РАСПРЕДЕЛЕНИЕ РЕСУРСОВ В СЕТЕВЫХ СТРУКТУРАХ С АКТИВНЫМИ ЭЛЕМЕНТАМИ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЫ

М.Х. Прилуцкий, У.С. Кулакович

Нижегородский госуниверситет им. Н.И. Лобачевского

Представлена общая математическая модель функционирования сложных производственных систем, проводится её исследование. В рамках построенной математической модели поставлены оптимизационные задачи объёмно-календарного планирования, предложены алгоритмы приближенного решения поставленных задач с использованием параллельной вычислительной среды.

Целью данной работы является построение общей математической модели распределения ограниченных ресурсов в сетевых структурах с активными элементами, в рамках которой формализуется широкий класс задач производственного планирования.

Как и в [1], пусть $G = (V,A)$, $A \subseteq V^2$, – ориентированный граф без петель и контуров, моделирующий процесс изготовления изделия. Множество вершин графа V соответствует элементам системы, моделирующим процесс выполнения операций по изготовлению изделий, множество дуг A – моделирует взаимозависимость выполнения операций. Обозначим через $K(j) = \{i \mid (i,j) \in A, i \in V\}$ - множество номеров вершин графа, непосредственное предшествующих вершине с j-ым номером, $j \in V$, а через $Q(j) = \{i \mid (j,i) \in A, i \in V\}$ - множество номеров вершин графа, непосредственно следующих после вершины с j-ым номером, $j \in V$. Разбиением множества V является совокупность множеств $V^{вх}$, $V^{вых}$, V^a, V^p , где

- $V^{вх}$ – множество входных элементов (определяют «начальные» операции по изготовлению изделий), $V^{вх} = \{j \mid K(j) = \varnothing, j \in V\}$;
- $V^{вых}$ – множество выходных элементов (определяют «конечные» операции по изготовлению изделий), $V^{вых} = \{j \mid Q(j) = \varnothing, j \in V\}$;
- V^a – множество активных элементов системы – операций, выполнение которых возможно с различной интенсивностью;
- V^p – множество пассивных элементов системы, под которыми будем понимать операции, выполняемые с заданной интенсивностью.

Не уменьшая общности (за счёт введения фиктивных элементов) будем полагать, что система имеет один входной и один выходной элементы. Будем считать, что для любого активного элемента время выполнения соответствующей операции может принимать любое значение от минимально возможного до максимального возможного.

Пусть I – множество характеристик, соответствующих элементам системы (например, время выполнения операции, её ресурсоёмкость). Пусть U^j – множество допустимых управлений, соответствующих активному элементу j, где $j \in V^a$; w^j – вектор, определяющий значения на входе j-го элемента системы, $j \in V$. Известны минимальные и максимальные возможные значения каждой из характеристик i на входе и выходе j-го элемента системы, $i \in I$, $j \in V$. Известны также вектор-функция, преобразующая входные характеристики каждого элемента системы в его выходные характеристики под воздействием допустимых управлений; вектор-функция, определяющая входные характеристики j-го элемента системы по выходным характеристикам всех элементов, непосред-

ственно предшествующих элементу j, $j \in V$. Задан директивный срок окончания изготовления изделия, а также величины возможного отклонения времени завершения изготовления изделия от заданного директивного срока.

В отличие от [1], здесь предполагается, что функционирование системы осуществляется по тактам планирования и в каждый из тактов известен объём ресурса, которым располагает система. Будем считать, что в системе распределяется только один ресурс, и он является нескладируемым. Известна функция, зависящая от интенсивности выполнения операции j, определяющая объём ресурса, который необходим для выполнения j-ой операции, т.е. известен минимально-необходимый объём ресурса для выполнения j-ой операции (при минимальной интенсивности) и максимально-необходимый (при максимальной интенсивности), $j \in V^a$.

