ПАРАЛЛЕЛЬНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМА СЕГМЕНТАЦИИ ИЗОБРАЖЕНИЙ RHSEG С УЧЕТОМ АРХИТЕКТУРНЫХ ОСОБЕННОСТЕЙ СИСТЕМЫ BLUEGENE/P

А.И. Серебрийский

Одной из часто встающих задач в области распознавания и обработки изображений является сегментация изображений на регионы по определенным критериям. Критерии, которым должна удовлетворять итоговая сегментация(ии) определяются в зависимости от задачи. Существующие алгоритмы для выполнения таких задач потребляют память и вычислительные ресурсы пропорционально качеству работы и строгости заявленных критериев. Поэтому остро встает вопрос об уменьшении времени работы алгоритма за счет малых потерь в качестве. Одним из решений этой задачи является модификация алгоритма таким образом, чтобы появилась возможность его параллельной реализации.

В данной работе рассматривается модифицированная параллельная реализация алгоритма Recursive Hierarchical Segmentation (RHSEG)[1]. Алгоритм RHSEG является модифицированной версией алгоритма Hierarchical Segmentation (HSEG)[1]. Алгоритм HSEG разбивает изображение, базируясь на значениях спектров элементов изображения, и является алгоритмом с потреблением памяти и вычислительной сложностью порядка $O(m^2 * n^2)$, где m и n - размерность картинки. Поэтому для него очень остро стоит проблема параллельной реализации. Целью работы является изменение предложенной в [1] реализации для учета архитектурных особенностей BlueGene /P. Основными такими особенностями были невозможность разделять физический процессор с другими пользовательскими программными процессами, быстрая коммуникационная среда между вычислительными узлами и медленная связь с хранилищем данных.

За всю историю развития данного направления обработки изображений, было разработано множество семейств алгоритмов сегментации.

Алгоритмы кластеризации.

Это достаточно общее направление, не завязанное непосредственно на обработку изображений. Однако методы, разработанные в рамках этого направления, можно с успехом применять и к сегментации изображений. Примером алгоритма кластеризации является, например, итеративный континуальный алгоритм K-средних, описанный в работе [2], который является улучшением стандартного алгоритма k-средних, предложенного Ллойдом. Параллельные реализации алгоритма k-средних рассматриваются, например, в работах [3][4].

Методы, основанные на гистограмме.

Основное преимущество данных алгоритмов - очень быстрая работа, зачастую требующая всего одного прохода по изображению. Однако сегментация, основанная на гистограмме не может учитывать информацию о положении пикселей в пространстве при объединении их в регионы, что делает подобную сегментацию неприменимой к многим типам задач.

Выделение границ.

Являясь самостоятельным классом алгоритмов в области обработки изображений, часть алгоритмов может с успехом применяться для сегментации изображений. Результатом применения алгоритмов выделения границ является множество кривых, указывающих на факты резкого изменения параметров изображения. Однако проблемой при построении сегментации, основанной на выявленных границах является тот факт, что выявленные границы часто фрагментированы, а их структура на естественных изображениях может быть достаточно сложна для легкого преобразования в регионы.

Разрастание регионов.

Методы данного семейства основаны на итеративном разрастании регионов по определенным критериям. Алгоритмы данного класса различаются, в том числе, и по тому, какое определение сегментации изображения используется.

Классическим определением сегментации изображения на регионы является следующее:

Пусть X - двухмерный массив, представляющий собой изображение. Сегментация X это набор попарно-непересекающихся множеств $X_1, X_2, \ldots, X_N$, удовлетворяющих следующим условиям:

- $\bigcup_{i=1}^{N} X_i = X$
- $X_i, i = 1,2,\ldots,N$ связанны
- $P(X_i) = true для i = 1,2,\ldots,N$
- $P(X_i \cup X_j) = false, i \neq j где X_i и X_j соседи.$

$P(X_i)$ предикат который присваивает значения true или false региону X_i, в зависимости от значений пикселей в регионе.

