

ГИБРИДНЫЙ ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ СОПОСТАВЛЕНИЯ СТЕРЕО-ИЗОБРАЖЕНИЙ В ЗАДАЧЕ РЕКОНСТРУКЦИИ 3D-СЦЕН

В.А. Фурсов¹, Е.В. Гошин², А.П. Котов¹

¹ Самарский государственный аэрокосмический университет им. академика С.П.Королева

² Институт систем обработки изображений РАН

1. Введение

Задача реконструкции 3D-сцен востребована в современных системах технического зрения. В ряде приложений, например, при обработке данных дистанционного зондирования Земли (ДЗЗ) ставится задача оперативной обработки чрезвычайно больших изображений. Технология реконструкции 3D-сцены основана на том, что для каждой точки на одном изображении стереопары осуществляется поиск соответствующей ей точки на другом изображении. Наиболее ресурсоемким в этой технологии является этап сопоставления фрагментов изображений с целью определения соответствующих точек. Поскольку обрабатываемые изображения ДЗЗ могут иметь разрешение более 100 миллионов пикселей, актуальной является задача построения быстродействующих параллельных алгоритмов нахождения соответствующих точек и их реализация на гибридных вычислительных устройствах, включающих графические процессоры.

Соответствующие точки на стереоизображениях находятся на так называемых эпиполярных линиях, которые могут быть определены с использованием заданной или вычисленной фундаментальной матрицы. Обычно этапу определения соответствующих точек предшествует этап ректификации изображений, в результате которого эпиполярные линии преобразуются в параллельные прямые. В настоящей работе используется рассматривавшийся в работе [1] алгоритм, в котором в явном виде ректификация не проводится. В работе [2] была предложена параллельная реализация этого алгоритма с использованием технологии CUDA.

Однако, в указанной работе внутренний параллелизм алгоритма не был реализован в полной степени, поскольку один из этапов (формирование векторов дескрипторов) не может быть эффективно распараллелен на графическом процессоре. В настоящей работе предлагается схема гибридной параллельной реализации, позволяющей ускорить выполнение алгоритма в целом. Предложенная схема обладает высокой степенью параллелизма и позволяет выполнять вычисления на нескольких CPU и одной GPU.

2. Описание алгоритма, формулировка задачи

В работе используется модель камеры-обскуры [3]. Предполагается, что реконструкция 3D-сцены осуществляется по стереоизображениям, полученным с двух камер. Пусть $\mathbf{M}$ – координаты некоторой точки в глобальной системе координат. Координаты этой точки в системах координат первой и второй камер определяются как

$$\mathbf{m}_1 = \mathbf{K}_1 [\mathbf{R}_1; \mathbf{t}_1] \mathbf{M}$$

$$\mathbf{m}_2 = \mathbf{K}_2 [\mathbf{R}_2; \mathbf{t}_2] \mathbf{M}$$

Здесь $\mathbf{R}_1$ $\mathbf{R}_2$ – матрицы размерности 3×3 , описывающие поворот систем координат первой и второй камер относительно глобальной, а $\mathbf{t}_1$ $\mathbf{t}_2$ – координаты начала глобальной системы координат в системах координат первой и второй камер. $\mathbf{K}_1$ $\mathbf{K}_2$ – матрицы внутренних параметров камер:

$$\mathbf{K}_1 = \begin{bmatrix} f_1 & 0 & u_{10} \\ 0 & f_1 & v_{10} \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{K}_2 = \begin{bmatrix} f_2 & 0 & u_{20} \\ 0 & f_2 & v_{20} \\ 0 & 0 & 1 \end{bmatrix}$$

где f_1 , f_2 – фокусные расстояния камер, (u_{10}, v_{10}) , (u_{20}, v_{20}) – координаты главных точек камер в системах координат, связанных с камерами [3].

Фундаментальная матрица задает соотношение между точками

$$\mathbf{m}_1 = \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}, \quad \mathbf{m}_2 = \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

на изображениях

$$\mathbf{m}_2^T \mathbf{F} \mathbf{m}_1 = 0$$

$$\text{где } \mathbf{F} = \begin{bmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{bmatrix}.$$

Используется тот факт, что каждой точке $\mathbf{m}_1$ с координатами (x_0, y_0) на первом изображении соответствует эппиполярная прямая $\mathbf{l}_2: a_2 x_2 + b_2 y_2 + c_2 = 0$ на втором:

$$(x_0, y_0) \rightarrow \begin{pmatrix} F_{11} & F_{12} & F_{13} \\ F_{21} & F_{22} & F_{23} \\ F_{31} & F_{32} & F_{33} \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

$$a = x_0 F_{11} + y_0 F_{12} + F_{13}$$

$$b = x_0 F_{21} + y_0 F_{22} + F_{23}$$

$$c = x_0 F_{31} + y_0 F_{32} + F_{33}$$

Полученной эппиполярной прямой $\mathbf{l}_2$ на втором изображении соответствует эппиполярная прямая $\mathbf{l}_1: a_1 x_1 + b_1 y_1 + c_1 = 0$ на первом изображении.

