

СЕГМЕНТАЦИЯ ИЗОБРАЖЕНИЙ НА ОСНОВЕ АЛГОРИТМА ВЫДЕЛЕНИЯ СООБЩЕСТВ НА ГРАФЕ

С.В. Белим

д.ф.-м.н., профессор кафедры инф. безопасности, e-mail: belimsv@omsu.ru

С.Б. Ларионов

аспирант, e-mail: me@stas-larionov.ru

Омский государственный университет им. Ф.М. Достоевского

Аннотация. В статье предложен алгоритм сегментации изображений, использующий его представление в виде неориентированного взвешенного графа, с последующим выделением сообществ на графе. Каждому сообществу соответствует сегмент изображения. В качестве меры веса дуг графа используется функция от разности интенсивности цветов в пространстве RGB, возрастающая по экспоненциальному закону. Для выделения сообществ построен жадный алгоритм. Для оценки качества разбиения графа на сообщества используется функция модульности, введённая Ньюманом. Проведён компьютерный эксперимент. Показано, что размеры сегментов зависят от значения параметра в функции веса рёбер. Предложенный алгоритм устойчив к импульсным шумам.

Ключевые слова: сегментация изображений, сообщества, графы.

Введение

Под сегментацией принято понимать разбиение изображения на группы пикселей по близким значениям некоторого показателя [1]. Необходимость в сегментации изображения возникает в ряде прикладных задач, таких как обработка медицинских и биологических снимков [2–4], обработка аэрокосмических снимков [5–7], контроль качества поверхности материалов [8], системах технического зрения [9, 10] и др.

На сегодняшний день разработано достаточно много методов сегментации, среди которых можно выделить несколько базовых, на которых основываются большое количество прикладных алгоритмов: метод пороговой сегментации [11, 12], метод центроидного связывания [13], метод водораздела [14] и др.

Основная идея всех алгоритмов сегментации состоит в выделении связной области изображения, яркость пикселей которой различается не более чем на некоторую величину. Данное число принято называть порогом сегментации. Ряд работ направлен на выработку алгоритмов автоматического выбора порога сегментации [15]. Кроме этого, большое распространение получили алгоритмы, основанные на выделении контурных границ с помощью вычисления первой

производной (операторы Собеля и др. [16–19]) или второй производной (оператор Лапласа [16, 17, 20]). В этом случае осуществляется поиск границ, которые в дальнейшем объединяются в контуры. В качестве сегментов выбираются области внутри замкнутых контуров. Также существуют методы, основанные на выращивании областей [21, 22], в которых происходит группировка пикселей в области вокруг «центров кристаллизации». Достаточно много методов основываются на представлении пикселей изображения в виде точек многомерного пространства признаков с последующей кластеризацией [23–30].

Существует ряд методов сегментации, основанных на разрезании графа. Основная идея состоит в сопоставлении пикселям изображения вершин графа. Между всеми вершинами строятся ребра, веса которых показывают меру сходства между вершинами по какой-либо характеристике. Далее граф разрезается на подграфы таким образом, чтобы внутри подграфа веса рёбер были значительно больше весов рёбер, связывающих подграфы. В работе [31, 32] предложен метод нормализованных разрезов. Применение данного метода к задаче выявления дефектов поверхности исследовано в работе [8]. Одним из лучших графовых алгоритмов на сегодняшний день считается GraphCut [33–35]. В данном алгоритме множество вершин состоит из пикселей изображения и двух дополнительных искусственно добавленных вершин — исток и сток. Пользователь в интерактивном режиме задаёт «семена» объекта — несколько пикселей, принадлежащих объекту, и «семена» фона — несколько пикселей, принадлежащих фону. Семена объекта связываются с истоком рёбрами, имеющими бесконечный вес. Семена фона связываются со стоком рёбрами, имеющими бесконечный вес. Для полученного графа, дополненного вершинами истока и стока, ищется минимальный разрез [36], делящий граф на две части. Подграф, содержащий исток, соответствует объекту. Подграф, содержащий сток, соответствует фону. Данный алгоритм работает в интерактивном режиме и позволяет выделять отдельный элемент изображения. Другой подход на основе теории графов, позволяющий проводить полную сегментацию изображения, предложен в работах [37, 38]. Основная идея сегментации состоит в сопоставлении изображению графа с последующей его кластеризацией на основе построения минимального остовного дерева.

