Исследование метода слияния областей в задаче цветовой сегментации[1]

Тимур Ханипов
Московский государственный университет имени М. В. Ломоносова
khanipov@gmail.com

Дмитрий Николаев
Институт проблем передачи информации имени А. А. Харкевича РАН
dimonstr@iitp.ru

Аннотация

В статье приводятся результаты исследования метода слияния областей в задаче цветовой сегментации для случая, когда процесс слияния начинается с отдельных пикселей изображения. Описаны особенности программной реализации метода: представление графа соседства областей, процедура слияния вершин и способ хранения ребер, обеспечивающие высокую скорость работы. Приведены результаты исследования зависимости качества сегментации от степени зашумленности изображения. Показана возможность применения метода для бинаризации текстовых изображений.

1 Введение

Задача цветовой объектной сегментации является важной задачей цифровой обработки изображений. Автоматическое выделение объектов применяется при сжатии видеопотоков [1], проведении УЗИ и МРТ диагностики [2], редактировании изображений и в алгоритмах шумоподавления. На сегодняшний день существует множество различных подходов к решению этой задачи: сегментация через выделение границ и оконтуривание [3], кластеризация цветового пространства [4], деление-слияние областей [5], соревнование областей [6] и другие методы. В этой работе рассматриваются особенности программной реализации метода слияния областей в случае, когда в роли начального разбиения выступают отдельные пиксели, исследуется влияние шума на качество сегментации и описывается применение алгоритма в задаче бинаризации текстовых изображений.

2 Описание метода слияния

Метод слияния областей позволяет получить целевые сегменты путем последовательного слияния более мелких соседних областей, полученных в результате так называемой предсегментации, с помощью какого-либо алгоритма (например, методом водоразделов). В этой работе стадия предсегментации была опущена, и в качестве элементов начального разбиения использовались отдельные пиксели с четырехсвязным соседством.

Вначале строится граф соседства областей с вершинами, соответствующими областям разбиения, а в нашем случае – пикселям, и ребрами, соответствующими отношению соседства. Каждому ребру приписан вес, отражающий «непохожесть» сегментов. На каждом шаге алгоритма объединяются две вершины, соединенные ребром с наименьшим весом. Процесс останавливается либо когда наименьший вес начинает превышать некоторое пороговое значение, либо когда достигается заданное число сегментов.

3 Критерий слияния

Во многих ситуациях (например, при распознавании символов на однородном фоне) достаточно использовать весьма простую модель изображения – *мондриан* с нормальным шумом. *Мондрианами* в обработке изображений называют рисунки, состоящие из небольшого числа сегментов постоянного цвета. Предполагается, что к каждой компоненте цвета мондриана прибавлен шум,

[1] Работа частично финансово поддержана грантом РФФИ 09-07-00473-а.

распределенный по нормальному закону с нулевым средним и постоянной дисперсией.

Каждый сегмент в цветовом пространстве RGB порождает сферическое нормальное распределение с центром в «основном» цвете сегмента. На рис. 1 и 2 приведены примеры мондриана с низкой и высокой дисперсиями шума соответственно. Справа на каждом рисунке показаны цветовые гистограммы двух областей: зеленой и красной (помечены стрелками в правом верхнем углу мондриана):

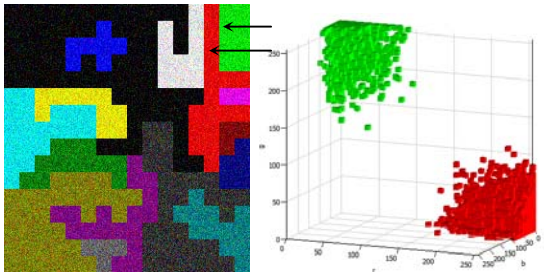

Рис. 1

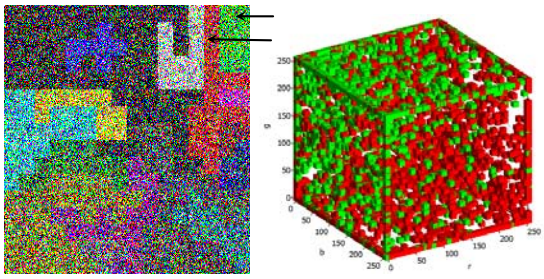

Рис. 2

Из рисунков видно, что, к примеру, метод кластеризации справился бы с сегментацией первого изображения, но дал бы плохие результаты для второго, поскольку он никак не учитывает взаимное расположение сегментов. Метод слияния областей, напротив, при правильном выборе критерия слияния показывает хорошие результаты как в первом, так и во втором случае.

