Enumeration of Regular Triangulations

中央大学理工学部情報工学科 今井桂子 (Keiko Imai)

東京大学理学部情報科学科　今井 浩 (Hiroshi Imai)

Abstract

Regular triangulations form a meaningful wide subclass of triangulations of points in general dimensions. They can be defined as a natural extension of the Delaunay triangulation and also of lexicographic triangulations, a subclass of triangulations well-known in the theory of oriented matroids. Moreover, regular triangulations have interesting algebraic aspects in connection with a famous paradigm of computer algebra, Gröbner bases. This paper proposes an output-size sensitive and work-space efficient algorithm to enumerate all regular triangulations by reverse search. The algorithm makes full use of the existing results on the secondary polytope [BFS 90, GKZ 94] whose vertices correspond to regular triangulations. Triangulations using all the points are called spanning, and an algorithm is given to enumerate all spanning regular triangulations. The diameter of the secondary polytope is investigated. Preliminary computational results are also shown.

1 Introduction

Triangulations have been one of main topics in computational geometry and other fields in recent years. Especially, some types of triangulations are found to bridge geometric issues and algebraic ones. Regular triangulations are of such a type [BFS 90, GKZ 94]. From the viewpoint of computational geometry, regular triangulations provide a good framework where many known results for triangulations of a planar point set can be generalized to higher dimensional case. For instance, in the planar case, any pair of triangulation can be transformed to each other by a sequence of so-called Delaunay flips, but, as far as the authors know, there is no known result for higher-dimensional triangulations. However, restricting ourselves to the class of regular triangulations in any dimensions, such a result is already shown (see [BFS 90, GKZ 94]). Also, there are several works in computational geometry on regular triangulations such as [ES 92, Fac 95].

Enumeration of all regular triangulations is interesting from the viewpoint of computer-aided mathematical research. Regular triangulations have connection with many mathematical concepts, and by enumerating them mathematical problems can be investigated through computational experiments (e.g., see [DeL 94, DST 95, St 95]). Also, for the three-dimensional case, through the enumeration algorithm, exhaustive and local search can be performed for triangulations of three-dimensional objects in computer graphics, finite element method, etc.

There have been proposed three algorithms for enumerating regular triangulations [BFS 90, DeL 94, Mas 94, Mas 95]. The algorithm by Billera et al. [BFS 90] reduces the problem to constructing the arrangement of $O(n^{d+1})$ homogeneous hyperplanes in the r-dimensional space in $O(n^{(d+1)(r-1)})$ time and space. This algorithm is worst-case optimal. However, in general, the number of regular triangulations may be much smaller than the complexity of the arrangement.

An output-size sensitive algorithm is given by De Loera [DeL 94]. It is based on the breadth-first search enumeration, and is implemented using `Maple` and `MACAULAY`. Since it is based on the breadth-first search, its work-space complexity is $\Omega(T)$, which becomes prohibitively large even for small-size problems. An output-size sensitive and work-space efficient algorithm is originally developed by Masada [Mas 94, Mas 95].

By extending the original work by Masada [Mas 94, Mas 95], this paper proposes an output-size sensitive and work-space efficient algorithm for enumerating regular triangulations of n points in the $(d-1)$-dimensional space. Its time complexity is $O(ds\text{LP}(r, ds)T)$, where where s is the upper bound of the number of simplices contained in one regular triangulation, i.e., $O(n^{\lfloor d/2 \rfloor})$, $\text{LP}(r, ds)$ denotes the time required for solving a linear programming problem with ds strict inequality constraints in $r = n - d$ variables, and T is the number of regular triangulations, which is bounded by $O(n^{(d+1)(r-1)})$. Its work-space complexity is $O(ds)$, which is best possible to retain one triangulation. Our time complexity is proportional to the output size T, and working space is quite small.

2 Regular Triangulations and Secondary Polytope

To simplify the discussion, we assume that n points $V = \{v_1, \cdots, v_n\}$ in general position are given in an affine space $\mathbf{R}^{d-1}$. The *convex hull* of a point set U is referred by conv U. First of all, let us review the definition of triangulations.

Definition 1 (Triangulations) *A triangulation Δ of V is a collection of sets of d points from V satisfying the following conditions:*

- conv $V = \cup_{\sigma \in \Delta}$conv V_σ, *where $\sigma = (\sigma_1, \cdots, \sigma_d)$ and $V_\sigma = \{v_{\sigma_1}, \cdots, v_{\sigma_d}\}$,*

- *for all $\sigma, \tau \in \Delta$, either conv V_σ, conv V_τ have no intersection or intersect in their common face.*

For this point set, regular triangulations are obtained in the following way.

