Журнал обчислювальної та прикладної математики | **2013, №1(111)** | Journal of Computational & Applied Mathematics

UDC 681.3 513, 004.6(075.8)

ПОСТРОЕНИЕ ТРИАНГУЛЯЦИИ МЕЖПОЛИТОПНОЙ ОБЛАСТИ В d-МЕРНОМ ПРОСТРАНСТВЕ

В. М. ТЕРЕЩЕНКО

Резюме. В статье предложен метод решения задачи триангуляцию области между выпуклыми многогранниками в d-мерном пространстве с использованием триангуляцию Делоне, за $O(N^2)$ времени. Следует отметить, что новизна работы заключается в использовании модифицированного алгоритма триангуляции Делоне с ограничениями.

Введение

Постановка проблемы. В работе рассматривается метод решения задачи триангуляции области между политопами в d-мерном Евклидовом пространстве. Большое количество задач вычислительной геометрии и компьютерной графики, а также их приложений сводиться к такой задаче. Особое распространение эта задача приобрела в геоинформационных системах при моделировании рельефа, в реконструкции изображений, распознавании образов и компьютерных играх.

Анализ последних исследований. В последнее время ставиться вопрос о возможности решения этой задачи за время $O(n + k^2)$ (n — общее количество точек, k — количество многогранников) [1]. При дальнейшем исследовании было обнаружено, что сложность решения зависит от формы участка между многогранниками [2]. Позже Б. Джо предлагает новый подход, связанный с локальным преобразованием многогранника, что улучшает качество триангуляции (триангуляция Делоне) [4]. Однако время выполнения такого алгоритма, $O(n^2 + k^2)$, далеко от желаемого результата.

Один из подходов к решению задачи состоит из двух основных этапов: построения самого участка и его триангуляции. Для каждого этапа используются различные методы. Например, часть методов поиска участка базируется на построении выпуклого многогранника вокруг заданных многогранников и затем удаление из него точек, принадлежащих заданным многогранникам. На этапе построения участка также используются различные алгоритмы построения выпуклой оболочки [3]. Для построения триангуляции тоже существуют различные способы [4, 5]. В частности, эту проблему можно решать как проблему триангуляции с ограничениями [6, 7] или проблему триангуляции невыпуклого многогранника [8]. Однако оптимального подхода еще не предложено, и проблема остается открытой.

Новизна и идеи. В данной работе предложена модификация подхода, базирующегося на построении выпуклой области и триангуляции с ограничениями.

Цель статьи. Предложить новый подход к решению задачи триангуляции области между многогранниками.

1. Постановка задачи и метод решения

Постановка задачи триангуляции участка. Пусть в d-мерном пространстве заданы k выпуклых многогранника с общим количеством N вершин, которые между собой не пересекаются. Необходимо триангулировать область, образованную заданными многогранниками (тетраедраезировать область).

Оределение 1. Область, ограниченная выпуклой оболочкой для множества многогранников и их взаимовыпуклыми гранями, образует участок, который необходимо триангулювать. Разделим задачу триангуляции участка на две подзадачи: 1) построение участка, образованного заданными многогранниками; 2) триангуляция построенного участка.

Рассмотрим метод решения задачи на примере трехмерного пространства.

Замечание 1. *Все грани являются треугольными, однако две смежные грани могут лежать на одной плоскости. На вход всех алгоритмов подаются именно такие грани, поскольку любую грань можно триангулировать.*

Замечание 2. *Введем отношение порядка для вершин. Вершина $v_1 < v_2$, тогда и только тогда, когда x-координата v_1 меньше x-координаты v_2, или если x-координата v_1 равна x-координате v_2 и y-координата v_1 меньше y-координаты v_2, или если x-координата v_1 равна x-координате v_2 и y-координата v_1 равна y-координате v_2 и z-координата v_1 меньше z-координаты v_2.*

Также дополнительно введем отношение порядка для граней. Пусть v_{11}, v_{12}, v_{13} — вершины, которые задают f_1, а v_{21}, v_{22}, v_{23} — вершины, которые задают f_2, причем $v_{11} < v_{12} < v_{13}$, и $v_{21} < v_{22} < v_{23}$.

Грань $f_1 < f_2$, тогда и только тогда, когда $v_{11} < v_{21}$ или $v_{11} = v_{21}$ и $v_{12} < v_{22}$ или $v_{11} = v_{21}$, $v_{12} = v_{22}$ и $v_{31} < v_{23}$.