По [1] эти условия можно интерпретировать следующим образом. Первое условие означает, что каждый пиксель должен принадлежать какому-то региону. Второе условие означает, что каждые регион должен быть связанным. Третье условие определяет критерий, т.е. каким условиям должны удовлетворять два пикселя изображения, чтобы считаться похожими. Четвертое условие говорит о том, что дальнейшее объединение любых двух соседних регионов будет нарушать третье условие.

Однако такое определение сегментации не лишено недостатков. Во-первых получаемая сегментация зависит от порядка прохода по изображению. Во-вторых, не определяется понятие оптимальности сегментации. Все сегментации, удовлетворяющие условиям, считаются одинаково хорошими.

Определение сегментации, которое нивелировало бы все вышеуказанные проблемы, может быть сформулировано таким образом:

Пусть X - двухмерный массив, представляющий собой изображение. Сегментация X это набор попарно-непересекающихся множеств $X_1, X_2, .., X_N$, удовлетворяющих следующим условиям:

- $¿_{i=1}^{N} X_i = X$
- $X_i, i = 1,2,\ldots,N$ связанны
- $\sum_i^N G(X_i), i = 1,2,\ldots,N$ минимальна среди всех возможных сегментаций X на N регионов.
- $G(X_i \cup X_j) > G(X_i) + G(X_j), i \neq j$ где X_i и X_j соседи.

$G(X_i)$ - функция эвристической оценки региона

Эти условия можно интерпретировать следующим образом: Первое условие означает, что каждый пиксель должен принадлежать какому-то региону. Второе условие означает, что каждые регион должен быть связанным. Третье условие означает, что это разбиение имеет минимальную эвристическую оценку среди всех возможных сегментаций изображения на N регионов. Четвертое условие означает, что дальнейшее объединение любых двух регионов увеличивает эвристическую оценку сегментации, полученную в третьем условии.

В таком определении результат не зависит от порядка прохода по изображению, потому что третье условие требует, чтобы сегментация была глобальным минимумом среди всех возможных сегментаций с тем же количеством регионов. Однако нахождение оптимальной сегментации почти невозможно, потому что требуется перебор всех возможных сегментаций. Также недостатком этого определения является то, что не определяется оптимальное N.

В работе [5] J.M.Beaulieu и M.Goldberg предложили алгоритм HSWO, который и лег в основу HSEG. Затем James C. Tilton разработал алгоритм RHSEG и его параллельную реализацию, модификацией которой и является реализация, предложенная в данной работе.

Описание алгоритма HSEG.

Данный алгоритм по заданному изображению строит иерархию сегментаций, каждая их которых представляет собой “улучшение” предыдущей. Для построения каждой следующей сегментации в иерархии используется алгоритм HSWO.

Определение сегментации, использованное в алгоритме HSWO следующее:

Пусть X - двухмерный массив, представляющий собой изображение и $X_1, X_2, .., X_N$ - это набор попарно-непересекающихся множеств , удовлетворяющих следующим условиям:

- $¿_{i=1}^{N} X_i = X$
- $X_i, i = 1,2,\ldots,N$ связанны

Пусть $G(X_i)$ - эвристическая функция оценки региона X_i. Переупорядочим разбиение $X_1, X_2, .., X_N$ так чтобы

$G(X_{N-1} \cup X_N) \leq G(X_i, X_j) \forall i \neq j$ где X_i и X_j - соседние регионы, и X_N и X_{N-1} - соседние регионы. Тогда сегментация X на N-1 регионов определяется как $X_1', X_2', \ldots, X_{N-1}'$, где $X_i' = X_i$ для $i = 1,2,\ldots,N-2$ и $X_{N-1}' = X_N \cup X_{N-1}$.

Отличия алгоритма HSEG от HSWO в том, что алгоритм позволяет объединять разделенные в пространстве регионы в один, чередуя объединение соседних регионов и разделенных.

