

АНАЛИЗ И СИНТЕЗ СИГНАЛОВ И ИЗОБРАЖЕНИЙ

УДК 681.786.2

ВЫСОКОСКОРОСТНАЯ ОБРАБОТКА ФАЗОВЫХ ИЗОБРАЖЕНИЙ С ИСПОЛЬЗОВАНИЕМ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ*

С. В. Двойнишников, В. Г. Меледин, В. А. Павлов

*Институт теплофизики им. С. С. Кутателадзе СО РАН,
630090, г. Новосибирск, просп. Академика Лаврентьева, 1
E-mail: dv.s@mail.ru*

Предложен алгоритм обработки фазовых изображений с использованием вычислительных мощностей графических процессоров при измерениях трёхмерной геометрии методом фазовой триангуляции. Показано, что реализация алгоритма расшифровки фазовых изображений на графическом процессоре работает более чем в 100 раз быстрее, чем на центральном процессоре. Практическая реализация этого алгоритма существенно улучшает быстродействие методов фазовой триангуляции.

Ключевые слова: фазовая триангуляция, структурированное освещение, графические процессоры.

DOI: 10.15372/AUT20170206

Введение. Сокращение времени расшифровки фазовых изображений при измерениях трёхмерной геометрии методами фазовой триангуляции [1, 2] является очень важным, поскольку позволяет уменьшить время измерений и повысить точность за счёт статистического накопления экспериментальных данных [3]. Расшифровка фазовых изображений сводится к решению системы трансцендентных уравнений вида [4]

$$I_i(x, y) = A(x, y)(1 + V(x, y) \cos(\varphi(x, y) + \delta_i(x, y))), \quad i \in 1, \dots, N, \quad (1)$$

где $I_i(x, y)$ — распределение интенсивности на i -м изображении контролируемого объекта; $A(x, y)$ — распределение фоновой интенсивности; $V(x, y)$ — средняя видимость; $\varphi(x, y)$ — искомое распределение разности фаз, кодирующее информацию дальности объекта; δ_i — вносимый сдвиг фазы между соседними изображениями структурированной засветки; N — число сдвигов.

Поскольку при обработке данных методом фазовых шагов анализируется фазовый сдвиг проецируемого гармонического сигнала в каждой точке на изображениях измеряемого объекта независимо от интенсивности изображения в соседних точках, перспективно выполнять обработку, используя современные методы распараллеливания вычислений.

Для эффективной реализации параллельной обработки фазовых изображений можно применять классические многопроцессорные системы и многомашинные вычислительные комплексы. Однако построение вычислительных систем требует немалых финансовых затрат. Другой путь решения состоит в использовании альтернативных, но более доступных многопроцессорных систем. Это видеокарты различных производителей, например фирм NVIDIA и AMD.

*Работа выполнена при поддержке Российского фонда фундаментальных исследований (гранты № 16-08-00838, № 16-08-01120).

Сегодня во многих областях науки и техники активно развиваются технологии повышения скорости обработки данных с помощью параллельных вычислений на основе графических процессоров (GPU), в частности методы обработки изображений для повышения производительности вычислительных методов [5]. Известны исследования, посвящённые обработке фазовых изображений и интерферограмм с использованием графических процессоров, которые могут быть применены, в свою очередь, к обработке изображений трёхмерных сцен в методах фазовой триангуляции [6]. Эти исследования ориентированы на узкий класс задач и не могут быть эффективными в методах фазовых шагов в условиях ограниченного по энергии динамического диапазона приёмника оптического излучения, большого количества изображений и неравномерных отражательных свойств поверхности объекта. Тем не менее встречаются публикации, посвящённые обработке фазовых изображений методом фазовых шагов для восстановления трёхмерной геометрии с помощью графических вычислительных процессоров [7, 8]. В них использован подход для определения фазы с помощью трёх фазовых изображений, что существенно ограничивает возможности уменьшения уровня погрешности и области потенциальных применений данных методов.

При реализации метода фазовой триангуляции для измерения трёхмерной геометрии объектов с неравномерными светорассеивающими свойствами поверхности перспективен известный устойчивый метод расшифровки фазовых изображений [9]. Такой метод основан на фильтрации недостоверных измерений и расшифровке фазового сдвига путём регрессионного анализа набора данных с известными произвольно вносимыми сдвигами фазы. Недостаток метода заключается в том, что он не предусматривает параллельную обработку информации.

