ресации под любой ОС. Она может быть как приложением пользовательского уровня, так и системным сервисом. Обнаружение утечек может проводиться также удалённо, с использованием протокола TCP/IP.

Заключение

Таким образом, разработчики программного обеспечения, пользуясь данными средствами, могут устранить утечки ресурсов в случае их наличия, повышая тем самым качество своего программного обеспечения, или убедиться, что утечки ресурсов отсутствуют в их программном обеспечении для тех режимов работы программы, при которых она испытывалась на наличие утечек ресурсов. Для этого необходимо скомпилировать и скомпоновать отлаживаемую программу с соответствующими файлами и осуществлять прогоны программы, анализируя результаты генератора отчётов и внося исправления в программу.

ЛИТЕРАТУРА

1. http://dmalloc.com/

УДК 519.688

АЛГОРИТМЫ ПОСТРОЕНИЯ И ОПТИМИЗАЦИИ СТРУКТУР ТРИАНГУЛЯЦИИ ДЕЛОНЕ С ПЕРЕБРОСКАМИ РЁБЕР

Д.А. Петренко, А.В. Скворцов

Томский государственный университет

E-mail: den@indorsoft.ru, skv@indorsoft.ru

Рассматриваются способы редактирования триангуляции Делоне с ограничениями. Предлагается новый тип исходных данных для триангуляции, описывающий изменения, выполненные в ручном режиме редактирования триангуляции. С целью недопущения разрастания объёма таких данных предлагается несколько алгоритмов их сокращения.

Ключевые слова: *триангуляция Делоне с ограничениями, цифровая модель рельефа, изолинии, перебросски рёбер, флипы.*

Триангуляция часто применяется в различных задачах машинной графики, вычислительной геометрии, методах конечных элементов, геоинформатике, системах автоматизированного проектирования, различных инженерных задачах.

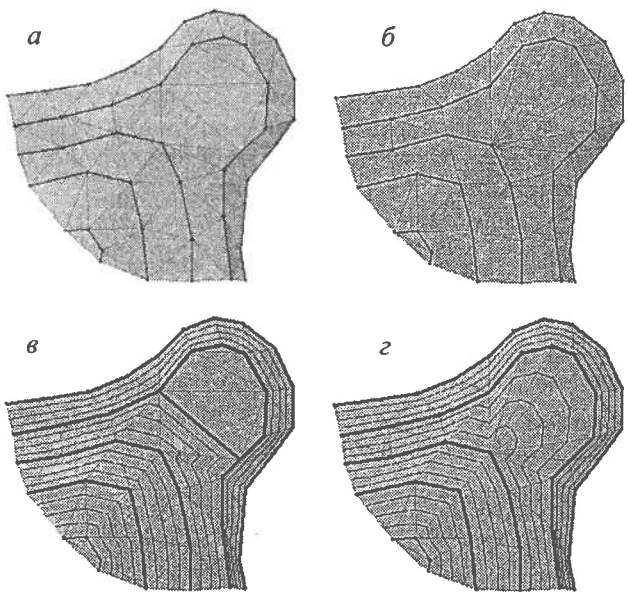

Рис. 1. Пример неправильного автоматического построения триангуляции: *а* – автоматически построенная поверхность; *б* – исправленная поверхность; *в* – изолинии на автоматически построенной поверхности; *г* – изолинии на исправленной поверхности

В большинстве случаев исходными данными для построения триангуляции являются точки, структурные линии и регионы [1].

Однако в ряде случаев построение триангуляции Делоне с ограничениями не даёт правильной картины поверхности, например, если исходными данными для триангуляции являются изолинии (рис. 1), полученные в результате обработки картографических материалов.

В приведённом примере поверхность, построенная по исходным данным автоматически (рис. 1, *а*), не соответствует «правильной» ситуации; скорее всего, поверхность должна иметь вид, показанный на рис. 1, *б*.

В некоторых случаях только человек может решить, как должны проходить рёбра триангуляции. Следовательно, требуется ручная доработка построенной триангуляционной модели. При этом возможны варианты: 1) при построении триангуляции вводить эмпирические ограничения, учитывающие природу исходных данных (например, как в приведённом случае, запрещать треугольники, в которых все узлы имеют одинаковую высоту; такого рода

ограничения рассматриваются в работах [2–10]); 2) добавлять в исходные данные структурные линии, исправляющие ошибку; 3) дать возможность пользователю в интерактивном режиме менять структуру триангуляции (перестраивать пары смежных треугольников, если такое перестроение возможно).

