

УДК 681.3.11

Р.А. Калачик

АЛГОРИТМ АВТОМАТИЧЕСКОЙ СЕГМЕНТАЦИИ С АГРЕГИРОВАНИЕМ СЕГМЕНТОВ В СИСТЕМАХ ПОИСКА ИЗОБРАЖЕНИЙ

Предлагается формировать результаты автоматической сегментации изображений в виде новой структуры данных – дерева агрегирования сегментов. Разработана модификация алгоритма Краскала, позволяющая генерировать такое дерево. Приведены результаты экспериментальных исследований модифицированного алгоритма.

Введение. При решении задачи построения системы поиска в базе изображений одной из важнейших процедур является сегментация, позволяющая представить изображение в виде совокупности сегментов, в идеале соответствующих отдельным объектам, образующим изображение.

С учетом количества изображений в типичной базе данных очевидно, что ручная сегментация в большинстве случаев оказывается невозможной из-за больших временных затрат на её выполнение. Алгоритмы автоматической сегментации существуют, однако результаты их работы в общем случае оказываются существенно отличными от результатов ручной сегментации. Это объясняется тем, что без априорных знаний о структуре изображения сложно или практически невозможно добиться правильного сегментирования объектов.

В статье предлагается отличный от традиционных подход к представлению результатов сегментации, в рамках которого предлагается отказаться от определения фиксированного набора сегментов и рассматривать результат сегментации как иерархическую структуру – *дерево агрегирования сегментов*, формируемое последовательно, в процессе слияния сегментов. Концептуальной основой для данного подхода послужила пирамида изображений Бурта и Адельсона[1]. Однако, в отличие от последней, в дереве агрегирования сегментов хранится не тривиальное разномасштабное изображение, а структурированная промежуточная информация о сегментах. Это позволяет извлечь из дерева сегментации тот или иной набор сегментов, соответствующий интересующему уровню детализации.

Заметим, что при предложенном подходе не приходится решать некорректную задачу выбора детализации сегментирования. Для реализации предложенного подхода был модифицирован алгоритм сегментации, основанный на теории

графов и алгоритме Краскала[2]. Его краткое описание сводится к следующему.

Алгоритм сегментации, основанный на теории графов и алгоритме Краскала. Пусть $G = (V, E)$ – ненаправленный граф с множеством вершин $v_i \in V$ и множеством граней $(v_i, v_j) \in E$. Множество вершин соответствует сегментируемому множеству. Каждая грань имеет вес $w((v_i, v_j))$, который представляет собой неотрицательную величину и выражает степень схожести между соседними элементами сегментируемого множества. В случае сегментации цифровых изображений элементы множества V являются пикселями, а веса граней определяют собой степень схожести пикселей, соединенных соответствующей гранью.

Сегментация S представляет собой разбиение множества V на такие компоненты $C \in S$, что каждому C соответствует подграф $G' = (V, E')$, где $E' \subseteq E$.

Предикат для оценивания необходимости разбиения двух компонентов при сегментации основан на измерении разницы между элементами вдоль границы двух компонентов относительно разницы между внутренними элементами каждого из двух компонентов.

Внутренняя разность компонента $C \subseteq V$ определяется как максимальный вес в кратчайшем остове графа компонента, $MST(C, E)$:

$$Int(C) = \max_{e \in MST(C, E)} (w(e)).$$

Разница между двумя компонентами $C_1, C_2 \subseteq V$ определяется как минимум весов граней, соединяющих два компонента:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j)).$$

Если нет ни одного ребра, соединяющего C_1 и C_2 , то $Dif(C_1, C_2) = \infty$. Данная мера разности может быть проблемой, поскольку опери-

рует только с самым маленьким весом между двумя компонентами. На практике было показано, что данный способ измерения работает достаточно хорошо, несмотря на это заметное ограничение.

