ИСПОЛЬЗОВАНИЕ СПЕЦИАЛИЗИРОВАННОЙ ЭВМ ДЛЯ РЕШЕНИЯ ЗАДАЧИ КОНТЕКСТНОГО ПОИСКА ИЗОБРАЖЕНИЙ

Е.А.Башков, Н.С. Шозда
Кафедра ПМиИ, ДонГТУ
Shozda@r5.dgtu.donetsk.ua

Abstract

Evgeny A. Bashkov , Natalia S. Shozda. Using of special- purpose computer for content-based color image retrieval. This article deals with the problem of parallel computing for different stages in content- based color image retrieval. Estimation of acceleration for parallel algorithms and comparison of time complexity are given.

Введение

Часто задача контекстного поиска изображений требует решения в режиме реального времени, поэтому при разработке такого рода систем требуется решение задач алгоритмического и аппаратного ускорения. Эффективным способом повышения производительности вычислительных систем является разработка новых параллельных архитектур. Часто вместо последовательных алгоритмов используются параллельные, реализуемые на многопроцессорных ЭВМ. В работе рассматриваются вопросы повышения производительности процесса контекстного поиска изображений, предполагающего архитектурную поддержку: решаются задачи сравнения цветовых гистограмм изображений и сортировки изображений путем распараллеливания вычислений с использованием многопроцессорной ЭВМ SIMD- архитектуры, анализируется временная сложность последовательных и параллельных алгоритмов.

1. Поиск изображений по цветовым характеристикам

Рассматриваемая задача сводится к выполнению следующих шагов:

1) Вычисление расстояний d_i между цветовой гистограммой изображения- образца и гистограммами всех изображений в базе данных (БД);
2) Сортировка полученных значений d_i по возрастанию;
3) Визуализация изображений, соответствующих первым M значениям упорядоченной последовательности.

Для решения задачи контекстного поиска изображений предлагается использовать специализированную вычислительную систему (рисунок 1). В БД хранятся как сами изображения, так и вычисленные для них на этапе заполнения БД цветовые гистограммы. Под управлением HOST- компьютера выполняется ввод образца поиска (это может быть новое изображение, созданный пользователем эскиз либо изображение из БД) и вычисление для него цветовой гистограммы (при использовании образца- изображения из БД его цветовая гистограмма извлекается из БД). После этого в подсистеме сравнения вычисляются расстояния между гистограммами изображений из БД и гистограммой образца. Через буферную память (БП) вычисленные значения передаются в подсистему сортировки, где определяются изображения, наиболее близкие по своему содержимому к изображению- образцу, после чего выполняется визуализация выбранных изображений. Проведенные исследования основаны на идеальной модели, предполагающей, что подсистемы сравнения и сортировки представляют собой процессоры SIMD- архитектуры, в которых каждый процессорный элемент (ПЭ) имеет локальную память, может

выполнить любую арифметическую операцию за один такт; временные затраты, связанные с обращением к запоминающему устройству, отсутствуют.

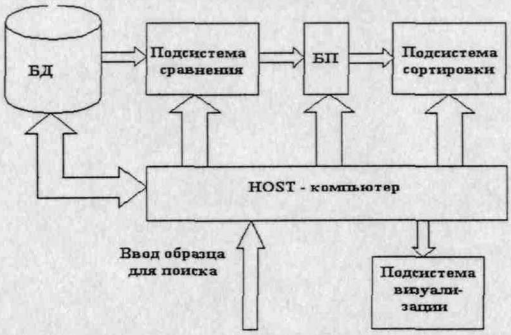

Рис. 1 Специализированная вычислительная система для решения задачи контекстного поиска изображений

2. *Подсистема сравнения цветовых гистограмм*

Цветовая гистограмма изображения представляет собой распределение цветов в трехмерном пространстве [1], в частности, для пространства RGB элементы гистограммы вычисляются таким образом:

$$H_{R,G,B}[r,g,b] = \frac{c(R = r, G = g, B = b)}{P} \qquad (1)$$

где $c\{R = r, G = g, B = b\}$-количество таких точек в изображении, для которых цвет определяется составляющими *r, g, b, P*- общее количество точек в изображении. Очевидно, что такая гистограмма является нормализованной, т.е. $|h| = \sum_{i=1}^{K1}\sum_{j=1}^{K2}\sum_{k=1}^{K3} h[i,j,k] = 1$, где *K1, K2, K3* -число уровней дискретизации цветового пространства соответственно по осям *R, G, B*. В дальнейшем предполагается, что *K1=K2=K3=K*. Для сравнения цветовых гистограмм вычисляют расстояние между ними в пространстве гистограмм по одной из метрик [1], приведенных в таблице 1.

