

УДК 004.9

UDC 004.9

05.00.00 Технические науки

Engineering

ПОВЫШЕНИЕ ЭФФЕКТИВНОСТИ ПРОГРАММНОЙ РЕАЛИЗАЦИИ АЛГОРИТМОВ РАСПОЗНАВАНИЯ ИЗОБРАЖЕНИЙ С ДЕТАЛЬНОЙ ОЦЕНКОЙ СОСТОЯНИЯ МАССОВОГО КОЛИЧЕСТВА ОБЪЕКТОВ

IMPROVING THE EFFICIENCY OF THE SOFTWARE IMPLEMENTATION OF IMAGE RECOGNITION ALGORITHMS WITH A DETAILED ASSESSMENT OF THE STATUS OF THE MASSIVE NUMBER OF OBJECTS

Остапов Дмитрий Сергеевич
Аспирант
Кубанский государственный технологический университет, Краснодар, Россия

Ostapov Dmitriy Sergeevich
graduate student
Kuban State Technological University, Krasnodar, Russia

Усатиков Сергей Васильевич
д.ф.-м.н., доцент
Кубанский государственный университет, Краснодар, Россия

Usatikov Sergey Vasilievich
Dr.Phis.-math., associate professor
Kuban State University, Krasnodar, Russia

Проведен вычислительный эксперимент с программной реализацией алгоритмов системы компьютерного зрения (СКЗ) для идентификации и детальной оценки состояния объектов природного происхождения и их массового количества, с высокой вариабельностью внутри классов и близостью самих классов. Оценена точность рабочего режима СКЗ. Показаны возможности снижения времени обработки видеокadra в рабочем режиме СКЗ оптимизацией последовательных вычислений и организацией параллельных вычислений

A computing experiment with software implementation of computer vision system (CVS) algorithms has been performed for identification and detailed condition assessment of objects of natural origin and their mass number, with high variability within classes and an affinity of the classes. An accuracy of the operating mode was assessed. The article also shows possibilities of decreasing of video frame processing in CVS operating mode through an optimization of sequential computation and organization of parallel computation

Ключевые слова: СИСТЕМЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ, ПАРАЛЛЕЛЬНЫЕ ВЫЧИСЛЕНИЯ, ЗАКОН АМДАЛА, КЛАСТЕРИЗАЦИЯ K-MEANS, ОПТИМИЗАЦИЯ, EMGU CV

Keywords: COMPUTER VISION SYSTEMS, PARALLEL COMPUTING, AMDAL'S LAW, K-MEANS CLUSTERIZATION, OPTIMIZATION, EMGU CV

Doi: 10.21515/1990-4665-124-037

Введение

Задача идентификации системами компьютерного зрения (СКЗ) массового количества объектов с получением информации о детальном состоянии их поверхности, с высокой вариабельностью внутри классов и близостью самих классов, актуальна для разработки систем поддержки принятия решений (СППР) в ходе управления технологическим процессом, при контроле продукции (инспекционный конвейер и т.д.), для решения задач фотосепарации и т.п.[1-3]. В работе [3] рассмотрено вычислительное

ядро такой системы. СКЗ состоит из нескольких подсистем: подсистемы бинаризации, выполняющей отделение объектов от фона и выделение единичных объектов из их массы; подсистема сегментации, которая осуществляет адаптивную сегментацию с использованием алгоритма X-means и подсистема идентификации сегментов, которая с использованием критерия Пирсона выполняет идентификацию каждого сегмента системой, даёт качественную и количественную оценку состояний каждого объекта и всей массы объектов как генеральной совокупности.

Эффективность системы оценивается в первую очередь по требованиям к точности и быстродействию (производительность с возможностью работы СКЗ в реальном масштабе времени). Например, при входном контроле фруктового сырья, для инспекционных конвейеров со скоростями движения ленты до 0.1 м/с масштабы времени обработки мгновенных видеоснимков составляют несколько сек., на конвейерах Buhler Group в оптических сортировщиках серии Sortex масштабы времени порядка 10^{-2} - 10^{-1} сек; для решения задач фотосепарации зерновой массы время обработки мгновенного изображения должно быть менее 10^{-2} сек.

