УДК 517.518.85+517.27
ББК 22.144

ОБ АЛГОРИТМЕ ПЕРЕЧИСЛЕНИЯ ТРИАНГУЛЯЦИЙ

Попов Владимир Валентинович

Кандидат физико-математических наук, доцент кафедры компьютерных наук и экспериментальной математики, Волгоградский государственный университет
popov_v_v@rambler.ru, knem03@mail.ru, kiem@volsu.ru
просп. Университетский, 100, 400062 г. Волгоград, Российская Федерация

Аннотация. Описывается алгоритм перебора всех триангуляций конечного набора точек трехмерного пространства. Приводятся некоторые результаты работы компьютерной программы, составленной по этому алгоритму. Обсуждается вопрос о распространении алгоритма на пространства размерности больше 3.

Ключевые слова: триангуляция, тетраэдр, симплекс, число триангуляций, выпуклая оболочка.

Вопросам построения триангуляций посвящена обширная литература (см., например, [2; 3]). В работе [1] предложен алгоритм построения всех триангуляций конечного множества на плоскости. В данной работе приводится модификация алгоритма для наборов точек трехмерного пространства.

Пусть $P = \{p_1, p_2, \ldots, p_N\}$ — конечный набор точек в трехмерном пространстве. Необходимо перечислить все триангуляции этого набора. Под триангуляцией набора точек P понимается такой набор тетраэдров S_1, S_2, …, S_k, для которого выполнены следующие условия:

(1) Каждая точка $p_i \in P$ является вершиной хотя бы одного из тетраэдров S_j.

(2) Вершины любого тетраэдра S_j лежат во множестве P.

(3) Если $i \neq j$, то пересечение $S_i \cap S_j$ или пусто, или является общей вершиной, или общим ребром, или общей гранью тетраэдров S_i и S_j.

(4) Объединение тетраэдров S_1, S_2, …, S_k совпадает с выпуклой оболочкой $\mathrm{conv}(P)$ множества P.

Под тетраэдром понимается выпуклая оболочка четырех точек (вершин тетраэдра), не лежащих в одной плоскости трехмерного пространства. Считаем, что точки множества P не лежат в одной плоскости — иначе триангуляций этого множества не существует.

На множестве упорядоченных четверок целых чисел будем рассматривать лексикографический порядок:

$$(i_1, i_2, i_3, i_4) \leq (j_1, j_2, j_3, j_4) \iff \begin{cases} i_1 < j_1 \text{ или} \\ i_1 = j_1, i_2 < j_2 \text{ или} \\ i_1 = j_1, i_2 = j_2, i_3 < j_3 \text{ или} \\ i_1 = j_1, i_2 = j_2, i_3 = j_3, i_4 \leq j_4. \end{cases}$$

Через $S(i_1, i_2, i_3, i_4)$ будем обозначать тетраэдр $\widetilde{S}$, вершинами которого являются точки P с номерами i_1, i_2, i_3 и i_4. Таким образом, вершины $\widetilde{S}$ — это точки p_{i_1}, p_{i_2}, p_{i_3} и p_{i_4} множества P.

Будем говорить, что тетраэдр S совместим с тетраэдром S', если эти тетраэдры или не имеют общих точек, или же пересекаются по общей вершине, или по общему ребру, или по общей грани.

Сначала рассмотрим вспомогательный алгоритм $\mathcal{A}$, который перечисляет все триангуляции, содержащие заданный тетраэдр.

1. Алгоритм $\mathcal{A}$

Пусть задан тетраэдр S_1 с вершинами из множества P, не содержащий точек из P, отличных от своих вершин. Опишем алгоритм $\mathcal{A}$, который перечисляет все триангуляции P, содержащие этот тетраэдр. На каждом шаге алгоритма будет определено число k построенных к данному моменту тетраэдров, а также два списка:

1) Список $\mathcal{E}$ построенных тетраэдров S_1, S_2, $\ldots$, S_k. Каждый тетраэдр представлен в этом списке упорядоченной четверкой чисел (i_1, i_2, i_3, i_4), состоящей из номеров вершин этого тетраэдра.