В работе [1,2] рассмотрен алгоритм нахождения соответствующих точек на парах эппиполярных линий. Кратко он сводится к следующему. На протяжении каждой эппиполярной линии с заданным шагом выбираются точки. Каждая точка описывается некоторым признаком или набором признаков. В качестве признаков используются значения яркостей и норм величин градиентов, вычисленных путем билинейной интерполяции значений этих характеристик по четырем ближайшим пикселям, расположенным в вершинах единичного квадрата:

$$f(x, y), \quad |\text{grad}_x f(x, y)|, \quad |\text{grad}_y f(x, y)|.$$

Таким образом, каждая точка характеризуется тремя параметрами, а эппиполярная линия представляет собой вектор таких троек, который будем называть вектором дескрипторов.

Фрагментом вектора дескрипторов являются последовательно расположенные N значений признаков. Сопоставление производится путем поиска наиболее сходных фрагментов для векторов первого и второго изображений. В качестве меры сходства используется расстояние между векторами. В качестве соответствующих точек $\mathbf{m}_1$, $\mathbf{m}_2$ принимаются точки на эппиполярных линиях, соответствующие центральным значениям дескрипторов на выбранных фрагментах.

Оценка $\hat{\mathbf{M}}$ пространственных координат точки $\mathbf{M}$ определяется как точка пересечения лучей $(\mathbf{c}_1, \hat{\mathbf{m}}_1)$, $(\mathbf{c}_2, \hat{\mathbf{m}}_2)$, где $\mathbf{c}_1$, $\mathbf{c}_2$ – центры камер. Множество таких точек, дающих полное описание трехмерной модели сцены, получают путем последовательного перебора эппиполярных линий.

В настоящей работе ставится задача эффективной гибридной параллельной реализации описанной выше технологии реконструкции 3D-сцен.

3. Гибридная параллельная схема реализации алгоритма

Рассмотрим общую схему вычислений в соответствии с технологией, описанной выше. Алгоритм вычисления решения является параллельным по обрабатываемым эппиполярным линиям. В работе [2], с помощью технологии CUDA на графическом процессоре, была реализована параллельная схема сопоставления фрагментов для уже сформированных эппиполярных линий. При этом формирование линий происходило последовательно, поскольку требовало большого числа операций интерполяции и загрузки в память всего изображения. Такая процедура не может быть эффективно реализована с использованием технологии CUDA. Для повышения быстродействия алгоритма предлагается формирование эппиполярных линий производить параллельно на нескольких CPU, а процесс сопоставления также выполнять на одной GPU. Таким образом, при формировании эппиполярных линий происходит декомпозиция исходных изображений по данным, которые рассылаются на узлы кластера. Далее выбираются координаты эппиполярных линий, и на каждом процессоре

вычисляются вектора дескрипторов для выбранной пары линий. На рисунке 1 приведена блок-схема гибридной параллельной реализации алгоритма.