Оптимальным критерием слияния в модели мондриана с нормальным шумом является критерий, основанный на следующем t-критерии Стьюдента проверки равенства средних для независимых выборок из нормального распределения с известной одинаковой дисперсией:

$$t = \frac{\| \vec{\mu}_A - \vec{\mu}_B \|}{\sigma \sqrt{\frac{1}{n_A} + \frac{1}{n_B}}}, \qquad (1)$$

где **A** и **B** – сегменты-кандидаты на слияние, n – площадь, а $\vec{\mu}$ – средний цвет сегмента (компоненты R, G и B), $\| \bullet \|$ – евклидова норма.

Т.к. для алгоритма важен лишь порядок слияния ребер и не важны абсолютные значения весов, то можно было модифицировать (1) и использовать следующий критерий:

$$dist(A,B) = \frac{n_A n_B}{n_A + n_B} \| \vec{\mu}_A - \vec{\mu}_B \|^2 \qquad (2)$$

Такой критерий позволяет не проводить оценку дисперсии в (1), а также не требует операции извлечения квадратного корня. Именно (2) был использован для присвоения весов ребрам в нашей работе.

На рис. 3 и 4 показаны результаты сегментации мондрианов, изображенных соответственно на рис. 1 и 2, с помощью критерия (2). Процесс останавливался, когда количество сегментов становилось равно 16 (т.е. сравнивалось с количеством «идеальных» сегментов). Зеленым обозначены границы найденных сегментов:

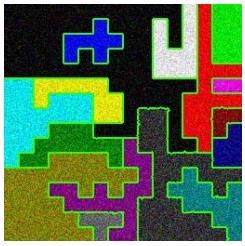

Рис. 3

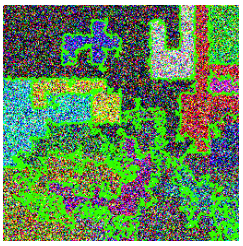

Рис. 4

Отметим, что вместо (2) можно использовать более сложные критерии, учитывающие физические особенности формирования сцены, как это сделано, например, в [7]. Важно лишь, чтобы они позволяли быстро, без прохода по точкам сегмента пересчитывать веса ребер после слияния. Этому условию удовлетворяют, например, критерии, вычисляющиеся на основе *аддитивных статистик*, примерами которых являются площадь и выборочная сумма, использующиеся для расчета средних в (2).

При исследовании качества изображений для сравнения также брался «наивный» критерий слияния, не учитывающий площади сегментов:

$$dist_1(A,B) = \| \vec{\mu}_A - \vec{\mu}_B \|^2 \qquad (3)$$

4 Программная реализация

При создании программы, реализующей сегментацию методом слияния областей, нужно было учесть несколько факторов. Во-первых, представление графа соседства областей должно было быть не очень объемным по памяти (для изображения размером 1024x1024 количество вершин в нем на начальном этапе есть 2^{20}), во-вторых, оно должно позволять быстро производить слияние вершин и, в-третьих, выбор ребра с наименьшим весом должен быть легкой процедурой.

С учетом первого требования было выбрано представление графа типа «список смежности», т.к.

на начальных стадиях граф является *разреженным*, и такой способ его хранения занимает меньше места, чем матрица смежности. Действительно, для графа с V вершинами и E ребрами списки смежности занимают $O(E)$ памяти, а матрица смежности – $O(V^2)$. На первом шаге, когда каждый пиксель соответствует отдельному сегменту, у изображения размером M на N пикселей $V = MN$, а $E = 2MN - M - N = O(V)$, и поэтому использовать списки смежности выгоднее.

