияния $(2N-1)$ сегментов минус число сегментов в изначальной сегментации. Если принять число сегментов в изначальной сегментации за N_i , то:

$$P = P_1 + N(P_2 + P_3 + P_4 + P_5 + P_6 + P_7) + P_8 \quad (6)$$

Вычислительная сложность этих блоков представлена на таблице.

N_g и M_g – число вершин и ребер в обрабатываемом подграфе.

Блок 6 выполняется либо за постоянное время (в случае преобразования дерева в последовательность в конце работы алгоритма), либо за время пропорциональное N (в случае когда последовательность формируется по ходу работы алгоритма).

Таким образом, в блоках алгоритма 2-7, доминирующим является блок 4 [11], который и определяет сложность этого участка алгоритма.

Номер блока	Сложность P_j
-------------	-----------------

1	$O(N)$
2	$O(1)$
3	$O(N_g + M_g)$
4	$O(N_g * M_g * \log(N_g * N_g / M_g))$
5	$O(M_g)$
6	$O(1)$ или $O(N)$
7	$O(1)$
8	$O(N)$

В худшем случае, когда на каждом этапе выделяется по одному неделимому сегменту, размер N_g изменяется от N до 1. Тогда вычислительная сложность алгоритма:

$$P = O\left(\sum_{n=N}^1 n M_g \log(n^2 / M_g)\right) \quad (7)$$

Исходя из структуры графа, можно сделать утверждение, что:

$$M_g \approx N_g \quad (8)$$

Это утверждение обосновывается тем, что каждый сегмент в сегментации в среднем имеет некоторое количество соседей, и с увеличением числа сегментов число границ растет линейно.

Тогда:

$$P = O\left(\sum_{n=1}^N n^2 \log(n)\right) = O(N^3 \log(N)) \quad (9)$$

В результате вычислительная сложность представленного алгоритма пропорциональна кубической степени числа сегментов в изначальной сегментации умноженной на логарифм числа сегментов.

Соответственно время работы алгоритма зависит от числа вершин в графе кубическим образом. Для обработки графа из 340 вершин (рис. 11) алгоритм тратит менее секунды, на граф из 2032 сегментов – 24 секунды.

9. РЕЗУЛЬТАТЫ

Разработанный алгоритм уменьшения детальности сегментации показал хорошие результаты как для аэрокосмических изображений, так и для изображений обычных сцен, снятых с камеры, установленной на мобильном роботе. На рис. 1,2,8 показаны исходное изображение, результат сегментации и результат уменьшения детальности для аэрокосмического изображения области Аральского моря, полученного путем совмещения трех спектральных каналов с прибора AVHRR спутника серии NOAA. Видно, что выделились водоемы, на изображении остались в основном границы, соответствующие наиболее заметным границам – береговым линиям. На рис. 10,11,12. – исходное изображение, результат сегментации и уменьшения детальности для изображения полученного с мобильного робота Pioneer-2AT. В обоих случаях использовался способ выбора стока и истока по разные стороны от наиболее сильной границы.

Для сравнения, на рис. 9 и 13 показаны результаты слияния сегментов с другим способом выбора стока и истока –

как сегментов наиболее различающихся по яркости. В целом способ выбора стока и истока по разные стороны от наиболее сильной границы показывает лучшие результаты. Как видно на рис. 9, при одинаковом числе сегментов с рис. 8 не выделились некоторые береговые линии, вместо них произошло отделение облаков. На рис. 12 плохо выделилась пачка чая, расположенная на самом верху.

В дальнейшем алгоритм планируется усовершенствовать путем добавления нового способа выбора стока и истока основанного на продлении границ с учетом характеристик формы линий. Также планируется учитывать более широкий набор характеристик сегментов при расчете веса ребер графа, сделать уточнение коэффициента, связывающий параметр D_i с дисперсией средних яркостей сегментов и сделать подбор этого параметра автоматическим, исходя из характеристик обрабатываемого изображения.

