



Рис. 3. Пример масок для найденных волокон

Маски помогают выделить каждое волокно в отдельный объект, что позволяет анализировать их геометрические параметры.

Таким образом, системы компьютерного зрения способствует значительному ускорению исследования полимерных волокон за счет применения сверточных нейросетей, которые автоматизируют процесс выделения объекта исследования из изображения, а благодаря битовым маскам, полностью повторяющих форму и размеры объекта, возможен их геометрический анализ.

В процессе разработки было создано приложение, позволяющее производить поиск волокон на фотографиях, выделять их и давать геометрический анализ. Приложение может записывать данные отмеченных волокон в текстовый файл.

ПРИМЕНЕНИЕ ТЕХНОЛОГИИ MPI ДЛЯ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ

А. В. Браим

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Н. В. Самовендюк

История технологии MPI начинается с начала 90-х гг. Для того времени написание приложений, реализующих параллелизм, для различных архитектур было достаточно сложной задачей. Несмотря на то, что существовало множество библиотек, способных облегчить разработку приложений, использующих параллельные вычисления, не было никакого единого стандарта. Одной из самых распространенных моделей работы этих библиотек была модель передачи сообщений между процессами. Суть этого метода заключается в том, что главный процесс может назначать работу подчиненным процессам, передавая им сообщение. В апреле 1992 г. состоялся семинар, где разработчики обсудили основные функции, необходимые для интерфейса передачи сообщений (MPI – Message Passing Interface), а в ноябре этого же года на конференции *Supercomputing* был принят стандарт. Первая версия стандарта была выпущена в июне 1994 г. Стандартизацией MPI занимается MPI Forum. На данный момент последняя версия стандарта MPI-3.1 вышла в июне 2015 г.

Существует множество реализаций MPI. Одни из самых популярных:

- MPICH;
- Open MPI;
- Intel MPI;
- Microsoft MPI;
- mpi4py.

MPICH и Open MPI – одни из самых первых свободных реализаций MPI, Intel MPI – коммерческая реализация MPI от Intel, Microsoft MPI – реализация MPI от Microsoft, mpi4py – реализация MPI для языка Python.

В основе MPI лежит понятие коммуникатора. Коммуникатор определяет группу процессов, которые могут общаться между собой. В этой группе каждому процессу присваивается уникальный ранг. Основа коммуникации основана на операциях отправки и получения сообщений. Процесс может отправить сообщение другому процессу, указав ранг процесса и уникальный тег для идентификации сообщения. Получатель, в свою очередь, получает сообщение от указанного процесса. Такие взаимодействия называются point-to-point связью. Примеры point-to-point команд: MPI_Send, MPI_Receive (отправление и получение сообщения, соответственно). В MPI есть так называемые коллективные функции. Например, функция MPI_Bcast, которая позволяет сделать рассылку сообщения с одного процесса на все остальные процессы в группе. Или MPI_Barrier, который останавливает задачи процесса до тех пор, пока не будет вызвана эта функция на остальных процессах, подсоединенных к указываемому коммуникатору.

При работе с MPI следует учитывать, что под параллельной программой понимается множество одновременно выполняемых процессов (ветвей). В общем случае процессы выполняются на разных процессорах. Каждый процесс параллельной программы порождается на основе копии одного и того же программного кода.

Рассмотрим преимущества и недостатки MPI.

К преимуществам MPI относится следующее:

- является стандартом, что означает единый набор функций для различных реализаций;
- помогает решить проблему переносимости параллельных программ между разными компьютерными системами;
- позволяет создавать хорошо масштабируемые параллельные программы.

Выделяют такие недостатки MPI:

- является низкоуровневым инструментом программиста;
- не предоставляет механизмов задания начального размещения процессов по процессорам;
- полномасштабная отладка MPI-программ затруднительна вследствие одновременного исполнения нескольких программных ветвей.