По своей постановке задача сегментации изображений близка к проблеме выделения сообществ (community) на графах. В обоих случаях необходимо выделить и сгруппировать элементы, близкие друг к другу по некоторому показателю. На сегодняшний день наблюдается всплеск интереса к проблеме поиска сообществ на графах [39], связанный, прежде всего, с развитием социальных сетей и других сообществ в глобальной сети Интернет. Однако предложенные алгоритмы могут найти применение и в других задачах близких по постановке, при соответствующем представлении данных.

Целью данной статьи является разработка и тестирование алгоритма сегментации, основанного на наиболее распространённом алгоритме поиска сообществ на графах, предложенного в работе [40].

1. Постановка задачи и представление данных

Задача кластеризации изображений состоит в разбиении множества точек изображения на подмножества, «близкие» по цвету. Основная проблема состоит в строгом определении степени близости точек. В большинстве прикладных задач необходимо выделить некоторые предметы на изображении, причём эти предметы человеческий глаз выделяет на общем фоне именно по цветовому признаку. То есть предмет идентифицируется как множество точек близких по цвету друг к другу, но отличающихся от окружающего фона. При этом цвет самого выделяемого предмета может плавно изменяться в широком диапазоне. В дальнейшем, при решении задачи кластеризации, ограничимся поиском только односвязных областей.

Пусть задано изображение, размеры которого составляют N пикселей по горизонтали и M пикселей по вертикали. Если цвет задан в градациях серого, то будем считать, что палитра содержит m цветов, то есть каждый пиксель характеризуется целым числом в интервале $[0, m - 1]$. Если изображение цветное, то будем использовать модель RGB. В этом случае каждому пикселю сопоставляется три целых числа в интервале $[0, m - 1]$. В дальнейшем будем рассматривать общий случай цветного изображения. Более простой случай монохромного изображения может быть получен из него простым обнулением двух из трёх цветовых координат.

Введём пятимерное пространство XYRGB. Построим в этом пространстве множество точек, соответствующих пикселям изображения. Первые две координаты будут определяться геометрическим положением пикселя на исходном изображении, оставшиеся три — его цветовыми характеристиками. В некоторых случаях имеет смысл провести нормировку координат так, чтобы по всем направлениям множество точек было ограничено величинами, сопоставимыми по порядку. Однако в рамках подхода, предлагаемого в данной статье, такое преобразование не является обязательным, так как в формуле для расстояния между точками геометрические и цветовые координаты входят не равноправно. Как будет видно далее, нормировка цветовых координат приводит лишь к переопределению значения параметра, определяемого пользователем. Будем считать, что пикселю, расположенному на изображении в точке с координатами (x, y) , в пространстве XYRGB соответствует точка с координатами $(x, y, r(x), g(x), b(x))$, где $r(x)$ — интенсивность красной составляющей цвета пикселя, $g(x)$ — интенсивность зелёной составляющей цвета пикселя, $b(x)$ — интенсивность синей составляющей цвета пикселя.

Значительное влияние на результат работы алгоритма кластеризации оказывает способ вычисления расстояния между точками в пространстве XYRGB. Величина расстояния между двумя точками должна быть чувствительна к изменению цвета, поэтому необходимо выбирать функцию, быстро растущую при переходе к соседнему пикселю с резко отличающимся цветом. Для координат, определяемых геометрическим положением пикселя на изображении, наоборот выберем зависимость, убывающую с расстоянием, для того, чтобы сгладить накапливание значения расстояния в рамках одного кластера при градиент-

ной заливке. По аналогии с работой [40] будем вычислять расстояние между точками $v_1=(x_1, y_1, r_1, g_1, b_1)$ и $v_2=(x_2, y_2, r_2, g_2, b_2)$ по формуле:

$$d(v_1, v_2) = f(x_1 - x_2, y_1 - y_2) \cdot \exp\left(\frac{1}{\delta} \sqrt{(r_1 - r_2)^2 + (g_1 - g_2)^2 + (b_1 - b_2)^2}\right).$$

Здесь функция $f(x, y)$ — зависит от геометрической удалённости двух точек друг от друга. Её выбор зависит от графа, сопоставляемого изображению на следующем шаге алгоритма. Параметр δ является выбираемым и влияет на скорость роста функции расстояния между двумя точками. Данный параметр является настраиваемым и, выбирая его, можно влиять на то, какие области признаются равномерно закрасненными, а какие нет.