Решением проблемы распределения ресурсов в системах с активными элементами является график выполнения операций, определяющий для каждого элемента системы такт времени начала и окончания операции при выбранных интенсивностях потребления ресурсов. График операций позволяет однозначно определить объём ресурса, необходимый для выполнения каждой из операций, вектор входных и выходных характеристик каждого элемента системы и объём ресурса, используемого системой в каждый из тактов планирования (не должен превышать заданного объёма ресурса, которым располагает система в каждый из тактов). Предполагается, что время окончания «последней» операции, соответствующей «выходному» элементу системы, должно определенным образом соответствовать заданному директивному сроку окончания изготовления изделия.

Построенная общая математическая модель, формализующая процесс распределения ресурсов, учитывает ограничения характеристик на входах и выходах элементов, согласует характеристики элементов, задает условия баланса между входными и выходными характеристиками элементов, а так же формализует условия допустимости функционирования системы, связанные с ограничениями на наличие ресурса, потребляемого активными элементами.

В рамках построенной общей математической модели ставится задача оптимального распределения ресурсов по критерию минимизации отклонения времени изготовления изделия от директивного срока, и задача минимизации потерь распределяемого нескладируемого ресурса.

Показано, что поставленные оптимизационные задачи относятся к классу NP-трудных и для их решения предлагаются приближенные алгоритмы, относящиеся к классу фронтальных алгоритмов. В основе фронтальных алгоритмов лежат процедуры упорядочения множества работ путем введения приоритетов работ. Строятся совокупности таких процедур, объединенные в «стратегии управления». Для решения поставленных задач предлагается параллельная схема, основанная на выборе различных стратегий, которые меняются за счет изменения приоритетов активных элементов системы.

Работа выполнена при частичной финансовой поддержке Минобрнауки России, государственное соглашение о представлении гранта №14.B37.21.0878.

Литература

1. Прилуцкий М.Х., Дикарев К.И. Распределение ресурсов в иерархических системах с активными элементами. – Вестник Нижегородского университета, 2012, №4.

ОБ ОДНОМ ИЗ АЛГОРИТМОВ ПОИСКА ОБЪЕКТОВ ПО БИОМЕДИЦИНСКИМ ДАННЫМ

А.Н. Решетников

Нижегородский госуниверситет им. Н.И. Лобачевского
E-mail: ANReshetnikov@mail.ru

Постановка задачи

На вход рассматриваемому алгоритму подаются данные компьютерной томографии или электронной микроскопии (всюду далее – CT- и EM-данные соответственно), представляющие собой трёхмерный массив вокселов. Каждому из элементов данного массива приписано определённое целочисленное значение плотности $density[i, j, k] \in Z$. При этом известны размеры каждого воксела вдоль осей **X**, **Y** и **Z** $ScaleX$, $ScaleY$ и $ScaleZ$ в некоторых метрических единицах. Таким образом, объём 1 воксела исходных данных равен $ScaleX \cdot ScaleY \cdot ScaleZ$ кубических единиц. Отметим, что данный 3D-массив можно интерпретировать двумя способами: как трёхмерное изображение и как последовательность полутоновых 2D-изображений (слоёв) одинаковой ширины и высоты.

По представленным данным требуется найти всевозможные объекты, при отсутствии какой бы то ни было априорной информации о них. Данная задача имеет важное практическое значение как один из промежуточных этапов в трёхмерной реконструкции объектов по таким данным и их последующем анализе конечным пользователем.

Многие из алгоритмов, решающие поставленную задачу, включают в себя 2 этапа:

1) сегментацию исходных данных для получения деталей искомых объектов или их начальных приближений;
2) обнаружение самостоятельных объектов по найденным сегментам.

Сегментация исходных данных

Учитывая специфику исходных биомедицинских данных (воксельная модель представления и скалярное поле значений плотности), для решения задачи сегментации использовался ***метод k-средних*** (или ***центров тяжести***) [1].