Definition 2 (Regular triangulations) *For the set V of points, we obtain a point set $W = \{(v_1, w_1), \cdots, (v_n, w_n)\} \subset \mathbf{R}^d$ by assigning weights $w_1, \cdots, w_n$ to $v_1, \cdots, v_n \in V$, respectively. Suppose the weights are assigned so that every lower facet (i.e., a facet whose outward normal vector has a negative d-th entry) be a simplex. Then, after projecting the lower facets onto conv V in $\mathbf{R}^{d-1}$, we obtain a triangulation of V. Triangulations constructed in this manner are called regular triangulations of V induced by an assignment of weights $w = (w_1, \cdots, w_n)$. See Fig.1.*

The members of W are called *lifted points*. Definition 2 states that, when an n-dimensional vector w is given, a regular triangulation of V can be constructed if the weights are assigned so that the lifted points be sufficiently generic. Notice that Definition 2 admits regular triangulations which do not use some of given points, while vertices of conv V are necessarily used. The regular triangulations using all points are treated in section 4. The next lemma is an implication of the well-known upper bound theorem of convex polytopes.

Lemma 1 *The number of (any dimensional) simplices in a regular triangulation of V is bounded from above by $O(n^{\lfloor d/2 \rfloor})$.*

For the rest, s denotes the maximum number of simplices in regular triangulations of V, and hence $s = O(n^{\lfloor d/2 \rfloor})$.

With each triangulation, not necessarily regular, a vector is associated as follows.

Definition 3 (Volume vector) *For a triangulation Δ, the volume vector φ of Δ is defined as an n-dimensional vector by:*
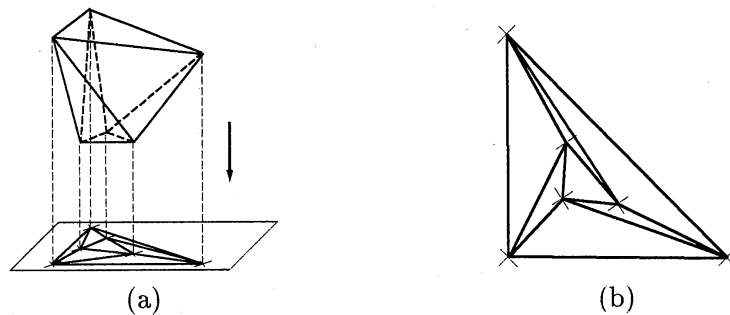
Figure 1: (a) A regular triangulation obtained by the projection, and (b) a nonregular triangulation

$$\varphi_i = \sum_{\sigma \in \Delta, \ v_i \in \sigma} \text{vol}(\sigma), \qquad \varphi = (\varphi_1, \cdots, \varphi_n)$$

where $\text{vol}(\sigma)$ *is the volume of the simplex* $\text{conv}(\{v_{\sigma_1}, \cdots, v_{\sigma_d}\})$.

Notice that i-th entry of the volume vector equals the sum of the volume of all simplices having v_i as its vertex. We adopt *lexicographic ordering* for comparing volume vectors, i.e., for $\varphi, \varphi' \in R^n$, $\varphi > \varphi'$ if and only if there is $i \in \{1, \cdots, n\}$ such that $\varphi_j = \varphi'_j$ for $1 \leq j < i$ and $\varphi_i > \varphi'_i$.

The *secondary polytope* is defined with the volume vectors of all triangulations and we can see properties of the secondary polytope from the following.

Definition 4 *By constructing a convex hull with the volume vectors of all triangulations of* V, *we obtain a convex polytope* $\Sigma(V)$, *called the secondary polytope of* V.

Theorem 1 ([BFS 90, GKZ 94]) *Vertices of the secondary polytope* $\Sigma(V)$ *correspond to regular triangulations one-to-one.*

Note that the volume vectors of non-regular triangulations fall into the interior of $\Sigma(V)$ or the relative interior of some face of $\Sigma(V)$.

Theorem 2 ([BFS 90, GKZ 94]) *For a point set* V *in general position, two distinct vertices in the secondary polytope are connected by an edge if and only if, for the corresponding two distinct regular triangulations* Δ_1 *and* Δ_2, *there exists a circuit* Z *satisfying the following conditions:*

(i) *There are no vertices of* V *inside* $\text{conv} \, Z$ *except for the elements of* Z *itself.*

(ii) $\text{conv} \, Z$ *is a union of the faces of the simplices of* Δ_1 *(and* Δ_2*) and* Δ_1 *and* Δ_2 *coincide outside* $\text{conv} \, Z$.