Построим неориентированный взвешенный граф, множество вершин которого совпадает с построенным множеством точек. Соединим рёбрами только вершины, соответствующие соседним пикселям на исходном изображении. Причём соседями будем считать не только пиксели, граничащие с данным по горизонтали и вертикали, но и по диагонали. Таким образом, каждый пиксель, расположенный не на границе изображения, имеет восемь соседей, с которыми он соединён рёбрами. Четыре точки по углам изображения имеют всего по три соседа. Точки, расположенные на границе изображения, имеют по пять соседей. Из построения графа видно, что выбор функции будет слабо влиять на результат, так как мы рассматриваем только соседние пиксели, расстояние между которыми фиксировано. В дальнейшем будем считать $f(x, y) = 1$, что равносильно работе в трёхмерном пространстве RGB.

Задача кластеризации изображения сводится к разбиению построенного графа на подграфы, обладающие определёнными свойствами. Неформально данное свойство подграфа определяется как отношение к одному кластеру на изображении. Одна из формализаций искомого свойства может быть построена на основе поиска сообществ в графе.

Таким образом, кластеризация изображения может быть осуществлена в три этапа. На первом этапе производится представление множества пикселей изображения в виде сначала множества точек пятимерного пространства, а потом взвешенного графа. На втором этапе выполняется разбиение графа на подграфы. На третьем этапе происходит восстановление кластеров изображения, соответствующих подграфам.

2. Алгоритм кластеризации

Как было сказано ранее, кластеризация изображения в нашем представлении свелась к разбиению графа на подграфы. Причём выделение областей одного цвета равносильно выделению подграфов, для которых ребра между собственными вершинами имеют больший вес, чем ребра связывающие с внешними вершинами. Такие подграфы принято называть сообществами (community).

Как было указано во введении, на сегодняшний день разработано достаточно много алгоритмов выделения сообществ в графах. Для кластеризации изображений используем один из них, предложенный Ньюманом [40].

Будем задавать построенный граф с помощью матрицы весов E . Элементы матрицы весов задаются следующим образом. Диагональный элемент E_{ii} показывает вес вершины с номером i (v_i). Элемент E_{ij} ($i \neq j$) показывает величину связи вершины v_i с вершиной v_j . Как показано в работе [40], более удобным является приведённый вид матрицы весов $e = E/m$, где $m = \sum E_{ij}$. В приведённой матрице весов элемент e_{ij} показывает долю веса заданного ребра в общем весе графа. В дальнейшем под матрицей весов будет пониматься именно приведённый вид. Легко увидеть, что $\sum e_{ij} = 1$.

Для выявления связанных структур введём величину модульности (modularity), зависящую от элементов матрицы смешения, и численно определяющую «силу» связности. Модульность задаётся в виде [40]:

$$Q(e) = \sum_{i=1}^{MN} (e_{ii} - p_{ii}),$$

где p_{ii} — «ожидаемая связность». В канонической модели [40] p_{ii} определяется через исходящую степень вершины a_i , и входящую степень вершины b_i :

$$p_{ii} = a_i b_i.$$

В этом случае модульность записывается в виде:

$$Q(e) = \sum_{i=1}^{MN} (e_{ii} - a_i b_i),$$

где

$$a_i = \sum_{j=1, j \neq i}^{MN} e_{ij}, \quad b_i = \sum_{j=1, j \neq i}^{MN} e_{ji}.$$

Определим более строго процедуру образования сообщества в графе. Начнём с алгоритма образования стяжек как преобразования графа G в граф G_1 . Выделим в графе G подграф G' и заменим все входящие в него вершины одной вершиной, при этом вершины подграфа $G \setminus G'$ остаются неизменными. Образованная вершина связана дугами с теми вершинами графа G_1 , с которыми были связаны вершины, вошедшие в стяжку. Вес вершины, вошедшей в стяжку, равен сумме весов вершин и дуг, вошедших в стяжку. Под сообществом будем понимать подграф исходного графа, который при образовании из него стяжки максимизирует модульность графа $Q(e)$. Нашей задачей является такое разбиение исходного графа изображения на сообщества, которое максимизирует $Q(e)$. Каждое такое сообщество представляет собой кластер исходного изображения.