Один из наиболее распространенных случаев, когда требуется доработка триангуляции, возникает тогда, когда триангуляция строится по структурным линиям, при этом возникает множество строго горизонтальных треугольников. Можно выделить три класса решения проблемы плоских треугольников [2].

1. *Превентивные методы.* В частности, было предложено генерализовать исходные контуры перед построением триангуляции [3], но это решение ухудшает детализацию рельефа, может привести к пересечению исходных контуров и не гарантирует избавления от плоских треугольников. Другой метод заключается в том, что в исходные данные заранее вставляются структурные рёбра, которые не дадут образоваться плоским треугольникам. Недостатком данного метода является то, что данных о структурных рёбрах, которые необходимо вставить, зачастую нет.

2. *Коррективные процедуры.* В качестве методов коррекции триангуляции обычно используются переброски рёбер и вставка новых точек. Brandli [4] в 1992 году предложил интерактивно редактировать триангуляцию для избавления от плоских треугольников, а там, где это невозможно, – вставлять новые точки. Vozenilek [5] в 1994 году предложил вставлять вспомогательные точки в середине рёбер, если перестроение пар треугольников невозможно.

3. *Специфические решения.* Например, использование специфических алгоритмов построения триангуляции (не Делоне) разработанных специально для работы на контурных данных [6, 7]. Также в 1987 году Christensen [8] предложил процесс построения триангуляции разделить на серию локальных триангуляций, строящихся между соседними контурами, а затем объединить их. При построении локальной триангуляции добавлялись дополнительные точки средней высоты, что позволяло избежать плоских треугольников. Существуют и другие методы решения данной проблемы [9, 10].

Все описанные методы были направлены на автоматическое или ручное избавление от плоских треугольников. В данной работе рассматривается вопрос ручного редактирования триангуляции в общем случае, не только применительно к плоским треугольникам.

Ручное редактирование триангуляции вряд ли потребуется при работе с большими объёмами данных в мелком масштабе (в геоинформационных системах), тем более что на больших объёмах данных вручную исправить всю триангуляцию практически невозможно. Но при обработке геодезических данных в крупных масштабах, когда неправильно проведённое ребро триангуляции может существенно изменить результат вычислений и, как следствие, стоимость работ, ручное редактирование триангуляции порой необходимо.

Интерактивное ручное редактирование триангуляции обычно сводится к локальному изменению пары соседних треугольников с помощью так называемой переброски ребра (рис. 2). Данная операция также в ряде источников называется флипом (англ. *to flip* – переворачивать) или своппингом (англ. *swapping* – обмен).

Ручное редактирование триангуляции позволяет человеку построить модель поверхности так, как он считает правильным. Однако если каким-то образом поменяется часть исходных данных (например, будут добавлены или удалены точки) и триангуляция будет перестроена заново, то результаты кропотливого труда по ручному редактированию триангуляции бесследно потеряются.

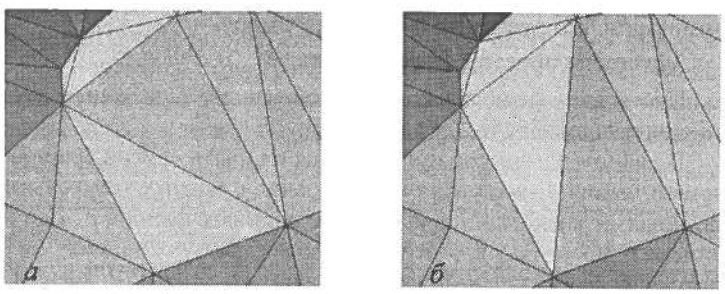

Рис. 2. Редактирование триангуляции с помощью флипов:
а – до применения флипа; *б* – после применения

В данной работе предлагается ввести новый тип исходных данных триангуляции, описывающий изменения, выполненные в ручном режиме редактирования триангуляции.

Постановка задачи

Пусть дана триангуляция T, построенная на множестве исходных точек, структурных линий и регионов. Для исправления автоматически построенной триангуляции Делоне требуется обеспечить возможности: *а*) интерактивного редактирования триангуляцию с помощью переброс рёбер; *б*) сохранения произведённых переброс рёбер таким образом, чтобы после полного перестроения триангуляции не требовалось исправлять заново исправленные в интерактивном режиме участки; *в*) не допускать безразмерного разрастания информации о переброс рёбер во избежание излишних затрат памяти и снижения скорости работы алгоритма построения триангуляции.