Here, $\text{conv} \, Z$ can be triangulated in two ways, which correspond to Δ_1 and Δ_2. These are called generalized flips, and are an extension of the original Delaunay flip in the two-dimensional case. Since the graph formed by vertices and edges of a polytope is connected, by a sequence of generalized flips, any two regular triangulations can be transformed to each other. It should be noted that a new triangulation Δ_2 obtained from a regular triangulation Δ_1 and a circuit satisfying the conditions is not necessarily regular, and hence the regularity of Δ_2 should be checked separately.

3 Our Algorithm

We first describe the data structure for representing a regular triangulation for efficient manipulation. First of all, we represent each simplex in the triangulation as a set of d points in V. For the set V of points in general position, a graph formed by simplices and facets of the triangulation suffices to represent the incidence relation of simplices of the triangulation. Each facet is a simplex of $d-1$ points, and we represent it by two points which are the complement of $d+1$ points of adjacent two simplices to the $d-1$ points. This data structure for representing the incidence relation requires $O(ds)$ space, where s is the maximum number of simplices of a regular triangulation of V. Note that the number of facets is $O(ds)$.

Besides this graph, we maintain all circuits satisfying the condition (i), (ii) of Theorem 2 for the triangulation. Each circuit is conceptually represented by a $(d+1)$-tuple of points in the circuit sorted in the increasing order of indices of points. Then, all the circuits are maintained by a list in the lexicographic order of the $(d+1)$-tuples, and this list can be maintained in $O(ds)$ space.

For each regular triangulation, we also maintain its volume vector. By updating triangulations by a generalized flip, we have to maintain these data structures. For example, the volume vector can be updated in $O(d^4)$ time by the flip. Also, to maintain the list of candidate circuits in the sorted order, at most d^2 circuits are deleted and inserted to the list. The list for a flip can be updated in $O(d^2 s)$ time. When a new triangulation is computed, we have to check its regularity by solving the linear programming problem in $O(\text{LP}(r, ds))$ time, as described below. In the sequel, we assume that the time complexity to update the data structure by a flip is dominated by $O(\text{LP}(r, ds))$, since d^4, $d^2 s = O(\text{LP}(r, ds))$ in general.

Now we show that it can be checked by linear programming whether a given triangulation is regular. In the existing literature, the regularity check is done in the dual space. We here give a simple primal approach. For each facet, not on the boundary of conv V, of a given triangulation of the set V of points, there are two simplices sharing the facet. Suppose two simplices have points $\{v_{\sigma_0}, \cdots, v_{\sigma_{d-1}}\}$ and $\{v_{\sigma_1}, \cdots, v_{\sigma_d}\}$. Then, for variables w_i for v_i as its weight, consider the following determinant for each facet where $(v_{\sigma_i,1}, \cdots, v_{\sigma_i,d-1})$ are the coordinate values of v_{σ_i}:

$$
D = \begin{vmatrix}
1 & v_{\sigma_0,1} & \cdots & v_{\sigma_0,d-1} & w_{\sigma_0} \\
1 & v_{\sigma_1,1} & \cdots & v_{\sigma_1,d-1} & w_{\sigma_1} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
1 & v_{\sigma_{d-1},1} & \cdots & v_{\sigma_{d-1},d-1} & w_{\sigma_{d-1}} \\
1 & v_{\sigma_d,1} & \cdots & v_{\sigma_d,d-1} & w_{\sigma_d}
\end{vmatrix} .
$$

Note that for two simplices having points $\{v_{\sigma_0}, \ldots, v_{\sigma_{d-1}}\}$ and $\{v_{\sigma_1}, \ldots, v_{\sigma_d}\}$ in a triangulation, the sign of the coefficient of w_{σ_0} is the same as w_{σ_d}'s in the determinant D.

We can prove that a given triangulation is regular if and only if there is a solution w satisfying the following strict linear inequalities defined for each facet:

$$
\text{sign}\left(\begin{vmatrix}
1 & v_{\sigma_0,1} & \cdots & v_{\sigma_0,d-1} \\
1 & v_{\sigma_1,1} & \cdots & v_{\sigma_1,d-1} \\
\vdots & \vdots & \ddots & \vdots \\
1 & v_{\sigma_{d-1},1} & \cdots & v_{\sigma_{d-1},d-1}
\end{vmatrix} \right) \cdot D > 0 .
$$

Thus, in a primal way, the regularity can be checked by linear programming. It is easy to see that for a fixed simplex we can set $w_{\sigma_i} = 0$ for each point of the simplex without changing the existence of the solution. Hence, this system of strict linear inequalities is to check the existence of a solution to $r = n - d$ variables and at most ds constraints. In general, a linear programming problem is formulated with non-strict inequalities, and a method of solving this type of problem is developed. It is easy to see that our problem with strict inequalities can also be solved by the general method. Denote by $LP(r, ds)$ the time required to solve this linear programming problem.

