

ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТИ ТЕХНОЛОГИИ CUDA В ЗАДАЧЕ РАСПРЕДЕЛЕННОЙ ПРЕДПЕЧАТНОЙ ПОДГОТОВКИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ

С.А. Бибииков, А.В. Никоноров, В.А. Фурсов, П.Ю. Якимов

Введение

Известно, что центральной при построении параллельных приложений является проблема отображения графа алгоритма на граф вычислительной системы. С ней связано, берущее начало у истоков развития вычислительной техники, противостояние специализации и универсальности. В этом контексте заслуживает внимания исследование места и значения модного в последнее время направления в развитии высокопроизводительных приложений, основанного на технологии CUDA.

Привлекательной при использовании этой технологии является, прежде всего, возможность увеличения вычислительной мощности ценой сравнительно небольших затрат. По оценкам Алексея Тутубалина для десктопных решений "экономика гигафлопса" при использовании карт NVidia различных модификаций выглядит следующим образом. Один гигафлопс одинарной точности стоит от \$1,0 до \$2,9, а один гигафлопс двойной точности - от \$ 4,8 до \$ 6,3.

Конечно, показатели стоимости вычислений являются важными, однако с точки зрения стоимости реализации конструкторского проекта в целом (с учетом возможного выигрыша за счет сокращения времени вывода продукта на рынок) не менее важно также выявление достижимых показателей масштабируемости и связанного с этим ускорения. В частности, достижение минимального значения стоимости вычислений имеет смысл лишь при выполнении требования допустимого времени счета. В этой связи представляет интерес, во-первых, выявление приложений, для которых использование технологий CUDA эффективно, во-вторых, получение для них количественных оценок достижимых показателей ускорения.

В настоящей работе приводятся описание реализаций и результаты исследований распределенных приложений, на основе технологии CUDA для решения задач предпечатной обработки цифровых изображений.

Задачи предпечатной подготовки цветных изображений.

Необходимость проведения предпечатной обработки изображений при производстве цифровых репродукций произведений живописи связана со спецификой фотографирования оригиналов. В процессе фотографирования возникают следующие артефакты: множественные блики, затенения части холста рамкой и так называемые матовые блики. Артефакты обусловлены неровностями нанесения краски на холст и способом установки дополнительного освещения.

Множественные блики возникают вследствие того, что отдельные малые участки слоя краски обладают таким углом наклона поверхности, что падающий свет источника отражается и регистрируется камерой в виде пятен. На цифровом изображении такие артефакты выглядят как области небольшого размера, обладающие выделяющейся на общем фоне светлостью.

Коррекция множественных бликов выполняется в два этапа: поиск областей обладающих высокой относительно некоей окрестности яркостью и замена цвета блика на цвета из его окрестности.

Затенение возникает, когда свет от осветительного прибора падает под большим углом к поверхности картины. При этом рама вокруг картины является экраном и создает на полотне затененную область. На изображении граница затененной области может обладать сложной формой, если рама имеет объемный рельеф.

Для устранения тени вначале необходимо определить ее границу на изображении. Сложность задачи состоит в том, что на картине могут быть участки "похожие" на границу тени. Затем в найденной области тени осуществляется приведение распределения цветов к виду распределения цветов из соседней освещенной области.

Матовый блик возникает в тех случаях, когда свет от лампы падает под малым углом. При этом на изображении возникает область, соответствующая размытому изображению осветительного прибора. Точки в этой области обладают повышенной яркостью и пониженной цветностью.

Решение данной задачи заключается в определении геометрических параметров матового блика и проведении цветовой коррекции изображения в соответствии с полученными параметрами.

Алгоритмы цветовой коррекции в процессе предпечатной подготовки.

В работе [1] был рассмотрен алгоритм удаления множественных точечных бликов с использованием вычислительной среды, позволяющей автоматизировать процесс и провести его с минимальным участием оператора.

Алгоритм устранения точечных артефактов разбивается на две фазы - фазу идентификации артефакта на изображении и фазу коррекции. Идентификация выполняется при помощи обработки всего изображения окном 16x16 пикселей с расчетом внутри окна локальных характеристик таких как гистограммы и средние

значения. На основе такой оконной обработки формируется битовая маска описывающая положение точечных бликов на изображении.

Коррекция бликов выполняется по полученной маске при помощи медианной фильтрации [3] в области блика. Для яркостной коррекции на границе блика на завершающей стадии обработки выполняется гауссово размытие изображения [3], результат которого накладывается на исходное изображение согласно маске.

Задача устранения на изображении тени решается в четыре этапа: поиск границы тени (1), определение параметров преобразования гистограмм (2), внесение изменений в область постоянной тени (3), внесение изменений в область размытой тени (4).