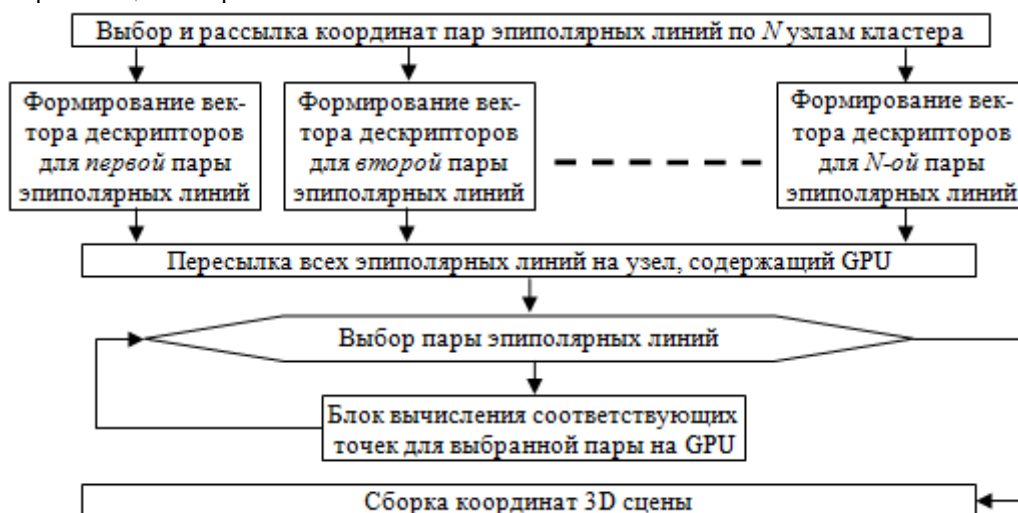


Рис. 1. Схема гибридной параллельной реализации алгоритма

После того как для всех линий сформированы вектора дескрипторов, информация полученная со всех узлов кластера пересылается на узел, содержащий GPU, чтобы избежать лишних пересылок между узлами. Таким образом, далее алгоритм выполняется на одном узле, но уже с использованием графического процессора. При решении задачи сопоставления фрагментов, для каждой пары эпиполярных линий последовательно будет выполняться запуск ядра CUDA на GPU. На рисунке 2 показана двухэтапная параллельная схема реализации блока алгоритма. В соответствии с приведенной схемой запуск CUDA-ядра осуществляется дважды.

На первом этапе формируется матрица сходства точек. В качестве элемента i -ой строки j -ого столбца формируется значение меры сходства дескрипторов i -ой точки на первом изображении и j -ой точки на втором изображении. Если дескрипторами являются значения яркости в точках, то в качестве меры сходства принимается разность между этими яркостями. В случае использования градиентов, как дополнительных дескрипторов, в качестве меры сходства принимается сумма разностей значений дескрипторов.



Рис. 2. Параллельная схема реализации блока алгоритма на CUDA

На каждой нити вычисляется разность значений дескрипторов для выбранной точки на одном изображении и точки на втором изображении. Результатом работы второго этапа является матрица сходства фрагментов, на основе которой осуществляется поиск всех соответствующих точек для выбранной пары эпиполярных линий.

4. Экспериментальные исследования

В качестве объекта эксперимента взята трехмерная модель плоскости с расположенными на ней предметами, использовавшаяся в работе [2]. Полученные координаты соответствующих точек моделируются в программе трассировки лучей POV-Ray. Камеры имеют параметры

$$\mathbf{K}_1 = \mathbf{K}_2 = \mathbf{K} = \begin{bmatrix} 400 & 0 & 400 \\ 0 & 400 & 300 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{R}_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_2 = \begin{bmatrix} 0,984808 & 0 & -0,173648 \\ 0 & 1 & 0 \\ 0,173648 & 0 & 0,984808 \end{bmatrix}, \quad \mathbf{c}_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{c}_2 = \begin{bmatrix} -1,25 \\ 0 \\ -1,25 \end{bmatrix}.$$

На рисунке 3 приведена пара полученных стереоизображений.

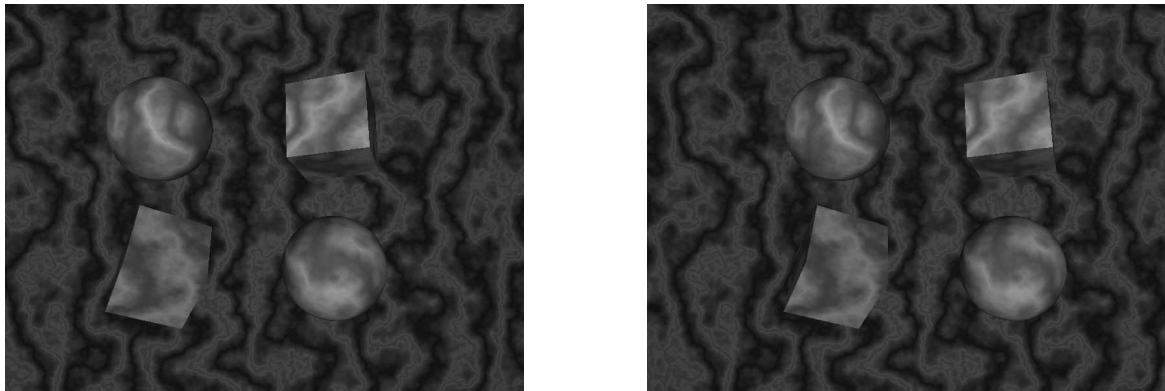
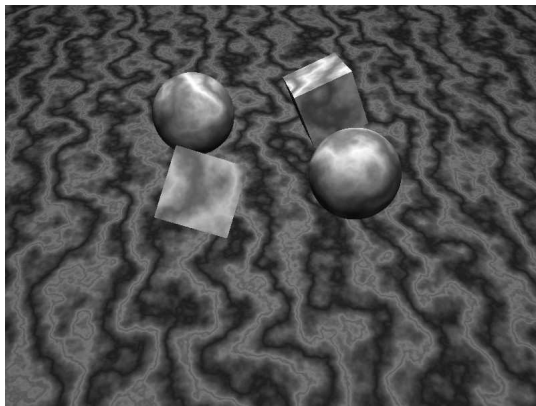