Для более детального представления сегментов, относящихся к разным объектам, предлагается использовать один из возможных нисходящих методов иерархической сегментации. Его суть состоит в том, что после сегментации исходных данных по значениям плотности можно ввести в рассмотрение сегменты более высокого уровня детализации, а именно, ***сегменты по расстоянию*** между вокселами и ***связные области*** (в пространственном смысле). Указанные сегменты в общем случае образуют иерархическую структуру, изображённую на рис. 1. При этом сегментацию по расстоянию можно осуществить итеративным алгоритмом, основанным на методе *k*-средних, рассматривая в качестве объектов пары индексов $\{(i, j)\}$, описывающих положение вокселов на одном и том же слое. В качестве меры рассогласования между вокселами x_1 и x_2 используется обычное евклидово расстояние, порождённое парами индексов. Связные области являются самым высоким уровнем детализации исходных данных. Для их поиска ис-

пользуется итеративный алгоритм разметки областей [2] в рамках конкретного сегмента по признаку 8-связности пикселов.

В случае послойной сегментации исходных данных (каждого слоя как 2D-изображения в отдельности) для дальнейшей идентификации объектов необходимо определить, какие сегменты на разных изображениях относятся к одним и тем же объектам. Для того, чтобы судить об этом, необходимо уметь сравнивать любые 2 сегмента друг с другом.

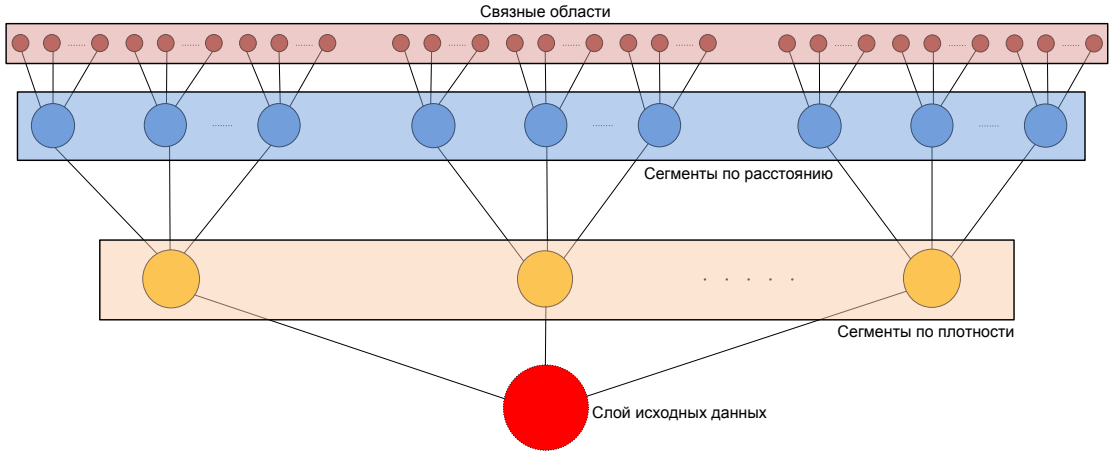

Рис. 1. Иерархическая структура сегментов после декомпозиции

Для построения дескриптора сегмента использовались следующие его атрибуты:

1) среднее значение плотности: $\bar{\mu} = \frac{1}{|S|}\sum_{i \in S} \mu_i$, (1)

где μ_i – значение плотности воксела с индексом i;

2) среднеквадратическое отклонение плотности: $\sigma_\mu = \frac{1}{|S|}\sum_{i \in S} \left(\mu_i - \bar{\mu}\right)^2$; (2)

3) геометрический центр масс $\left(\bar{x}, \bar{y}\right)$: $\bar{x} = \frac{1}{|S|}\sum_{i \in S} x_i$, $\bar{y} = \frac{1}{|S|}\sum_{i \in S} y_i$, (3)

где $\left(x_i, y_i\right)$ – координаты воксела с индексом i;

4) дисперсии координат вокселов (X и Y):

$$\sigma_X = \frac{1}{|S|}\sum_{i \in S} \left(x_i - \bar{x}\right)^2, \quad \sigma_Y = \frac{1}{|S|}\sum_{i \in S} \left(y_i - \bar{y}\right)^2; \quad (4)$$

5) объём сегмента в используемых кубических единицах:

$$V = |S| \cdot ScaleX \cdot ScaleY \cdot ScaleZ\ . \quad (5)$$

Атрибуты, вычисляемые по формулам (1) – (5), образуют признаковое описание сегмента S, или, по-другому, его ***дескриптор***, который далее будем обозначать $d(S)$.