На скорость работы СКЗ влияют множество факторов: характеристики и архитектура вычислительных мощностей, состояние операционной системы и среда программирования, оптимальность программного кода и степень параллельности вычислений. В данной работе рассмотрена организация параллельных вычислений и некоторые аспекты оптимальности программного кода. Проведён вычислительный эксперимент с программной реализацией алгоритмов СКЗ [3] на изображениях с инспекционного конвейера с фруктовым сырьём. Оценена точность рабочего режима такой СКЗ. Показаны возможности снижения времени обработки видеокadra в рабочем режиме СКЗ оптимизацией последовательных вычислений и организацией параллельных вычислений.

1. Особенности программной реализации алгоритмов системы и организация параллельных вычислений.

Алгоритмы СКЗ [3] реализованы с использованием языка программирования C# в среде Visual Studio и библиотеки обработки изображений EmguCV, кросс-платформенной реализации библиотеки OpenCV. СКЗ состоит из подсистемы бинаризации и выделения единичных объектов (основана на алгоритмах k-means, Оцу и ELSD), сегментации (основана на модифицированном алгоритме k-means) и идентификации сегментов (основана на использовании критерия Пирсона).

Для повышения эффективности кода расчетные формулы видоизменялись. Например, прирост скорости на 30% дало: при расчете средних квадратических отклонений (СКО) вычитание квадрата математического ожидания после окончания цикла; в общей формуле с отношением СКО исключение использования квадратного корня и операции деления на объём выборки. В алгоритме присутствует участок, связанный с автоматическим определением числа сегментов системой. В качестве критерия останова используется энтропия и СКО. Вместо использования отдельных методов объекта для расчета энтропии и СКО, когда дважды будет происходить перебор по всем пикселям объекта, данные показатели рассчитываются в одном методе в одном цикле. В реализуемом алгоритме используется большое число простых для процессора промежуточных расчетов, которые могут занимать большое количество памяти. Чтобы время на выделение и очистку памяти не было во много раз большим, чем время расчетов, следует максимально аккуратно относиться к работе с массивами, стараться как можно меньше выделять новую память, без необходимости не использовать переменные типа double (число двойной точности, 64 бита), а использовать вместо них float (число с плавающей точкой после запятой, точность 7 знаков после запятой, 32 бита).

Основу разработанной в [3] системы составляет модифицированный алгоритм k-means: он используется при бинаризации и сегментации изображения. Скорость работы СКЗ зависит как от перечисленных выше факторов, так и от самого языка программирования. Так, при проведении ряда экспериментов, было выявлено, что выполнение кластеризации с использованием алгоритма k-means на C++ срабатывает в 3.15 раза быстрее, чем аналогичные действия на C#, что объясняется особенностями работы платформы .Net Framework.

Алгоритм k-means относится к итерационным алгоритмам, каждую итерацию которого легко распараллелить. Параллельность реализуется в самом узком месте алгоритма: расчету расстояний от каждой точки до центра кластера[4-7]. В реализованных алгоритмах СКЗ итерации выполнялись параллельно в следующих местах: в k-means при расчете расстояний до центров кластеров; при переборе по всем пикселям и определении нечеткого кластера при работе подсистемы контроля качества бинаризации; при сегментации и обработке каждого объекта отдельно. В блоке сегментации при проведении экспериментов использовался непараллельный k-means, так как избыточная параллельность, которая превышает число ядер на процессоре, ведет к замедлению работы системы. Однако в случае использования процессоров с большим числом ядер, можно осуществлять параллельность не только по объектам, но и рассчитывать среднеквадратическое отклонение, энтропию и выполнять итерации цикла по определению числа кластеров и использовать параллельный k-means. В блоке идентификации сегментов объектов каждый объект обрабатывался отдельно и параллельно.

2. Условия проведения вычислительного эксперимента и точность предлагаемой СКЗ.

Эксперименты проводились на компьютере со следующими характеристиками: процессор Intel i7-3930K, тактовая частота 3.2ГГц;

количество ядер: 6; количество потоков: 12; кэш-память 12 Мб; оперативная память DDR3 16 Гб 1333 МГц; материнская плата MSI X79A-GD45 (8D) (MS-7760); жёсткий диск SSD Kingston SNM225-S2/160GB; операционная система Windows 7; установленная платформа .NET Framework 4.5.