При образовании стяжки изменяется граф и, как следствие, меняется значение функции $Q(e)$. Пусть произошло объединение вершин v_k и v_l в одну вершину v'_k , в результате чего граф G преобразовался в граф G' . Изменение функции модульности будет иметь вид:

$$\Delta Q(e) = Q(e') - Q(e) = 2(e_{kl} - a_k a_l).$$

Таким образом, изменение функции модульности графа может быть легко найдено из параметров графа G .

Разбиение графа на сообщества будем производить с помощью «жадного» алгоритма. Выберем одну из вершин графа v_0 , которая будет служить зерном сообщества C_0 . Пусть на некотором этапе построена часть сообщества C_k , содержащая вершины $\{v_{i1}, \dots, v_{ik}\}$. Рассмотрим точки, являющиеся ближайшими соседями каждой из вершин, входящих в C_k и не включённые в сообщество. Для каждого из ближайших соседей сообщества рассчитаем величину ΔQ_{kl} , показывающую изменение функции модульности при присоединении к C_k вершины v_l . Присоединим к сообществу ту вершину, для которой ΔQ_{kl} максимально, и при этом $\Delta Q_{kl} > 0$. Полученный граф обозначим C_{k+1} , модульность графа увеличим на ΔQ_{kl} и повторим процедуру. Продолжаем алгоритм до тех пор, пока будут находиться вершины, присоединение которых увеличивает модульность графа. Если в некоторый момент такие вершины отсутствуют, то сообщество считается сформированным и выбирается новое зерно, не входящее ни в одно сообщество, для формирования нового сообщества. Алгоритм продолжается до тех пор, пока все вершины не будут включены в какое-нибудь из сообществ. Трудоемкость данного алгоритма $O(M^2N^2)$.

Выбор зерна сообщества зависит от режима работы алгоритма кластеризации изображения. При автоматическом разбиении изображения на кластеры зерно нового сообщества может выбираться случайным образом. При интерактивном режиме работы зерно указывает пользователь.

3. Компьютерный эксперимент

Эксперимент проводился как на искусственных изображениях, так и на фотографиях различных объектов. Рассмотрим результаты работы алгоритма на искусственных изображениях с чёткими контурами и равномерной заливкой (рис.1). Как хорошо видно из рисунка, алгоритм абсолютно точно разделя-

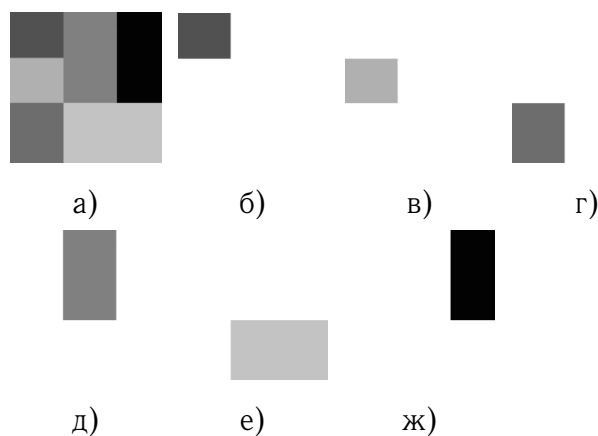


Рис. 1. Результат работы алгоритма по разбиению искусственного изображения с равномерной заливкой на связные области одного цвета: а) исходное изображение, б-ж) выделенные области.

