



Fig. 5. Processed with the algorithm based on Hough transform

More comprehensive information about the line can be estimated with the „mean points of the line” representation. The representation lies in a division of the detected line pixels into some groups and then finding the mean coordinates of these groups of pixels. The least-squares method can be used to get only one straight line fully describing the observed part of the defined route.

Fig. 6. „Mean points of the line” representation (marked with „+”) and a straight line gotten with the least-squares method

All described above algorithms were realized and tested in MatLab v.7 for a set of images containing an insulation tape with the symbols „maxon motors” as a line. The algorithm based on Gaussian function was implemented in C++ and used for the motion control algorithms of the robot „MAX” [5] on a task of line-following. Visually the most precise detection was brought with K-means algorithm and the CIELAB colorspace but it couldn't be used in real time systems because of the time of his work isn't determinate. The solution is to get at the beginning the high quality line color characteristics with K-means algorithm and then to use the nearest neighborhood classification at the same time recalculating initial characteristics. As the conclusions the combination of K-means algorithm and Hough transform can be used for the robust line detection in real time.

This work has done in cooperation of the Intelligent Information Technologies Department, Brest State Technical University and the System Engineering Laboratory, Hochschule Ravensburg-Weingarten and was supported by the Landesstiftung Baden-Württemberg gGmbH.

References

1. http://en.wikipedia.org/wiki/Automated_Guided_Vehicle (15-10-2007).
2. http://en.wikipedia.org/wiki/Gaussian_function (15-10-2007).
3. J. B. MacQueen. Some methods for classification and analysis of multivariate observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press, Berkeley, USA. 1967.
4. Richard O. Duda, Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. Published in the Comm. ACM, Vol. 15, No. 1, pp. 11-15. January, 1972.
5. Zajac M., Paczynski A., Stetter R., Ucinski D. Sensor/Actuator network concept for robust control of a four-wheel-steering mobile robot. Paper. Hochschule Ravensburg-Weingarten, Weingarten, Germany. 2006.

ОРГАНИЗАЦИЯ РАСПРЕДЕЛЁННОЙ ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ СРЕДСТВАМИ ПАКЕТА R

Курочка К.С.

Гомельский государственный технический университет им. П. О. Сухого, г. Гомель

Задачи анализа и распознавания цифровых изображений приводят к обработке сотен гигабайт информации, что накладывает значительные требования на используемую вычислительную технику, от которой требуется не только высокая суммарная производительность обрабатывающих процессоров, но и большая оперативная память, достаточная для одновременного хранения сотен и тысяч изображений [1]. Даже самые быстродействующие современные ЭВМ последовательной архитектуры оказываются неспособными решать задачи такой сложности [2]. Таким образом, наиболее эффективным способом решения данных задач оказалось их распараллеливание и распределённая обработка.

Существуют специальные процессоры обработки графической информации, которые являются весьма дорогостоящими и малораспространёнными. Поэтому для решения указанных выше задач широкое применение находят распределённые вычисления на суперкомпьютерах с параллельной архитектурой векторных процессоров (PVV - Parallel Vector Process) или на кластерных конфигурациях вычислительных средств, созданных на основе локальных или глобальных компьютерных сетей [3].

Для создания программного обеспечения таких систем разработан стандарт MPI (Message Passing Interface), обеспечивающий связь между ветвями параллельного приложения [4]. В рамках MPI для решения поставленной задачи разрабатывается одна программа, и эта единственная программа запускается одновременно на выполнение на всех имеющихся процессорах или системах. При этом каждый процесс может обрабатывать свой набор исходных данных. В MPI существует целое множество операций передачи данных. Они обеспечивают разные способы пересылки данных, реализуют практически все коммуникационные операции. Подобный способ организации параллельных вычислений получил наименование модели «одна программа множество процессов» (Single Program Multiple Processes – SPMP).

В многопроцессной обработке цифровых изображений выделяются следующие фундаментальные задачи: коммуникация данных, организация управляющей структуры, статическое и динамическое распределение вычислительных ресурсов, синхронизация и оптимизация процессов [2]. Причём наиболее узким местом в таких вычислительных системах из-за объёмов передаваемой информации будет коммуникация данных. Поэтому следует использовать различные методы, позволяющие минимизировать объёмы передаваемой информации. Одним из способов решения данной проблемы является организация многоуровневых приложений, где работой клиентов управляет специальный компьютер или процесс, называемый сервером заданий (рис. 1) [3].