В настоящей работе предлагается ввести дополнительный вид исходных данных для построения триангуляции, а результаты интерактивного редактирования запоминать в этих данных. Для контроля над разрастанием объема таких данных предлагается несколько алгоритмов их сокращения.

Авторами предлагается все переброски рёбер триангуляции, производимые при её ручном редактировании, заносить в список для возможного его последующего применения после перестроения всей триангуляции. Следует отметить, что если пользователь будет производить большое количество флипов, подбирая оптимальный с его точки зрения результат, то указанный список будет постоянно расти, что будет замедлять работу программы при перестроении триангуляции (например, если изменилась часть исходных данных) и потребует дополнительных затрат памяти на хранение информации о сделанных флипах.

В качестве информации, добавляемой в список флипов, предлагается запоминать координаты четырёх узлов триангуляции, на которые пользователь перебросил ребро: координаты двух узлов, образовывавших ребро триангуляции до перестроения, и координаты двух узлов, на которых образовалось новое ребро триангуляции в результате перестроения. Необходимо заметить, что после перестроения триангуляции применение флипов должно производиться строго последовательно, в том порядке, в котором эти действия производил пользователь при ручном редактировании, поскольку сразу после перестроения в триангуляции может не существовать рёбер, которые хранятся в конце списка флипов, но появятся при последовательном их применении к перестроенной триангуляции.

В результате изменения исходных данных триангуляции (например, после удаления некоторых исходных точек) часть списка флипов может оказаться «не у дел», например, если после перестроения в триангуляции перестали существовать узлы, координаты которых хранились во флипе. В этом случае данный флип невозможно будет применить и его придётся исключить из списка.

Возможна ситуация, когда пользователь, последовательно редактируя триангуляцию в ручном режиме (и добавляя тем самым в список флипов новые записи), в результате многочисленных переброс рёбер триангуляции приведёт её к исходному виду, т.е. такому, который получился бы в результате автоматического её построения, при этом список флипов будет непустым. В этом случае для экономии памяти и времени построения триангуляции следует оптимизировать список флипов так, чтобы по возможности уменьшить его до минимального размера, не нарушив при этом информации, необходимой для восстановления отредактированной человеком триангуляции.

Алгоритмы сокращения списка переброс ребер

Для решения задачи сокращения списка флипов в данной работе предлагаются три алгоритма оптимизации: 1) простая; 2) локальная; 3) глобальная.

Известно [1], что любую триангуляцию за линейное время можно перестроить посредством переброс рёбер смежных треугольников так, что в результате получится триангуляция Делоне. Алгоритм простой оптимизации базируется на данном утверждении.

Алгоритм простой оптимизации

Пусть имеется построенная триангуляция T, заданная множеством треугольников и узлов. Также задан список флипов $F=(f_1,\ldots,f_i)$.

В данном алгоритме предлагается очищать список флипов F и затем восстанавливать его из имеющейся триангуляции T, выполняя перестроение смежных треугольников до получения триангуляции Делоне и запоминая делающиеся перестроения в список флипов F. Все перестроения необходимо добавлять в начало списка F, поскольку в ходе построения триангуляции сначала будет получена триангуляция Делоне и только потом последовательно будут применяться флипы. В данном же случае при работе этого алгоритма сокращения списка флипов мы идём по обратному пути: из имеющейся триангуляции восстанавливаем триангуляцию Делоне.

Стоит отметить, что поскольку одна и та же триангуляция может быть получена с помощью различных последовательностей флипов, то в результате работы этого алгоритма список флипов может не только сократиться, но и увеличиться, при этом триангуляции, построенные на одних и тех же исходных точках, но на разных списках флипов, будут эквивалентными. Возможность данного ухудшения объясняется тем, что поскольку порядок узлов и треугольников в двух эквивалентных триангуляциях может отличаться [12], восстановление построенной триангуляции до триангуляции Делоне может начаться с другого элемента. Соответственно список флипов может получиться совершенно другой.

В общем случае, после работы данного алгоритма размер списка флипов будет составлять $O(N)$, где N – число узлов в триангуляции, независимо от количества перестроений, сделанных пользователем интерактивно.

Чтобы избежать увеличения списка флипов, предлагается использовать следующий алгоритм локальной оптимизации.