Общее описание алгоритма HSEG можно представить в такой форме:

1. Задать начальную сегментацию изображения. Эта сегментация может предоставляться пользователем или генерироваться автоматически. Каждому пикселю изображения присваивается номер, обозначающий регион, к которому он принадлежит. В случае автоматической генерации начальной сегментации каждый пиксель становиться регионом и получает уникальный номер.
2. Для каждой пары соседних регионов вычисляется расстояние между ними.

3. Находится минимальное расстояние из вычисленных на предыдущем шаге, и все регионы, расстояние между которыми равно минимальному, объединяются в один регион.
4. Вычисляются расстояния между всеми парами разделенных регионов.
5. Все такие пары, расстояния между которыми оказались меньше или равными величине, вычисленной на шаге 3, объединяются в один регион.
6. Если количество регионов оказалось меньше заданной величины, то управление переходит на шаг 7, в противном случае повторяются шаги, начиная со второго.
7. Вычисляется оценочная функция текущей сегментации и прошлой сегментации(Если прошлой сегментации еще нет, то алгоритм прогоняется еще раз с шага 2 и таким образом получается вторая сегментация). Если отношение значения оценочной функции на текущей сегментации к значению на предыдущей сегментации больше указанного пользователем порога, то предыдущая сегментация является следующем уровнем в иерархии и возвращается пользователю. Затем алгоритм повторяет шаги, начиная со второго, и получает новую текущую сегментацию, для которой выполняются те же самые сравнения. Алгоритм заканчивает работу в тот момент, когда количество регионов в текущей сегментации станет меньшим или равным 2.

Рассмотрим основные понятия:

- Расстояние между регионами - это мера их непохожести. В качестве расстояния между регионами используется можно использовать любую оценку расстояния между их средними значениями. Часто используются ¿||_1, ¿||_2 или ¿_infinity.
- Оценочная функция - это эвристическая функция для оценки сегментации. Она используется для выявления существенных изменение в последовательности сегментаций. Для вычисление оценочной функции считается расстояние от каждого пикселя текущей сегментации(в качестве значения берется среднее значение региона) до соответствующего пикселя исходного изображение и берется среднее от суммы вычисленных расстояний.

Алгоритм HSEG очень требователен к ресурсам, в частности потребление памяти составляет $O(s^2)$, где s - площадь картинки,количество сравнений также составляет $O(s^2)$. Поэтому запуск алгоритма на больших изображениях затруднителен. Для преодоления этих сложностей был разработан алгоритм RHSEG и его параллельный вариант.

Существуют другие параллельные алгоритмы этого класса, описанные, например, в работе[3].

Описание алгоритма RHSEG.

Алгоритм RHSEG был создан для преодоления непомерных на больших изображениях требований алгоритма HSEG. В общем виде алгоритм RHSEG может быть описан следующим образом:

- Если уровень рекурсии не последний, то:
- Изображение делится на 4(в общем случае - n) части.
- Для каждой части рекурсивно вызывается RHSEG.
- Полученные сегментации частей собираются в одну сегментацию.
- Над полученной сегментацией вызывается HSEG без последнего шага(Т.е. алгоритм прекращает работу при достижении в сегментации указанного числа регионов) или, в случае если это первый уровень рекурсии, полный HSEG.
- иначе, если этот уровень рекурсии последний, то изображение не делится на части, и просто вызывается HSEG без последнего шага.

Описание несбалансированной версии.

В [1] была предложена параллельная реализация RHSEG, которая легла в основу разработанной реализации.

Как уже замечалось алгоритм RHSEG является рекурсивным алгоритмом. Каждый задача порождает 4 подзадачи. Таким образом при описании алгоритма можно использовать понятие "уровень рекурсии x" и "задачи на уровне рекурсии x". Уровни рекурсии нумеруются с 1. На каждом уровне рекурсии начиная со второго количество задач = 4*(количество задач на предыдущем).