При определении границы тени для уменьшения влияния мазков, проводится усреднение значений светлости вдоль нее, в результате формируется вектор среднего перепада светлости:

$$S^i(x) = \frac{1}{h} \sum_{y=1}^h W^i(x,y)$$

где $W(x,y)$ - значения светлости изображения в точке (x,y) , h - высота окна.

Далее определяется средняя светлость, для чего значения полученного вектора последовательно усредняются, значениям, задавая набор векторов:

$$x^{\#} : \min_x |S^{\#}(x) - M^i|, \quad M^i = \frac{1}{2} (\max(S^{\#}) + \min(S^{\#}))$$

Определяя для каждого из векторов S

и усредняя значения x , получают значение, определяющее положение линии границы тени. При этом используется также специальный алгоритм, обеспечивающий связность линии. Затем в полученной области осуществляется приведение распределения цветов к виду распределения цветов из освещенной области.

Сам алгоритм цветовой коррекции основан на эквализации гистограмм распределения цветов [3]. По гистограммам затененных и освещенных участков строится преобразование P , переводящее значения светлости затененной области в значения освещенной области.

Алгоритмы устранения матовых бликов также строятся на основе сравнения функций распределения яркости компонент в цветовом пространстве Lab. Основная проблема в данном случае состоит в том, что искажения изображений этого типа являются существенно неизопланатичными.

Описанные выше алгоритмы устранения множественных бликов, теней и матовых бликов весьма требовательны к вычислительным мощностям и их реализация на персональном компьютере при размерах изображения, характерных для полиграфии, выполняется неприемлемо долго. Так, например, последовательная реализация алгоритма из работы [1] выполняется порядка 5 минут для изображения 5000x5000 пикселей.

Архитектура системы

В работе использован следующий принцип распределенной реализации, предложенный в работе [1] для решения задачи ретуши точечных артефактов (бликов) при подготовке цветных изображений к печати. В данном случае, наряду с устранением точечных бликов разработанный по этому же принципу алгоритм устранения тени от рамки реализован в виде модуля расширения (plug-in) к системе обработки изображений Adobe Photoshop. Общая схема организации распределенных вычислений в системе, интегрированной с системой Adobe Photoshop, показана на рисунке 1.



Рис.1 Архитектура системы

Обрабатываемая часть изображения декомпозируется на фрагменты по числу компьютеров выполняющих вычисления. Декомпозицию, рассылку и последующую сборку изображения из фрагментов выполняет корневой Java-процесс, инициировавший распределенные вычисления. Вычисления выполняются native-модулями. Комбинация Java и native-кода в рамках GRID системы позволяет получить приемлемое сочетание производительности, масштабируемости и дает возможность повторного использования кода.

Характеристики ускорения и эффективности для конкретной локальной сети существенным образом зависят от вычислительной сложности корневого процесса ретуши изображений и интенсивности искажений. Ниже описывается, реализованный в указанной GRID-системе алгоритм ретуши тени, осуществляющий автоматическую идентификацию и коррекцию области тени на изображениях.

Рассмотренные задачи позволяют про вести не только распределение вычислений по компонентам GRID системы, но и провести более глубокое, многопоточное распараллеливание. Многопоточность задействуется как на многоядерных CPU так и на GPU (Graphic Processing Unit) - графических сопроцессорах. Возможность декомпозиции задачи по данным позволяет использовать технологию NVIDIA CUDA [4]. Технология CUDA позволяет задействовать симметричные многопоточные вычисления (SMT) над однотипными данными. Для вычислений, как на стадии идентификации так и на стадии коррекции может быть использовано более тысячи потоков на одном графическом вычислителе. Для рассматриваемых задач коррекции цветных изображений такое распараллеливание позволяет получить сокращение времени вычислений в 3-5 раз.

Так как стоимость CUDA вычислителей достаточно высока, выгодно комбинировать технологии путем создания GRID/SMT системы. Такая система позволяет контролировать загрузку GPU вычислителей и распределять нагрузку между GPU и CPU ресурсами.

Обсуждение результатов

В развитие результатов работы [1] удалось получить существенное ускорение работы алгоритмов за сет использования универсальных графических процессоров Nvidia и технологии CUDA. Сравнительные оценки эффекта от использования тех или иных CPU и GPU для задачи идентификации точечных бликов [1] приведены на рисунке 2. На диаграмме представлены затраты времени на идентификацию бликов на изображениях разного размера (3, 5 и 18 мегапикселов).

