Combinatorial Properties of One-Dimensional Arrangements

Frédéric Cazals

CONTENTS

Motivated by problems from computer graphics and robotics—namely, ray tracing and assembly planning—we investigate the combinatorial structure of arrangements of segments on a line and of arcs on a circle. We show that there are, respectively, $1\times 3\times 5\times\cdots\times(2n-1)$ and $(2n)!/n!$ such arrangements; that the probability for the i-th endpoint of a random arrangement to be an initial endpoint is $(2n-i)/(2n-1)$ or $\frac{1}{2}$, respectively; and that the average number of segments or arcs the i-th endpoint is contained in are $(i-1)(2n-i)/(2n-1)$ or $(n-1)/2$, respectively. The constructions used to prove these results provide sampling schemes for generating random inputs that can be used to test programs manipulating arrangements.

We also point out how arrangements are classically related to Catalan numbers and the ballot problem.

1. INTRODUCTION AND MOTIVATION

Consider a set of n intervals in the real line, and assume that all $2n$ endpoints are distinct. We will be interested in the combinatorial properties of such arrangements, that is, the properties that depend solely on the order in which the endpoints occur, rather than their precise position. Specifically, we will count the number of possible arrangements and determine two statistics (averaged over all possible arrangements) for the i-th endpoint in the sequence: the average number of intervals that this point belongs to, and the probability that it is an initial, rather than terminal, endpoint. We also consider the analogous problem for arcs in a circle.

The overview of the paper is the following. Section 1 briefly discusses the applications that led to this investigation. Sections 2 and 3 deal with the linear and circular cases, respectively. Section 4 lists some interesting open problems.

Work on this paper was accomplished while the author was visiting the Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305 USA.

Keywords: combinatorics, computational geometry, algorithms and data structures.

1058-6458/97 $0.50 per page

Assembly Sequencing and Arrangements

Assembly sequencing is a domain of robotics whose purpose is, given a collection of mechanical parts that fit together in a certain way and a class of motions that these parts can be subjected to, to compute a way, if one exists, to get the single parts from the whole assembly. For example, in the simple assembly in Figure 1, if we restrict ourselves to translations in the plane, it is clear that P_1 and P_2 can only be taken apart by a horizontal motion, whereas P_3 and P_4 can be taken apart by motions within an interval of directions.

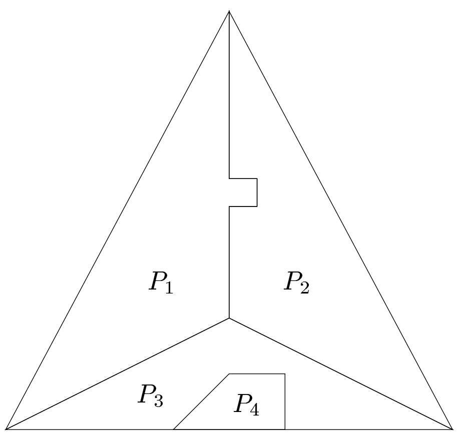

FIGURE 1. A simple assembly.

Analyzing assembly sequences can be of great use in many ways: for example, to check that the product can be disassembled, to ensure that the parts that may be serviced often are easily accessible, or to facilitate recycling by clustering parts made of the same material. Of major practical interest, assembly sequencing is also a difficult algorithmic problem since it is intractable in its general form; see [Natarajan 1988], for example. Restricted, yet interesting, versions of the problem have been shown to have polynomial-time algorithms.

For example, consider the case of planar polygonal assemblies where the only class of motions allowed is infinite translations and where each split results in two subassemblies [Wilson and Latombe 1994; Latombe et al. 1996]. The space of motions is described by the unit circle S^1, since a translation corresponds to a unit vector in the plane. Given two parts, the set of directions along which one can be translated without colliding with the other is described by an arc on S^1, determined by a cone on the Minkowski difference [Latombe 1991]; see Figure 2.

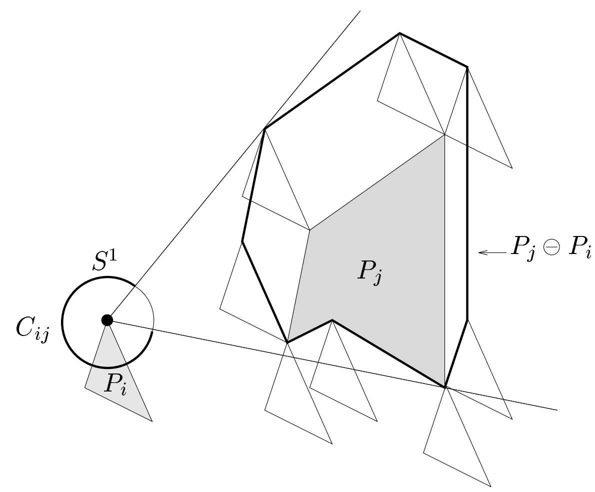

FIGURE 2. The arc of directions of movement of P_i that lead to collision with P_j is given by the cone on the Minkowski difference set $P_j \ominus P_i$.

