

Алгоритм сегментации изображений на основе поиска сообществ на графах

С.В. Белим
sbelim@mail.ru

С.Б. Ларионов
me@stas-larionov.ru

Омский государственный университет им. Ф.М. Достоевского, Омск, Россия

Аннотация

В статье предложен новый алгоритм сегментации изображений. Изображению сопоставляется взвешенный граф. Вершинами графа являются пиксели. Ребрами соединены только ближайшие соседи. Вес ребер зависит от разности цветов пикселей. Построенный граф разбивается на сообщества. Каждому сообществу сопоставляется сегмент изображения.

Введение

Подход к решению задачи сегментации изображения с помощью сопоставления ему взвешенного графа активно развивается в последние годы [1, 2, 3, 4]. Проблема разбиения изображения на непересекающиеся области точек, близких друг к другу по цветовым характеристикам, заменяется на эквивалентную ей задачу разбиения графа на подграфы. Для численной оценки качества разбиения графа Ньюманом [5] была введена функция модульности графа. В настоящее время данная функция активно используется для выделения сообществ на графах. Сообщества представляют собой набор вершин графа наиболее тесно связанных друг с другом. Сегментация изображения может быть сведена к поиску сообществ на графах.

Основной проблемой, с которой приходится сталкиваться при применении алгоритмов выделения сообществ к сегментации изображений состоит в высоких требованиях к ресурсам. Использование алгоритмов Fast Greedy [7] и Label Propagation [8] в задаче сегментации изображений было предложено в статье [6], однако не нашло широкого применения из-за высокой трудоемкости. Для снижения трудоемкости было предложено использовать суперпиксели. Под суперпикселями [9] понимается некоторое объединение соседних реальных пикселей изображения. В дальнейшем изображению сопоставляется граф суперпикселей, на котором и осуществляется поиск сообществ. Данный подход используется на этапе предобработки изображений и зависит от алгоритма формирования суперпикселей [9, 10, 11]. Например, в статье [12] предложен алгоритм формирования квазиуниверсальных суперпикселей, предъявляющий невысокие требования к вычислительным ресурсам. Однако ненулевые размеры суперпикселей оказывают существенное влияние на результаты сегментации и не позволяют получать точные решения.

Алгоритм сегментации изображения с применением трех различных алгоритмов поиска сообществ на графах предложен в работе [13]. Построение ребер графа, соответствующего изображению, основано на выборе порогового значения для разности цветовых координат пикселей, что делает его не чувствительным к малым изменениям цвета. Компьютерный эксперимент показал, что такой подход дает хорошие результаты только для монохромных изображений. В статье [14] поиск сообществ на графах используется для проверки и уточнения результатов сегментации, полученных другими методами. Аналогичный

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: Sergey V. Belim, Nadezda F. Bogachenko (eds.): Proceedings of the Workshop on Data, Modeling and Security 2018 (DMS-2018), Омск, Russia, October 2018, published at <http://ceur-ws.org>

подход используется в работе [15] для выделения отдельных предметов на изображении. В ряде работ поиск сообществ на графах используется на этапе постобработки сверхсегментированных изображений для объединения мелких сегментов [16, 17, 18]

Таким образом, на сегодняшний день алгоритмы выявления сообществ в задачах сегментации изображений используются в основном как дополнение к другим методам. Целью данной статьи является разработка алгоритма сегментации изображений с использованием поиска сообществ на графах без привлечения других алгоритмов.

1 Постановка задачи и алгоритм решения

В качестве входного данного будем рассматривать изображение размером $N \times M$ пикселей. Положение каждого пикселя определяется парой координат (x, y) , где x – целое число из отрезка $[0, N - 1]$, y – целое число из отрезка $[0, M - 1]$. Цвет пикселя с координатами (x, y) задается с помощью трех цветовых составляющих: $r(x, y)$ – интенсивность красного цвета, $g(x, y)$ – интенсивность зеленого цвета, $b(x, y)$ – интенсивность синего цвета.

Сопоставим изображению неориентированный взвешенный граф G . Каждому пикселю изображения сопоставим свою вершину графа. Ребрами соединим только вершины графа, соответствующие пикселям, которые являются ближайшими соседями на изображении. Вес ребра, соединяющего вершины $v_i = (x_i, y_i)$ и $v_j = (x_j, y_j)$, будем вычислять по формуле:

$$d(v_i, v_j) = \exp\left(-\frac{1}{h} \sqrt{(r_i - r_j)^2 + (g_i - g_j)^2 + (b_i - b_j)^2}\right).$$

$$r_i = r(v_i, v_j), \quad g_i = g(v_i, v_j), \quad b_i = b(v_i, v_j).$$

Параметр h , определяемый пользователем, служит задания порога, определяющая переход от одного сегмента к другому.