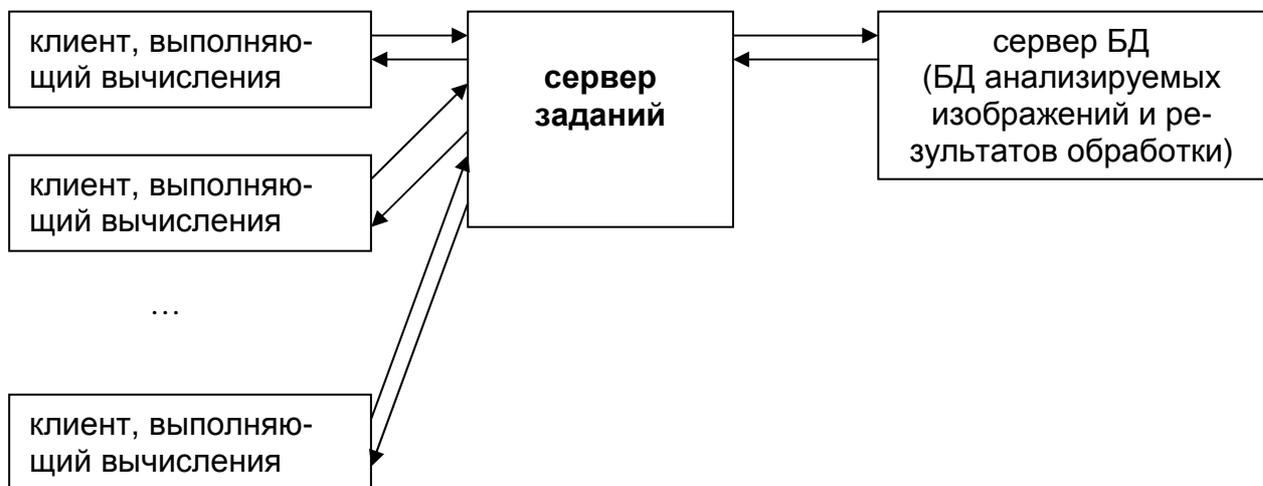


Рис. 1. Трёхуровневое приложение распределённой обработки

Применение программных пакетов, удовлетворяющих спецификации MPI, наряду с преимуществами простой реализации распараллеливания вне зависимости от архитектуры используемой вычислительной системы имеет и недостатки. Разработчику приходится реализовывать ряд уже стандартных алгоритмов и методов преобразований цифровых изображений (в частности, фильтрация на основе свертки и корреляция, а также базовые методы линейной алгебры), которые, обладая хорошей структурированностью, регулярностью и рекурсивностью, допускают эффективное распараллеливание. Все данные алгоритмы реализованы в специализированных пакетах обработки цифровых изображений, например, таких как «R» (<http://www.r-project.org/>). Поэтому весьма актуальной является задача организации распределённой обработки цифровых изображе-

ний с использованием средств специальных пакетов. Для этого предлагается использовать несколько изменённую архитектуру вычислительной системы (рис. 2). В качестве клиентов могут выступать процессы (например, отдельные запущенные копии программы R), выполняемые как в рамках одной многопроцессорной ЭВМ, так и выполняемые на различных компьютерах вычислительной сети. В качестве сервера заданий может выступать также экземпляр R, выполняющий функции TSP сервера. Взаимодействие между элементами распределённой системы осуществляется на сетевом и транспортном уровне сетевых протоколов (согласно модели OSI). При использовании стека протоколов TCP/IP это взаимодействие реализуется посредством сокетов. В качестве сервера БД может выступать любая СУБД (например, MySQL).