Таблица 1 Метрики для вычисления расстояния в пространстве цветовых гистограмм

Обозначение	Название	Формула для ненормализованных гистограмм	Формула для ненормализованных гистограмм
D1	**Конъюнкция гистограмм**	$D1 = 1 - \frac{\sum_{i=1}^{K1}\sum_{j=1}^{K2}\sum_{k=1}^{K3} \min(h_1[j,j,k], h_2[i,j,k])}{\min(\lvert h_1\rvert, \lvert h_2\rvert)}$	$D1 = \sum_{i=1}^{K1}\sum_{j=1}^{K2}\sum_{k=1}^{K3} \lvert h_1[i,j,k] - h_2[i,j,k]\rvert$
D2	**Евклидово расстояние**	$D2 = \sum_{i=1}^{K1}\sum_{j=1}^{K2}\sum_{k=1}^{K3} (h_1[i,j,k] - h_2[i,j,k])^2$	$D2 = \sum_{i=1}^{K1}\sum_{j=1}^{K2}\sum_{k=1}^{K3} (h_1[i,j,k] - h_2[i,j,k])^2$
D3	**Косинусное расстояние**	$D3 = 1 - \cos\theta = 1 - \frac{\lvert h_1\rvert + \lvert h_2\rvert - D2}{2\lvert h_1\rvert \cdot \lvert h_2\rvert}$	$D3 = 1 - \cos\theta = \frac{D2}{2}$
D4	**Квадратичное расстояние**	$D4 = (h_1 - h_2)^T \cdot A \cdot (h_1 - h_2)$	$D4 = (h_1 - h_2)^T \cdot A \cdot (h_1 - h_2)$

Подсистема сравнения в предложенной вычислительной системе реализована в виде SIMD- процессора с кубической сеткой, содержащей K^3 процессорных элементов. Параллельные алгоритмы вычисления расстояний между гистограммами по метрикам из таблицы 1, а также оценка временных затрат рассмотрены в [2]. В таблице 2 приведены формулы для определения временных затрат и вычисления ускорения[3] соответствующих алгоритмов.

Таблица 2 Временные затраты и ускорение параллельных алгоритмов для вычисления расстояний между цветовыми гистограммами

Метрика	Временные затраты на вычисление с использованием последовательной ЭВМ	Временные затраты на параллельное вычисление	Ускорение параллельного алгоритма
D1,D2, D3	$T_2 = 2 \cdot K^3 \cdot t_{op}$	$T_1 = +3\log_2 K + 3 \cdot (K-1)/S) \cdot t_{op}$	$E = \frac{2K^2}{(3/S) \cdot \log_2 K}$
D4	$T_2 = (2 \cdot K^6 + 2 \cdot K^3 - 1)t_{op}$	$T1 = ((2+1/S) \cdot (K^3 - 1) + 3 + \log_2 K^3) \cdot t_{op}$	$E = \frac{2K^6 + 2K^3 - 1}{(2+1/S) \cdot (K^3 - 1) + 3 + \log_2 K^3}$

В приведенных формулах константа S характеризует, во сколько раз быстрее выполняется операция сдвига по сравнению с арифметическими операциями. Наиболее трудоемким является вычисление расстояния по метрике *D4* (квадратичное расстояние). На рисунке 2 показана зависимость временных затрат на вычисление этой величины на ЭВМ последовательной архитектуры и на предлагаемом SIMD-процессоре от количества цветов в изображении. Аналогичные зависимости могут быть построены и для вычисления расстояния по другим метрикам [2].

3. Подсистема сортировки

Функции подсистемы сортировки заключаются в определении *M* изображений, наиболее сходных с образцом (или, что то же самое, в выборе из полученной от подсистемы сравнения последовательности расстояний $D=(d_1, d_2, ..., d_n)$ *M* элементов с минимальными значениями). С этой целью предлагается использовать параллельный

Рис. 2 Зависимость временной сложности вычисления квадратичного расстояния между гистограммами от числа цветов в изображении

алгоритм, базирующийся на алгоритме линейного выбора с подсчетом [4]. Суть алгоритма в том, что для каждого элемента определяется количество элементов, меньших рассматриваемого; и это число характеризует положение элемента в