Будем выделять сегменты методом выращивания областей, последовательно присоединяя пиксели к уже сформированной части сегмента. Используем для этого графовое представление изображения. Задача выделения сегментов сводится к разбиению графа на подграфы. Причем вершины, относящиеся к одному подграфу, близки друг другу по цветовым характеристикам. В качестве меры близости можно использовать «цветовое расстояние» между пикселями $d(v_i, v_j)$. Поиск подмножества «близких» вершин может быть осуществлен с помощью методов выделения сообществ на графах.

Пусть изображение представлено с помощью графа G . Для численной оценки правильности разбиения графа на сообщества может быть использована функция модульности Ньюмана [5, 7].

Аргументом функции модульности разбиения графа G служит приведенная матрица весов e . Для ее определения введем матрицу исходных весов E . Диагональные элементы матрицы весов E_{ii} показывают вес i -ой вершин. До разбиения на подграфы все вершины имеют нулевой вес. Недиagonalные элементы E_{ij} ($i \neq j$) равны цветовому расстоянию между вершинами v_i и v_j . Граф неориентированный, поэтому матрица E будет симметричной относительно главной диагонали. Приведенную матрицу весов определим как

$$e = E/m,$$

где

$$m = \sum_{i,j=1}^{MN} E_{ij}.$$

Элемент приведенной матрицы весов e_{ij} равен доле веса заданного ребра в общем весе графа. Для приведенной матрицы весов выполняется очевидное равенство:

$$\sum_{i,j=1}^{MN} e_{ij} = 1.$$

Функция модульности графа вычисляется по следующей формуле [5, 7]:

$$Q(G) = \sum_{i=1}^K e_{ij} - \sum_{i=1}^K a_i b_i.$$

где K – общее количество вершин графа, a_i – приведенная исходящая степень вершины v_i :

$$a_i = \sum_{j=1, j \neq i}^K e_{ij},$$

b_i – приведенная входящая степень вершины v_i :

$$b_i = \sum_{j=1, j \neq i}^K e_{ji}.$$

В связи с тем, что граф G неориентированный, входящие и исходящие степени вершин будут равны между собой

$$a_i = b_i, \quad i = 1, \dots, K.$$

Функция модульности в этом случае примет более простой вид:

$$Q(G) = \sum_{i=1}^K e_{ij} - \sum_{i=1}^K a_i^2.$$

Для выявления сообществ на графе будем использовать процедуру образования стяжек. Под стяжкой понимается преобразование, при котором некоторый подграф H графа G заменяется одной вершиной v_H . Если какая-то из вершин подграфа H была связана ребром с вершиной v из подграфа $G \setminus H$, то вершина v_H также будет связана ребром того же веса с вершиной v . Вес новой вершины будет равен сумме весов ребер и вершин, входящих в подграф H . Новый граф обозначим G_H . Подграф H будем считать сообществом, если $Q(G_H) > Q(G)$. Следует отметить, что при образовании стяжки уменьшается число вершин графа K .

Поиска сообществ с помощью метода выращивания областей может быть записан в виде следующей последовательности шагов:

1. Выбираем произвольный пиксель изображения v , не входящий ни в одно сообщество.
2. Для выбранного пикселя v поочередно рассматриваем ближайшие соседние пиксели $v^{(i)}$ ($i = 1, \dots, 8$) и пытаемся объединить их в сообщество. При каждом объединении вычисляем изменение функции модульности ΔQ_i ($i = 1, \dots, 8$).
3. Если объединение данной вершины в сообщество с другими пикселями выгодно ($\Delta Q_i < 0$, $i = 1, \dots, 8$), то соответствующий пиксель присоединяется к сообществу.
4. Если присоединение всех соседних пикселей не выгодно, то считаем сообщество сформированным и переходим к шагу 1.

На выходе алгоритма получим разбиение графа на сообщество, каждое из которых соответствует одному сегменту изображения.

Изменение функции модульности может быть вычислено достаточно быстро на основе текущих характеристик графа. При объединении вершин v_i и v_j изменение модульности будет равно:

$$\Delta Q = 2(e_{ij} - a_i a_j).$$

Очевидно, что предложенный алгоритм имеет квадратичную трудоемкость от числа пикселей.

2 Компьютерный эксперимент

Для определения эффективности работы предложенного алгоритма протестируем его результаты на основе библиотеки изображений для сегментации Беркли (Berkeley Segmentation Data Set and Benchmarks 500, BSDS500) [19]. Данная коллекция на текущий момент представляет собой набор изображений, для которых проведена ручная сегментация группой экспертов. Численность группы составляет от 5 до 8 экспертов. В ручном режиме проведена полная сегментация изображений. Следует отметить, что результаты ручной сегментации, проведенные различными экспертами, отличаются друг от друга, как по количеству сегментов, так и по очертаниям самих сегментов. Как хорошо известно, задача сегментации не имеет точного решения.