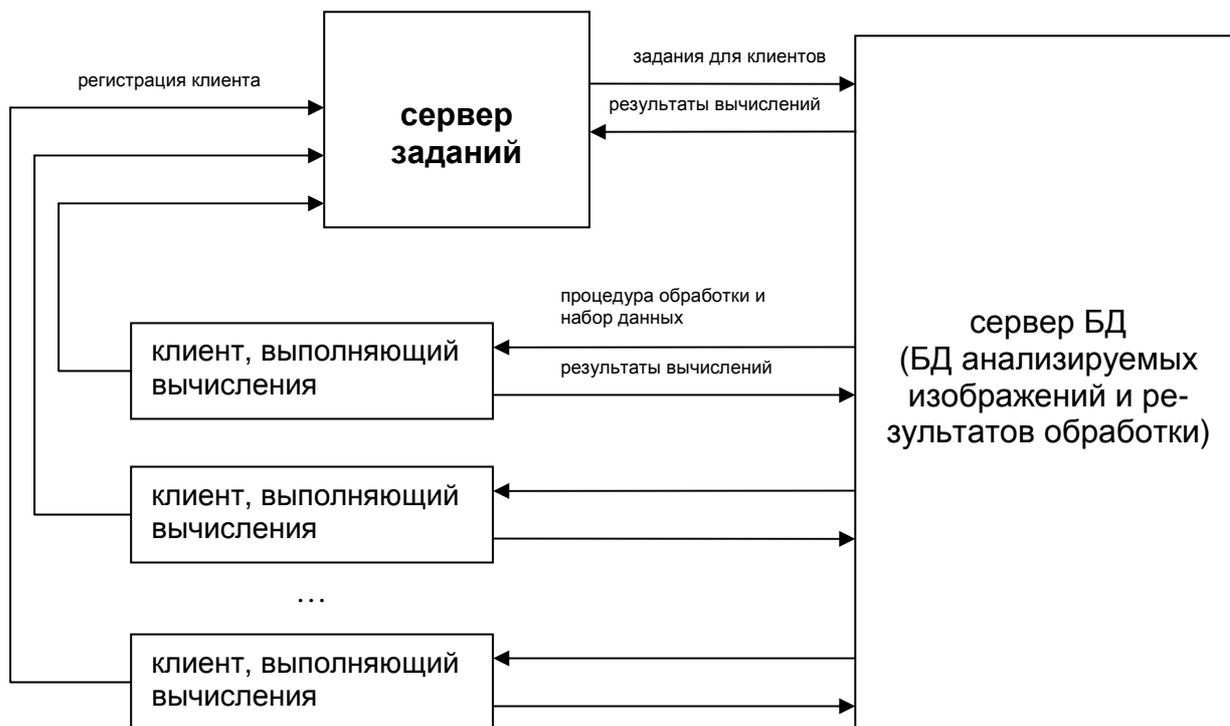


Рис. 2. Модифицированная схема распределённой обработки при анализе биомедицинских изображений

При такой структуре (рис. 2) вычислительной системы минимизируется внутрисетевой трафик за счёт исключения передачи данных между планировщиком задач и вычислительными клиентами. Кроме того, имеется возможность организации сервера БД в виде кластера компьютеров, что ещё значительно позволит увеличить производительность системы.

Была проведена апробация предлагаемой методики организации распределённых вычислений при анализе биомедицинских изображений. В качестве тестовой задачи рассматривалась фильтрация растрового изображения с использованием функции t-test. Обрабатываемое изображение разделялось на части и, используя возможности стека протоколов TCP/IP, передавалось на рабочие станции по локальной вычислительной сети. Каждая рабочая станция обрабатывала свою часть изображения и отправляла результаты на сервер. Для обработки изображений на клиентских компьютерах и передачи информации на сервер использовалась прикладная программа «R». Рассматривалось растровое изображение размером 1024 на 768 32-битных пикселей, которое группами по несколько десятков строк передавалось на обрабатывающие станции. Окрестность определялась в форме квадрата 55 на 55 пикселей. Поэтому в случае 10 компьютеров на каждый передавалось по 187 строк. Передача информации осуществлялась последовательно каждому компьютеру. Таким же образом было организовано и получение результатов.



Рис. 3. Зависимость времени распределённой обработки от количества используемых компьютеров

Время распределённой обработки 1 изображения на 3 персональных компьютерах в среднем составило порядка 252 с, при однопоточной обработке – 357 с (рис. 3). На управляющем компьютере расчётов не производилось. Рабочие станции работали под управлением ОС Windows 2000, CPU Celeron 1000, ОЗУ 256 Мб. Сеть – Fast Ethernet 100 Мбит/с. Время передачи обрабатываемой информации при 10 узлах составило порядка 0.197 с. Время обработки на одном узле при 10 станциях составило порядка 8 с.

Литература

1. Tuzlukov V. P. Signal and image processing in navigational systems –CRC press, 2005 (ISBN 0-8493-1598-0)
2. William K. Pratt Digital image processing – John Wiley & Sons, Inc., Hoboken, New Jersey, 2007 (ISBN: 978-0-471-76777-0)
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.
4. Marc Snir, Steve Otto, Steven Huss-Lederman, David Waker and Jack Dongarra MPI: The Complete Reference, MIT Press

РАСКРАШИВАНИЕ ЧЕРНО-БЕЛОГО ИЗОБРАЖЕНИЯ СРЕДСТВАМИ C#

Новиков В.А., Харитонов Д.С.

Международный институт трудовых и социальных отношений, г. Минск

Известно, что в изображении глаз наиболее чувствителен к цветовой, а не к яркостной составляющей. На использовании этой особенности, в частности, основано сжатие изображения в формате .jpg. В то же время графический пакет Photoshop при раскрашивании областей использует только один фиксированный цвет в модели Lab. В результате цветное раскрашенное изображение оказывается “безжизненным”. Это особенно заметно в областях тела и одежды. Реально эти области изменяют свой цвет в зависимости от освещенности, и не учитывать эту особенность нельзя при получении цветного изображения.

Объектная система C# содержит в себе все средства для полноценной обработки изображения в формате .bmp [1]. В том числе средства C# позволяют работать с альфа-каналом.

Идея раскрашивания черно-белого изображения заключается в задании для фиксированного набора оттенков серого соответствующих цветов в модели RGB. Промежуточные цвета получаются путем линейной интерполяции по каждой составляющей RGB. Подобное раскрашивание возможно как для всего изображения, так и для выделенной области с использованием альфа-канала. Алгоритм раскрашивания состоит в предварительном выводе на экран границ шкалы серого в исходном изображении. Далее с ис-