1.1. Решение задачи построения области

Для построения области, нам необходимо сначала построить выпуклую оболочку на множестве входных многогранников. Для этого применим стратегию "Разделяй и властвуй" [9]. В результате получим выпуклую оболочку множества заданных многогранников. Следующим шагом является удаление множества точек, принадлежащих входным многогранникам:

1. Отсортируем грани входящих многогранников (обозначим это множество, как F_v).

2. Отсортируем грани построенного выпуклого многогранника (F_m).
3. F_a-множество граней области.
4. Определяем наименьшую грань f среди всех граней, принадлежащих $F_v \cup F_m$.
5. Если грань $f \in F_v \cap f \in F_m$, то f не принадлежит области, и просто удаляется из F_m и F_v. Иначе f прибавляется к F_a и удаляется из F_v (если $f \in F_v$) или $F_m (f \in F_m)$.
6. Если $F_v \cup F_m = \emptyset$, то останавливаемся, иначе п.4

В результате множество F_a содержать все грани, принадлежащие области, а следовательно, область построена.

1.2. Решение задачи триангуляции области

Триангуляцию полученной фигуры будем рассматривать как проблему триангуляции с ограничениями в виде поверхностей.

Оределение 2. Триангуляция множества точек с ограничениями в виде поверхностей — это построение триангуляции, в которой ребра построенных тетраэдров не пересекают поверхности ограничений (вершины тетраэдров могут лежать на поверхностях ограничений).

Для решения задачи триангуляции с ограничениями применим известный алгоритм C. Hazlewood [6], который состоит из четырех шагов.

1. Триангуляция Делоне без ограничений.
2. Восстановление ребер ограничений.
3. Восстановления поверхностей ограничений.
4. Удаление тетраэдров, которые находятся вне заданной области.

Все эти шаги подробно описаны в [6]. Но, учитывая специфику поставленной нами задачи, первый шаг будет иметь свои особенности. И поэтому, опишем модифицированный алгоритм построения триангуляции Делоне без ограничений с учетом участка ограниченного многогранниками.

1.2.1. Алгоритм построения триангуляции Делоне без ограничений

В качестве узлов триангуляции будем использовать вершины области. Алгоритм построения триангуляции состоит из следующих шагов.

1. Создание множества U — набор заданных узлов.

2. Создание "суперструктуры" — тетраэдра, в средине которого лежат все заданные узлы.

3. Формирование пустого множества G.

4. Поиск для суперструктуры центра и радиуса описанной сферы.

5. Выбор любого узла с U и перенос его в G. Удаляются все тетраэдры, для которых q лежит внутри описанных вокруг них сфер. В результате, в построенной сетке образуется многогранник, который в общем случае имеет звездную форму, причем любой луч, который выходит из q, должен пересекать контур этого многогранника только в одной точке. Если существуют тетраэдры (смежные с нашим многогранником), для которых q

лежит в плоскости одной из граней, то такие грани также необходимо удалить. Фактически ребро или грань удаляются только в том случае, если удаляются все смежные с ним тетраэдры, при этом грани и ребра суперструктуры не удаляются никогда. Новые тетраэдры образуются добавлением ребер между q и вершинами полученного многогранника. Доказано [10], что при этом получим триангуляцию Делоне.

6. Поиск центров и радиусов описанных сфер для новых тетраэдров.

7. Если множество U непустое, то переходим на шаг 5, иначе — конец алгоритма.

Результатом работы алгоритма будет триангуляция Делоне вершин участка, находящегося в средине тетраэдра.

2. Сложность алгоритма

Теорема 1. *Задачу триангуляции области на множестве многогранников (суммарное число вершин — N) предложенным методом можно решить за $O(N^2)$ времени.*

Доказательство. Основные этапы построения триангуляции области между многогранниками следующие:

1. Алгоритм построения выпуклой оболочки множества многогранников (алгоритм «разделяй и властвуй»).
2. Алгоритм построения триангуляции области (алгоритм построения триангуляции Делоне с ограничениями).