СКЗ работает в двух режимах: режиме обучения и рабочем режиме. Задача режима обучения – заполнить базу, состоящую из распределений вероятностей сегментов, которая в дальнейшем необходима для идентификации состояний с использованием критерия Пирсона χ^2 . Обучение производилось на 15 снимках с инспекционного конвейера с фруктовым сырьём со 150 объектами (яблоки). На рис.1 представлен пример одного из мгновенных снимков рабочей зоны роликового конвейера. После того как система обучена, возможно её функционирование в рабочем режиме. Эксперименты проводились только по определению состояния поверхности единичных объектов - считалось, что тип и вид объекта уже известен и траты времени на их распознавание не оценивались.



Рисунок 1 – Пример видеокadra инспекционного конвейера с 14 единицами фруктового сырья.

Вычислительный эксперимент по оценке точности проводился на изображениях, не участвовавших в процессе обучения СКЗ. Точность рассчитывалась как отношение количества неверно идентифицированных сегментов системой к общему количеству сегментов. В результате эксперимента выявлено, что на данном виде объектов СКЗ имеет точность 98.5%.

Известен теоретический предел возможностей распараллеливания вычислительного процесса [4-5]. В соответствии с законом Амдала-Уэра, оптимизация по параллельности ограничена временными затратами на выполнение последовательного кода. Предел ускорения с использованием параллельных вычислений ограничен долей последовательного кода -

коэффициентом α , который рассчитывается по формуле: $\alpha = \frac{t_1}{t_2}$, где t_1 - время, потраченное на выполнение последовательного кода, t_2 - время, потраченное на выполнение кода, который можно распараллелить. Ускорение, которое может быть получено на многопроцессорной системе,

не может превышать величину $s_p = \frac{1}{\alpha + \frac{1-\alpha}{p}}$, где p - число процессоров в системе. Время, равное результату деления времени работы на одном процессоре на ускорение s_p , является минимальным временем, которое можно получить за счет организации параллельных вычислений. Для расчета времени выполнения параллельных и последовательных участков кода, использовался таймер класса `StopWatch` пространства имён `System.Diagnostics`. Последовательно выполнялся код загрузки изображения в оперативную память, сохранение результатов бинаризации, сохранение результатов контроля качества бинаризации; выделение единичных объектов. Операции загрузки изображения и сохранения результатов выполнения промежуточных шагов можно также распараллелить, однако для этого необходимо использование потокобезопасных элементов, для чего необходимо реализовывать свои классы работы с изображениями и не пользоваться готовым набором методов `Emgu CV`.

Коэффициент α зависит от разрешения изображения (т.е. число пикселей, равное произведению числа пикселей по длине и ширине), количества объектов и числа сегментов на изображении. Первоначально

расчеты проведены на одном процессоре и на основании этих данных было получено значение коэффициента α . Значение коэффициента α в зависимости от числа пикселей изображения представлено на рисунке 2.

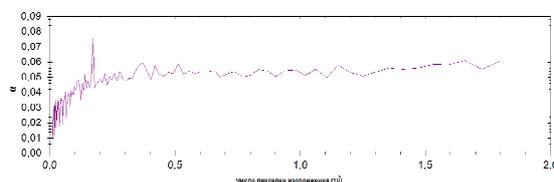


Рисунок 2 – Значение параметра закона Амдала α в зависимости от разрешения изображения при заданном числе объектов, равном 14.

Непостоянство значения α на рис.2 объясняется следующими причинами. Коэффициент α показывает долю кода, которую невозможно распараллелить - чем меньше значение α , тем лучше будет эффект от увеличения числа процессоров. В зависимости от разрешения, разный процент времени будет затрачен на участки кода, которые нельзя распараллелить по сравнению с общим временем выполнения кода; также изменение разрешения может повлиять на критерий останова в алгоритме сегментации.

3. Многопроцессорный вычислительный эксперимент

При проведении вычислений с данной программной реализацией СКЗ варьировались: разрешение изображения, число потоков (соответствует числу процессоров p), количество объектов на изображении. Число сегментов на изображении при подаче изображения на вход заранее не известно и зависит от числа объектов, состояния каждого объекта и разрешения изображения.