Среди множества параметров предложенной реализации надо выделить следующие:

- rnblevels - общее количество уровней рекурсии. Задача, находящаяся на уровне рекурсии < rnblevels будет рекурсивно вызывать RHSEG от подчастей своей части изображения. Задача, находящаяся на уровне рекурсии == rnblevels по сути представляетс собой алгоритм HSEG.
- inblevels - Все задачи, выполняемые на уровне рекурсии < inblevels создают свои подзадачи на новом вычислительном узле. Задачи на уровне inblevels и больше выполняются на том же узле, что и их родитель. Таким образом все задачи, находящиеся на уровнях рекурсии от 1-го до inblevels находятся каждая на своем вычислительном узле.
- fnblevels - Это уровень рекурсии, на процессорах которого хранится изображение. Это связанно с тем, что размер изображения может быть велик и не помещаться в память одного вычислительного узла.
- minregions - Минимально количество регионов. При достижении количества регионов в сегментации этого числа происходит возврат обработанной части изображения родительской задаче либо начинается процесс генерации иерархии.

Таким образом предложенную реализацию можно описать следующим образом:

•Порождаются процессы находящиеся на уровнях рекурсии вплоть до fnblevels

•В память вычислительных узлов находящихся на уровне рекурсии fnblevels загружается картинка.

•Выполняется RHSEG. Выполняться он начинает с уровня рекурсии fnblevels однако возврат из рекурсии происходит вплоть до уровня 1. Следует также отметить что шаг 7 HSEG выполняется только процессом на первом уровне рекурсии.

Более детальное описание представлено в исходной работе [1].

Выделим те особенности предложенной реализации, которые негативно влияют на производительность алгоритма на системе BlueGene/P:

- Во первых следует обратить внимание на переход от шага 2 к шагу 3 в алгоритме RHSEG. Между этими двумя шагами уровне рекурсии N происходит вычисление всех порожденных этим родителем задач, т.е. в общем случае процесс на уровне рекурсии N будет ждать пока завершаться процессы на уровнях рекусрии от N+1 до inblevels. В случае если вычислительный узел находится под управлением ОС с разделением времени, то значительного простоя происходить не будет. Однако на системе BlueGene/P вычислительный узел будет простаивать существенную часть времени.
- В системе BlueGene/P одним из узких мест является дисковая подсистема ввода-вывода. Напротив, система коммуникаций между вычислительными узлами обладает высокой пропускной способностью. Поэтому более эффективным представляется не распределение изображение рекурсивно вниз начиная с малого числа узлов, а распределение изображений по всем участвующим узлам заранее .

Описание предложенной реализации

В предложенную в [1] реализацию были внесены изменения, которые можно вкратце описать следующим образом:

- Уход от рекурсии к итерационному процессу.
- Предварительное распределение изображения по всем участвующим вычислительным узлам до начала работы алгоритма.

Часть архитектурных решений вызвана тем, что реализация алгоритма была проведена в рамках разработки объемлющей библиотеки. По этому, например, алгоритм начинает работу с момента, когда изображение уже загружено на вычислительные узлы и не занимается распределением изображения самостоятельно. Такой подход, в частности, позволили абстрагироваться от того, как именно будет загружаться изображение на систему, что позволяет делать это наиболее эффективным для данной системы способом, никак не влияя на работоспособность фильтра. Учитывая особенности системы ввода/вывода BlueGene/P, это представляется особо актуальным.

В связи с предварительной загрузкой изображений на вычислительные узлы, а также с целью минимизировать время простоя вычислительных узлов, рекурсивный процесс был заменен на итеративный, в котором обработанные части изображения постепенно собираются и обрабатываются далее. Возможность на уровне пользователя регулировать параметры распределения, количество вычислительных узлов на каждом уровне рекурсии позволяет оптимизировать время простоя на каждой конкретной системе, как на системах под управлением ОС с разделением времени (кластеры) так и системах подобных BlueGene /P.

