

3. Bradski Gary, Kaehler Adrian Learning OpenCV. – O’Reilly, 2008.
4. Intel® Threading Building Blocks. Tutorial. Revision: 1.13. – Intel Corporation, 2007.
5. OpenCL Best Practices Guide. Version 1.0. – NVIDIA Corporation, 2009.
6. The OpenCL specification. Version: 1.0. – Khronos OpenCL Working Group, 2009.
7. Мееров И.Б., Сиднев А.А., Сысоев А.В. Библиотека Intel Threading Building Blocks – краткое описание. – Н. Новгород, 2007.
8. Федоткин М.А. Основы прикладной теории вероятностей и статистики. – М.: Высшая школа, 2006.
9. Эндрюс Г.Р. Основы многопоточного, параллельного и распределенного программирования. – М.: Вильямс, 2003.

Б.Г. Севрюков, А.А. Лукьяница

Московский государственный университет им. М.В. Ломоносова

ИСПОЛЬЗОВАНИЕ GPU В ЗАДАЧАХ ТРЕХМЕРНОЙ РЕКОНСТРУКЦИИ СЦЕН

Задачи определения пространственной структуры реальных объектов или сцен по двумерным изображениям ставятся в разнообразных областях, таких как: компьютерное моделирование, навигация роботов по заранее неизвестной местности, кино и телевидение, медицина, архитектура. За последнее десятилетие методы решения этого класса задач получили существенное развитие, однако значительный объем вычислений, необходимый для получения качественной реконструкции, не позволяет использовать их в системах реального времени.

С другой стороны, вычислительная мощь современных видеокарт, невысокая цена и общая доступность делают их привлекательными для решения указанных задач. Поэтому нами было проведено исследование возможности переноса вычислений на GPU. Устройство, предназначенное для синтеза двумерных изображений, мы будем использовать для решения обрат-

ной задачи. Мы рассматриваем класс методов реконструкции, использующих два и более изображения. Общий процесс вычислений может быть представлен в виде конвейера, схема которого, а также методы и оценка возможности их эффективной реализации на видеокарте, приведены в следующей таблице.

Общий процесс вычислений

Номер этапа	Название этапа	Методы	GPU
0	Формирование входного набора изображений, $n \geq 2$		
1	Выделение особых точек	SIFT	+
		SURF	+
		Детектор углов Харриса	+
2	Поиск соответствий между изображениями	Трекер особых точек Лукаса-Канаде	+
		Оптический поток	+
		KD-дерево	+
3	Вычисление фундаментальных матриц, факторизационные методы	SVD	+
		QR	+
		RANSAC	+/-
4	Самокалибровка в случае неизвестного фокусного расстояния	SVD	+
5	Выравнивание пучков (Bundle adjustment)	-	+/-
6	Создание плотной реконструкции	Триангуляция Делоне	+
		Ректификация и сопоставление в случае стерео	+
		Поиск минимального сечения на графе	+
7	Наложение текстуры и отрисовка	-	+

В зависимости от условий процесс получения решения может меняться – некоторые этапы конвейера могут отсутствовать либо быть заменены другими. Например, при отсутствии информации о камерах для получения евклидовой реконструкции за этапом 3 должен следовать этап самокалибровки.

Отметим, что в данной задаче параллелизм может быть двух типов: массивный параллелизм по данным – когда распараллеливаются операции внутри каждого из блоков реконструкции для фиксированного набора ракурсов и параллелизм по наборам ракурсов, т.е. когда весь конвейер или отдельные его блоки выполняются параллельно для наборов ракурсов, а затем результаты реконструкции «склеиваются». Яркий пример второго случая – последовательная реконструкция по видео. Естественно, при должной организации процесса возможно комбинирование.

Основным препятствием на пути эффективного ускорения метода RANSAC является то обстоятельство, что он требует случайной выборки значений из памяти видеокарты, в результате чего доступ к памяти перестает быть согласованным между нитями. Для борьбы с этим можно предложить заранее помещать в память $n \cdot k$ значений, где n – количество итераций алгоритма, k – длина вектора значений (например, 7 двумерных точек в случае вычисления фундаментальной матрицы).

Процедура выравнивания пучков – процесс, построенный на методе Левенберга-Марквардта. Предположительно, он может быть достаточно эффективно распараллелен на GPU, однако довольно сложен и достоин отдельного исследования. Пока же стоит отметить, что он не является необходимым этапом конвейера реконструкции.

Результаты наших исследований показали, что состояние дел в области вычислений общего GPU позволяет использовать видеокарты для достижения эффективной производительности при решении задач реконструкции трехмерных сцен по изображениям.