Для того чтобы быстро извлекать лучшее ребро, использовалась бинарная куча, упорядоченная по весу [8]. Это позволяло находить минимальное ребро за $O(1)$, а перевзвешивать и удалять ребра – за $O(\log E)$ (E уменьшается на единицу на каждом шаге).

Быстрое слияние вершин обеспечивалось с помощью модификации традиционного списка смежности путем введения *линков*. Поясним это на примере. Пусть сливаются две вершины, **A** и **B**, имеющие в сумме n соседей, среди которых максимальный порядок равен k (напомним, что *порядком* вершины называется количество смежных с ней вершин). Тогда в случае традиционной реализации списков смежности, когда элементами списка являются лишь указатели на соседние вершины, слияние вершин **A** и **B** потребует $O(nk)$ операций. Действительно, в списке смежности каждого из соседей **A** и **B** необходимо отразить факт слияния, изменив соответствующий элемент списка (содержащий **A** или **B**) на новую вершину. Поиск этого элемента в списке длины k займет $O(k)$ операций, и такие замены надо сделать n раз.

Сложность слияния $O(nk)$ является неприемлемой для сколько-нибудь больших изображений, ее необходимо улучшить. Это было сделано следующим образом: вместо того, чтобы хранить в списке смежности вершины **A** только указатели на соседние вершины, мы вместе с этими указателями хранили также указатель на позицию **A** в списках смежности соседей. Структуру, состоящую из указателя на вершину и указателя на позицию в списке смежности, мы назвали *линком*. Рис. 5 иллюстрирует идею:

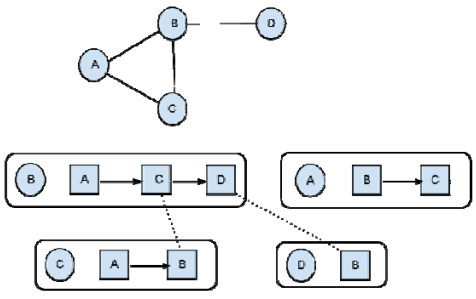

Рис. 5

Вверху изображен граф, состоящий из четырех вершин, а внизу – его представление. Стрелками соединены элементы списков смежности, а пунктирной линией показаны указатели на позицию в списке для вершин, соединенных с **B** (другие пунктирные линии не проведены, чтобы не загромождать рисунок). *Линк* здесь представляется как объединение стрелки (указателя на следующий линк), пунктирной линии и названия вершины в прямоугольнике.

Если вершина **A** объединяется с вершиной **B**, то отразить это в вершине **C** можно сразу, не осуществляя поиск нужной позиции – она известна из линка. Таким образом, вместо сложности $O(nk)$ достигается сложность $O(n)$. В реализации линк также содержал указатель на соответствующее ребро, чтобы можно было сразу приписать ему новый вес без поиска по множеству ребер.

Нужно заметить, что выше ничего не говорилось о необходимости удаления дублей: например, вершина **C** не должна встречаться дважды в списке смежности объединенной вершины **AB**. Для удаления повторяющихся вершин с сохранением сложности слияния $O(n)$ был использован следующий прием. В начале работы алгоритма заводится массив M размера $O(V)$, сопоставляющий каждой вершине целое число, и заполняется нулями. На каждом шаге алгоритма при слиянии вершин **A** и **B** вначале элементы **A** помечаются в M: в соответствующей позиции ставится текущий номер шага. После этого при проходе по **B** к новой вершине присоединяются только те элементы, которые не помечены в M (т.е. такие вершины v, у которых $M[v]$ меньше текущего номера шага). Такой способ позволяет избавиться от дублей за $O(n)$ операций.