ет изображение на области, причём результат не зависит от выбора значения параметра δ . Рассмотрим результаты работы алгоритма на искусственном изображении с градиентной заливкой (рис. 2). При уменьшении параметра

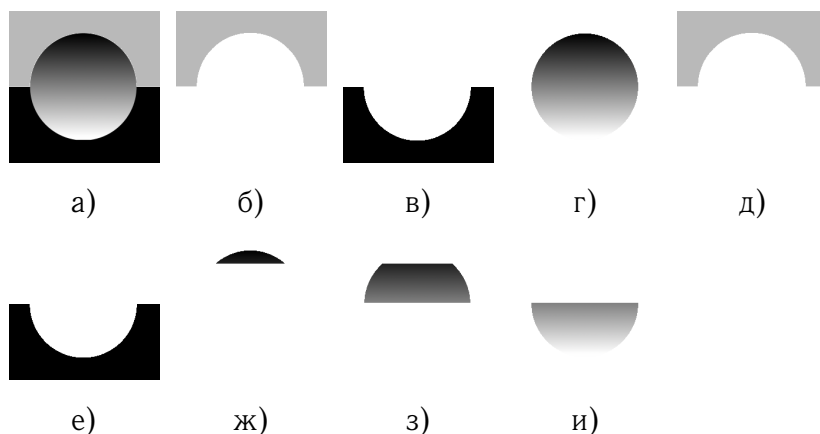


Рис. 2. Результат работы алгоритма по разбиению искусственного изображения с градиентной заливкой на связанные области одного цвета (размеры изображения 256x256): а) исходное изображение, б-г) выделенные области при $\delta = 0.04$, д-и) выделенные области при $\delta = 0.03$.

δ увеличивается количество кластеров, на которые разбивается изображение. Выделим отдельный кластер, например рисунок 3г, и проследим его разбиение при уменьшении параметра δ от значения 0.05 до значения 0.04. Результат представлен на рисунке 4.

Как видно из рисунка 2, при $\delta = 0.04$ внутренний круг с градиентной заливкой определяется как одна область, а при $\delta = 0.03$ он разбивается на четыре части. При уменьшении δ круг с градиентной заливкой разбивается на всё большее количество частей. Следует отметить, что при $\delta > 0.1$ всё изображение воспринимается как один кластер. Таким образом, выбор параметра δ влияет на значение скорости изменения цвета, которое определяется как граница между областями.

На рисунке 3 приведены наиболее крупные кластеры, получаемые на фотографическом изображении «Перцы» при $\delta = 0.05$, общее количество выделяемых кластеров равно 174.

Как видно из сравнения рисунков 3 и 4 для фотографических изображений, как и для искусственных изображений, варьирование параметра δ позволяет выбирать величину перепада цветов, которая считается границей области. На рисунке 5 приведён график зависимости количества кластеров, на которые разбивается изображение «Перцы», в зависимости от параметра δ .

Как видно из рисунка 5, количество кластеров достаточно быстро убывает с ростом значения δ , что свидетельствует о высокой чувствительности предложенного метода к выбору данного параметра. Начиная со значения $\delta = 0.1$, количество кластеров становится равным 1, то есть всё изображение воспринимается как один кластер.

Проведём исследование предложенного алгоритма на устойчивость к наличию случайно повреждённых пикселей. Компьютерный эксперимент состоял в

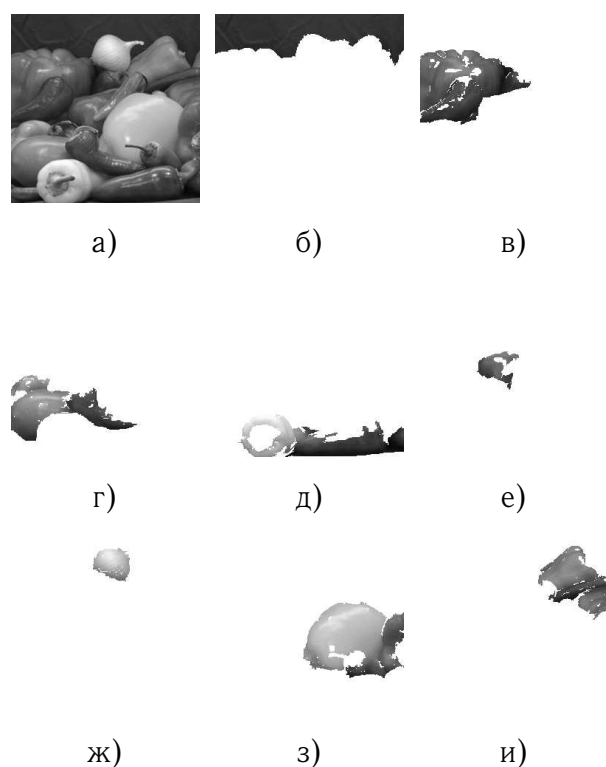


Рис. 3. Результат работы алгоритма по разбиению фотографического изображения «Перцы» при $\delta = 0.05$: а) исходное изображение, б-и) наиболее крупные выделенные области.