Рассмотрим эффективность работы MPI на примере решения системы линейных алгебраических уравнений методом Гаусса. Метод Гаусса хорошо подходит для параллельных вычислений. Он содержит несколько участков для распараллеливания. Первый участок – это прямой ход, когда систему приводят к ступенчатому виду. Второй участок – обратный ход, суть которого заключается в том, чтобы выразить все базисные переменные через небазисные. Есть модификация метода Гаусса, где нужно находить минимальный элемент в столбце перед прямым ходом. Нахождение минимального элемента в столбце тоже можно распараллелить.

Для оценки эффективности технологии MPI были решены системы линейных алгебраических уравнений с различным количеством неизвестных. Для каждого количества неизвестных было проведено более 25 опытов.

Перейдем к результатам решения СЛАУ на компьютере с 8-ядерным процессором, которые указаны на рис. 1.

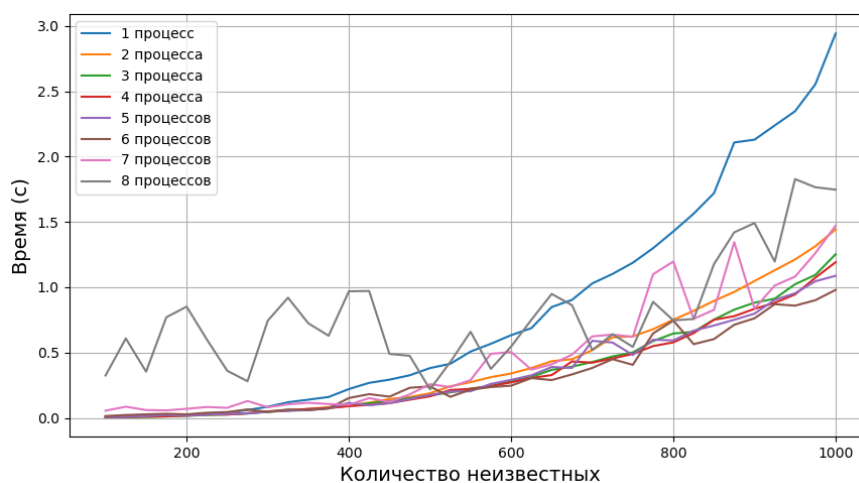


Рис. 1. Результаты решения СЛАУ

Можно заметить неоднозначность результатов для решений с помощью 7 и 8 процессов. Эта обусловливается тем, что мы используем почти все ядра компьютера, на котором были произведены расчеты, что дает местами неоднозначный результат по времени из-за загруженности процессора. Лучший результат для 1000 неизвестных показало решение с помощью 6 процессов. Если сравнить с решением, в котором распараллеливание не применялось, то получилось решить систему приблизительно в три раза быстрее.

3D МОДЕЛИРОВАНИЕ ТЕХНИЧЕСКИХ ОБЪЕКТОВ СРЕДСТВАМИ OPENSCAD

В. Г. Земченко

Учреждение образования «Гомельский государственный технический университет имени П. О. Сухого», Республика Беларусь

Научный руководитель Т. А. Трохова

В современном мире достаточно сложно найти сферу деятельности человека, в которой в той или иной степени не присутствовал бы этап моделирования. Именно от методов исследования модели объекта для полноты описания его поведения зависит качество будущего продукта. В цифровом веке все шире и шире применяются 3D технологии, в том числе для создания визуального представления модели, преимущество которого в простоте восприятия размера, формы, внешнего вида объекта.

Задачи, решаемые при моделировании технических объектов: получение адекватной математической модели; создание компьютерной модели на основе математической модели; визуализация результатов моделирования в графическом виде; 3D моделирование динамической системы в реальном времени.

В качестве инструмента для решения поставленных задач могут выступать платформа Unity3D Engine, редактор Blender или система проектирования OpenSCAD. Система автоматизированного проектирования OpenSCAD – это бесплатное программное обеспечение с открытым исходным кодом для создания 3D моделей.