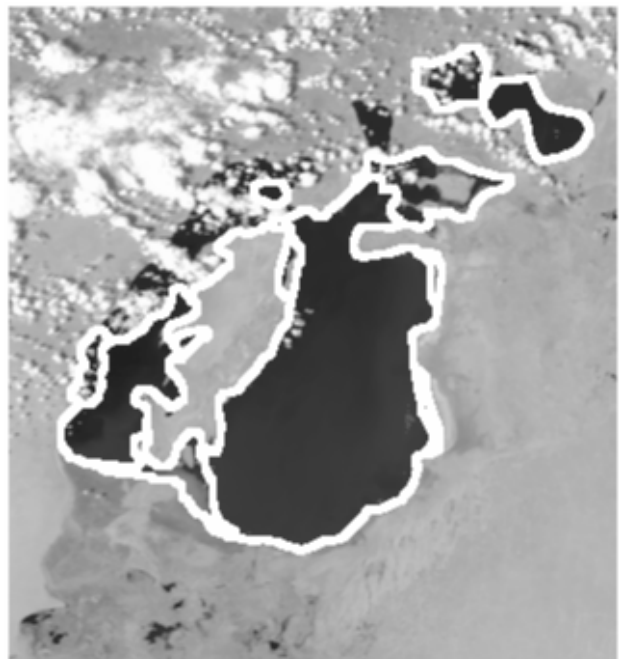


Рисунок 8. Область Аральского моря, результат уменьшения детальности (выбор стока и истока по разные стороны от наиболее сильной границы)

10. ЛИТЕРАТУРА

- [1] Непомнящий П.В., Юрин Д.В. Поиск опорных точек на векторных изображениях путем детектирования угловых структур с помощью статистической оценки гипотез. // В сб. Труды конференции. 12-я Международная Конференция по Компьютерной Графике и Машинному Зрению ГрафиКон'2002 –С. 442–448. Нижний Новгород, 16-21 сентября 2002 г. (Nepomnyaschy P.V., Yurin D.V. Reference Points Search on Vector Images by Corner Detection via Statistical Hypotheses Estimation. // In Conference Proceedings. 12-th International Conference on Computer Graphics GraphiCon'2002 –P. 442–448. Nizhny Novgorod, September 16-21, 2002).

[2] Минченков М.В., Юрин Д.В., Хельвас А.В. Алгоритм автоматической сегментации изображений на основе релеевого детектора границ двумерных объектов. //В сб. Труды конференции. 12-я Международная Конференция по Компьютерной Графике и Машинному Зрению ГрафиКон'2002 –С. 243–250. Нижний Новгород, 16-21 сентября 2002 г. (Mintchenkov M.V., Yurin D.V., Helvas A.V. Unsupervised Algorithm for Images Segmentation on the Base of the Rayleigh 2D Objects Boundaries Detector. // In Conference Proceedings. 12-th International Conference on Computer Graphics GraphiCon'2002 –P. 243–250. Nizhny Novgorod, September 16-21, 2002).

[3] Янова Н.В., Юрин Д.В. Итеративный алгоритм восстановления трехмерных сцен, движения и фокусного расстояния камеры в перспективной проекции, основанный на факторизации матриц. //В сб. Труды конференции. 12-я Международная Конференция по Компьютерной Графике и Машинному Зрению ГрафиКон'2002 –С. 123–129. Нижний Новгород, 16-21 сентября 2002 г. (Yanova N.V., Yurin D.V. Matrix Factorization Based Iterative Algorithm for 3D Shape, Motion and Camera Focal Length Recovery in Perspective Projection. // In Conference Proceedings. 12-th International Conference on Computer Graphics GraphiCon'2002 –P. 123–129. Nizhny Novgorod, September 16-21, 2002).

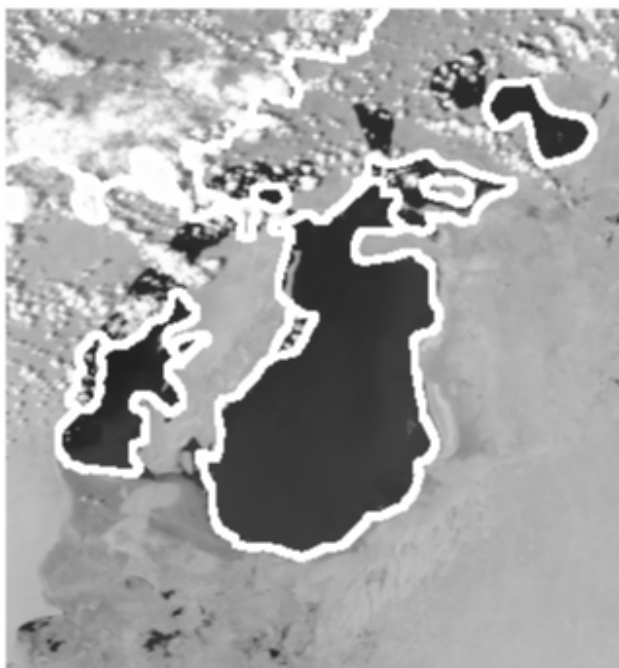


Рисунок 9. Область Аральского моря, результат уменьшения детальности (выбор стока и истока как двух сегментов наиболее отличающихся по яркости)