Одним из крупных этапов вычислений является бинаризация «фон-объект». Время работы подсистемы бинаризации при обработке изображений разного разрешения при разной степени параллельности для фиксированного числа объектов на изображении представлено на рис.3а.

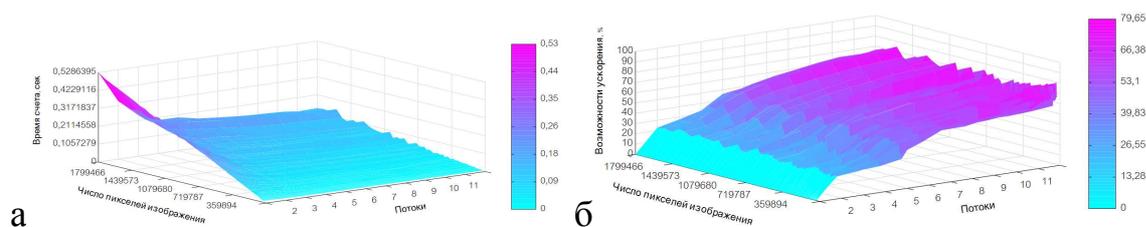


Рисунок 3. а - Время работы подсистемы бинаризации для числа объектов на изображении, равном 14; б - теоретический % снижения времени счёта подсистемы бинаризации.

Поверхность теоретического предела снижения времени счета, представленная на рис.3б, рассчитывается по формуле:

$$a_t = \frac{T_R - T_A}{T_R} * 100\% \quad (1)$$

где a_t – теоретический % снижения времени счёта, T_R –реальное время работы, T_A – теоретическое время работы в соответствии с законом Амдала. В данном случае T_R и T_A определялись для подсистемы бинаризации. Из рис.3б видно, что имеется значительный резерв для ускорения, который растёт с числом потоков.

Время работы всей системы и возможности её ускорения представлены на рис.4.

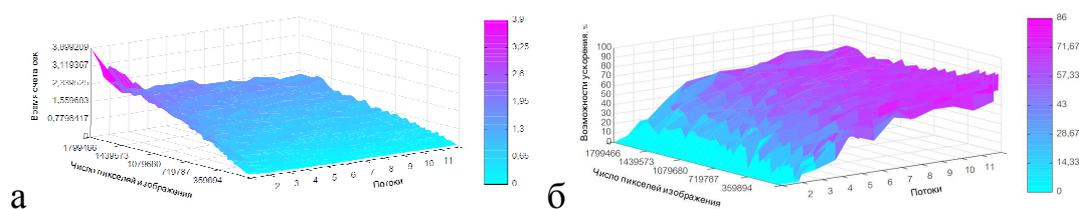
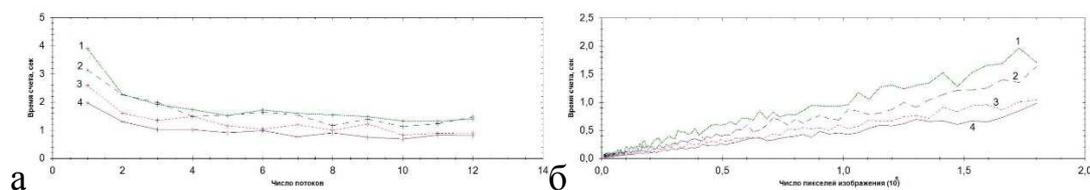


Рисунок 4. а - Время работы СКЗ для числа объектов на изображении, равном 14; б - теоретический % снижения времени счёта системы



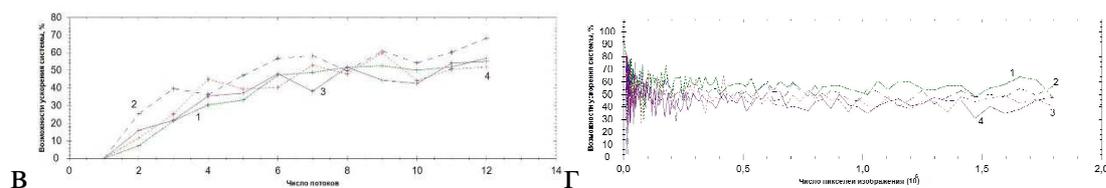


Рисунок 5 – Время работы системы (а, б) и возможности ускорения (в,г) в зависимости от числа потоков (с фиксированным количеством пикселей 1799467) (а,в) и от числа пикселей (при 6 потоках) (б,г): 1 – для изображения с 14 объектами, 2 – 12, 3 – 6, 4 – 4 объектами.