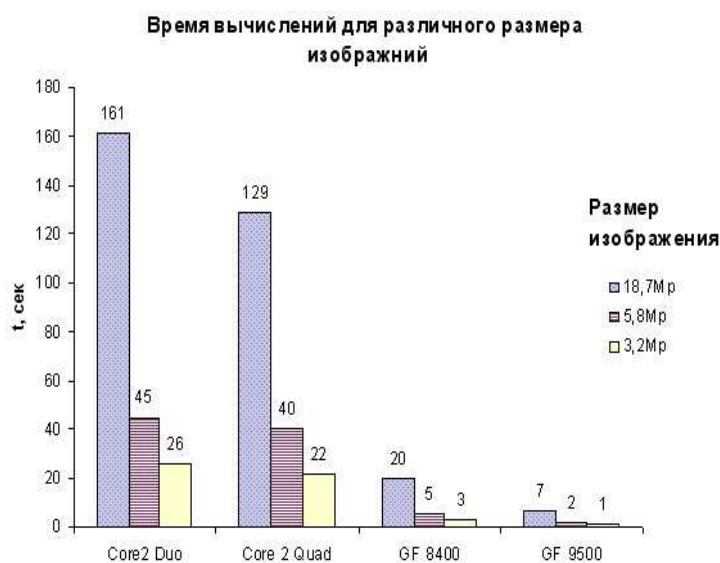


Рис.2 Эффективность CPU и GPU реализаций.

На рисунке 2 представлены результаты для CPU-реализации алгоритма идентификации бликов на процессорах Core2Duo 2.16Ghz и Core2Quad 2.4Ghz и для SMT CUDA-версии на графических ускорителях NVIDIA GeForce 8400 и 9500 с 8 мультипроцессорами, следует отметить что оба этих ускорителя относятся к категории бюджетных. Из рисунка, что даже GPU среднего класса позволяет получить прирост производительности в 10 раз по сравнению с CPU. Использование профессиональных GPU с 32 и более мультипроцессорами позволит получить еще более значительный прирост скорости. Однако, такие ускорители достаточно дороги для установки их на каждом рабочем месте цветокорректора. К тому же, при такой организации вычислительные мощности будут большую часть времени простаивать.

Использование GRID архитектуры позволяет организовать коллективный доступ к многопоточным вычислительным ресурсам непосредственно из системы обработки изображений (PhotoShop). Таким образом, GRID архитектура позволяет оптимизировать загрузку вычислительных мощностей при минимальных затратах на аппаратную часть комплексов.

Использование CUDA очевидно позволяет существенно ускорить обработку изображений, однако переход от CPU программирования к GPU программированию должен учитывать особенность технологии CUDA которые существенно осложняют программирование. Придем те особенности, которые нам пришлось учесть при разработке CUDA реализации алгоритмов ретуши.

CUDA приложения программируются на C++ с использованием специального SDK и представляют собой комбинацию кода который выполняется на CPU и т.н. ядер (kernels)? функций которые в SMT режиме выполняются на GPU. Разработка осложнена отсутствием средств отладки и обработки критических ошибок в GPU коде. Кроме того, т.к. GPU с точки зрения операционной системы (Windows XP) является интерфейсным устройством, то в случае если GPU не отвечает в течении 5 секунд производится перезагрузка драйвера устройства. Таким образом, параллельно выполняемые процедуры не должны выполняться более пяти секунд. Дополнительно при разработке CUDA функций необходимо учитывать наличие на устройстве быстрой и медленной памяти. Следовательно, желательно чтобы все локальные вычисления работали в режиме записи с небольшим объемом (не более 16Кб) локальной памяти.

С учетом сказанного, использование CUDA очень эффективно с точки зрения производительности, но несколько сложной в программировании для задач обработки цветных изображений. Несомненное достоинство CUDA в доступности GPU NVIDIA для массового использования.

Работа выполнена при поддержке Российско-американской программы "Фундаментальные исследования и высшее образование" (BRHE) и РФФИ (проекты № 07-07-00210-а, № 09-07-00269-а).

ЛИТЕРАТУРА:

1. Никоноров А.В., Фурсов В.А. Распределенная вычислительная среда коррекции цифровых изображений. Труды XV Всероссийской научно-методической конференции Телематика 2008, том 1, с.88-89.
2. Nikonogov A. V., Iterative improvement of estimations using multifractal spectra. Proc. of the conf. Concurrent Engineering Resesarch and Application, Dallas, USA July 2005, pp. 407-411.
3. Сойфер, В.А. и др., Методы компьютерной обработки изображений. - М.: ФИЗМАТЛИТ, 2003. - 784 с.
4. Фролов В.В., Введение в технологию CUDA <http://cgm.computergraphics.ru/issues/issue16/cuda>