Рисунок 10. Снимок с мобильного робота Pioneer-2AT



Рисунок 11. Результат исходной сегментации снимка с мобильного робота Pioneer-2AT



Рисунок 12. Результат уменьшения детальности сегментации на снимке с мобильного робота Pioneer-2AT (выбор стока и истока по разные стороны от наиболее сильной границы)

[4] Cheng, V. Wu, R. Collins, A. Hanson, and E. Riseman. Maximum-Weight Bipartite Matching Technique and Its Application in Image Feature Matching, SPIE Conference on Visual Communication and Image Processing, 1996.



Рисунок 13. Результат уменьшения детальности сегментации на снимке с мобильного робота Pioneer-2AT (выбор стока и истока как двух сегментов наиболее отличающихся по яркости)

- [5] Прэтт У. Цифровая обработка изображений: Пер. с англ. – М.: Мир, 1982. – Кн.1 – 312 с., ил., Кн.2 – 480 с., ил.
- [6] Koepfler G., Lopez C., Morel J.M. A Multiscale Algorithm for Image Segmentation by Variational Method // SIAM Journal on Numerical Analysis. — 1994 — V.31 — N 1. — P. 282–299.
- [7] Т. Кормен, Ч. Лейзерсон, Р. Ривест. Алгоритмы, построение и анализ. М.: МЦНМО, 2001. – 960 с., 263 ил.
- [8] Huseyin Tek, Benjamin B. Kimia Volumetric Segmentation of Medical Images By Three-Dimensional Bubbles // CVIU. — 1997.
- [9] J. Canny. A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8, No. 6, Nov. 1986.
- [10] Olga Veksler. Image Segmentation by Nested Cuts. IEEE Computer Vision and Pattern Recognition, June 2000 (CVPR'00), p.339-344.
- [11] B. V. Cherkassky, A. V. Goldberg. On Implementing Push-Relabel Method for the Maximum Flow Problem. Technical Report STAN-CS-TR-94-1523, Department of Computer Science, Stanford University, Stanford, CA, September 1994.
- [12] A. V. Goldberg and S. Rao. Length Functions for Flow Computations, Technical Report 97-055, NEC Research Institute, Inc., March 1997.
- [13] A. V. Goldberg and R. E. Tarjan. A New Approach to the Maximum Flow Problem, Journal of ACM, Vol. 35, No. 4, pages 921-940, 1988.
- [14] Dmitry Chetverikov and Zsolt Szabo. A Simple and Efficient Algorithm for Detection of High Curvature Points in Planar Curves, Proc. 23rd Workshop of the Austrian Pattern Recognition Group, 1999, pp.175-184.
- [15] Bedekar A.S., Haralick R.M., Ramesh V. and Zhang X. A Bayesian Corner Detector: Theory and Performance Evaluation, ARPA94 (I:703-715), 1994.
- [16] Mei Han and Takeo Kanade, Scene Reconstruction from Multiple Uncalibrated Views, Technical Report CMU-RI-TR-00-09, Robotics Institute, Carnegie Mellon University, Jan, 2000.
- [17] Boost Graph Library. <http://www.boost.org/>

[18] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(11):1101--1113, 1993

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. In IEEE Conference on Computer Vision and Pattern Recognition, pages 731v737, 1997.

[20] W. Skarbek and A. Koschan. Colour image segmentation - a survey. Technical report, Technical University Berlin, 1994

Об авторах

Непомнящий Павел Владимирович, студент бго курса Московского Физико-Технического Института, факультет Физической и Квантовой Электроники, кафедра Центр Открытых Систем и Высоких Технологий.

E-mail: npv@cos.ru

Юрин Дмитрий Владимирович, кфмн.

E-mail: yurin@cos.ru