На *этапе* 1 применяется стратегия «разделяй и властвуй», основными шагами которой будут: алгоритм поиска первой грани процедуры слияния (нижнего моста) и рекурсивное построение выпуклой оболочки [9]. Временная сложность алгоритма поиска нижнего моста будет равна $O(N)$ так, как все операции: построение проекции на плоскость XOY, построение отрезков, поиск медианы [11] и удаления выполняются за линейное время. На каждом шаге гарантированно удаляется $N/4$ вершин. Тогда время алгоритма нахождения моста определяется следующим образом:

$$O\left(N + \left(\frac{3}{4}\right)N + \left(\frac{3}{4}\right)^2 N + \dots\right) = O(N).$$

Что касается рекурсивного построения выпуклой оболочки, то сортировка вершин занимает время — $O(NlogN)$, слияние двух выпуклых многогранников занимает время $O(N)$. Сложность нахождения триангуляции — O(количество ребер $CH(A \cup B)$). Сложность удаления скрытых граней — O(количество граней $CH(A)$ + количество граней $CH(B)$). Из формулы Эйлера для выпуклого многогранника следует $N - M + L = 2$, где N — количество вершин, M — количество ребер, L — количество граней, следует, что сложность слияния двух выпуклых многогранников $O(N)$. Так что в общем случае сложность построения выпуклой оболочки — $O(NlogN)$. Формирование выпуклой области занимает $O(NlogN)$ времени. Так, в частности, сортировка граней занимает время $O(NlogN)$, а слияние двух списков производится за линейное время.

На *этапе* 2 применяется алгоритм построения триангуляции Делоне с ограничениями, сложность которого $O(N^2)$. Так, в частности, построение суперструктуры занимает линейное время. Вставка одной точки занимает линейное время от количества тетраэдров. Количество тетраэдров составляет $O(N)$. Чтобы вставить все вершины в триангуляцию необходимо потратить $O(N^2)$ времени. Время восстановления одного ребра триангуляции — линейное от количества тетраэдров. Поэтому восстановление всех ребер также займет $O(N^2)$ времени. Аналогично, время восстановления поверхностей ограничений занимает $O(N^2)$. Проверка на принадлежность точки многограннику занимает $O(N)$ времени, поэтому общее время проверки для всех тетраэдров будет равно $O(N2)$ времени. Таким образом, суммарная сложность всех шагов будет следующей:

$$O(NlogN + NlogN + N^2) = O(N^2).$$

$\square$

3. Практическая часть

Программа реализована на языке программирования Java. Для отображения используется библиотека Java3D, а для представления данных используется структура данных — "узлы", "ребра" и "треугольники". Ввод данных осуществляется с помощью следующих средств:

1. Xml-файл, в котором каждый многогранник представляется списками вершин и граней.

2. Генерация многогранников, которая позволяет сгенерировать до 10 многогранников с суммарным количеством вершин 10000.

Программа дает возможность просматривать процесс решения программы поэтапно (результат построения выпуклой области, ее вид, результат триангуляции). Также есть возможность просмотреть элементы триангуляции по очереди. На рис. 1–5 показан пример программной реализации основных шагов алгоритма.

Заключение

В работе предложен метод решения задачи построения и триангуляции области в d-мерном пространстве. Алгоритм можно модернизировать, используя другую структуру данных, чтобы уменьшить затраты памяти, но это может повлиять на быстродействие при выполнении некоторых шагов алгоритма. Также необходимо обратить внимание на то, что алгоритм является достаточно чувствительным к точности, поскольку решаются плохо обусловленные задачи (поиск центра и радиуса описанной сферы, точек пересечения и т.д.). Поэтому для повышения точности можно использовать точную арифметику, но за счет увеличения сложности вычислений элементарных операций это повлияет на быстродействие всего алгоритма.