В соответствии с законом Амдала при увеличении степени параллельности при заданных остальных характеристиках системы, время работы должно строго убывать, однако из поверхности, представленной на рис.4а, видно, что присутствуют участки роста времени. Это объясняется тем, что на процессоре i7-3930К, на котором проводились эксперименты, присутствуют 6 физических и 12 логических ядер (потоков), и операционная система нагружает каждое физическое ядро в соответствии со своими приоритетами, которые не задаются с использованием языка высокого уровня C#.

На рис.4б прослеживается рост возможностей для ускорения с ростом числа потоков. Также видно, что график времени при переменном числе потоков не является монотонно убывающим. Это объясняется тем, что операционная система сама перераспределяет нагрузку между ядрами и в определенных случаях 2 потока обрабатываются на 1 ядре, что является неоптимальным, так как затрачивается дополнительное время на управление параллельностью.

Из рис.5 видно, что время работы системы в среднем примерно пропорционально числу объектов; в то время как число объектов слабо влияет на возможный % снижения времени работы СКЗ.

Как видно из рис.4б, код является неоптимальным и его выполнение возможно значительно ускорить. Загрузка ядер процессора на

изображении с 14 объектами, разрешения 1799467 пикселей, при использовании 12 потоков, представлена на рис.6.

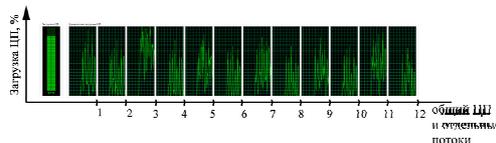


Рисунок 6 –Общая загрузка центрального процессора и загрузка отдельных ядер процессора.

Как видно из рис.6, все ядра никогда не нагружаются полностью, на 100%. Это объясняется тем, что значительную часть времени программа тратит на выполнение операций с памятью (её выделение, очистку, операции чтения и записи). И оптимизация программы, основанная на снижении расходов памяти, позволит с большей интенсивностью выполнить загрузку процессора. Однако в связи с тем, что EmguCV для работы предоставляет методы, не учитывающие специфику решаемой задачи, не оптимизированные под её нужды, первым этапом оптимизации является собственная реализация использованных методов в рамках работы СКЗ.

Заключение

1. Проведён вычислительный эксперимент с программной реализацией СКЗ [3] для оценки эффективности системы, на изображениях с инспекционного конвейера с фруктовым сырьём. Точность рассчитывалась как отношение количества неверно идентифицированных СКЗ сегментов к общему количеству сегментов. Выявлено, что на данном виде объектов СКЗ имеет точность 98,5%.

2. Из известных факторов, влияющих на работу программного обеспечения (установленный тип процессора, тип и размер оперативной памяти, среда разработки программного обеспечения, оптимальность кода, использование методов параллельного программирования), выявлено

влияние использования языка программирования C++ вместо C#: это позволяет увеличить скорость работы СКЗ более чем в 3 раза.

3. Оценена эффективность СКЗ [3] по требованиям к быстродействию (производительность с возможностью работы СКЗ в реальном масштабе времени). Показаны возможности снижения времени обработки видеокadra в рабочем режиме СКЗ оптимизацией последовательных вычислений и организацией параллельных вычислений: исследовано влияние размера изображения, количества единичных объектов на изображении и числа процессоров. Разработанное программное обеспечение является неоптимизированным по памяти и тратит много времени на работу с памятью, в течение которого процессор простаивает. Осуществление оптимизации по памяти позволит значительно увеличить нагрузку на процессор и в следствие этого, повысить скорость работы программы.