Цель предлагаемой работы — развить устойчивый метод расшифровки фазовых изображений с произвольными пошаговыми сдвигами для эффективного распараллеливания выполняемых вычислений. Это позволит существенно снизить время обработки экспериментальных данных при измерениях трёхмерной геометрии методами фазовой триангуляции в случае применения технологий параллельных вычислений.

Описание метода. Согласно устойчивому методу расшифровки фазовых изображений [9] сначала проводят фильтрацию экспериментальных данных в целях исключения из анализируемого ансамбля недостоверных измерений, которая выполняется пороговым фильтром с параметром, соответствующим энергетическому диапазону работы фотоприёмника. Пример отфильтрованного фазового изображения поверхности пропеллера вентилятора представлен на рис. 1.

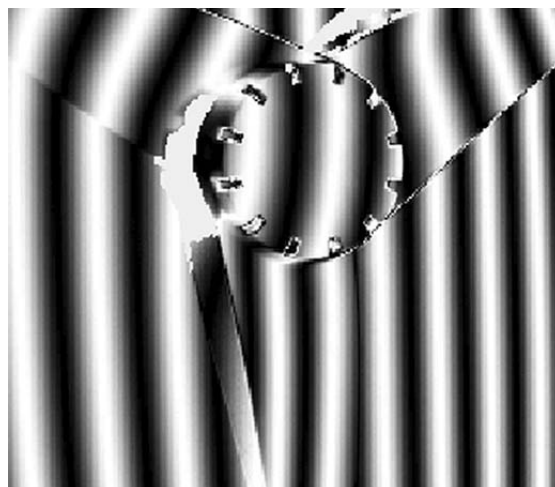


Рис. 1. Фазовое изображение поверхности пропеллера бытового вентилятора после фильтрации недостоверных измерений

Представим выражение (1) в виде

$$I_i(x, y) = K(x, y, i)(A(x, y) + B(x, y) \sin(\delta_i(x, y)) + C(x, y) \cos(\delta_i(x, y))), \quad (2)$$

$$\varphi(x, y) = -\arctan(B(x, y)/C(x, y)), \quad (3)$$

где $K(x, y, i)$ — функция фильтрации, принимающая значение 1 в точках с достоверными результатами измерения и 0 в точках, где результаты измерения не достоверны.

Коэффициенты $A(x, y)$, $B(x, y)$, $C(x, y)$ определяются из условия минимизации функционала невязки между экспериментальными и теоретическими данными $S(A, B, C, x, y)$ для каждого значения (x, y) :

$$S = \sum_{i=1}^N K(x, y, i)(I_i(x, y) - A(x, y) - B(x, y) \sin(\delta_i(x, y)) - C(x, y) \cos(\delta_i(x, y)))^2. \quad (4)$$

Условие минимизации $S(A, B, C)$ — равенство 0 всех частных производных:

$$\frac{\partial S(x, y)}{\partial A} = 0; \quad \frac{\partial S(x, y)}{\partial B} = 0; \quad \frac{\partial S(x, y)}{\partial C} = 0. \quad (5)$$

Поскольку x, y — координаты пикселя на анализируемых фазовых изображениях размерности (X, Y) , то задача сводится к решению (XY) -систем из трёх линейных уравнений:

$$\begin{cases} k_1(x, y)A(x, y) + k_2(x, y)B(x, y) + k_3(x, y)C(x, y) = k_7(x, y), \\ k_2(x, y)A(x, y) + k_4(x, y)B(x, y) + k_5(x, y)C(x, y) = k_8(x, y), \\ k_3(x, y)A(x, y) + k_5(x, y)B(x, y) + k_6(x, y)C(x, y) = k_9(x, y), \end{cases} \quad (6)$$