Рис. 3. Пара стереоизображений

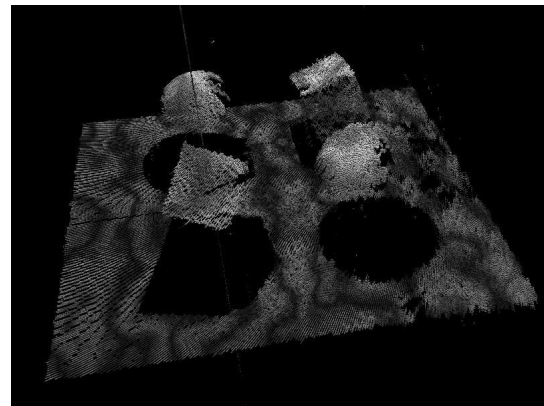
По известным параметрам камер была вычислена фундаментальная матрица

$$\mathbf{F} = \begin{bmatrix} 0 & -5,725 \cdot 10^{-15} & 0,017175 \\ 4,94192 \cdot 10^{-5} & 0 & -0,0540064 \\ -0,0148258 & 0,050673 & 1 \end{bmatrix}$$

С помощью показанного на рисунке 1 алгоритма было сформировано множество пар соответственных точек и вычислены соответствующие этим точкам координаты сцены в трехмерном пространстве. На рисунках 4, а и б для иллюстрации работы алгоритма приведены изображения исходной и восстановленной сцены.



а



б

Рис. 4. Исходное (а) и восстановленное (б) изображения

Эксперименты проводились на суперкомпьютере «Сергей Королёв» с использованием процессоров Intel Xeon X5670 и графического процессора Nvidia Tesla c2070. Результаты экспериментов продемонстрированы в таблице 1. При последовательной реализации алгоритм полностью выполнялся на одном CPU, при реализации CPU+GPU происходило распараллеливание процедур, указанных в строчках 2 и 3, по схеме описанной в работе [2]. В данной работе дополнительно распараллелена процедура в строчке 1 на нескольких CPU.

Таблица 1. Время работы последовательной и параллельных реализаций

№ п/п	Выполняемая процедура	Время работы (мс)		
		Последовательная реализация (CPU)	Реализация CPU+GPU	Реализация 12 CPU + GPU

1	Предварительный этап формирования векторов дескрипторов эпиполярных линий	149	149	26
2	Формирование матрицы сходства единичных фрагментов	697	37	37
3	Формирование матрицы сходства составных фрагментов	457	36	36
4	Выбор соответственных точек	342	342	342

Применение гибридной параллельной схемы (12 CPU + GPU) позволило значительно сократить время выполнения процедур, указанных в строках 1,2,3. Общее время на одном CPU составило 1645 мс, а на гибридной системе – 441 мс. Таким образом, получено ускорение в четыре раза.

5. Заключение

Развита параллельная схема алгоритма реконструкции 3-D сцены по стереопаре изображений, опубликованная в работе [2], в частности, предложена гибридная параллельная схема алгоритма с использованием MPI и CUDA. В отличие от указанной работы реализовано распараллеливание предварительного этапа формирования векторов дескрипторов эпиполярных линий с помощью нескольких CPU. Время выполнения этого этапа уменьшено в шесть раз. Результаты для этапов сопоставления фрагментов, реализация которых, как и прежде, производится на одной GPU, остались прежними. Таким образом, в целом при использовании гибридной вычислительной схемы достигнуто ускорение в четыре раза.

Благодарности

Работа выполнена при поддержке РФФИ (проекты № 12-07-00581, № 13-07-12030 офи_м, № 13-07-13166).

ЛИТЕРАТУРА:

1. Фурсов В.А., Гошин Е.В., Бибилов С.А. Реконструкция 3D-сцен на пучках эпиполярных плоскостей стереоизображений // Мехатроника, автоматизация, управление, 2013, №9 (150), С. 19-24
2. Фурсов В.А., Гошин Е.В., Котов А.П. Параллельная CUDA-реализация алгоритма сопоставления стереоизображений // Вычислительные методы и программирование, 2014, №15, С. 154-161
3. Форсайт Д., Понс Ж. Компьютерное зрение. Современный подход. М.: Издательский дом "Вильямс", 2004. - 928 с.
4. Грузман И.С., Киричук В.С., Косых В.П., Перетягин Г.И., Спектор А.А. Цифровая обработка изображений в информационных системах: Учебное пособие. Новосибирск: Изд-во НГТУ, 2002. - 352 с.
5. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. М.: Изд-во ДМК Пресс, 2010. – 234 с.