The blocking relations for all the pairs of parts are thus described by $n(n-1)$ arcs in S^1. Together, they constitute an *arrangement of arcs* that divides S^1 into *endpoints* and *intervals*, as shown on Figure 3. This arrangement is called the *non-directional blocking graph*, or NDBG, since it gives the blocking relations for any pair of parts and any direction. To each endpoint of the arrangement corresponds a directed graph, called the *directional blocking graph*, having a vertex for each part and an edge between vertices i and j if part i collides with part j when translated along this direction. A topological sorting of the strongly connected components of this directed graph gives the removable subassemblies along this direction. Starting with the full assembly, the disassembly algorithm consists in recursively removing translatable subassemblies with the previous scheme.

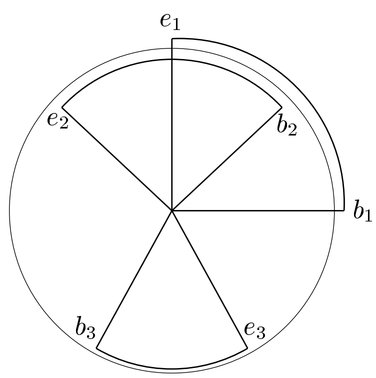

FIGURE 3. Arrangement of arcs on S^1.

FIGURE 4. A ray-traced scene.

Performing a worst-case analysis of this algorithm is pretty easy. Indeed, the NDBG has $O(n^2)$ vertices and each DBG has size $O(n^2)$, which gives a space requirement of $O(n^4)$. The time complexity of the recursive disassembly is $O(n^5)$ since there are at most n levels of recursion, and each level requires examining $O(n^2)$ DBGs for which the reduced graph (graph of the strongly connected components) and a topological sorting have to be computed.

The average-case analysis is much more challenging. Firstly, a precise understanding of the combinatorics of arc arrangements is required. Secondly, some random graph structure is needed for the directional blocking graphs. The latter question is difficult since the number of edges of a DBG depends on the geometric information encoded in the relative position of the pairs of parts, which requires some definition of random assemblies. This goes beyond the scope of this paper. By contrast, the first problem is better defined and raises precise questions such as the generation of a random arrangement (see also [Zimmermann 1994]), the probability of a given endpoint to be an initial or terminal endpoint, the average number of arcs a given endpoint of an arrangement is contained in, and so on. These questions will be addressed in Section 3.

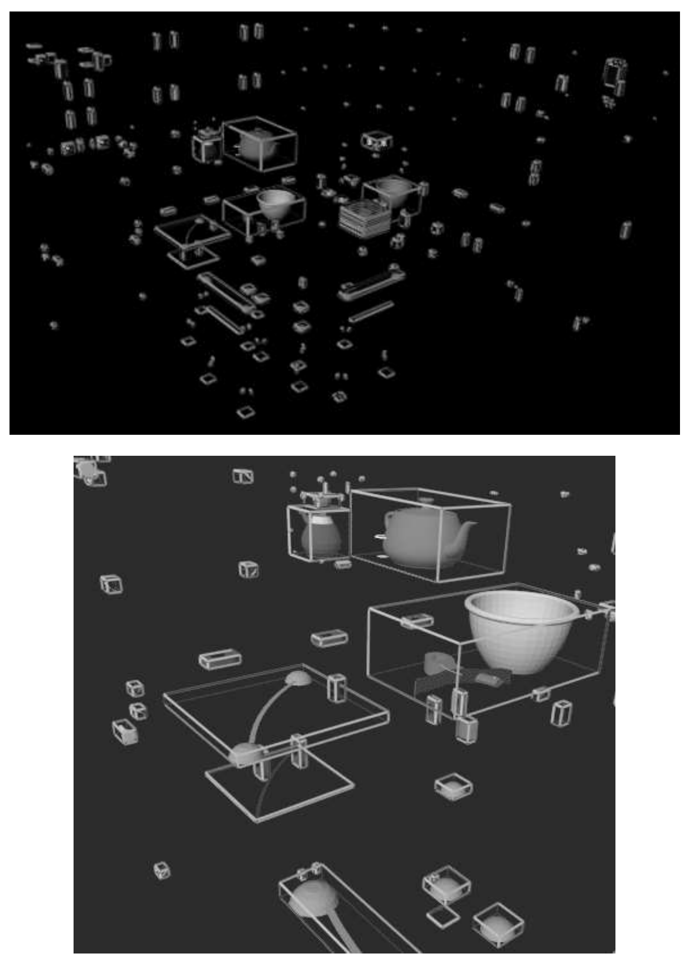

FIGURE 5. Clusters found in the scene. Bottom: detail.