том, что p процентов случайно выбранных пикселей изображения заменялись на пиксели со случайным цветом. После чего осуществлялась кластеризация. При кластеризации не учитывались кластеры размером в один пиксель. Результаты эксперимента для изображения «Перцы» при $p = 10\%$ и $\delta = 0.5$ приведены на рисунке 6.

Как видно на рисунке 6, наличие повреждённых пикселей не влияет на выделение крупных кластеров, так подавляющее большинство повреждённых пикселей идентифицируется как кластеры размером один пиксель.

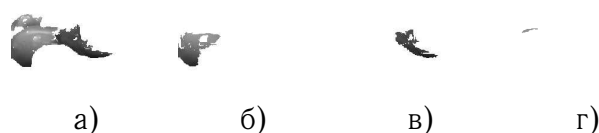


Рис. 4. Разбиение кластера на более мелкие кластеры при переходе от $\delta = 0.05$ к $\delta = 0.04$. а) один из кластеров, выделенных на изображении «Перцы» при $\delta = 0.05$, б-г) наиболее крупные кластеры, на которые разбивается исходный кластер.

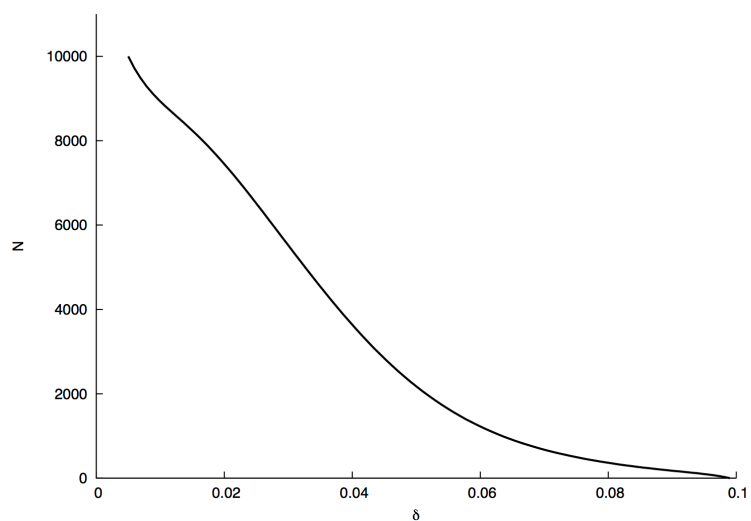


Рис. 5. График зависимости количества кластеров N , на которые разбивается изображение «Перцы», от параметра δ .

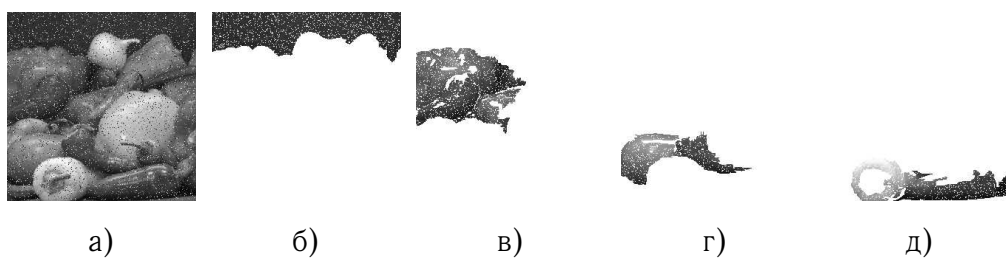


Рис. 6. Результат работы алгоритма по разбиению фотографического изображения «Перцы» повреждённого на 10% при $\delta = 0.05$: а) зашумлённое изображение, б-д) наиболее крупные кластеры.

