

И. М. ГОСТЕВ, А. Г. ПОДГОРБУНСКИЙ

## РАСПРЕДЕЛЕНИЕ ПОТОКОВ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ В СИСТЕМАХ РАСПОЗНАВАНИЯ ОБРАЗОВ В РЕАЛЬНОМ МАСШТАБЕ ВРЕМЕНИ

Рассматриваются два алгоритма управления видеопотоками в распределенной системе по обработке изображений и распознаванию образов на основе теории графов. Представлен алгоритм управления распределенной системой посредством контроллера потока. Приведено математическое описание статического (граф определен до начала работы системы) и динамического (граф может быть перестроен в процессе работы системы в зависимости от особенностей видеопотока) распределения информационных потоков.

**Ключевые слова:** алгоритм управления, графическая информация, распознавание объектов, реальный масштаб времени.

**Введение.** Несмотря на постоянное повышение быстродействия компьютеров построение систем обработки изображений и распознавания образов в реальном масштабе времени сопряжено с преодолением ряда проблем. В работе [1] показано, что быстродействие вычислительной системы для обработки видеопотока должно быть пропорциональным размеру изображения, для телевизионного формата оно превышает мощность одного компьютера примерно в 25—30 раз. Для решения этой проблемы были разработаны параллельные алгоритмы, позволяющие реализовать такую систему в распределенной среде, при этом каждый из элементов системы выполняет свою операцию по обработке изображения и распознавания образов. Для управления всей системой обработки видеопотока и распознавания объектов используется контроллер, который распределяет задания по вычислительным узлам системы, осуществляет контроль и синхронизацию их выполнения.

В работах [1, 2] была предложена и развита идея построения распределенной системы обработки и распознавания изображений в режиме реального времени, основанной на принципах конвейера и разбиения потока на участки. В настоящей работе рассматривается способ моделирования параллельных вычислений, основанный на применении ориентированных графов [3].

**Математическая модель процессов.** Разделим методы распределения потока на статические и динамические. В первом случае маршрут потоков информации определяется на этапе начальной инициализации системы и далее остается неизменным; размер фрагмента потока, его маршрут и методы обработки заранее определены. При динамическом распределении маршрут и размеры фрагментов определяются на стадии обработки.

Рассмотрим односвязный ориентированный ациклический граф, имеющий один исток и один сток, вершинами которого являются узлы вычислительной сети, реализованные в виде экземпляров метода, а дугами — каналы передачи потока. Методом назовем процедуру обработки изображения, реализованную в виде библиотечной функции. Входными данными метода будут фрагменты изображения и его параметры, необходимые для процедуры обработки. В результате работы метода получаются выходное изображение и/или некоторые другие данные, полученные в результате применения процедуры, формат которых зависит от конкретного метода. С помощью методов осуществляются разделение исходного потока на фрагменты, их параллельная обработка и последующее объединение нескольких потоков в один — результирующий. Всем процессом управляет компонент, называемый „контроллером потока“.

Поскольку процесс обработки изображения представляет собой последовательность (цепочку) нескольких методов, то в вычислительной сети каждому методу будет сопоставлен один или несколько узлов, его реализующих — экземпляров. Совокупность одинаковых экземпляров образует *ярус* сети, соответствующий данному методу цепочки. Входная информация поступает на некоторый узел только с узлов яруса, соответствующего предыдущему методу цепочки. Все выходы узла направлены только на узлы яруса, соответствующего следующему методу в цепочке. Первый и последний ярусы содержат по одному узлу. Каждый метод характеризуется относительной вычислительной сложностью процедуры, началом и концом смещения фрагмента изображения по высоте  $(y^-, y^+)$  и начальными условиями. Например, смещение по высоте  $(y^-, y^+)$  означает, что для метода  $l$  с параметрами начала и конца фрагмента  $[s_1, s_2]$  необходим фрагмент  $l[s_1 - y^-, s_2 + y^+]$ .