Ray Tracing and Clustering

Ray tracing is a technique from computer graphics that consists in computing views of scenes defined by geometric primitives. Very often these primitives are polygons defined by their geometry and color, a given object of the scene being defined by a set of such polygons. As an example, consider Figure 4, where the kitchen model consists of about 25,000 polygons, and objects such as the bowl on the table or the teapot are made of about 1000 polygons. To sketch the ray-tracing algorithm (see [Foley et al. 1990] for details), let a ray be defined by a point and a direction in three dimensions. Rays are used to simulate the light received by the observer's eye, so that the key operation of the whole algorithm consists in finding, for a given ray, the closest object hit in order to plot the corresponding color on the screen of the computer where the algorithm is run.

Reducing the number of ray-polygon intersection tests has ever been a challenging issue. The main paradigm consists in partitioning the volume containing the scene into small boxes, in order to test for intersection only those polygons stored in the boxes of the partition crossed by the ray of interest. An example of such partitioning, the *uniform grid*, is based on a regular grid aligned with the three coordinate axes. (See [Cazals et al. 1995] for a discussion of grid-like data structures.)

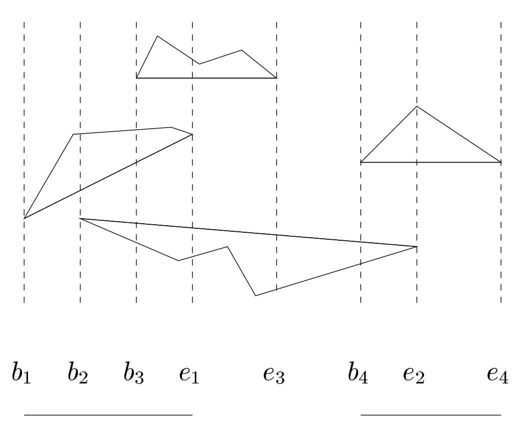

FIGURE 6. Arrangement of line segments.

The problem with this approach is that whenever too many polygons fall into the same box the spatial partitioning does not result in data partitioning, so the number of ray-polygon intersection tests is not reduced significantly. To remedy this problem, it was observed in [Cazals et al. 1995] that using uniform grids for densely populated areas of the scene called clusters could partially solve the problem. Examples of clusters are the neighborhoods of the bowl, teapot, or door knobs, and are depicted on Figure 5. More precisely, a cluster is defined as a subset of objects whose projection along the three axis x, y and z is almost-connected. And, since the projection of a polygon on a line is a line segment, the clustering algorithm analysis turns out to be closely related to the combinatorics of arrangement line segments, as in Figure 6. Thus, the results presented in Section 2 of this paper were recently used in [Cazals and Sbert 1997] in conjunction with integral geometry techniques to define statistics aiming at characterizing standard scenes types such as natural models, architectural scenes, etc.

2. THE LINEAR CASE

Notations and Previous Work

Consider an set of n segments on the line. Let the $2n$ endpoints, which are assumed distinct, be indexed in order by $(1 \,.\,.\, 2n) = \{1, 2, 3, \ldots, 2n\} \subset \mathbb{Z}$, an orientation having been fixed in advance. From the combinatorial point of view, the arrangement of segments is specified completely by an involution a of $(1 \,.\,.\, 2n)$ without fixed points. More precisely, a segment joining endpoints i and j is denoted $[i, j]$, if $i < j$; the endpoint-pairing involution maps i to j and j to i, and we call i and j the initial and terminal endpoints of the pair. For instance, the three possible arrangements of two segments are shown in Figure 7: they are $\{[1, 2], [3, 4]\}$, $\{[1, 3], [2, 4]\}$,

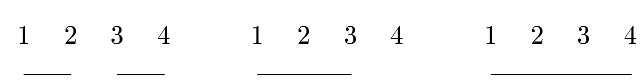

FIGURE 7. The possible arrangements of two segments.

and $\{[1,4], [2,3]\}$. The arrangement $\{[1,2], [3,4]\}$ is also thought of as the pairing $1 \leftrightarrow 2$, $3 \leftrightarrow 4$.