4. Ориентировочно, работы СКЗ с одним центральным компьютером достаточно, например, для инспекционного конвейера. Для задач фотосепарации может возникнуть необходимость разработки другой вычислительной архитектуры - системы специализированных вычислительных плат, (например, платы Siemens 6ES7195-3BA10-0YA0). Это один из способов увеличения производительности СКЗ переносом вычислений с CPU на специализированные платы. При работе на классическом ПК данные постоянно помещаются в оперативную память и происходит частое обращение к ней. В случае использования специализированных плат, оснащённых памятью, будет происходить более редкое обращение к оперативной памяти. В результате этого, сократится время транспортировки данных по системным шинам материнской платы, на каждой плате будет обрабатываться свой блок данных, и в оперативную память будет помещаться только результат вычислений с этим блоком.

Литература

1. Зиятдинова В.А., Шаззо А.Ю., Усатиков С.В., Погорелова И.И. Оценка качества риса с использованием современных методов анализа цветовых характеристик единичных зерен // Изв.ВУЗов. Пищевая технология, 2015, № 2-3. - с.100-104
2. D. Weimer, H. Thamer and K.-D. Thoben, GPU architecture for unsupervised surface inspection using multiscale texture analysis, pp. 278-284, 2014
3. Остапов Д.С. Подходы к идентификации состояния 3D-поверхности по 2D-изображениям единичных объектов и их массового количества // Вестник АГТУ. Сер.УПРАВЛЕНИЕ, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ИНФОРМАТИКА, 2016, № 4 (октябрь), с.55-64
4. Воеводин В.В. Вычислительная математика и структура алгоритмов // М.: Изд-во МГУ, 2010, - 162 с.
5. Карпов В.Е. Введение в распараллеливание алгоритмов и программ. // Компьютерные исследования и моделирование, 2010. Т. 2, № 3, с. 231-272.
6. Речкалов Т.В. Параллельный алгоритм кластеризации для многоядерного процессора Intel Xeon Phi // Международная научно-техническая конференция "Суперкомпьютерные дни в России 2015" 26-27 сентября 2015 г. - М.: Федеральное агентство научных организаций России, 2015. - С. 632 - 640
7. Третьяков Ф.И., Серебряная Л.В. Распараллеливание алгоритмов классификации и кластеризации данных. // Вестник БГУ. Серия 1, Физика. Математика. Информатика. 2013. № 2. С. 105-109.

References

1. Ziyatdinova V.A., SHazzo A.YU., Usatikov S.V., Pogorelova I.I. Ocenka kachestva risa s ispol'zovaniem sovremennyh metodov analiza cvetovyh karakteristik edinichnyh zeren // Izv.VUZov. Pishchevaya tekhnologiya, 2015, № 2-3. - s.100-104
2. D. Weimer, H. Thamer and K.-D. Thoben, GPU architecture for unsupervised surface inspection using multiscale texture analysis, pp. 278-284, 2014
3. Ostapov D.S. Podhody k identifikacii sostoyaniya 3D-poverhnosti po 2D-izobrazheniyam edinichnyh ob"ektov i ih massovogo kolichestva // Vestnik AGTU. Ser.UPRAVLENIE, VYCHISLITEL'NAYA TEKHNIKA I INFORMATIKA, 2016, № 4 (oktyabr'), s.55-64
4. Voevodin V.V. Vychislitel'naya matematika i struktura algoritmov // M.: Izd-vo MGU, 2010, - 162 s.
5. Karpov V.E. Vvedenie v rasparrallelivanie algoritmov i programm. // Komp'yuternye issledovaniya i modelirovanie, 2010. T. 2, № 3, s. 231-272.
6. Rechkalov T.V. Parallel'nyj algoritm klasterizacii dlya mnogoyadernogo processora Intel Xeon Phi // Mezhdunarodnaya nauchno-tekhnicheskaya konferenciya "Superkomp'yuternye dni v Rossii 2015" 26-27 sentyabrya 2015 g. - M.: Federal'noe agentstvo nauchnyh organizacij Rossii, 2015. - S. 632 - 640
7. Tret'yakov F.I., Serebryanaya L.V. Rasparrallelivanie algoritmov klassifikacii i klasterizacii dannyh. // Vestnik BGU. Seriya 1, Fizika. Matematika. Informatika. 2013. № 2. S. 105-109.