2) Список $\mathcal{F}$ граней тетраэдров из списка $\mathcal{E}$. Каждая грань представлена упорядоченной тройкой (q_1, q_2, q_3), состоящей из номеров вершин этой грани.

В список $\mathcal{F}$ помещаются такие грани, к которым в дальнейшем могут приклеиваться другие тетраэдры. Эти грани называются граничными. Если к граничной грани приклеивается тетраэдр, то грань становится внутренней. После удаления приклеенного тетраэдра грань снова становится граничной. В начале работы алгоритма списки $\mathcal{E}$ и $\mathcal{F}$ пусты.

Пусть вершинами тетраэдра S являются точки p_{i_1}, p_{i_2}, p_{i_3} и p_{i_4}, то есть $S_1 = S(i_1, i_2, i_3, i_4)$. Алгоритм $\mathcal{A}$ состоит из следующих шагов:

(1) Полагаем $k = 1$.

(2) Помещаем тетраэдр S_1 в список $\mathcal{E}$ (в качестве единственного элемента). Этот тетраэдр будет представлен в списке упорядоченной четверкой чисел (i_1, i_2, i_3, i_4).

(3) Упорядоченные грани (i_1, i_2, i_3), (i_1, i_2, i_4), (i_1, i_3, i_4) и (i_2, i_3, i_4) тетраэдра S_1 помещаем в список $\mathcal{F}$ и объявляем все эти грани граничными.

(4) Просматривая список $\mathcal{F}$ и давая переменной j значения $1, 2, \ldots, N$, пытаемся найти такую граничную грань $(q_1, q_2, q_3) \in \mathcal{F}$ и такое число j, $1 \le j \le N$, что тетраэдр $\widetilde{S} = S(q_1, q_2, q_3, j)$ не содержит точек из P, отличных от своих вершин, и совместим с каждым из тетраэдров списка $\mathcal{E}$. Если нужной четверки чисел (q_1, q_2, q_3, j) не найдено, то идем к пункту (6). Если же такая четверка есть, то переходим на пункт (5).

(5) Среди упорядоченных четверок чисел (q_1, q_2, q_3, j), удовлетворяющих условиям пункта (4), находим минимальную четверку $(q_1^0, q_2^0, q_3^0, j^0)$ (в смысле лексикографического порядка). За тетраэдр S_{k+1} принимаем $S(q_1^0, q_2^0, q_3^0, j^0)$. Далее осуществляем следующие действия:

(a) тетраэдр S_{k+1} (то есть четверку чисел $(q_1^0, q_2^0, q_3^0, j^0)$) помещаем в конец списка $\mathcal{E}$;

(b) три грани (q_1^0, q_2^0, j^0), (q_1^0, q_3^0, j^0) и (q_2^0, q_3^0, j^0) тетраэдра S_{k+1} помещаем в конец списка $\mathcal{F}$. Объявляем эти грани граничными;

(c) грань (q_1^0, q_2^0, q_3^0) тетраэдра S_{k+1} (находящуюся в списке $\mathcal{F}$) объявляем внутренней и запоминаем, что она была использована при построении тетраэдра S_{k+1};
(d) увеличиваем k на 1;
(e) идем к пункту (4).

(6) Построена очередная триангуляция. Запоминаем данные об этой триангуляции. Счетчик числа триангуляций увеличиваем на 1.

(7) Из конца списка $\mathcal{E}$ удаляем тетраэдр S_k, а из конца списка $\mathcal{F}$ удаляем три грани тетраэдра S_k. Просматривая получившийся список $\mathcal{F}$, находим четвертую грань тетраэдра S_k и объявляем ее граничной.

(8) Уменьшаем k на 1.