где

$$k_1(x, y) = \sum_{i=1}^N K(x, y, i);$$

$$k_2(x, y) = \sum_{i=1}^N K(x, y, i) \cos(\delta_i(x, y));$$

$$k_3(x, y) = \sum_{i=1}^N K(x, y, i) \sin(\delta_i(x, y));$$

$$k_4(x, y) = \sum_{i=1}^N K(x, y, i) \cos^2(\delta_i(x, y));$$

$$k_5(x, y) = \sum_{i=1}^N K(x, y, i) \sin(\delta_i(x, y)) \cdot \cos(\delta_i(x, y));$$

$$k_6(x, y) = \sum_{i=1}^N K(x, y, i) \sin^2(\delta_i(x, y));$$

$$k_7(x, y) = \sum_{i=1}^N K(x, y, i) I_i(x, y);$$

$$k_8(x, y) = \sum_{i=1}^N K(x, y, i) I_i(x, y) \cos(\delta_i(x, y));$$

$$k_9(x, y) = \sum_{i=1}^N K(x, y, i) I_i(x, y) \sin(\delta_i(x, y)).$$

Для эффективного использования параллельных вычислений необходимо минимизировать или совсем исключить одновременный доступ разных потоков к одним и тем же данным, поскольку коллизии доступа между потоками могут привести к снижению производительности. При расшифровке фазовых изображений требуется независимый анализ фазы на различных пикселях изображения. Это достигается за счёт параллельного решения линейных уравнений для различных пар (x, y) в выражении (6) путём размещения их в различные потоки.

Экспериментальные результаты. Продемонстрирована работоспособность предложенного подхода. Созданы две программы, реализующие устойчивый метод обработки фазовых изображений с произвольным пошаговым сдвигом: для выполнения на центральном процессоре (CPU) и на GPU с применением технологии CUDA. В обоих случаях использованы числа с плавающей точкой одинарной точности (тип float, размер 4 байта). Данный тип выбран в связи с тем, что графический процессор NVIDIA GeForce GTX 970 оптимизирован для выполнения арифметических операций именно с такими переменными. Производительность вычислений существенно ниже для переменных с плавающей точкой двойной точности.

Анализ работы алгоритма проводился на компьютере с центральным процессором Intel Core i7-4790K 4 GHz, оперативной памятью 16 Гб и графическим процессором NVIDIA GeForce GTX 970. Корректность расчётов реализаций осуществлялась путём прямого сравнения результатов работы алгоритмов между собой. Установлено, что результаты вычислений полностью совпали.

Проведены эксперименты по оценке времени выполнения алгоритма на GPU в зависимости от параметров запуска GPU-вычислений. Размер анализируемых фазовых изображений составляет 320×240 пикселей. Количество анализируемых фазовых изображений с различными сдвигами начальной фазы равно 200. Графический процессор NVIDIA GeForce GTX 970 имеет 1664 вычислительных ядра CUDA, физически расположенных в 13 мультипроцессорах [10]. Для вычисления фазовых сдвигов на изображениях с разрешением 320×240 необходимо выполнить не более 76800 вычислений, которые можно эффективно распараллелить. Поскольку измеряемый объект, как правило, занимает часть анализируемого изображения, часто количество анализируемых точек M оказывается существенно меньше, чем величина XY . Далее выполнялось вычисление фазы при $M = 10000$. Такое число точек близко к типичному количеству измерений методом фазовой триангуляции.

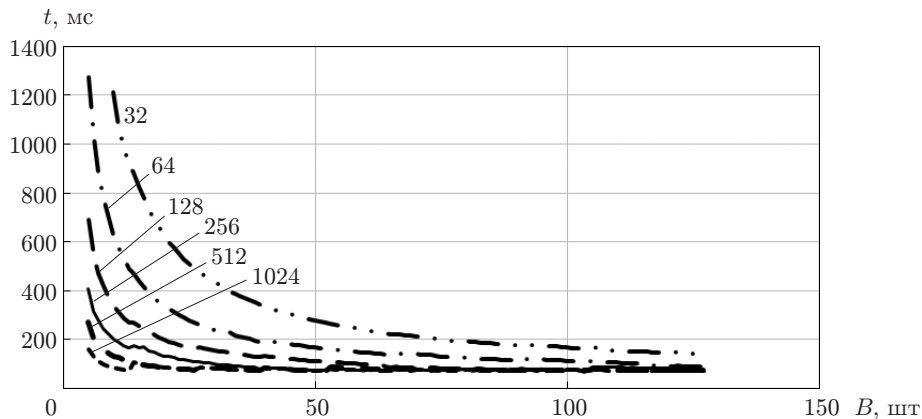


Рис. 2. Зависимость времени выполнения алгоритма t на GPU от количества используемых блоков B . Кривые соответствуют различным количествам потоков в каждом блоке

Так как независимых вычислений нужно выполнить существенно больше, чем количество ядер CUDA, требуется оптимально распределить вычисления по ядрам. Программный интерфейс CUDA предусматривает объединение потоков в блоки [11], имеющие распределённые ресурсы. Ввиду того что внутренняя архитектура мультимикропроцессоров CUDA является закрытой, аналитически определить оптимальное количество потоков и блоков для запуска вычислений затруднительно. Более достоверно в этом случае выполнить экспериментальный анализ зависимости времени обработки данных от количества исполняемых блоков и потоков.