Let S_n be the set of all arrangements of n segments, and let $s_n = |S_n|$, where the bars denote cardinality; thus $s_2 = 3$ (compare Figure 7). In general, we have

$$s_n = 1 \times 3 \times \cdots \times (2n{-}1),$$

as can easily be seen: the pairing can take 1 to any of the $2n-1$ remaining indices; it can take the lowest of the remaining $2n-2$ indices into any of the remaining $2n-3$; and so on.

For a particular arrangement $a \in S_n$ and for $i \in (1\,..\,2n)$, we define $a[i]$ to be B or E according to whether endpoint i begins or ends the respective segment, that is, according to whether $a(i) > i$ or $a(i) < i$. For fixed i, the statistics we are interested in are the probability that $a[i] = B$ (or $a[i] = E$), as a ranges over all of S_n, and the *overlap number* of i, that is, the average number of arcs or line segments in whose interior endpoint i is contained. Formally, we define

$$\begin{aligned}
\beta_i^{(n)} &= |\{a \in S_n : a[i] = B\}|,\\
\varepsilon_i^{(n)} &= |\{a \in S_n : a[i] = E\}|,\\
\tau_i^{(n)} &= \sum_{a \in S_n} |\{(b,e) \in a : b < i < e\}|.
\end{aligned}$$

The corresponding vectors as i ranges over $(1\,..\,2n)$ are denoted $\vec{\varepsilon}^{(n)}$, $\vec{\beta}^{(n)}$, and $\vec{\tau}^{(n)}$. Thus for $n = 2$ we have $\vec{\beta}^{(2)} = [3,2,1,0]$, $\vec{\varepsilon}^{(2)} = [0,1,2,3]$, $\vec{\tau}^{(2)} = [0,2,2,0]$ (Figure 7).

The numbers s_n have appeared in the literature in several forms, in particular in [Touchard 1950; Riordan 1975], which deal with the stamp-folding problems. The value of s_n is given by Touchard. Riordan mentions that the number of pairings of $2n$ points on a circle is also s_n, since such pairings, too, can be seen as involutions of $(1\,..\,2n)$. (More geometrically, one can open up the circle at an arbitrary point; then a pair of points on S^1 corresponds to a segment in the resulting interval, and vice versa.) Finally, a look at the very nice book [Sloane and Plouffe 1995, M3002] shows that the sequence s_n has long been known in connection with the expression of Wallis integrals.

Riordan [1975] also points out the interesting relation between the number of pairings on a circle and the Catalan numbers: pairings where chords are not allowed to intersect give rise to the Catalan numbers $C_n = \binom{2n}{n}/(n+1)$, while pairings that allow crossings between the chords lead to s_n. Riordan cites a correspondence between the Catalan numbers and the ballot problem, also known as the subdiagonal random walks problem [Comtet 1974; Yaglom and Yaglom 1964; Knuth 1973].

Initial and Terminal Endpoints

Theorem 2.1. *For any $i = (1\,..\,2n)$ we have*

$$\beta_i^{(n)} = \frac{2n-i}{2n-1}\, s_n = (2n-i)s_{n-1} = s_n - (i-1)s_{n-1}.$$

Therefore the probability that the i-th endpoint is initial is $(2n{-}i)/(2n{-}1)$, and the probability that it is final is $(i{-}1)/(2n{-}1)$.

Proof. We use the recursion

$$\beta_i^{(n+1)} = (i{-}1)\beta_{i-1}^{(n)} + s_n + (2n{+}1{-}i)\beta_i^{(n)}, \quad (2.1)$$

for $i = (1\,..\,2n{+}1)$, with initial condition $\beta_{2n+2}^{(n+1)} = 0$. This recursion can be verified as follows. Given an element of S_{n+1}, let $i \in (1\,..\,2n{+}1)$ be the initial point of the segment whose terminal endpoint is $2n+2$. If we remove the pair $[i, 2n{+}2]$ and renumber, we get a well-defined element of S_n. Conversely, a choice of $a \in S_n$ and $i \in (1\,..\,2n{+}1)$ yields a unique element $a' \in S_{n+1}$, by the addition of a segment that starts between position $i-1$ and i of a and ends at the far right. (Incidentally, this is another way to derive the value of s_n, since it shows that $|S_{n+1}| = (2n{+}1)|S_n|$.) Because of the renumbering, we have