Обнаружение объектов по найденным сегментам

Если исходные данные рассматривать как 3D-изображение, то результатом их сегментации является разбиение исходного множества вокселов на подмножества, соответствующие различным искомым объектам. Однако в том случае, когда применяется 2D-сегментация каждого слоя данных по отдельности, для идентификации объектов необходимо различить, какие сегменты на разных слоях относятся к одному и тому же или разным объектам. После этого для каждого объекта можно построить множество

точек, состоящее из соответствующих сегментов. Рассмотрим более детально метод, позволяющий решить данную проблему.

С помощью дескрипторов, которые ставятся в соответствие сегментам по формулам (1) – (5), можно определить степень рассогласования между сегментами s_1 и s_2 как расстояние ***Махаланобиса*** между их дескрипторами $d(s_1)$ и $d(s_2)$:

$$f(s_1, s_2) = \sqrt{\sum_{i=1}^{n} \frac{(d_i(s_1) - d_i(s_2))^2}{\sigma_i^2}}, \quad (6)$$

где

n – количество рассматриваемых атрибутов сегмента,

d_i – значение i - ого атрибута,

σ_i – среднеквадратическое отклонение i -го атрибута.

Определение. Последовательность сегментов на соседних слоях $(s_{i_{k_0}}, s_{i_{k_0+1}}, \ldots, s_{i_{k_0+K-1}})$, где i_j – индекс сегмента на j - м слое, k_0 – начальный слой, а K – количество слоёв, назовём ***соответствием*** между данными сегментами.

Таким образом, задача идентификации объектов сводится к поиску всевозможных соответствий между сегментами, расположенными на разных слоях исходных данных. Поиск таких соответствий можно рассматривать как многошаговый ***динамический процесс***, в котором каждый момент времени определяется номером соответствующего слоя данных, состояниями являются сегменты, распределённые по слоям, а управлениями – переходы с сегментов текущего слоя на соседний. Следует отметить, что множество финальных состояний совпадает с множеством всех предварительно найденных 2D-сегментов, поскольку объект может начинаться и заканчиваться на любом слое исходных данных.

В процессе поиска соответствий рассматриваются сегменты самого верхнего уровня детализации, то есть листья иерархической структуры, описанной в разделе «Сегментация исходных данных». Все сегменты делятся на ***занятые*** (те, которые уже включены в одно из найденных соответствий) и ***свободные***. В самом начале работы алгоритма все сегменты помечаются как свободные. Как только сегмент оказался включён в какое-либо соответствие, он получает статус «занят». Поиск очередного соответствия начинается с первого свободного сегмента на слое с наименьшим индексом. Для того чтобы не рассматривать слишком маленькие и/или большие детали, полученные в результате иерархической декомпозиции сегментов, можно ввести ограничения на их объём *V*, который должен принадлежать диапазону $[V_{\min}; V_{\max}]$.

Определим стоимость перехода с текущего слоя на соседний как рассогласование между дескрипторами соответствующих сегментов (состояний), тогда значение функции $f(s_1, s_2)$ вычисляется по формуле (6), где s_1 – сегмент текущего слоя, а s_2 – сегмент соседнего слоя.