Пороговая функция D используется для определения необходимости объединения двух компонентов:

$$D(C1, C2) = \begin{cases} 1, & Dif(C1, C2) > MInt(C1, C2) \\ 0, & Dif(C1, C2) < MInt(C1, C2), \end{cases}$$

где

$$MInt(C1, C2) = \min(Int(C1) + \tau(C1), Int(C2) + \tau(C2)).$$

Пороговая функция τ определяет величину, на которую должны отличаться два компонента, чтобы представлять разные сегменты. Пусть $\tau(C) = k / |C|$, где k – эмпирическая константа.

В качестве исходных данных для работы алгоритма используется граф $G = (V, E)$, состоящий из n вершин и m ребер. Результатом работы является сегментация множества V на компоненты $S = (C_1, \dots, C_r)$.

Шаг 1. Сортировать E по возрастанию весов. В результате получим упорядоченный набор ребер $\pi(o_1, o_2, \dots, o_m)$.

Шаг 2. Задать начальную сегментацию S^0 таким образом, что каждой вершине соответствует свой компонент в сегментации.

Шаг 3. Повторять шаг 4 для каждого $q = 1, \dots, m$.

Шаг 4. Построить S^q из S^{q-1} . Имеем $o_q = (v_i, v_j)$. Вершины v_i и v_j принадлежат компонентам из S^{q-1} : C_i^{q-1} и C_j^{q-1} соответственно. Если $C_i^{q-1} \neq C_j^{q-1}$ и $w(o_q) \leq MInt(C_i^{q-1}, C_j^{q-1})$, то S^q получается объединением C_i^{q-1} и C_j^{q-1} в один компонент, иначе $S^q = S^{q-1}$.

Шаг 5. Вернуть $S = S^m$.

Основным недостатком данного алгоритма является необходимость ручного подбора константы k для каждого изображения. Это существенно ограничивает область его применения в задачах автоматической сегментации, поскольку для получения приемлемых результатов необходимо участие человека. Ниже предлагается модификация алгоритма, позволяющая избежать необходимости подбора константы k и, таким образом, обеспечить возможность автоматической сегментации.

Агрегирование сегментов. Как упоминалось выше, сегментация полутоновых изображений является неоднозначной задачей. На рисун-

ке 1 изображен фрагмент полутонового изображения. В зависимости от сцены сегментация данного фрагмента будет происходить по-разному. Слева на рисунке изображена поверхность вращения. Рассматриваемый фрагмент представлен одним сегментом. Справа на рисунке рассматриваемый фрагмент разбивается на два сегмента – объект и фон.



Рисунок 1 – Пример разной сегментации яркостного клина

Неоднозначность сегментации полутоновых изображений затрудняет использование результатов сегментации в качестве низкоуровневых признаков в системах поиска графической информации. Требование выдачи фиксированного набора сегментов в существующих алгоритмах автоматической сегментации приводит к тому, что форма сегментов не всегда повторяет форму объектов на изображении. Возможны два типа ошибок сегментации.

1. Избыточная детальность сегментации. Сегменты маленького размера и их количество значительно превышают количество объектов на сегментируемом изображении.

2. Недостаточная детальность сегментации. Сегменты большого размера и их количество меньше, чем количество объектов на сегментируемом изображении.

Решение этой задачи заключается в отказе от формирования результата сегментации в форме фиксированного набора сегментов и переходе к выдаче дерева агрегирования сегментов, содержащего промежуточную информацию о сегментах, формируемых на разных этапах обработки (рисунок 2).