Отметим, что в первоначальной реализации алгоритма был применен другой способ: объединение упорядоченных списков. Списки линков перед слиянием должны были быть отсортированы по номеру вершины, после чего за $O(n)$ операций осуществляется объединение с удалением дублей, причем новый список автоматически оказывается упорядоченным. Единственная ситуация, при которой упорядоченность нарушается – модификация элементов вершин, смежных с объединяемыми. Поддерживая флаг упорядоченности вершины, можно не выполнять сортировку вершин, если они уже упорядочены («ленивая» сортировка). При таком способе в лучшем случае слияние осуществляется за $O(n)$, а в худшем – за $O(n \log n)$ операций.

С учетом того, что перевзвешивание ребра требует $O(\log E)$, итоговая сложность слияния вершин при использовании массива составляет $O(n \log E)$, а при использовании «ленивой» сортировки – $O(n \log E)$ или $O(n \log nE)$.

В следующей таблице показано время сегментации в секундах двух изображений: «Тюльпаны» (рис. 6, 1024x768) и «Текст» (рис. 8, 640x256). Эксперименты проводились на компьютере с процессором Intel Core 2 Duo с тактовой частотой 2 ГГц и 4 Гб ОЗУ.

	сортировка	**массив**
«Тюльпаны»	22,0	12,0
«Текст»	4,1	2,2

Рис. 6

Хотелось бы отметить, что построенная реализация никак не привязана к критерию слияния. Важно только, чтобы можно было быстро вычислять веса ребер после слияния вершин.

5 Исследование влияния шума

Представляло интерес исследование зависимости качества сегментации от степени зашумленности изображения. Для его проведения нужно было, во-первых, зафиксировать модель изображения, а во-вторых, строго определить, что понимать под качеством.

Моделью изображения был выбран «мондриан» с нормальным шумом, именно для такой модели использованный нами критерий слияния является оптимальным. С помощью специальной программы генерировалась серия мондрианов с известным заранее количеством сегментов *N*, на них накладывался шум с разной дисперсией, после чего запускался процесс сегментации, который останавливался, когда количество сегментов совпадало с *N*. Для определения качества сегментации вначале устанавливалось соответствие между «идеальными» сегментами незашумленного изображения A_i и «реальными», полученными с помощью программы сегментами B_j: для каждой пары (i, j) подсчитывалась *похожесть* $s_{ij} = s(A_i, B_j)$ областей A_i и B_j по формуле

$$s(A,B) = 1 - \frac{|A \cap B|}{|A \cup B|},$$

после чего с помощью алгоритма Куна находилась такая биекция $\tau(i)$ между идеальными и реальными сегментами, которая максимизировала суммарную похожесть

$$S = \sum_{i=1}^{N} s_{i,\tau(i)}.$$

Под *качеством* сегментации понималась величина

$$q = \frac{S}{N}.$$

На рис. 7 приводятся графики зависимости качества сегментации от среднеквадратичного отклонения шума для оптимального критерия слияния (2) и «наивного» критерия (3) красным и синим цветами соответственно:

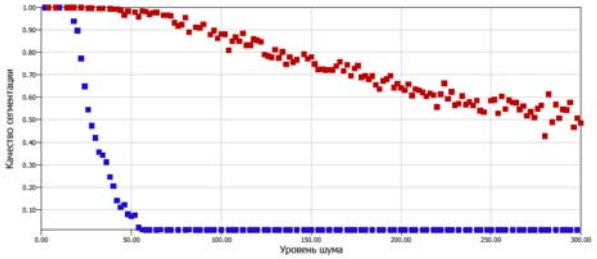

Рис. 7

Видно, что с нарастанием дисперсии качество сегментации «наивным» критерием резко падает до 0, в то время как оптимальный критерий дает хороший результат даже для сильно зашумленных изображений.

6 Применение в задаче бинаризации

В [9] предлагается способ бинаризации, опирающийся на сегментацию изображения методом слияния областей с предсегментацией методом водоразделов. В этой работе была осуществлена попытка применить аналогичную технику в случае предразбиения на отдельные пиксели. Бинаризация проводилась в три шага:

1. Сегментация.
2. Выделение фона как объединения всех сегментов достаточной большой площади.
3. Удаление «полостей» из букв «а», «р» и др.