**Алгоритм построения вычислительной среды.** *Статическое распределение потока.* Рассмотрим алгоритм статического распределения потока. Пусть исходный поток представляет собой одно длинное изображение — „ленту“. Фрагмент потока — непрерывный участок изображения, содержащий целое число строк. Промежуточные потоки состоят из последовательности фрагментов. Если поток состоит из фрагментов одинаковой длины, следующих через равные промежутки, то он может быть описан схемой потока.

Схема фрагмента потока представляет собой сочетание  $\langle S, L, P \rangle$  где  $S$  — начальная строка первого фрагмента,  $L$  — длина каждого фрагмента в строках,  $P$  — промежуток от конца предыдущего фрагмента до начала следующего. Таким образом,  $\langle S, L, P \rangle$  определяет последовательность фрагментов

$$[S + (L + P)i, S + (L + P)i + L], i \in N.$$

Две схемы называются *смежными*, когда при объединении двух соседних фрагментов их описание (состоящее из всех фрагментов первого и второго потока) также определяется схемой потока. Результирующая схема потока носит название *объединенной*. Символом  $\Xi_i(\langle S, L, P \rangle)$  обозначается схема потока, скорректированная в соответствии со смещениями по  $y$  для  $i$ -го метода (необходимого методу на входе, чтобы производить поток  $\langle S, L, P \rangle$  на выходе):

$$\Xi_i(\langle S, L, P \rangle) = \langle S - y_i^-, L + y_i^- + y_i^+, P - y_i^- - y_i^+ \rangle.$$

Алгоритм состоит из двух шагов — прямого и обратного. На первом происходят определение количества вершин в каждом ярусе и построение связей между ними путем равномерного последовательного разделения потока. На втором шаге происходит коррекция соединений и разметки с целью исключения каналов, по которым передаются очень малые относительно среднего потока на ярусе фрагменты.

**Входом алгоритма** являются цепочка методов  $\langle M_1, \dots, M_N \rangle$ , относительная вычислительная сложность  $c_1, \dots, c_N$ , пропускная способность эталонного метода  $q_e$ , требуемая пропускная способность цепочки  $q$ ; **выходом** — построенный граф вычислительной сети  $G = \{V, E\}$ , ребра и вершины которого размечены схемами потоков  $M \rightarrow \{\langle S, L, P \rangle\}$ ,  $M' \rightarrow \{\langle S, L, P \rangle\}$ .

**Шаг 1.** Для каждого  $i \in [1, N]$  определить ярус  $G_i = \{V_i, \emptyset\}$ , где

$$V_i = \left\{ v_i^k \mid k = 1 \dots \left\lceil \frac{q_e c}{q} \right\rceil \right\}$$

— набор упорядоченных узлов методов, являющихся экземплярами метода  $M_i$ .

Положить  $V = \bigcup_{i \in [1, n]} V_i$ ;  $E = \emptyset$ . Приписать истоку схему потока  $\langle 0, L, 0 \rangle$ , где  $L$  — исходная длина блока, задаваемая контроллером потока на этапе конфигурации. Положить  $M = \emptyset, M' = \left\{ (v_1^1, \langle 0, L, 0 \rangle) \right\}$ , для каждого узла  $v_i^k$  разметить его следующим образом:

$$\left\langle \left\lfloor \frac{k-1}{|V_i|} \right\rfloor, \left\lfloor \frac{k}{|V_i|} \right\rfloor - \left\lfloor \frac{k-1}{|V_i|} \right\rfloor, L - \left\lfloor \frac{k}{|V_i|} \right\rfloor + \left\lfloor \frac{k-1}{|V_i|} \right\rfloor \right\rangle.$$

Для каждой пары узлов  $v_i^k$  и  $v_{i+1}^{k'}$ ,  $i \in [1, N-1]$ ,  $k \in [1, |V_i|]$ ,  $k' \in [1, |V_{i+1}|]$  в случае если  $M'(v_i^k) \cap \Xi_i(M'(v_{i+1}^{k'})) \neq \emptyset$ , добавить дугу между узлами  $E = E \cup (v_i^k, v_{i+1}^{k'})$  и пометить ее схемой-пересечением:

$$M = M \cup \left\{ (v_i^k, v_{i+1}^{k'}), M'(v_i^k) \cap \Xi_i(M'(v_{i+1}^{k'})) \right\}.$$