Обобщение метода.

Примененные в процессе разработки архитектурные решения не завязаны исключительно на алгоритм RHSEG и могут быть с успехом использованы в других рекурсивных алгоритмах. Однако необходимо учитывать баланс между скоростью работы коммуникационной среды и производительностью вычислительных узлов. Существует также иное решение проблемы рекурсивных алгоритмов на подобных BlueGene/P системах, которое также позволяет уменьшить время простоя вычислительного узла - когда вместо того, чтобы отдавать все n части изображения на обработку отдельным вычислительным узлам, отдаются n-1 часть, а оставшаяся одна часть обрабатывается на том же узле. Таким образом некоторые вычислительные узлы одновременно находятся на нескольких уровнях рекурсии. Однако такой метод требует больших коммуникаций между узлами(передача изображения) и неприменим в условиях когда изображение уже было загружено на вычислительные узлы.

Полученные результаты и заключение.

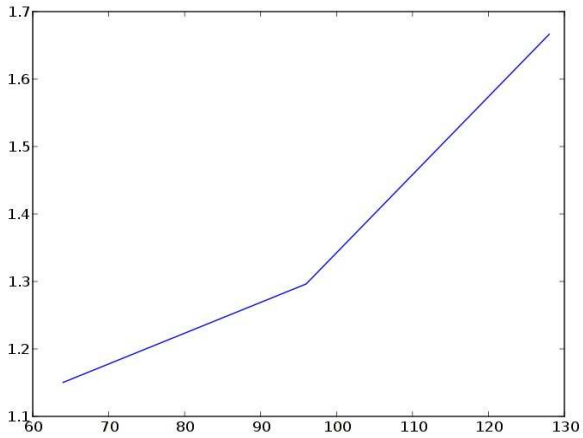

Рис. 1 Предварительные результаты тестирования
предложенного алгоритма на BlueGene /P.

На Рис. 1 приведен график тестирования предложенного алгоритма на BlueGene /P. Данный график показывает ускорение работы программы на одинаковом массиве данных по сравнению с минимально-возможным количеством вычислительных узлов, на котором такие данные могут быть обработаны за удобоваримое время. В связи с требованиями алгоритма по памяти вычисления не проводились на однопроцессорной машине, поэтому данный график не отражает классическое понимание ускорения, однако дает общее представление о динамике. В данном случае такое количество узлов равнялось 32.

Основные результаты работы:

- Проведен обзор алгоритмов HSEG и RHSEG, выделены и проанализированы их особенности, важные с точки зрения архитектуры BlueGene/P.
- Предложена реализация алгоритма RHSEG учитывающая выделенные особенности.
- Проведено предварительно тестирование на системе BlueGene/P.

Работа выполнена в рамках НОЦ «Суперкомпьютерные технологии для решения задач обработки, хранения, передачи и защиты информации» при поддержке Федеральной целевой программы "Научные и научно-педагогические кадры инновационной России" на 2009 - 2013 годы и грантов РФФИ 08-07-00445-а, 09-07-12068-офи_м.

ЛИТЕРАТУРА:

1. J.C. Tilton.Image segmentation by iterative parallel region growing and splitting. In Quantitative remote sensing: An economic tool for the Nineties; Proceedings of IGARSS '89 and Canadian Symposium on Remote Sensing, 12th, Vancouver, Canada;
2. V. Faber. Clustering and the continuous kmeans algorithm.Los Alamos Science, 1994.
3. M.N. Joshi. Parallel k-means algorithm on distributed memory multiprocessors. Technical report, University of Minnesota, 2003.
4. A. Prasad. Parallelization of k-means clustering algorithm. Technical report, Department of Electrical And Computer Engineering,University of Colorado.
5. J.M. Beaulieu. Hierarchy in picture segmentation: A stepwise optimization approach. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1989.