Theorem 3 *Whether a given triangulation is regular can be judged in* $LP(r, ds)$ *time.*

Our algorithm requires a regular triangulation to start. This can be an arbitrary regular one. For conceptual simplicity and some technical merits, we may consider two candidates for the initial one. One is a regular triangulation whose volume vector is lexicographically maximum among all volume vectors. The other is the Delaunay triangulation. In any case, the time necessary for obtaining the initial regular triangulation is negligible in comparison with the time necessary for the rest of the enumeration algorithm.

Suppose we have a lexicographically maximum regular triangulation. The reverse search technique considers a rooted tree R of the graph of vertices and edges of the secondary polytope, with the root corresponding to this triangulation. The rooted tree can be defined as follows.

Definition 5 (Reverse search tree) *The reverse search tree R of the secondary polytope with respect to the lexicographic maximization of volume vectors is a subgraph of the graph, formed by vertices and edges of the polytope, such that from each vertex except the lexicographically maximum one a directed edge emanates to a vertex adjacent to it whose volume vector is lexicographically maximum among those of adjacent vertices.*

The following Lemma can be obtained from linear programming theory (the same fact is used in Rote [Rot 92]).

Lemma 2 *R is a directed tree with a unique root with lexicographically maximum volume vector.*

The reverse search technique traverses this tree R in a depth-first manner. To perform it, we need to arrange children in some order for each vertex in the tree. Each child of a vertex is obtained by flipping with respect to some circuit. For the regular triangulation corresponding to the parent vertex, we maintain all the circuits in the sorted order, and this ordering is adopted in the search. Of course, in performing the depth-first search, we go down to the youngest unvisited child from a vertex.

Now that we have a rooted tree with ordered brother relations, the depth-first search can be performed efficiently with the data structure described above. Then we obtain the following.

Lemma 3 (1) *For a vertex in the reverse search tree, its parent can be computed in $O(dsLP(r, ds))$ time and $O(ds)$ space.*

(2) *During the whole traverse by the depth-first search, the time to spend for listing children of a vertex in the order is $O(dsLP(r, ds))$ time in $O(ds)$ space.*

Summarizing the above discussion we now have the following.

Theorem 4 *Regular triangulations of n points in R^{d-1} in general position can be enumerated in $O(dsLP(r, ds)T)$ time and $O(ds)$ working space.*

4 Spanning Regular Triangulations

We call a regular triangulation using all points *spanning*. The first question concerning spanning regular triangulations is whether their corresponding vertices are connected by edges in the secondary polytope. To investigate this, consider the weight vector w_D with $w_i = \|v_i\| = \sum_{j=1}^{d-1}(v_{i,j})^2$. The corresponding regular triangulation is the Delaunay triangulation. Consider transforming a spanning regular triangulation into the Delaunay one. Then, the following holds.

Lemma 4 *From a spanning regular triangulation, we can generate a sequence of regular triangulations to the Delaunay triangulation by generalized flips such that*

(1) *all the regular triangulations appearing in this process are spanning, and the inner product of w_D and the volume vector of a regular triangulation is strictly decreasing, and furthermore*

(2) *a circuit used in a generalized flip in the sequence is never used again in this process.*

Theorem 5 *All the spanning regular triangulations can be enumerated in $O(ds\mathrm{LP}(r, ds)T')$ time and $O(ds)$ working space, where T' is the number of spanning regular triangulations.*

We have the following theorem from arguments in the proof of Lemma 4.

Theorem 6 *The diameter of the secondary polytope is $O(n^{d+1})$.*

5 Preliminary Computational Results

We here describe computational results for randomly generated points. Concerning the results for regularly structured point sets which are interesting from mathematical viewpoints, see Masada [Mas 94, Mas 95]. These are still preliminary results and we just show them here.

Our algorithm is implemented in C language. The experiments are done on Sun SPARCstation 10 with 64MB memory. Exact arithmetics are realized by GNU MP library for arbitrary precision integer and rational number arithmetic. Linear programming problems are solved by a simplex method with Bland's rule. The space complexity is a little more than $O(ds)$ for speeding up the computation in this implementation. Our implementation also works for degenerate inputs.