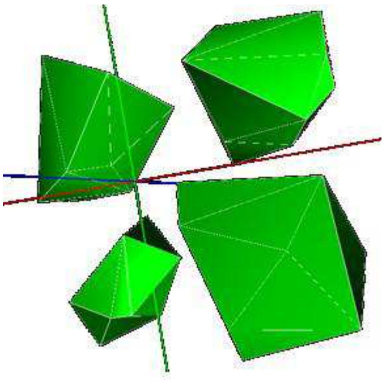

Рис. 1. Пример заданного множества многогранников

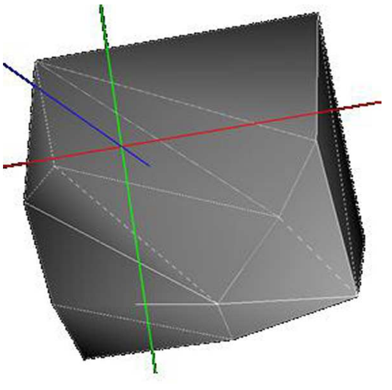

Рис. 2. Выпуклая оболочка области триангуляции

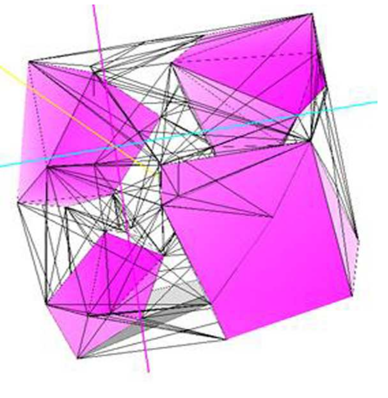

Рис. 3. Триангуляция области

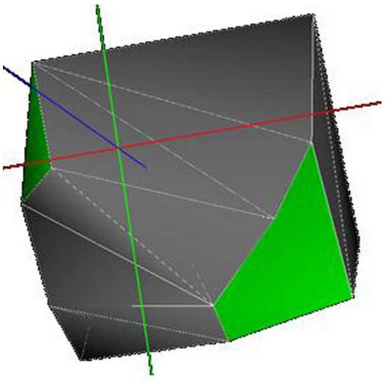

Рис. 4. Выделение триангулированой области между многогранниками

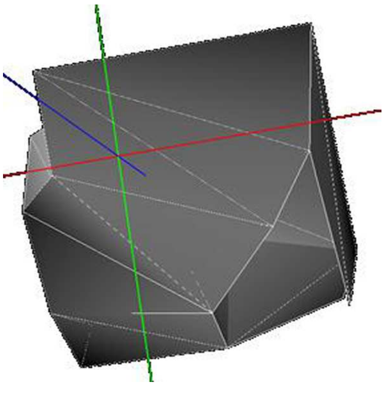

Рис. 5. Результирующая триангулированная область

Литература

1. Goodman J. E. Handbook of Discrete and Computational Geometry. / J. E. Goodman and J. O'Rourke, eds. — Chapman and Hall:CRC Press, 2004. — 1453 p.
2. Chazelle B. Bounds on the Size of Tetrahedralizations. / B. Chazelle, N. Shouraboura. // Discrete & Computational Geometry — 1995. — 14. — P. 429–444.
3. Preparata F. P. Computational Geometry:An introduction. / F. P. Preparata, M. I. Shamos — Berlin: Springer-Verlag, 1985. — 478 p.
4. Kraus M. Simplification of Nonconvex Tetrahedral Meshes. / M. Kraus, T. Ertl // Mathematics and Visualization — 2003. — P. 185–195.
5. Ruppert J. On the difficulty of tetrahedralizing 3-dimensional non-convex polyhedra. / J. Ruppert, R. Seidel // Discrete & Computational Geometry — 1992. — 7. — P. 227–253.
6. Hazlewood C. Approximating constrained tetrahedralizations. / C. Hazlewood. // Computer Aided Geometric Design — 1983. — 10. — P. 67–87.
7. Skvortsov A. V. Delaunay Triangulation and Its Application. / A. V. Skvortsov — Tomsk: Izd. Tomsk. Gos. Univ., 2002.— 478 p.
8. Chazelle B. Triangulating a Nonconvex Polytope. / B. Chazelle, L. Palios. // Discrete & Computational Geometry — 1990. — 5. — P. 505–526.

9. Preparata F. P. Convex hulls of finite sets of points in two and three dimensions. / F. P. Preparata and S. J. Hong. // Comm. ACM — 1977. — 2. — P. 87–93.
10. Joe B. Construction Of Three-Dimensional Delaunay Triangulations Using Local Transformations. / B. Joe. // Computer Aided Geometric Design — 1991. — 8. — P. 123–142.
11. Cormen T. H. Introduction to Algorithms. / T. H. Cormen, C. E. Leiserson, R. L. Rivest — Cambridge: MIT Press, 2009. — 1312 p.

Факультет кибернетики, Киевский национальный университет имени Тараса Шевченко, ул. Владимирская, 64, Киев, 01601, Украина.

Поступила 03.04.2012