Тогда критерием качества найденного соответствия можно считать суммарное рассогласование между всеми соседними сегментами, образующими соответствие:

$$F = \sum_{k=k_0}^{k_0+K-2} f(s_{i_k}, s_{i_{k+1}}). \quad (7)$$

Таким образом, поиск соответствий между сегментами сводится к решению задачи минимизации функции F при перемещении от начального слоя к конечному по соответствующим сегментам (рис. 2).

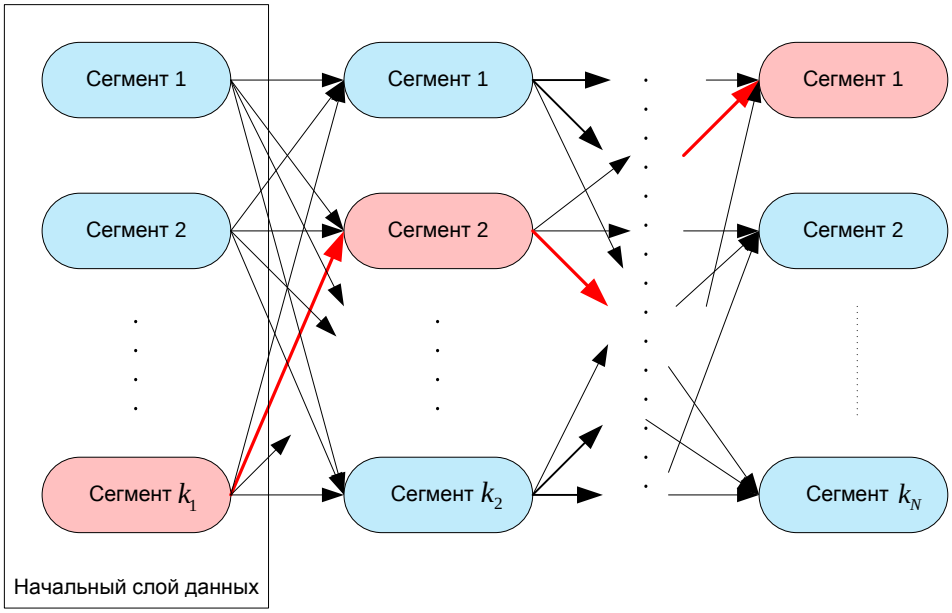

Рис. 2. Схема поиска соответствий между 2D-сегментами

Поскольку функция F аддитивна по промежутку слоёв, на которых расположены сегменты, и удовлетворяет принципу Беллмана в форме достаточного условия [3], для решения рассматриваемой задачи можно применить алгоритм пошаговой минимизации этой функции, основанный на методе рекуррентных соотношений Беллмана [3].

Приведём более подробное описание данного алгоритма применительно к рассматриваемой задаче, используя следующие обозначения:

s – массив сегментов,
IsSegmentUsed – булевы переменные (флаги) использования сегментов,
layer_index – индекс текущего слоя данных (служебная переменная),
Disparity – текущее значение минимальной стоимости переходов по соответствию,
N – общее количество слоёв исходных данных.

Шаг 1. Полагаем для всех сегментов **IsSegmentUsed = false**.
Шаг 2. В цикле по **I** от 0 до **N-1** и по сегментам **s[I][J]**, принадлежащим слою **I**, если ***IsSegmentUsed[I][J] == false***, выполняем ***Шаг 3 – Шаг 5***.
Шаг 3. Полагаем **layer_index = I, Disparity = 0**.
Шаг 4. Создаём новое соответствие и добавляем в него сегмент с индексом **J** на слое **layer_index**.
Шаг 5. Пока **layer_index<N**, выполняем ***Шаг 6 – Шаг 10***. После этого переходим на ***Шаг 2***.
Шаг 6. В цикле по **K** по сегментам слоя **layer_index + 1** выбираем тот, объём которого принадлежит заданному диапазону $[V_{\min}; V_{\max}]$ и который доставляет минимум величины **Disparity + *F(s[layer_index][J], s[layer_index+1][K])***. Обозначим его **L**.
Шаг 7. Добавляем сегмент с индексом **L** в созданное соответствие.
Шаг 8. Полагаем **Disparity = Disparity + F(s[layer_index][J], s[layer_index+1][L])**.
Шаг 9. IsSegmentUsed[layer_index+1][L] = true.
Шаг 10. Увеличиваем **layer_index** на 1 и переходим на ***Шаг 5***.