упорядоченной последовательности. При этом для последовательности из n элементов используется n просмотров. На первом просмотре выполняется $(n-1)$ операция сравнения, на втором- $(n-2)$ операции и т.д. Общее число сравнений при последовательном выполнении такого алгоритма составит

$$t_1 = (n-1)+(n-2)+(n-3)+\ldots+1 = \frac{(n-1)+1}{2}\cdot(n-1) = \frac{n^2-n}{2} \qquad (2)$$

После каждого сравнения выполняется сложение, поэтому общее время выполнения последовательного алгоритма составит

$$T_0 = (n^2-n)\cdot t_{op} \qquad (3)$$

Параллельная реализация этого алгоритма предполагает, что подсистема сортировки представляет собой SIMD- процессор с линейкой процессорных элементов, каждый из которых связан с двумя соседними.

Рассмотрим параллельный алгоритм, предполагающий, что подсистема сортировки состоит из n ПЭ (n- число упорядочиваемых элементов последовательности $D=(d_1, d_2, \ldots, d_n)$), и в локальной памяти ПЭ имеются ячейки для хранения сравниваемых элементов и счетчика сравнений. Первоначально в i-й ПЭ загружаются значения d_i и d_{i+1} (в ПЭ с номером n загружаются элементы d_1 и d_{n+1}), ячейки, соответствующие счетчикам, обнуляются. После этого в каждом ПЭ выполняется сравнение элементов d_i и d_{i+1}, и, если первое значение больше, то наращивается счетчик. Затем выполняется циклический сдвиг влево значений, находящихся в ячейке 2. Такая последовательность сравнений и сдвигов повторяется n раз, после чего ячейка-счетчик каждого ПЭ содержит номер соответсвующего значения d_i в упорядоченной по возрастанию последовательности. После этого остается только визуализировать те изображения, для которых счетчик не превышает M.

Для описанного алгоритма выполняется n операций сдвига, $(n-1)$ операция сравнения и столько же операций сложения. Время выполнения алгоритма составит

$$T_1 = n\cdot t_{shift} + 2\cdot(n-1)\cdot t_{op} = \left(\frac{n}{S}+2n-2\right)t_{op} \qquad (4)$$

Ускорение параллельного алгоритма равно

$$E = \frac{n^2-n}{n/S+2n-2}. \qquad (5)$$

Несмотря на хорошие характеристики, этот алгоритм нереализуем в задаче контекстного поиска изображений, поскольку число изображений в БД практически неограничено; необходимо модифицировать рассмотренный алгоритм для случая фиксированного числа процессорных элементов P, $P<n$, т. е. необходимо разместить n виртуальных ПЭ на P реальных. При этом коэффициент виртуализации [5] составит

$$K_v = (n \operatorname{div} P)+1 \qquad (6)$$

Первая модификация алгоритма предполагает, что в локальную память произвольного i-го ПЭ $(i=1..P)$ загружаются следующие значения:

- массив $D=(d_1, d_2, \ldots, d_n)$;
- значение $a=D[i]$;
- счетчик $c=0$.

После этого в каждом ПЭ выполняется такая последовательность действий:

```
For j:=1 to n do
  If a>D[j] then c:=c+1;
```

При этом выполняется n операций наращивания счетчика цикла, n сравнений этого счетчика с границей цикла, n сравнений и n сложений внутри цикла, и все эти действия повторяются K_v раз. Суммарные временные затраты составят

$$T_2^1 = 4\cdot K_v\cdot n\cdot t_{op}, \qquad (7)$$

а ускорение параллельного алгоритма может быть вычислено по формуле

$$E = \frac{n^2 - n}{4 \cdot K_v \cdot n} = \frac{n-1}{4 \cdot K_v}. \quad (8)$$

Вторая модификация алгоритма заключается в следующем. В локальную память произвольного *i*-го ПЭ *(i=1..P)* загружаются:

- значение *b= D[i+1]*;
- значение *a=D[i]*;
- счетчик *c=0*.

