

0.1. Назаров Н.А. Torch PIV: Фреймворк для анализа двумерных PIV экспериментов с поддержкой CUDA

Метод PIV (Particle Imaging Velocimetry или лазерная анемометрия по изображениям частиц) широко используется для бесконтактного измерения полей скорости потока. Этот метод основан на вычислении перемещения групп специальных трассеров на последовательных изображениях посредством корреляционного анализа для определения скорости течения среды. PIV эксперименты зачастую требуют анализа большого количества изображений для определения статистических характеристик потока, поэтому для ускорения анализа данных в этой области активно применяются графические процессоры. Одна из первых GPU реализаций кросскорреляционного метода PIV была представлена еще в 2004 году [1]. Для первых реализаций GPU PIV все этапы корреляционного алгоритма были реализованы без использования специализированных библиотек. Позже, с появлением технологии CUDA, стали появляться более высокоуровневые, гибкие и сложные решения задачи PIV, такие как OpenPIV [2] или [3] предназначенные непосредственно для анализа экспериментальных данных. Однако в таких проектах отсутствуют стабильные GPU версии. Например, GPU версия OpenPIV библиотек все еще находится в beta режиме и давно не развивается. От пользователя требуется сборка библиотек, и некоторые сторонние ресурсы, необходимые для работы GPU OpenPIV, на сегодняшний день не поддерживаются.

В работе продемонстрированы возможности применения высокоуровневых библиотек для анализа данных с поддержкой GPU вычислений на языке Python. Была разработана библиотека Torch PIV для анализа 2D PIV экспериментов на основе фреймворка глубокого обучения PyTorch с поддержкой CUDA. Библиотека реализует итерационный кросскорреляционный БПФ PIV алгоритм со смещением окна поиска, последующей валидацией векторов скорости, а также интерполяцией эффекта потери пары. Выбранная реализация не требует от пользователя сборки, имеет компактную кодовую базу, способна работать как на CPU, так и на GPU в зависимости от выбора пользователя, а также обладает гибкостью Python модуля. Реализация на Python позволяет использовать модуль в качестве бэкенда для более сложных систем обработки экспериментальных данных, а базовая библиотека для данного метода является одной из самых надежных и распространенных в области машинного обучения. Кроме того, в библиотеку встроен упрощенный Qt интерфейс, позволяющий визуализировать полученные поля скорости.

В ходе работы было проведено сравнение производительности CPU версии разработанного метода

с Open Source аналогами. Показано, что основные функции разработанного модуля могут быть выполнены на GPU со скоростью CUDA реализаций. Разработанная библиотека прошла классический цикл валидации на синтетических изображениях и открытых базах экспериментальных данных [4]. Было получено хорошее соответствие между полученными распределениями скоростей и тестовыми значениями.

Работа выполнена при финансовой поддержке Мегагрант (Соглашение № 075-15-2021-575).

Научный руководитель — д.ф.-м.н. Терехов В. В.

Список литературы

- [1] SCHWIETZ T., WESTERMANN R. GPU-PIV // Proc. Intern. Conf. «Vision, Modeling And Visualization». Stanford, USA: Ios Press, 2004.
- [2] DALLAS C., WU M., CHOU V., ET AL. Graphical Processing Unit-Accelerated Open-Source Particle Image Velocimetry Software for High Performance Computing Systems // J. Fluids Eng. 2019. Vol. 141. N. 11. P. 111401.
- [3] AGUILAR-CABELLO J., PARRAS L., DEL PINO C. DPIVSoft-OpenCL: A multicore CPU-GPU accelerated open-source code for 2D Particle Image Velocimetry // SoftwareX. 2022. Vol. 20. N. 3. P. 101256.
- [4] NEAL D.R., SCIACCHITANO A., SMITH B.L., ET AL. Collaborative framework for PIV uncertainty quantification: the experimental database // Meas. Sci. Technol. 2015. Vol. 26. N. 7. P. 074003.