**Шаг 2.** Для каждого яруса  $V_i$ ,  $i = N, \dots, 2$  для каждого узла  $v_i^k \in V_i$ ,  $k \in [1, |V_i|]$ .

**а)** Убедиться, что объединенная схема выходов совпадает со схемой узла:

$$\bigcup_{(v_i^k, v) \in E} \left\{ M(v_i^k, v) \right\} = M'(v_i^k).$$

В случае, если это не так и объединенная схема включает схему узла, то вычислить множество недостающих схем

$$\bigcup_{(v_i^k, v) \in E} \left\{ M(v_i^k, v) \right\} \setminus \Xi_i(M'(v_i^k))$$

и каждый недостающий элемент этого множества прибавить к одной из схем входящих ребер узла, имеющих схему, смежную с недостающей, и наибольшую длину.

**б)** Удалить входные дуги, длина схемы которых меньше определенного порога, и схемы этих дуг приписать дугам наибольшей длины, имеющим смежные схемы.

*Динамическое распределение потока.* При динамическом распределении потока между обрабатываемыми узлами маршрут, по которому пройдет конкретный фрагмент изображения, определяется на стадии выполнения. При этом из-за смещений по высоте для обработки данного фрагмента может потребоваться несколько строк предыдущего или последующего фрагмента, непосредственно прилегающего к данному. А так как заранее неизвестно, какие именно фрагменты пройдут через данный экземпляр метода, невозможно сказать, требуется сохранять этот фрагмент или нет. Кроме того, чтобы обеспечить обработку данного фрагмента с учетом смещений по высоте, требуется разместить предыдущий и последующий фрагменты непосредственно до и после него. Указанные факторы усложняют процедуру управления памятью и требуют реализации дополнительных условий маршрутизации на этапе обработки потока, включающих взаимодействие с контроллером потока, который координирует в этом случае маршрутизацию. Это противоречит одной из основных концепций, положенных в основу проектирования системы, согласно которой взаимодействие контроллера потока с методами на этапе обработки должно быть минимальным. Однако обе проблемы могут быть решены с помощью передачи избыточного количества фрагментов, так чтобы каждый фрагмент на каждом ярусе сети мог быть обработан независимо. В этом случае освободить память, занимаемую фрагментом, можно сразу после его обработки. Единственным и очевидным недостатком такого метода является повышение общего объема пересылаемой по сети

информации, однако при больших значениях длины пересылаемых фрагментов и небольшом количестве ярусов эти издержки сравнительно малы.

Приведем алгоритм построения сети с динамическим избыточным распределением потока. При этом в отличие от статического метода, где схема потока, приписанная конкретному узлу или дуге сети, определяет, какие именно фрагменты изображения и в каком порядке пройдут через данную дугу или узел, эта схема определяет, какие фрагменты могут пройти через узел или дугу. Все узлы одного яруса являются равнозначными в том смысле, что любой узел предыдущего яруса может передать обработанный фрагмент любому из узлов следующего яруса.

**На вход** алгоритма поступает цепочка методов  $\langle M_1, \dots, M_N \rangle$  с относительной вычислительной сложностью  $c_1, \dots, c_N$ , пропускной способностью эталонного метода  $q_e$ , требуемой пропускной способностью цепочки  $q$ . **На выходе** получаем построенный граф вычислительной сети  $G = \{V, E\}$ , ребра и вершины которого размечены схемами потоков  $M \rightarrow \{\langle S, L, P \rangle\}$ ,  $M' \rightarrow \{\langle S, L, P \rangle\}$ .

Для каждого  $i \in [1, N]$  определить ярус  $G_i = \{V_i, \emptyset\}$ .

Положить  $V = \bigcup_{i \in [1, n]} V_i$ ;  $E = \emptyset$ . Приписать стоку схему потока  $\langle 0, L, 0 \rangle$ , где  $L$  — исходная длина блока, задаваемая контроллером потока на этапе конфигурации. Положить разметку  $M = \emptyset, M' = \{\langle v_N^1, \langle 0, L, 0 \rangle \rangle\}$ .