Был проведён ряд экспериментов по получению оптимального количества блоков и потоков при запуске параллельных вычислений с помощью CUDA в целях минимизации времени обработки данных. На рис. 2 представлена зависимость времени обработки от количества блоков. Из графиков видно, что для снижения времени обработки необходимо увеличение количества используемых блоков и потоков.

На рис. 3 показана зависимость времени выполнения алгоритма на GPU от большего количества блоков. Здесь приведена конфигурация с количеством 32 потока в каждом блоке. График демонстрирует, что при выборе количества блоков больше чем 512 время обработки данных не меняется.

На рис. 4 изображены графики зависимости времени обработки данных от количества используемых потоков. Из графиков следует, что эффективным является применение более 400 потоков.

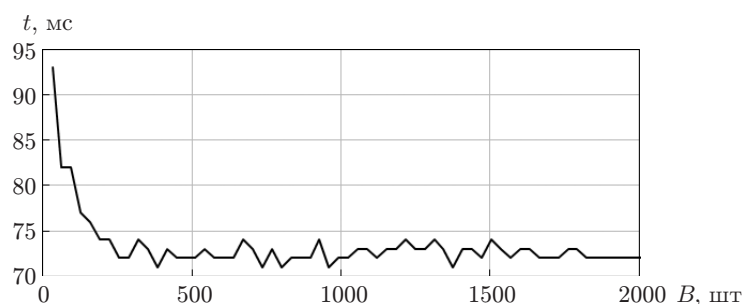


Рис. 3. Зависимость времени выполнения алгоритма t на GPU от количества используемых блоков B

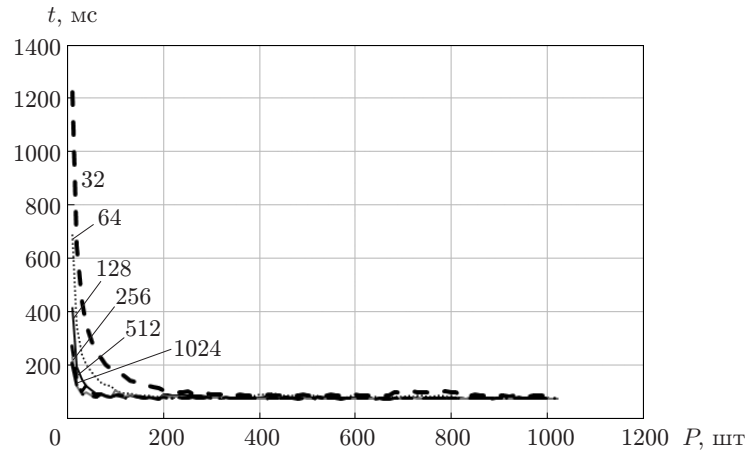


Рис. 4. Зависимость времени выполнения алгоритма t на GPU от количества используемых потоков в каждом блоке P . Кривые соответствуют различным количествам блоков

На рис. 5 представлены зависимости времени обработки данных на CPU и GPU от линейного размера квадратных изображений, которые обрабатывались с помощью CUDA с использованием 512 блоков и 1024 потоков в каждом блоке.

Полученные экспериментальные результаты показывают, что для реализации устойчивого метода расшифровки фазовых изображений с произвольными пошаговыми сдвигами наиболее оптимальным оказывается запуск вычислений на графических процессорах в конфигурации 512 блоков и более 400 потоков в каждом блоке. При этом важно, чтобы выполнялось условие

$$BP \geq M. \quad (7)$$

В случае если неравенство не решается, в каждом ядре осуществляется цикл обработки данных в нескольких точках фазовых изображений. Такой подход оказался менее эффективным, чем запуск большего количества потоков, каждый из которых обрабатывает не более одной точки. Этот эффект наблюдается на рис. 2–4: увеличение времени обработки данных в левой части графиков связано с режимом работы программы, когда условие (7) не выполняется.