После этого в каждом ПЭ выполняется такая последовательность действий:

```
If a>b then  c:=c+1;
```

Затем выполняется сдвиг влево на одну позицию значений, находящихся в ячейке *b*, а ПЭ с номером *P* считывает из буферной памяти следующее по порядку значение *b*. Таким образом, на каждом шаге выполняется одна операция сравнения, одна операция сложения и одна операция сдвига. С учетом коэффициента виртуализации суммарные временные затраты составят

$$T_2^2 = K_v \cdot n \cdot (2 \cdot t_{op} + t_{shift}) = K_v \cdot n \cdot \frac{2 \cdot S + 1}{S} \cdot t_{op}, \quad (9)$$

а ускорение параллельного алгоритма равно

$$E = \frac{n^2 - n}{K_v \cdot n \cdot \frac{2 \cdot S + 1}{S}} = \frac{n-1}{K_v \cdot \frac{2 \cdot S + 1}{S}}. \quad (10)$$

Учитывая, что *S*- целочисленная константа, *S>1*, можно сказать, что с точки зрения ускорения две рассмотренные модификации практически равносильны. Поэтому все рассуждения, приведенные ниже, справедливы для обоих рассмотренных алгоритмов. На рис. 3 показана зависимость числа выполняемых операций от количества ПЭ в подсистеме сортировки для БД, содержащей 1 млн. изображений. Из графика видно, что временные затраты рассматриваемого алгоритма сверху ограничены временем выполнения последовательного алгоритма, а снизу- временем выполнения алгоритма на линейке с неограниченным числом ПЭ.

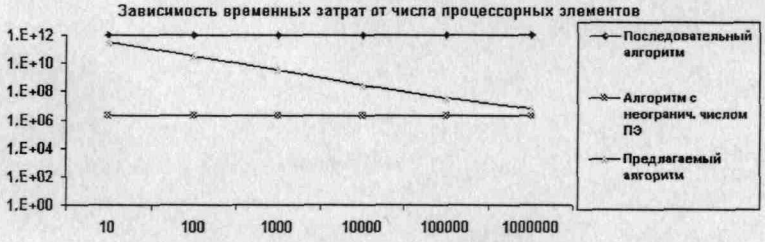

Рис. 3 Сравнение зависимостей временных затрат от количества процессорных элементов для БД из 1 млн. изображений

Изображенная на рис. 4 зависимость времени выполнения алгоритма от числа изображений в БД (количество ПЭ равно 1000) показывает рациональность распараллеливания вычислений.

4. Оценка совместного времени функционирования подсистем сравнения и сортировки

Для оценки совместного времени функционирования подсистем сравнения и сортировки необходимо просуммировать временные затраты подсистем. На рисунке 5 показана зависимость временных затрат от числа изображений в БД (в качестве метрики выбрано квадратичное расстояние в пространстве гистограмм, в составе подсистемы сравнения- 1000 ПЭ).

Рис. 4 Сравнение зависимостей временных затрат от числа изображений в БД для 1000 процессорных элементов

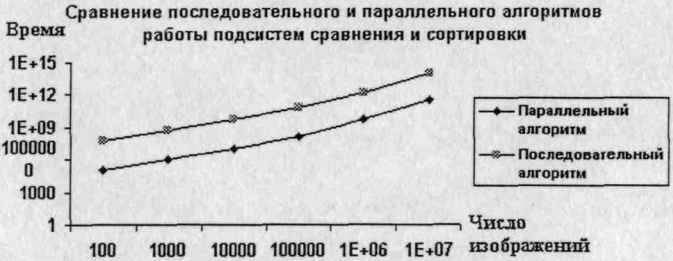

Рис. 5 Зависимость временных затрат от числа изображений в БД

Заключение

В данной работе предложена организация специализированной ЭВМ для контекстного поиска изображений, проанализирована возможность распараллеливания вычислений на различных этапах работы системы контекстного поиска изображений и показана рациональность такого подхода. Среди направлений дальнейшего усовершенствования данной вычислительной системы следует указать ее расширение за счет введения подсистемы вычисления цветовых гистограмм на этапе занесения изображения в БД, также реализованной в виде SIMD- процессора.

Литература

1. John R. Smith and Shih-Fu Chang.Tools and Techniques for Color Image Retrieval ACM Multimedia 1996: Boston, MA
2. Шозда Н. С. Поиск в базе данных изображений земной поверхности.- Тезисы всеукраинской молодежной научно- практической конференции «Людина і космос». - Днепропетровск, 1999.
3. Ортега Дж. Введение в параллельные и векторные методы решения линейных систем: Пер. с англ.-М.: Мир, 1991-367 с.
4. Лорин Г. Сортировка и системы сортировки: Пер с англ.- М.: Наука, 1983- 384 с.
5. Бройнль Т. Паралельне програмування (пер. з німецької В.А.Святного).-Київ: Вища школа, 1997.