(9) Пусть тетраэдр S_k представлен в списке $\mathcal{E}$ упорядоченной четверкой чисел (i_1, i_2, i_3, i_4). Давая переменной j значения $i_4 + 1, i_4 + 2, \ldots, N$, пытаемся найти такое число j, что $i_4 < j \leq N$, а тетраэдр $\widetilde{S} = S(i_1, i_2, i_3, j)$, не содержащий точек из P, отличных от своих вершин, и совместим с каждым из тетраэдров S_1, S_2, $\ldots$, S_{k-1} из списка $\mathcal{E}$.
Если число j с указанным свойством найдено, то идем к пункту (10).
Если нужное j не найдено, то при $k > 2$ идем к пункту (7), а при $k = 2$ идем к пункту (12).

(10) Полагаем $S_k = \widetilde{S} = S(i_1, i_2, i_3, j)$, для чего в списке $\mathcal{E}$ последнюю четверку (i_1, i_2, i_3, i_4) заменяем четверкой (i_1, i_2, i_3, j). Кроме того, заменяем в списке $\mathcal{F}$ последние три грани (старого) тетраэдра S_k гранями (i_1, i_2, j), (i_1, i_3, j) и (i_2, i_3, j) (нового) тетраэдра $S_k = \widetilde{S} = S(i_1, i_2, i_3, j)$ и объявляем все эти грани граничными.

(11) Идем к пункту (4).

(12) Конец работы.

2. Алгоритм $\mathcal{B}$

Пусть $P = \{p_1, p_2, \ldots, p_N\}$ — конечный набор точек в трехмерном пространстве. Алгоритм $\mathcal{B}$ перечисляет все триангуляции этого набора. Если все точки лежат в одной плоскости, то триангуляций нет. Пусть теперь выпуклая оболочка $\mathrm{conv}(P)$ множества P — многогранник. Не теряя общности, считаем, что несколько первых вершин P (а именно — вершины p_1, p_2, $\ldots$, p_{kv} и только они) лежат в одной грани G этого многогранника, а отрезок $[p_1, p_2]$ является ребром этой грани (или лежит на ребре) и не содержит точек из P, отличных от p_1 и p_2.

Пусть T — произвольная триангуляция множества P. Тогда найдется (и единствен) входящий в эту триангуляцию тетраэдр S_1, одна из граней которого лежит в G, а p_1 и p_2 являются вершинами этой грани. Ясно, что S_1 имеет вид $S(1, 2, i_3, i_4)$, где $2 < i_3 \leq kv$ и $kv < i_4 \leq n$. Алгоритм $\mathcal{B}$ состоит из следующих шагов:

(1) Полагаем $i_3 = 3$.

(2) Полагаем $i_4 = kv + 1$.

(3) Если тетраэдр $\widetilde{S} = S(1, 2, i_3, i_4)$ содержит точки из набора P, отличные от своих вершин, то идем к пункту (5).

(4) Полагаем $S_1 = \widetilde{S}$ и с помощью алгоритма $\mathcal{A}$ перечисляем все триангуляции, содержащие тетраэдр S_1.

(5) Увеличиваем i_4 на 1. Если $i_4 \leq N$, то идем к пункту (3).

(6) Увеличиваем i_3 на 1. Если $i_3 \leq kv$, то идем к пункту (2).

(7) Конец.

Замечание 6. Если какая-либо грань тетраэдра S_1 лежит на грани выпуклой оболочкой $\mathrm{conv}(P)$ множества P, то при выполнении пункта (3) эту грань можно не заносить в список граничных граней, так как к ней нельзя будет в дальнейшем приклеить новый тетраэдр. Таким же образом можно поступать при выполнении подпункта (b) пункта (5). При этом, однако, надо будет запоминать номер того тетраэдра, гранью которого является грань, помещаемая в список $\mathcal{F}$.

Отметим также, что если в пункте (4) алгоритма $\mathcal{A}$ не учитывать лексикографический порядок на множестве граней, то триангуляции могут повторяться.

Несложно убедиться, что алгоритм $\mathcal{B}$ перечисляет все триангуляции на P. Проверим, что каждая триангуляция встретится ровно один раз. Пусть $T = S_1, S_2, \ldots, S_k$ и $T' = S'_1, S'_2, \ldots, S'_l$ — две триангуляции, построенные на различных шагах алгоритма $\mathcal{B}$. Покажем, что эти триангуляции различны. Если $k \neq l$, то это очевидно. Поэтому в дальнейшем считаем, что $k = l$.