Поскольку физически количество вычислительных ядер на использованном процессоре существенно меньше, то видно, что технология CUDA перераспределяет вычислительные ресурсы близко к оптимальному способу. Следует заметить, что в данном случае мы задействовали кэш-память первого и второго уровней, память графического устройства и

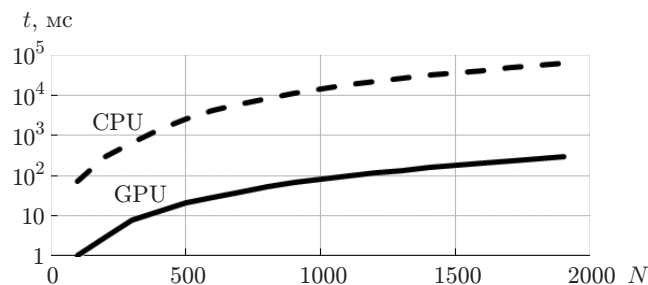


Рис. 5. Сравнение времени обработки данных t на CPU и GPU в зависимости от линейного размера фазовых изображений N

не использовали разделяемую память в пределах одного блока. Очевидно, что при реализации алгоритмов обработки данных, применяющих разделяемую память, оптимальные конфигурации будут другими.

Заключение. Предложен алгоритм параллельной обработки фазовых изображений для измерения трёхмерной геометрии методом фазовой триангуляции. Этот алгоритм основан на устойчивом методе расшифровки фазовых изображений с фильтрацией экспериментальных данных, методах гармонической регрессии и распараллеливания вычислений фазы в различных точках анализируемых изображений. Такой алгоритм может быть эффективно использован для вычислений на графических процессорах. Показано, что алгоритм расшифровки фазовых изображений на GPU работает в 100 раз быстрее, чем на CPU. Полученные результаты демонстрируют перспективность применения данного подхода для практической реализации методов фазовой триангуляции.

СПИСОК ЛИТЕРАТУРЫ

1. **Двойнишников С. В., Куликов Д. В., Меледин В. Г.** Оптоэлектронный метод бесконтактного восстановления профиля поверхности трёхмерных объектов сложной формы // Метрология. 2010. № 4. С. 15–27.
2. **Двойнишников С. В., Меледин В. Г., Шпольвинд К. В.** Метод компенсации нелинейности тракта источник-приёмник оптического излучения при 3D-измерениях на основе фазовой триангуляции // Измерительная техника. 2012. № 2. С. 12–16.
3. **Чугуй Ю. В.** Трёхмерные оптико-электронные измерительные системы и лазерные технологии для научных и промышленных применений // Автометрия. 2015. **51**, № 4. С. 76–91.
4. **Gruber M., Häusler G.** Simple, robust and accurate phase-measuring triangulation // Optik. 1992. **89**, N 3. S. 118–122.
5. **Asano S., Maruyama T., Yamaguchi Y.** Performance comparison of FPGA, GPU and CPU in image processing // Proc. of the 19th Intern. Conf. on Field Programmable Logic Applications. Prague, Czech Republic, 2009. P. 126–131.
6. **Kang H., Yamaguchi T., Yoshikawa H. et al.** Acceleration method of computing a compensated phase-added stereogram on a graphic processing unit // Appl. Opt. 2008. **47**, Is. 31. P. 5784–5789.
7. **Karpinsky N., Hoke M., Chen V., Zhang S.** High-resolution, real-time three-dimensional shape measurement on graphics processing unit // Opt. Eng. 2014. **53**, Is. 2. 024105.
8. **Espinosa-Romero A., Legarda-Saenz R.** GPU based real-time quadrature transform method for 3-D surface measurement and visualization // Opt. Express. 2011. **19**, Is. 13. P. 12125–12130.
9. **Двойнишников С. В.** Устойчивый метод расшифровки интерферограмм с пошаговым сдвигом // Компьютерная оптика. 2007. **31**, № 2. С. 21–25.
10. **Видеокарта GeForce GTX 970.** URL: <http://www.nvidia.ru/object/geforce-gtx-970-ru> (дата обращения: 06.02.2017).
11. **Farber R.** CUDA Application Design and Development. Waltham: Morgan Kaufmann, 2011. 336 p.

Поступила в редакцию 12 июля 2016 г.