Алгоритм локальной оптимизации

В этом алгоритме предлагается перед применением алгоритма простой оптимизации сохранить имеющийся список флипов F. Затем последовательно применять алгоритм простой оптимизации до тех пор, пока размер получающегося списка флипов F_1 не перестанет уменьшаться. После каждой итерации, если размер нового списка флипов F_1 стал меньше уже имеющегося списка F, список F заменяется на F_1. Как только размер списка F_1 стал больше либо равен размеру списка F, в качестве сокращённого списка флипов берётся наименьший из ранее достигнутых списков – список F.

В результате работы данного алгоритма мы, хотя и можем получить список флипов лучше того, что получился бы в результате работы алгоритма простой оптимизации, но этот список не обязательно будет наименьшим из всех возможных. Для получения самого лучшего (в смысле размера) списка потребовалось бы перебрать все возможные пути восстановления триангуляции Делоне из существующей триангуляции и выбрать из них вариант с наименьшим количеством перестроений треугольников.

Задача перебора всех возможных путей восстановления триангуляции Делоне является *NP*-полной. Это следует из экспоненциальной зависимости количества всех возможных триангуляций от числа исходных точек [1]. На практике, когда количество исходных точек составляет тысячи и миллионы, полный перебор всех вариантов, конечно, не приемлем. Таким образом, встает задача поиска приближенного решения, имеющего относительно небольшую трудоемкость (разумной будет не более чем линейно-логарифмическая трудоемкость алгоритма).

Тем не менее, алгоритм полного перебора можно несколько улучшить, если применить комбинацию алгоритмов локальной оптимизации и полный перебор.

Алгоритм глобальной оптимизации

В данном алгоритме предлагается, разделив некоторым образом триангуляцию на непересекающиеся минимальные (независимые) участки, в каждом из которых присутствуют флипы, провести сокращение списка флипов в каждом их этих участков. По завершении обработки всех участков полученные списки флипов можно в произвольном порядке склеить между собой. Склейка списков флипов не изменит структуру триангуляции, поскольку участки, на которых сокращались списки флипов, не пересекаются, и, следовательно, перестроения ребер в одном участке никак не повлияют на перестроения в другом.

В качестве алгоритма сокращения списка флипов на участке триангуляции предлагается использовать полный перебор с ограничением по глубине поиска n. При нахождении решения с длиной списка $n_1 < n$ дальнейший поиск лучших решений ограничивается глубиной n_1. В качестве начального n_0 можно использовать длину списка, полученного в результате работы алгоритма локальной оптимизации, рассмотренного выше.

Для полного перебора потребуется проверить $N = k^n$ различных способов восстановления триангуляции в триангуляцию Делоне, где k – количество внутренних ребер в участке; n – ограничение по глубине поиска, полученное приближённым алгоритмом сокращения списка флипов. Однако на практике не все пары смежных треугольников можно перестроить (например, пара треугольников не образует выпуклого четырёхугольника или общее для пары треугольников ребро является структурным), поэтому число k можно уменьшить в среднем примерно в два раза. Если же применить ещё и ограничение, не допускающее перестроение пар треугольников, уже удовлетворяющих условию Делоне, то количество вариантов перебора сокращается в среднем до величины $N \approx (k/10)^n$.

Таким образом, алгоритм полного перебора вариантов будет работать приемлемое (неэкспоненциальное) время только тогда, когда число треугольников в каждом независимом участке будет относительно невелико (в пределах 30–50). Поэтому в алгоритме необходимо вставить ограничение по максимально возможному количеству вариантов, которое можно будет проанализировать.

Заключение

Предложенные в данной работе новый тип исходных данных для триангуляции и алгоритм их сокращения позволяют решить поставленную в начале работы задачу сохранения и недопущения разрастания информации об изменениях триангуляции, сделанных человеком.

Используя приведённые в данной работе алгоритмы сокращения списка флипов, можно быть уверенным в том, что этот список не будет разрастаться бесконечно по мере редактирования триангуляции.

Предложенные в данной работе новый тип исходных данных (переброски ребер) для построения триангуляции и алгоритм сокращения списка флипов реализованы на практике и внедрены в систему автоматизированного проектирования IndorCAD и геоинформационную систему IndorGIS, разработанные в ООО «ИндорСофт», г. Томск.

Алгоритмы сокращения списка флипов прошли апробацию в реальных проектах по проектированию автомобильных дорог, выполняемых в ООО ИДЦ «Индор», г. Томск, и подтвердили высокую скорость работы.

