

Векторизация бинарных изображений на многоядерном процессоре

Кудинов П. Ю., Местецкий Л. М.
pkudinov@gmail.com, l.mest@ru.net
Москва, МГУ

К задаче векторизации граничных контуров (аппроксимации их многоугольниками) предъявляются очень высокие требования по скорости. Появление многоядерных процессоров является ресурсом для повышения эффективности этих алгоритмов. Предлагаемый метод параллельной векторизации включает в себя четыре шага: разделение изображения на части, прослеживание каждой части параллельно, объединение результатов прослеживания, вытягивание многоугольных фигур минимального периметра [2].

Алгоритм прослеживания

Разработанный метод позволяет использовать произвольные алгоритмы, прослеживающие граничные пары точек контуров на изображении. Алгоритм должен выявить все пары соседних по горизонтали разноцветных точек, относящихся к одному контуру, и упорядочить эти пары вдоль контура. Примером является метод симплексного прослеживания, описанный в [2]. На Рис. 1 показан пример работы такого алгоритма. Разноцветными кружками обозначены пиксели изображения, а крестиками — точки, входящие в след трассировки.



Рис. 1. След трассировки контура (слева) и маршрут движения симплекса (справа).

Описание метода

Разделение изображения на части. Изображение разделяется на N частей таким образом, чтобы части имели три общие строки.

Среднюю из этих трех строк будем называть *линией разреза*. Интерес представляют контуры, в следе трассировки которых есть точки, общие с линией разреза, так как именно эти контуры оказались *разрезанными*. На Рис. 3 (справа) изображены разрезанные контуры K1, K2, K3,

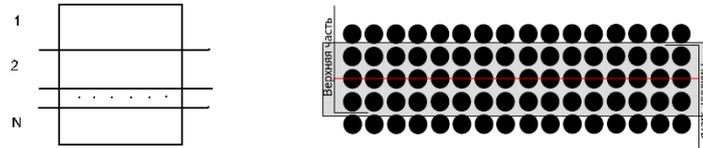


Рис. 2. Разделение изображения на части.

по которым нужно восстановить два контура: внешний и внутренний для фигуры серого цвета.

После того, как изображение разделено на части, проводится анализ линий разреза. В каждой линии регистрируются пары разноцветных точек — *места склейки*, на Рис. 3 (слева) они обведены прямоугольниками.

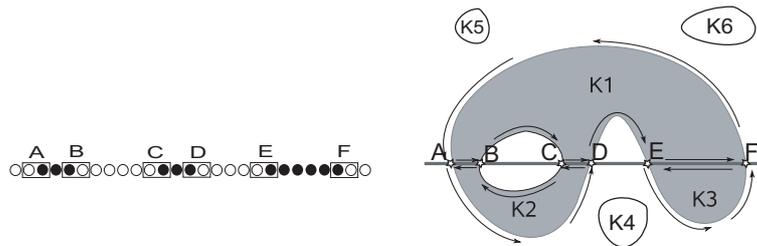


Рис. 3. Пары разноцветных точек в линии разреза (слева), разрезанные и независимые контуры (справа).

Прослеживание частей. Каждая часть прослеживается параллельно выбранным алгоритмом. В процессе прослеживания регистрируются выходы на линии разреза в парах разноцветных точек. Это позволяет установить соответствие между местами склейки и контурами, которым они принадлежат. Каждому месту склейки соответствует два контура — в верхней и нижней частях, а каждому контуру — последовательность мест склейки, упорядоченных по направлению прослеживания.

Операция склейки. Прослеживание разрезанных частей вообще говоря меняет топологию контуров. На Рис. 3 изображена фигура серого цвета, образованная двумя контурами: внешним черного цвета и внутренним — белого. После разреза образовалось три контура: K1, K2, K3 из которых и нужно получить исходные контуры.

Операция заключается в последовательном просмотре массива мест склейки, в котором места склейки упорядочены слева направо. На каж-

дой итерации из установленного на этапе прослеживания соответствия выбираются два контура из верхней и нижней части и соединяются в месте склейки (происходит перенаправление ссылок в списках прослеживания). Затем определяется следующее место склейки (после A следующим будет место склейки, отмеченное как D , Рис. 3), выбираются соответствующие контуры $K1$, $K2$ и производится соединение. Аналогично будут обработаны E , F , после чего следующим местом склейки станет A . На этом процедура восстановления внешнего контура завершена. Осталось две пары разноцветных точек: B , C . По ним так же восстанавливается контур.

Операция восстановления контуров линейна по количеству мест склейки, что позволяет рассчитывать на небольшое время ее выполнения на большинстве изображений.

Вытягивание многоугольника минимального периметра.

Воспользовавшись независимостью следов трассировки контуров, можно запустить процедуру восстановления непрерывных границ параллельно на нескольких процессорах. Речь идёт о построении многоугольника минимального периметра, лежащего целиком в коридоре, образованном разноцветными точками вдоль контура.

Результаты

Предложенный метод прослеживания и склейки контуров является эффективным и не влияет на скорость обработки частей изображения, что позволяет рассчитывать на N -кратное увеличение производительности вычислений при использовании N процессоров (или N -ядерного процессора). Метод был реализован в полном объеме с использованием технологии OpenMP [1] на языке C++. Результаты были подтверждены экспериментально.

Литература

- [1] Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002. — 608 с.
- [2] Местецкий Л. М. Непрерывный скелет бинарного растрового изображения // Тр. конф. «Графикон-98», Москва, 1998.