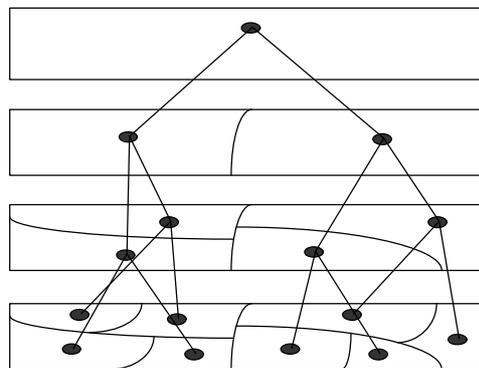


Рисунок 2 – Дерево агрегирования сегментов

Дерево агрегирования предлагается строить следующим образом: на первом этапе существующим алгоритмом Краскала производится сегментация предъявленного изображения с заведомо высокой детальностью сегментации. Сегменты, получившиеся в результате данной операции, соответствуют нижнему слою в дереве агрегирования. Каждый последующий слой в дереве получается объединением соседних сегментов предыдущего слоя. Верхний слой дерева агрегирования будет содержать единственный сегмент, соответствующий самой низкой детальности сегментации. В результате дерево агрегирования будет содержать наиболее вероятные варианты сегментации предъявленного изображения, оставляя возможность выбора детальности сегментации на последующих этапах. Как показано на рисунке 1, окончательный выбор детальности сегментации должен производиться на основе семантики изображения, которая, как правило, не известна на этапе сегментации. Важным положительным результатом является то, что для изменения детальности сегментации нет необходимости перезапускать процедуру сегментации, поскольку всю необходимую информацию дерево агрегирования уже содержит. Сформулируем окончательно алгоритм автоматической сегментации с агрегированием сегментов.

Алгоритм автоматической сегментации с агрегированием сегментов. Пусть известно, что площадь объектов на изображении не меньше определенного размера μ . В этом случае нет смысла хранить информацию о формировании сегментов, площадь которых не превышает μ .

Определим $C_a = C_1 \cup C_2$ – агрегированный сегмент, полученный объединением сегментов C_1 и C_2 . Обозначим дерево агрегирования сегментов $G_a = \{C_a, V_a\}$.

После модификации шагов 2, 4 и 5 получим следующий алгоритм.

Шаг 1. Сортировать E по возрастанию весов. В результате получим упорядоченный набор ребер $\pi(o_1, o_2, \dots, o_m)$.

Шаг 2а. Задать начальную сегментацию S^0 таким образом, что каждой вершине будет соответствовать свой компонент в сегментации.

Шаг 2б. Проинициализировать C и V_a пустыми множествами.

Шаг 3. Повторять шаг 4 для каждого $q = 1, \dots, m$.

Шаг 4а. Построить S^q из S^{q-1} , построить дерево агрегирования сегментов. Имеем

$o_q = (v_i, v_j)$. Вершины v_i и v_j принадлежат компонентам из S^{q-1} : C_i^{q-1} и C_j^{q-1} соответственно.

Шаг 4б. Если $C_i^{q-1} \neq C_j^{q-1}$ и

$w(o_q) \leq MInt(C_i^{q-1}, C_j^{q-1})$, то:

$$C_a^q = C_i^{q-1} \cup C_j^{q-1},$$

$$S^q = (S^{q-1} / \{C_i^{q-1}, C_j^{q-1}\}) \cup C_a^q.$$

Шаг 4в. Если $(|C_i^{q-1}| > \mu) \& (|C_j^{q-1}| > \mu)$, то

$$C_a = C_a \cup C_a^q \cup C_i^{q-1} \cup C_j^{q-1},$$

$$V_a = V_a \cup (C_a^q, C_i^{q-1}) \cup (C_a^q, C_j^{q-1}), \quad \text{иначе}$$

$$S^q = S^{q-1}.$$

Шаг 5. Вернуть $G_a = \{C_a, V_a\}$.

Рассматриваемый алгоритм работает таким образом, что пока площадь объединяемых сегментов не превышает μ , его работа не отличается от алгоритма сегментации Краскала, если площадь объединяемых сегментов превышает μ , т.е. выполняется условие $(|C_i^{q-1}| > \mu) \& (|C_j^{q-1}| > \mu)$, формируется дерево агрегирования сегментов.

Коэффициент k в модифицированном алгоритме не влияет на конечный результат сегментации, он определяет минимальный размер сегмента нулевого уровня в дереве агрегирования сегментов.

Реализация предложенного алгоритма.

Предлагаемый алгоритм был реализован средствами языка C++ в среде разработки Microsoft Visual Studio 2005. Пример работы программы на тестовом изображении представлен на рисунке 3.