Для каждого яруса  $i \in [N - 1, 1]$  разметить все его вершины схемами

$$\left\langle -\sum_{j=i}^N y_j^-, L + \sum_{j=i}^N (y_j^- + y_j^+), -\sum_{j=i}^N (y_j^- + y_j^+) \right\rangle.$$

Для каждой пары узлов  $v_i^k$  и  $v_{i+1}^{k'}$ ,  $i \in [1, N - 1]$ ,  $k \in [1, |V_i|]$ ,  $k' \in [1, |V_{i+1}|]$  добавить дугу между ними

$$E = E \cup (v_i^k, v_{i+1}^{k'})$$

и пометить ее схемой  $\langle S, L, P \rangle$ :

$$\left\langle -\sum_{j=i+1}^N y_j^-, L + \sum_{j=i+1}^N (y_j^- + y_j^+), -\sum_{j=i+1}^N (y_j^- + y_j^+) \right\rangle.$$

В построенной таким образом сети каждый узел каждого яруса соединен дугой с каждым узлом последующего яруса. При большом количестве узлов в соседних ярусах это может привести к значительной дополнительной нагрузке на сетевые ресурсы вычислительной системы (каждая дуга соответствует двум дополнительным потокам в процессах отправляющего и принимающего узлов). Для снижения нагрузки при вычислении множества дуг можно использовать первый алгоритм, а для разметки дуг и узлов — второй. Тогда каждый узел может послать обработанный фрагмент любому из узлов следующего яруса, и простейшим способом маршрутизации обработанных фрагментов из данного узла станет чередующаяся рассылка фрагментов последовательно каждому из узлов следующего яруса. При этом будет достигаться равномерное распределение загрузки между всеми узлами каждого яруса в предположении, что каждый из них обладает равным количеством ресурсов.

**Обсуждение и выводы.** Подводя итог, можно выделить следующие преимущества и недостатки статического и динамического способа распределения потока.

Распределенная система обладает высокой отказоустойчивостью — при выходе из строя одного из узлов сети его функции перераспределяются между остальными узлами того же яруса. Поэтому при динамическом распределении потоков неисправный узел будет заменен. При статическом методе распределения отказ узла приведет к появлению в выходном изображении необработанных участков.

Динамическое распределение дает возможность более гибкой балансировки загрузки узлов вычислительной сети. Если какой-то узел перегружен, его работу можно распределить на другие узлы того же яруса. Статическое распределение не позволяет этого делать. При динамическом распределении суммарный объем пересылаемых данных возрастает вследствие избыточной передачи строк, появляется дополнительная нагрузка на узлы, связанная с маршрутизацией и балансировкой исходящих фрагментов. По сравнению со статическим методом возрастают число потоков приема и передачи фрагментов и число сетевых соединений.

В целом преимущества динамического метода превосходят его недостатки.

### СПИСОК ЛИТЕРАТУРЫ

1. *Gostev I. M., Sevastianov L. A.* About construction distributed calculation in real time system of image processing and pattern recognition // Proc. Int. Conf. „Distributed computation and GRID in science and education“. Dubna JINR. June, 2004. P. 79—85.
2. *Gostev I. M., Sevastianov L. A.* Internal Architecture of Distributed Real-time System of Image Processing and Pattern Recognition // Proc. XX Int. Symp. on Nuclear Electronics & Computing (NEC 2005). Varna, Bulgaria. September, 2005.
3. *Мелихов А. Н.* Ориентированные графы и конечные автоматы. М.: Наука, 1971. 416 с.

#### *Сведения об авторах*

**Иван Михайлович Гостев**

— д-р техн. наук; доцент; Московский государственный институт электроники и математики (Технический университет), кафедра кибернетики; E-mail: igostev@gmail.com

**Александр Глебович Подгорбунский**

— аспирант; Московский государственный институт электроники и математики (Технический университет), кафедра кибернетики; E-mail: shurikp@gmail.com

Рекомендована кафедрой  
вычислительной техники

Поступила в редакцию  
12.09.08 г.