We here show the number of simplices of regular triangulations when the points are randomly generated in the $(d-1)$-cube with the edges of length 1000. Every coordinate is an integer less than or equal to 500 and more than -500.

- $r = 3$; this is, so to speak, the first non-trivial case, since in the case of $r = 2$ all triangulations are regular.

    - $n = 5, d = 2$: Each of 20 configurations has 8 regular triangulations.
    - $n = 6, d = 3$: 2 of 20 configurations have 16 regular triangulations, 6 of them have 15 ones, and 12 of them have 14 ones.
    - $n = 7, d = 4$: 2 of 20 configurations have 27 regular triangulations, and 18 of them have 25 ones.
    - $n = 8, d = 5$: 3 of 20 configurations have 40 regular triangulations, 3 of them have 41 ones, 7 of them have 42 ones, 3 of them 43 ones, and the other four have 44 ones.

- $r = 4$;

  - $n = 6, d = 2$: Each of 20 configurations has 16 regular triangulations.
  - $n = 7, d = 3$: The number of regular triangulations is quite various. 4 of 20 configurations have 42 ones, 9 of them have 46 ones, 2 of them have 50 ones, one of them has 51 ones, and 2 have 55 one, and two other configurations have 56 regular triangulations, respectively.
  - $n = 8, d = 4$: In this case the number regular triangulations varies from 128 to 168 with some small peak around 133.

Acknowledgments

References

[AF 92] Avis, D., and K. Fukuda, A Pivoting Algorithm for Convex Hulls and Vertex Enumeration of Arrangements and Polyhedra, *Discrete & Computational Geometry*, Vol.8 (1992), pp.295–313.

[BFS 90] Billera, L. J., P. Filliman, and B. Sturmfels, Constructions and Complexity of Secondary Polytopes, *Advances in Mathematics*, Vol.83 (1990), pp.155–179.

[Cha 91] Chazelle, B., An Optimal Convex Hull Algorithm and New Results on Cuttings, *Proc. 32nd Annual IEEE Symposium on Foundations of Computer Science*, 1991, pp.29–38.

[DeL 94] De Loera, J., Computing Regular Triangulations of Point Configurations, Preprint, May 1994.

[DST 95] De Loera, J., B. Sturmfels and R. R. Thomas, Gröbner Bases and Triangulations of the Second Hypersimplex, *Combinatorica*, Vol.15, No.3 (1995), pp.409–424.

[ES 92] Edelsbrunner, H., and N. R. Shah, Incremental Topological Flipping Works for Regular Triangulations, *Proc. 8th Annual ACM Symposium on Computational Geometry*, 1992, pp.43–52.

[Fac 95] Facello, M.A., Implementation of a Randomized Algorithm for Delaunay and Regular Triangulations in Three Dimensions, *Computer-Aided Geometric Design*, Vol.12, No.4 (1995), pp.349–370.

[GKZ 94] Gelfand,I. M., M. M. Kapranov and A. V. Zelevinski, *Discriminants, Resultants and Multidimensional Determinants*, Birkhäuser, Boston, 1994.

[Lee 91] Lee, C., Regular Triangulations of Polytopes, in *Applied Geometry and Discrete Mathematics*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol.4, 1991, pp.443–456.

[Mas 94] Masada, T., An Output Size Sensitive Algorithm for the Enumeration of Regular Triangulations, Technical Report 94-1, Department of Information Science, University of Tokyo, November 1994.

[Mas 95] Masada, T., *An Algorithm for the Enumeration of Regular Triangulations*. Master's Thesis, Department of Information Science, University of Tokyo, March 1995.

[Rot 92] Rote, G., Degenerate Convex Hulls in High Dimensions Without Extra Storage, *Proc. 8th Annual ACM Symposium on Computational Geometry*, 1992, pp.26–32.

[Sei 86] Seidel, R., Constructing Higher-Dimensional Convex Hulls at Logarithmic Cost per Face, *Proc. 18th Annual ACM Symposium on the Theory of Computing*, 1986, pp.404–413.

[St 91] Sturmfels, B., Gröbner Bases of Toric Varieties, *Tohoku Math. Journal*, Vol.43, No.2 (1991), pp.249–261.

[St 95] Sturmfels, B., Gröbner Bases and Convex Polytopes, Expanded Lecture Notes from the Holiday symposium at New Mexico State University, April 1995.