Допустим, что $S_1 \neq S'_1$. По построению $S_1 = (1, 2, i_3, i_4)$ и $S'_1 = (1, 2, i'_3, i'_4)$ для некоторых i_3, i_4, i'_3, i'_4. При этом точки p_1, p_2, p_{i_3} и $p_{i'_3}$ лежат в грани G выпуклой оболочки $\mathrm{conv}(P)$ множества P, а отрезок $[p_1, p_2]$ является ребром этой грани (или лежит на ребре). Кроме того, точки p_1, p_2, p_{i_3} являются вершинами тетраэдра S_1, а точки p_1, p_2, $p_{i'_3}$ — вершинами тетраэдра S'_1. Поэтому пересечение тетраэдров S_1 и S'_1 имеет внутренние точки и потому эти тетраэдры не могут входить в одну триангуляцию. Следовательно, $T \neq T'$.

Пусть теперь $S_1 = S'_1$ и $S_2 \neq S'_2$. Пусть $S_2 = (q_1, q_2, q_3, j)$ и $S'_2 = (q'_1, q'_2, q'_3, j')$, где (q_1, q_2, q_3) и (q'_1, q'_2, q'_3) — граничные грани тетраэдра S_1. Допустим, что эти грани различны. Тогда одна из них, например, первая, меньше второй (при лексикографическом порядке). Но тогда при построении триангуляции T' в результате выполнения пунктов (4) и (5) алгоритма $\mathcal{A}$ за S'_2 был бы принят тетраэдр S_2 — противоречие. Поэтому (q_1, q_2, q_3) и (q'_1, q'_2, q'_3) — одна и та же грань. При этом вершины p_j и $p_{j'}$ лежат по одну сторону от этой грани. Поэтому тетраэдры S_2 и S'_2 пересекаются по многограннику и потому не могут входить в одну триангуляцию. Следовательно, $T \neq T'$. Продолжая подобные рассуждения, несложно убедиться, что $T \neq T'$.

Описанные алгоритмы несложно модифицировать так, чтобы они были применимы для конечных наборов точек P из пространства $\mathbf{R}^n$, $n > 3$. Для этого нужно вместо тетраэдров рассматривать симплексы размерности n.

Алгоритмы $\mathcal{A}$ и $\mathcal{B}$ позволяют получить, например, следующие результаты:

(1) Пусть P — набор вершин куба. Тогда число триангуляций равно 74.

(2) Пусть к вершинам куба добавлена точка на ребре куба. Тогда имеется 276 различных триангуляций.

(3) Пусть P — набор вершин куба, к которому добавлен центр грани. Тогда число различных триангуляций равно 150.

(4) Если к вершинам куба добавить середины двух соседних ребер, то число триангуляций увеличится до 1016.

При получении очередной триангуляции можно провести анализ содержащихся в ней тетраэдров (при выполнении пункта (6) алгоритма $\mathcal{A}$). Пусть, например, $\alpha = \alpha(T)$ — наименьший из плоских углов тетраэдров, входящих в триангуляцию T. Тогда в случае

(4) $\alpha > 15^o$ для всех 1016 триангуляций, $15^o < \alpha < 16^o$ для 356 триангуляций, $18^o < < \alpha < 19^o$ для 612 триангуляций и $\alpha > 19^o$ для 48 триангуляций.

Автор признателен В.А. Клячину за полезные обсуждения.

СПИСОК ЛИТЕРАТУРЫ

1. Клячин, В. А. Метод цепей для организации хранения многомерных триангуляций / В. А. Клячин, В. В. Попов // Вестник Волгоградского государственного университета. Серия 1, Математика. Физика. — 2013. — № 2 (19). — С. 71–79.

2. Препарата, Ф. Вычислительная геометрия: Введение / Ф. Препарата, М. Шеймос. — М. : Мир, 1989. — 478 с.