4. Обсуждение результатов и выводы

Таким образом, предложенный подход к сегментации изображений, состоящий в представлении изображения в виде графа с последующим поиском сообществ (community), даёт достаточно хорошие результаты как на искусственно сформированных графических объектах, так и на фотографиях. Следует отметить, что разработанный алгоритм тестировался нами не только на монохромных изображениях, представленных в статье, но и на цветных рисунках и фотографиях и продемонстрировал хорошие результаты.

В сравнении с другими методами сегментации следует отметить, что предложенный подход использует разность интенсивностей цветов соседних пикселей, характерную для дифференциальных методов сегментации. Однако получаемые кластеры имеют чёткие границы, которые свойственны методам, основанным на абсолютном значении интенсивности, например, методе водораздела. Дополнительной положительной характеристикой предложенного алгоритма является устойчивость к импульсным шумам, которая состоит в малом влиянии испорченных пикселей на внешние границы крупных кластеров.

ЛИТЕРАТУРА

1. Шапиро Л., Стокман Дж. Компьютерное зрение. М. : Бином, 2006. 752 с.
2. Tek H., Kimia B.B. Volumetric Segmentation of Medical Images By Three-Dimensional Bubbles // CVIU – 1997. V. 64(2). P. 246–258.
3. Dogdas B., Shattuck D.W., Leahy R.M. Segmentation of Skull in 3D Human MR Images Using Mathematical Morphology // Proceedings of SPIE Medical Imaging Conference. 2002. V. 4684(180). P. 1553–1562.
4. D.L. Pham, C. Xu and J.L. Prince. Current Methods in Medical Image Segmentation // Annual Review of Biomedical Engineering. 2000. V. 2. P. 315–337.
5. Koepfler G., Lopez C., Morel J.M. A Multiscale Algorithm for Image Segmentation by Variational Method // SIAM J. on Numerical Analysis. 1994. V. 31, No. 1. P. 282–299.
6. Newsam S., Bhagavathy S., Fonseca L., Kenney C., Manjunath B. S. Object based representations of spatial images // Acta Astronautica. 2001. V. 48, No. 5–12. P. 567–577.
7. Siebert A. Segmentation based Image Retrieval // Proc. SPIE 3312, Storage and Retrieval for Image and Video Databases VI. 1997. V. 14. doi:10.1117/12.298444.
8. Цапаев А.П., Кретинин О.В. Методы сегментации изображений в задачах обнаружения дефектов поверхности // Компьютерная оптика. 2012. Т. 36, № 3. С. 448–452.
9. Мошкин В.И., Титов В.С., Якушенков Ю.Г. Техническое зрение роботов / Под общ. ред. Ю.Г. Якушенкова. М. : Машиностроение, 1990. 272 с.
10. Bruce J., Balch T., Veloso M. Fast and in-expensive color image segmentation for interactive robots // Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00). 2000. V. 3. P. 2061–2066.
11. Павлидис Т. Алгоритмы машинной графики и обработки изображения. М. : Радио и связь, 1986. 400 с.