Данная коллекция не может быть использована для тестирования предложенного алгоритма целиком, так как для большого количества изображений экспертами выделены сегменты, не удовлетворяющие требованиям, установленным при построении алгоритма. В формулировке решаемой задачи под сегментом понималась область изображения, точки которой имеют близкие цвета. В коллекции BSDS500 достаточно много результатов ручной сегментации, не удовлетворяющих этому требованию. Очевидно, что предложенный алгоритм не позволит получить все результаты, выделенные экспертами вручную. Следует отметить, что тестирование других, широко распространенных, алгоритмов сегментации также не позволяет получать результаты, полностью согласующиеся с предложенными экспертами.

В связи с этим коллекция была проанализирована и в ней выделена подколлекция из 97 изображений, в которых выделенные экспертами фрагменты удовлетворяют нашему определению сегмента. Исходная коллекция содержит 200 изображений. Кроме этого в отобранной подколлекции были отброшены результаты сегментации отдельных экспертов, которые сильно отличаются от остальных результатов на данном изображении. Следует отметить, что даже среди отобранных изображений результаты экспертов могут существенно отличаться.

Сравнение работы алгоритма с результатами ручной сегментации осуществлялось на основе одного центрального сегмента, присутствующего у всех экспертов. Определялись три величины:

N_i^h – количество точек в сегменте, выделенном i -ым экспертом в ручном режиме ($i = 1, \dots, b$, b – количество экспертов),

N^a – количество точек в аналогичном сегменте, выделенном алгоритмом,

N_i^{ah} – количество точек, присутствующих, как в сегменте, выделенном i -ым экспертом, так и в сегменте, полученном в результате работы алгоритма ($i = 1, \dots, b$).

Результаты работы алгоритма будем характеризовать двумя параметрами: эффективностью и процентом ложных срабатываний. Эффективность eff определяется как среднее значение количества пикселей, выделенных экспертами и определенных алгоритмом одновременно:

$$eff = \frac{1}{b} \sum_{i=1}^b \frac{N_i^{ah}}{N_i^h}.$$

Процент ложных срабатываний err определим как среднее количество точек, которые есть в сегменте, выделенном алгоритмом, но отсутствуют в сегменте, выделенном экспертом:

$$err = \frac{1}{b} \sum_{i=1}^b \frac{N^a - N_i^{ah}}{N^a}.$$

Кроме алгоритма, предложенного в данной главе также тестировались три алгоритма сегментации: алгоритм водораздела [20, 21], алгоритм k -средних [22, 23] и алгоритм сегментации Отсу [24, 25]. Для всех алгоритмов вычислялось значение эффективности и процент ложных срабатываний для каждого алгоритма в среднем по всем экспертам.

Будем считать, что выделение сегмента осуществлено алгоритмом успешно, если количество верно выделенных точек сегмента превышает пороговое значение eff_0 , при этом процент ложных срабатываний ниже порогового значения err_0 . Эти два условия должны выполняться одновременно. Если большое количество точек определено верно, но при этом высок также и процент ложных срабатываний, то сегмент не выделен.

В дальнейшем для факта успешного выделения кластера установим пороговые значения $eff_0 = 50\%$, $err_0 = 25\%$. При этих значениях полученный алгоритмом сегмент может быть визуальным сопоставлен с сегментом, выделенным экспертом и при этом визуальным понятно, какой фрагмент изображения может быть отнесен к сегменту.

Результаты компьютерного эксперимента на выделенной подколлекции показали, что предложенный в данной работе алгоритм успешно выделяет сегменты на 44% изображений, алгоритм водораздела – на 3%, алгоритм k -средних – на 30%, алгоритм Отсу – на 10%. Как видно из сравнения результатов, предложенный алгоритм значительно эффективней всех методов, с которыми производилось сравнение. Полученное преимущество обусловлено главным образом низким процентом ложных срабатываний.

3 Анализ результатов и выводы

Предложенный в статье алгоритм кластеризации изображений, основанный на графовом представлении с последующим выделением сообществ, позволяет получать хорошие результаты как на искусственных графических объектах, так и на фотографиях. Особенностью предложенного метода является использование разности интенсивностей цветов, характерную для дифференциальных методов сегментации. При этом границы сегментов имеют четкие границы, что характерно для методов, основанных на значении интенсивности, например, для метода водораздела. Сравнение с другими методами сегментации продемонстрировало значительно более низкий процент ошибок ложного срабатывания при сопоставимой эффективности. При этом следует отметить квадратичную трудоемкость алгоритма, что делает его конкурентноспособным по сравнению с большим количеством широко распространенных методов сегментации.