Пользователем задавались порог остановки сегментации и максимальная площадь символа. Полости из букв удалялись согласно следующей эвристике: полостью считался любой сегмент, не

граничащий с фоном, но имеющий соседа, который с ним граничит.

На рис. 8 показаны последовательные этапы бинаризации описанным методом: исходное изображение, границы сегментов, выделение фона и результат после удаления полостей. Видно, что даже текст в области слабого контраста был выделен правильно.

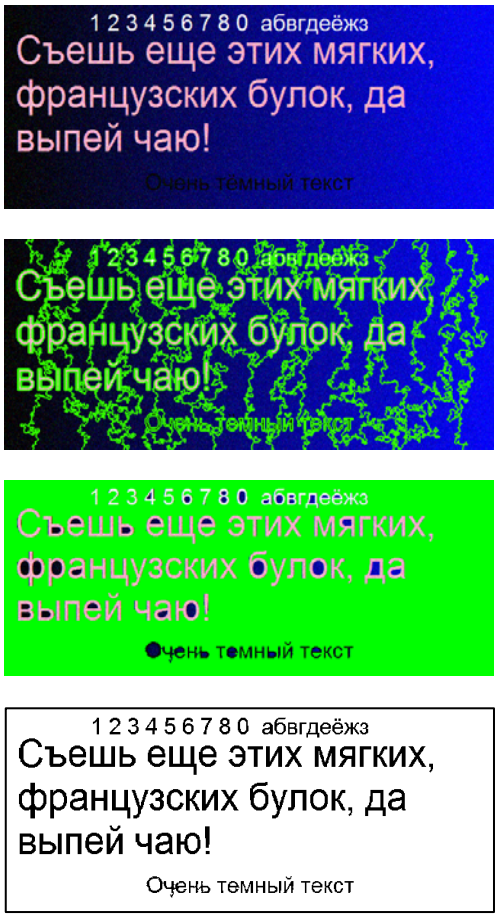

Рис. 8

7 Заключение

В работе показана возможность использования исходного разбиения изображения на отдельные пиксели. Построена программная реализация метода слияния областей, учитывающая повышенные требования к вычислительной эффективности, связанные с большим количеством вершин графа соседства областей на начальном этапе сегментации.

Исследование влияния шума на качество сегментации показало устойчивость оптимального критерия к шуму.

Продемонстрировано применение сегментации методом слияния областей в задаче бинаризации изображений цветного текста со сложным контрастом.

8 Список литературы

[1] Gary J. Sullivan, Thomas Wiegand, “Rate-distorsion optimization for video compression”, *IEEE Signal Proc.,* Nov. 1998.

[2] L.P.Clarke and others, “MRI segmentation: methods and applications”, *Magnetic Resonance Imaging,* 13 (3), 1995, pp. 343-368.

[3] Parent P., Zucker S., “Trace inference, curvature consistency and curve detection”, *IEEE Trans. Pattern Analysis Mach. Intel*, V. 11, 1989.

[4] Шапиро Л., Стокман Дж., *Компьютерное зрение*. М.: БИНОМ, 2006.

[5] Pietikainen M., Rosenfeld A., Walter I, “Split and link algorithm for image segmentation”, *Pattern Recogn.*, V. 15, 1982.

[6] Zhu S., Yuille A., “Region competition” *IEEE Trans. Pattern Analysis Mach. Intel*, V. 18, 1996.

[7] Николаев Д. П., “Алгоритмы цветовой сегментации, применимые в условиях сложного освещения сцены”. *Автореферат на соискание ученой степени кандидата физико-математических наук*. Москва. Физический факультет МГУ им. М. В. Ломоносова, 2004.

[8] T. Kurita, “An efficient agglomerative clustering algorithm using a heap”, *Pattern Recognition*, 24 (3), March, 1991, pp. 205-209.

[9] Nikolaev D. P., “Segmentation-based binarization method for color document images”. - In proceedings: *The 6th German-Russian Workshop "Pattern recognition and image understanding" (OGRW-6)*. Novosibirsk, 2003, pp. 190-193.