ЛИТЕРАТУРА

1. *Скворцов А.В.* Триангуляция Делоне и её применение. – Томск: Изд-во Том. ун-та, 2002. – 127 с.
2. *J.Mark Ware.* A procedure for automatically correcting invalid flat triangles occuring in triangulated contour data // Computers & Geosciences. – 1998. – V. 24. – No. 2. – P. 141–150.
3. ARC/INFO User's Guide: Surface Modeling With TIN – ESRI Inc., NY, 1992. – 240 p.
4. *Brandli M.* A triangulation-based method for geomorphological surface interpolation from contour lines // Proceeding of 3rd European Conference and Exhibition on GIS. – Münich, 1992. – V. 1. – – P. 691–700.
5. *Vozenilek V.* Generating surface models using elevations digitised from topographical maps // Proc. of 5th European Conference and Exhibition on GIS. – Utrecht, 1994. – V. 1. – P. 972–982.
6. *Fuchs H., Kedem Z.M., Uselton S.P.* Optimal surface reconstruction from planar contours // Communications of the ACM. – 1977. – V. 20. – No. 10. – P. 693–702.
7. *Ekoule A.B., Peyrin F.C., Odet L.* A triangulation algorithm from arbitrary shaped multiple planar contours // ACM Transactions on Graphics. – 1991. – V. 10. – No. 2. – P. 182–199.
8. *Christensen A.H.J.* Fitting a triangulation to contour lines // Proc. 8th Intern. Symp.of Computer Assisted Cartography. – Baltimore, Maryland, 1987. – P. 57–67.
9. *Ganthapy S., Dennehy T.G.* A new general triangulation method for planar contours // ACM Computer Graphics. – 1982. – V. 13. – No. 3. – P. 311–319.
10. *Tipper J.C.* A method and Fortran program for the computerized reconstruction of three dimensional objects from serial sections // Computer & Geosciences. – 1977. – V. 3. – No. 4. – P. 579–599.
11. *Фукс А.Л., Костюк Ю.Л.* Построение цифровой модели рельефа местности на основе структурных линий и высотных отметок // Вестник ТГУ. – 2003. – Т. 280.
12. *Петренко Д.А., Скворцов А.В., Куленов Р.О.* Сравнение триангуляций с помощью хеш-функций // Вестник ТГУ. – 2003. – Т. 280. – С. 306–309.

УДК 519.688

ОБЗОР АЛГОРИТМОВ ПОСТРОЕНИЯ ВЫПУКЛОЙ ОБОЛОЧКИ НА ПЛОСКОСТИ

Р.В. Чаднов, А.В. Скворцов, Н.С. Мирза

Томский государственный университет

E-mail: chadnov@newmail.ru; skv@indorsoft.ru; mirza@indorsoft.ru

Рассматриваются различные алгоритмы построения выпуклой оболочки на плоскости. Проводится сравнение различных алгоритмов. Показываются преимущества и недостатки алгоритмов на различных наборах данных. Предлагается использовать комбинированный алгоритм для достижения максимальной эффективности при построении выпуклой оболочки вне зависимости от видов исходных данных.

Ключевые слова: *выпуклые оболочки, вычислительная геометрия.*

Задача построения выпуклых оболочек является одной из центральных для вычислительной геометрии. Она позволяет разрешить целый ряд других, иногда с первого взгляда не связанных с ней вопросов: построение диаграмм Вороного, построение триангуляций и т.д. Построение выпуклой оболочки конечного множества точек на плоскости довольно широко исследовано и имеет множество приложений в распознавании образов, обработке изображений, в задаче раскроя и компоновки материалов. Очень широко алгоритмы построения выпуклой оболочки используются в геоинформатике и геоинформационных системах.

В настоящее время известно достаточно большое число алгоритмов построения выпуклой оболочки, однако существует проблема, связанная с недостаточным количеством работ, посвященных анализу и/или обзору этих алгоритмов.

В настоящей работе выполнен анализ наиболее распространенных современных алгоритмов построения выпуклой оболочки. Кроме того, в работе приведены результаты сравнения скорости работы алгоритмов в различных условиях.

1. Постановка задачи

Понятие выпуклой оболочки определяется следующим образом [1]:

Определение 1. *Выпуклой оболочкой* (ВО) множества точек S называется наименьшее выпуклое множество, содержащее S.

В случае, когда S – конечное множество точек на плоскости, это определение можно представить наглядно. Предположим, что множество точек охвачено большой растянутой резиновой лентой. Эта лента имеет форму выпуклой оболочки.