Список литературы

- [1] I.K. Park, I.D. Yun, S.U. Lee. Color image retrieval using hybrid graph representation. *Image and Vision Computing*, 17(7):465–474, 1999.
- [2] B. Xu, J. Bu, C. Chen, C. Wang, D. Cai, X. He. Emr: a scalable graph-based ranking model for content-based image retrieval. *IEEE Transactions on knowledge and data engineering*, 27(1):102–114, 2015.
- [3] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, L. Fei-Fei. Image retrieval using scene graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*:3668–3678, 2015.
- [4] S.V. Belim, P.E. Kutlunin. Boundary extraction from images using clustering algorithm. *Computer Optics*, 39(1):119–124, 2015.
- [5] M.E. Newman. Analysis of weighted networks. *Physical Review E*, 70(5):056131, 2004.
- [6] O. Cuadros, G. Botelho, F. Rodrigues, J.B. Neto. Segmentation of Large Images with Complex Networks. *SIBGRAPI*:24–31, 2012.
- [7] A. Clauset, M.E.J. Newman, C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.
- [8] U.S. Raghavan, R. Albert, S. Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76:036106, 2007.
- [9] X. Ren, J. Malik. Learning a classification model for segmentation. *Proceedings on the Ninth IEEE International Conference on Computer Vision*, 1:10–17, 2003.
- [10] A. Levinstein, A. Stere, K.N. Kutulakos, D.J. Fleet, S.J. Dickinson, K. Siddigi. Turbopixels: Fast superpixels using geometric flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:2290–2297, 2009.
- [11] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Susstrunk. Slic superpixels. *EPFL Technical Report 149300, Tech. Rep.*, 2010.
- [12] C. Cigla, A.A. Alatan. Efficient graph-based image segmentation via speeded-up turbo pixels. *17th International Conference on Image Processing*:26–29, 2010.
- [13] Y. Mouchid, M.E. Hassouni, H.A. Cherif. New Image Segmentation Approach using Community Detection Algorithms. *International Journal of Computer Information Systems and Industrial Management Applications*, 8:195–204, 2016.
- [14] R.S. Nair, K.V. Vineetha. Modularity Based Color Image Segmentation. *IJIREEICE*, 3(1):109–113, 2016.
- [15] P. Pourian, S. Karthikeyan, B.S. Manjunath. Weakly supervised graph based semantic segmentation by learning communities of image-parts. *The IEEE International Conference on Computer Vision (ICCV)*:1359–1367, 2015.
- [16] S. Li, D.O. Wu. Modularity-Based Image Segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(4):570–581, 2015.

- [17] S. Verma, A. Chugh. An Increased Modularity based Contour Detection. *International Journal of Computer Applications*, 135(12):41–44, 2016.
- [18] A. Browet, P.-A. Absil, P. Van Dooren. Community Detection for Hierarchical Image Segmentation. *IWCIA*, LNCS 6636:358–371, 2011.
- [19] D. Martin, C. Fowlkes, D. Tal, J. Malik. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics. *Proc. 8th Int'l Conf. Computer Vision*, 2:416–423, 2001.
- [20] K. Malik. Modified Watershed Algorithm for Segmentation of 2D Images. *Journal of Information science & Information Technology*, 6(3):546–552, 2009.
- [21] M. Cui, Sh. Sun, Y. Pan. An Image Region Merging Algorithm Based on Modified Fast Watershed Transform. *Journal of Computer Aided Design & Computer Graphics*, 17(3):546–552, 2005.
- [22] J. MacQueen. Some methods for classification and analysis of multivariate observations. *In Proc. 5th Berkeley Symp. on Math. Statistics and Probability*:281–297, 1967
- [23] D. Arthur, S. Vassilvitskii. How Slow is the k-means Method?. *Proceedings of the 2006 Symposium on Computational Geometry (SoCG)*, 2006.
- [24] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 9:62–66, 1979.
- [25] P.-S. Liao, T.-S. Chen, P.-C. Chung. A Fast Algorithm for Multilevel Thresholding. *J. Inf. Sci. Eng.*, 17:713–727, 2001.

Images Segmentation Algorithm on the Basis of Searching the Communities on Graphs

Sergey V. Belim, Stanislav B. Larionov

In article the new algorithm of images segmentation is suggested. The weighing graph is corresponded to the image. The vertexes of the graph are pixels. Edges connect only the closest neighbors. Weight of edges depends on the difference of pixels colors. Constructed graph divide into communities. The image segment is corresponded to each community.