Конец алгоритма

На рис. 3 в верхнем ряду показаны несколько последовательно расположенных слоёв одной из томограмм, а в нижнем – результаты сегментации каждого из слоёв. При этом одинаковыми цветами раскрашены сегменты, образующие один и тот же объект.

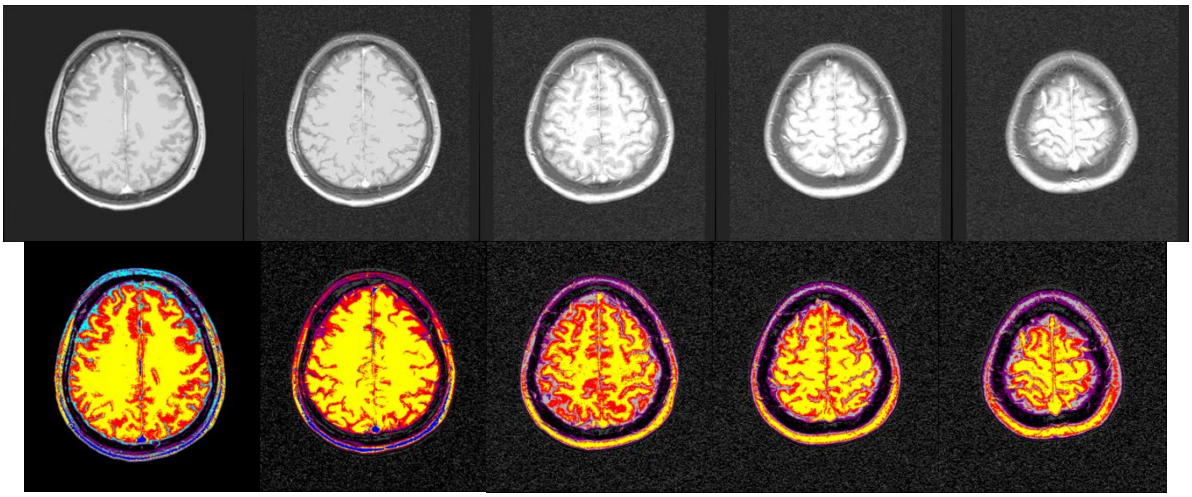

Рис. 3. Пример результатов поиска соответствий между 2D-сегментами

Повышение производительности вычислений

Вышеописанный алгоритм использовался при разработке программной системы Neocortex. В ходе этой разработки были реализованы как последовательная, так и параллельная (для GPU, с использованием технологии OpenCL) версии сегментации методом *k*-средних для решения поставленной задачи, которые используются для получения сегментов первого и второго уровней декомпозиции. При этом на каждом из этих двух этапов декомпозиции было получено ускорение от 4 до 10 раз в сравнении с последовательными реализациями [4].

Дальнейшее повышение производительности связано с разработкой параллельной схемы разметки связных областей (третий уровень декомпозиции сегментов) и алгоритма поиска соответствий между сегментами. Для этого целесообразно использовать возможности как центрального, так и графических процессоров.

Литература

1. Lloyd S. Least square quantization in PCM’s. Bell Telephone Laboratories Paper, 1957.
2. Форсайт Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 928 с.
3. Беллман Р. Динамическое программирование. – М.: Изд-во иностранной литературы, 1960.
4. Решетников А.Н. Параллельная реализация сегментации компьютерных томограмм методом центров тяжести с использованием технологии OpenCL // Высокопроизводительные параллельные вычисления на кластерных системах: Доклад на XI Всероссийской конференции, Н. Новгород, 1-3 ноября 2011. – С. 258–261.