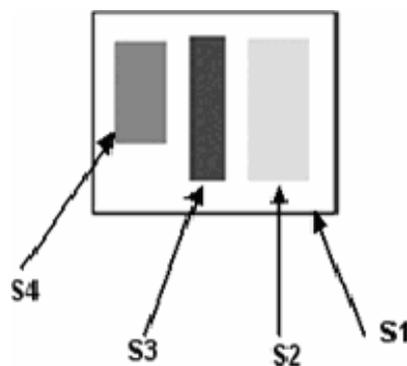


Рисунок 3 – Тестовое изображение и сегменты первого уровня дерева сегментации

В результате самой детальной сегментации получено 4 сегмента: S1, S2, S3, S4, показанные на рисунке 4. При построении дерева агрегирования получено еще три сегмента: S5, S6, S7. Таким образом, полученное на выходе алгорит-

ма дерево агрегирования сегментов позволяет изменять детальность сегментации без повторного выполнения процедуры сегментации.

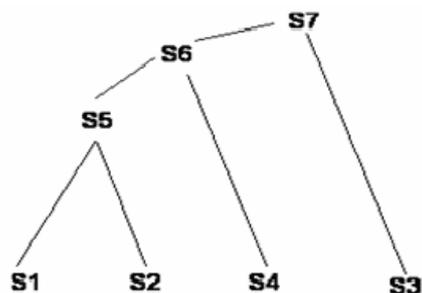


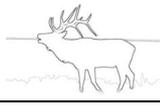
Рисунок 4 – Дерево агрегирования для тестового изображения

Для оценки качества работы алгоритмов сегментации в университете Беркли (США) была проведена работа по созданию базы изображений, сегментированных людьми вручную. Из базы изображений Cogel, широко используемой для тестирования систем машинного зрения, были отобраны 1000 изображений. Каждое из выбранных изображений было сегментировано вручную десятью разными людьми. База изображений и вариантов их ручной сегментации получила название Berkeley Segmentation Dataset. Результаты данного исследования опубликованы в работе [3]. Для иллюстрации работы алгоритмов из Berkeley Segmentation Dataset было выбрано несколько изображений. Данные изображения представлены в первом столбце таблицы. Для каждого изображения в Berkeley Segmentation Dataset существует 10 вариантов ручной сегментации. В столбце 2 таблицы представлен обобщенный вариант ручной сегментации. Толщина линии пропорциональна числу вариантов ручной сегментации, в которых встречается данная граница. В столбце 3 таблицы представлена сегментация тестовых изображений предлагаемым алгоритмом. Поскольку на выходе предложенного алгоритма можно получить любой уровень детализации сегментирования, для анализа выбирался уровень из дерева агрегирования сегментов, число сегментов в котором максимально близко к числу сегментов в соответствующей ручной сегментации.

Как видно из таблицы, результат работы предложенного алгоритма близок к результатам ручной сегментации. Данные результаты можно получить и на алгоритмах с фиксированной детальностью сегментации путем ее подбора для

каждого из изображений. Но для этого потребуется участие человека. Сегментация всех тестовых изображений происходила за один проход без изменений в параметрах алгоритма.

Тестовые изображения и результаты сегментации

Исходное изображение	Ручная сегментация	Результат работы модифицированного алгоритма
		
		
		
		
		

Выводы. Предложенный модифицированный алгоритм осуществляет формирование за один проход дерева агрегирования сегментов, которое в дальнейшем позволяет извлечь сегменты *любого уровня детализации* из заданного диапазона значений. Это делает его более эффективным в задаче определения низкоуровневых признаков в системах поиска графической информации по сравнению с существующими алгоритмами.

Библиографический список

1. P.J. Burt et A.E. Adelson, "The Laplacian pyramid as a compact image code", IEEE Trans. on Communications, 31, pp. 532-540, 1983.
2. Efficient Graph-Based Image Segmentation Algorithm / Pedro F. Felzenszwalb, Daniel P. Huttenlocher. IEEE Trans. on Communications, 4, pp. 342-359, 1994.
3. A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms / David Martin, Charles Fowlkes, Doron Tal, Jitendra Malik, ICCV, Jul. 2001.