12. Прэтт У. Цифровая обработка изображений. Кн. 1. М. : Мир, 1982. 312 с.
13. Анисимов Б.В., Курганов В.Д., Злобин В.К. Распознавание и цифровая обработка изображений. М. : Высшая школа, 1983. 295 с.
14. Vincent L., Soille P. Watersheds in Digital Spaces: An Efficient Algorithm Based on Immersion Simulations // IEEE Trans. on Pattern Analysis and Machine Intelligence. 1991. V. 13, No. 6. P. 583–598.
15. Самойленко Д. Е. Структурная сегментация изображений // Искусственный интеллект. 2004. № 4. С. 512–529.
16. Гонсалес Р., Вудс Р. Цифровая обработка изображений. М. : Техносфера, 2005. 1072 с.
17. Rosenfeld A., Kak A.C. Digital Picture Processing. New York : Academic press, 1976.
18. Робертс Л. Автоматическое восприятие трёхмерных объектов // В кн.: Интегральные роботы. Т. 1. М. : Мир, 1973. С. 162–208.
19. Prewitt J.M.S. Object Enhancement and Extraction // in Picture Processing and Psychopictories. New York : Academic Press, 1970. P. 75–150.
20. Clark J. J. Authenticating Edges Produced by Zero-Crossing Algorithms // IEEE Trans. Pattern Analysis Machine Intelligence. 1989. V. 12, No. 8. P. 830–831.
21. Fu K.S., Mui J.K. A Survey of Image Segmentation // Pattern Recognition. 1981. V. 13, No. 1. P. 3–16.
22. Haralick R.M., Shapiro L.G. Image Segmentation Techniques // Computer Vision, Graphics, Image Processing. 1985. V. 29, No. 2. P. 100–132.
23. Дуда Р., Харт П. Распознавание образов и анализ сцен. М. : Мир, 1976. 511 с.
24. Jain A.K., Dubes R.C. Algorithms for Clustering Data. Prentice Hall, 1988.
25. Matas J. Kittler J. Spatial and Feature Space Clustering: Applications in Image Analysis // 6th Int. Conf. on Computer Analysis and Patterns. Prague, Czech Republic, 1995.
26. Яне Б. Цифровая обработка информации. М. : Техносфера, 2007.
27. Ohta Y., Kanade T., Sakai T. Color Information for Region Segmentation // Computer Graphics, Image Processing. 1980. V. 13. P. 224–241.
28. Pal N.K., Pal S.K. A Review on Image Segmentation Techniques // Pattern Recognition. 1993. V. 26, No. 9. P. 1277–1293.
29. Haralick R.M. Image Texture Survey // Fundamentals in Computer Vision. Cambridge. 1983. P. 145–172.
30. Миронов Б.М., Малов А.Н. Сегментация изображений кластерным методом и алгоритмом случайных скачков: сравнительный анализ // Компьютерная оптика. 2010. Т. 34, № 1. С. 132–137.
31. Форсайт Д.А. Компьютерное зрение. Современных подход. М. : Издательский дом «Вильямс», 2004.
32. Jitendra M., Belongie S., Leung T., Shi J. Contour and Texture Analysis for Image Segmentation // International Journal of Computer Vision. 2001. V. 43, No. 1. P. 7–27.
33. Boykov Y., Jolly M.-P. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images // Proc. of the International Conference on Computer Vision. 2001. V. 1. P. 105–112.
34. Boykov Y., Jolly M.-P. Interactive organ segmentation using graph cuts // Medical Image Computing and Computer-Assisted Intervention. 2000. P. 276–286
35. Felzenszwalb P.F., Huttenlocher D.P. Efficient graph-based image segmentation //

- International Journal of Computer Vision. 2004. V. 59(2). P. 167–181.
36. Boykov Y., Kolmogorov V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004. P. 1124–1137.
 37. Белим С.В., Кутлуниин П.Е. Использование алгоритма кластеризации для разбиения изображения на односвязные области. // Наука и образование: электронное научно-техническое издание. 2015, N. 3, URL: <http://technomag.bmstu.ru/doc/759275.html>, DOI:10.7463/0315.0759275 (Дата обращения 13.06.2016).
 38. Белим С.В., Кутлуниин П.Е. Выделение контуров на изображениях с помощью алгоритма кластеризации // Компьютерная оптика. 2015, Т. 39, № 1, С. 119–124.
 39. Fortunato S. Community detection in graphs // Physics Reports 486. 2010. P. 75–174
 40. Newman M.E.J. Mixing patterns in networks // Phys. Rev. E. 2003. V. 67. P. 026126.

SEGMENTATION OF IMAGES ON THE BASIS OF ALGORITHM OF COMMUNITIES SEPARATION ON THE GRAPH

S.V. Belim

Dr.Sc. (Phys.-Math.), Professor, e-mail: belimsv@omsu.ru

S.B. Larionov

Postgraduate Student, e-mail: me@stas-larionov.ru

Dostoevsky Omsk State University

Abstract. In the article we propose the image segmentation algorithm using its representation as an undirected weighted graph, followed by separation of the communities on the graph. Each community corresponds to the image segment. The function of the difference between the intensity of the colors in the space of RGB increasing exponentially is used as the measure of arcs weight. To separate the communities a greedy algorithm is built. The modularity function introduced by Newman is used to assess the quality of the graph separation into communities. Computer experiment is provided. It is shown that segments sizes depend on the parameter value in the edges weight function. The proposed algorithm is robust to impulse noise.

Keywords: segmentation of image, community, graphs.