3. Скворцов, А. В. Триангуляция Делоне и ее применение / А. В. Скворцов. — Томск : Изд-во Том. ун-та, 2002. — 128 с.

REFERENCES

1. Klyachin V.A., Popov V.V. Metod tsepey dlya organizatsii khraneniya mnogomernykh triangulyatsiy [Chanes Method For Storage of Multidimensional Triangulation]. *Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 1, Matematika. Fizika* [Science Journal of Volgograd State University. Mathematics. Phisics], 2013, no. 2 (19), pp. 71–79.

2. Preparata F., Sheymos M. *Vychislitelnaya geometriya: Vvedenie* [Computational Geometry. An Introduction]. Moscow, Mir Publ., 1989. 478 p.

3. Skvortsov A.V. *Triangulyatsiya Delone i ee primenenie* [Delaunay Triangulation and its Application]. Tomsk, Izd-vo Tom. un-ta Publ., 2002. 128 p.

ON ALGORITHM OF NUMBERING OF TRIANGULATIONS

Popov Vladimir Valentinovich

Candidate of Physical and Mathematical Sciences, Associate Professor, Department of Computer Sciences and Experimental Mathematics,
Volgograd State University
popov_v_v@rambler.ru, knem03@mail.ru, kiem@volsu.ru
Prosp. Universitetsky, 100, 400062 Volgograd, Russian Federation

Abstract. In [1] the algorithm of numbering of all triangulations of a finite set on the plane is offered. This paper describes the modification of this algoritm for subset of points in three-dimensional space.

Let $P = \{p_1, p_2, \ldots, p_N\}$ is a finite set of points in three-dimensional space. The triangulation of this set is such a sequence S_1, S_2, ..., S_k of tetrahedrons with vertexs from P, which union is equal to convex hull $\mathrm{conv}(P)$ of P, and the intersection $S_i \cap S_j$, $i \neq j$ is or empty, or is the common vertex or the common edge or the common face of tetrahedrons S_i and S_j, and each point p_i is the vertex of some S_j.

At first, the algorithm $\mathcal{A}$ is described, which gives us the list of all triangulations on P, containing some tetrahedron S_1 with vertexs from a set of P without points from P other than his vertexs. Let at some k the list of tetrahedrons be already defined S_1, S_2, ..., S_k which can be completed to some triangulations for P, but the union $V = S_1 \cup S_2 \cup \ldots \cup of S_k$ is not equal to conv(P). Then there will be such two-dimensional face F a set V and such

tetrahedron of S with vertexs from P which doesn't contain points of a set P other than his vertexs, and $V \cap S = F$. Among tetrahedrons S, which can be constructed by this way, we choise such S, that the sequence (i_1, i_2, i_3, i_4) of numbers of his vertex has a minimum value with respect to lexicographic order. Now we put $S_{k+1} = S$. After some such a steps we get a triangulation of P. To build other triangulations it is necessary to delete tetrahedrons with big numbers and add new tetrahedrons before receiving new triangulations.

Let G be a boundary of the set $\mathrm{conv}(P)$. We assume that $G \cap P = \{p_1, p_2, \ldots, p_l\}$, where $l \geq 3$. and segment $[p_1, p_2]$ doesn't contains some points from P other than p_1 and p_2. Let $2 < i_3 \leq l < i_4$ and S is the tetrahedrons with vertexes p_1, p_2,p_{i_3},p_{i_4}, which does not containes the points from P other than his vertexs. Applying algorithm $\mathcal{A}$ to S, we obtain some triangulations of a set P. Changing i_3, i_4, we get all the triangulations. By this way we realies algorithm $\mathcal{A}$.

Algorithms $\mathcal{A}$ and $\mathcal{B}$ allows us to receive, for example, the following results:

(1) Let P is the set of vertexs of a cube. Then the number of triangulations of P is equall 74.

(2) Let $P' = P \cup \{p\}$, where p is the point of the edge of cube. Then P' has 276 triangulations.

Key words: triangulation, tetrahedron, simplex, number of triangulations, convex hull.