$$a'[j] = \begin{cases} a[j] & \text{if } j < i,\\ a[j{-}1] & \text{if } i < j < 2n+2i; \end{cases}$$

moreover $a'[i] = B$ and $a'[2n+2] = E$. Analyzing the contribution to each $\beta_j^{(n+1)}$ from each value of i, we can write:

$$\begin{array}{llllllll}
 & [\,\beta_1^{(n+1)} & \beta_2^{(n+1)} & \beta_3^{(n+1)} & \ldots & \beta_{2n+1}^{(n+1)} & \beta_{2n+2}^{(n+1)}\,] & \\
= & [\quad s_n & \beta_1^{(n)} & \beta_2^{(n)} & \ldots & \beta_{2n}^{(n)} & 0\quad] & (i=1) \\
+ & [\quad \beta_1^{(n)} & s_n & \beta_2^{(n)} & \ldots & \beta_{2n}^{(n)} & 0\quad] & (i=2) \\
 & & & & \vdots & & & \\
+ & [\quad \beta_1^{(n)} & \beta_2^{(n)} & \beta_3^{(n)} & \ldots & s_n & 0\quad] & (i=2n+1)
\end{array}$$

Summation by columns gives the desired recurrence relation (2.1). (Note that in this relation the undefined quantities $\beta_{i-1}^{(n)}$ when $i=0$ and $\beta_i^{(n)}$ when $i = 2n+1$ are multiplied by zero, so the equation still makes sense.)

We now prove the closed-form expression for $\beta_i^{(n)}$. We certainly have $\beta_1^{(n)} = 1$; assume by induction that $\beta_i^{(n)} = s_{n-1}(2n-i)$ for $i \in (1\,..\,2n)$. We get, for any $i \in (2\,..\,2n{+}1)$:

$$\begin{aligned}
\beta_i^{(n+1)} &= (i-1)s_{n-1}(2n-i+1) + s_n \\
&\qquad\qquad + (2n-i+1)s_{n-1}(2n-i) \\
&= s_n + s_{n-1}(2n-1)(2n-i+1).
\end{aligned}$$

But $s_n = s_{n-1}(2n-1)$, which completes the proof for $i \in (1\,..\,2n{+}1)$. The case $i = 2n+2$ is trivial.

The probability that endpoint i is initial in an n-point arrangement is of course $\beta_i^{(n)}/s_n$, and the probability that it is terminal is the complement. This proves the theorem. □

The Overlap Number

Theorem 2.2. *For any $i = (1\,..\,2n)$ we have $\tau_i^{(n)} = (i-1)(2n-i)s_{n-1}$. Thus, the average overlap number of the i-th endpoint in an n-segment arrangement is $(i-1)(2n-i)/(2n-1)$.*

It is possible to prove this using recursion, much like Theorem 2.1; but here a nicer direct proof:

Proof. Endpoint i is covered by segments of the form $[j,k]$ for $j \in (1\,..\,i{-}1)$ and $k \in (i{+}1\,..\,n)$, and there are $(i-1)(2n-i)$ such segments. Each of them appears exactly s_{n-1} times in the s_n arrangements, since once we have fixed segment $[j,k]$ we are left with an arrangement of $n-1$ segments. □

3. THE CIRCULAR CASE

We now turn to arrangements of arcs in the circle, and answer the same questions that were posed in Section 2 for linear segments. Because all endpoints are equivalent on S^1, the situation is easier.

We start with the number of arrangements:

Theorem 3.1. *The number r_n of arrangements of n arcs on a circle is equal to $(2n)!/n!$.*

Proof. An arrangement of n arcs is specified by a pairing of the $2n$ points, together with n independent binary choice, one for each of pair of endpoints (either arc determined by the pair may appear in the arrangement; see Figure 8). Therefore $r_n = 2^n \cdot s_n = 2^n((2n-1) \times (2n-3) \times \cdots \times 3 \times 1) = (2n)!/n!$. □

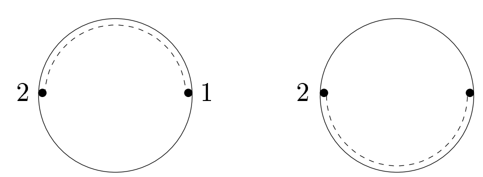

FIGURE 8. Two arcs are determined with equal probability by a choice of two endpoints.

The classification of the arrangements into pairings also yields the probability that a given endpoint is initial. Because, for a given pair of endpoints, each of the two choices of a segment with those endpoints occurs in half the arrangements that include this pair of endpoints, the probability that a fixed endpoint is initial is $\frac{1}{2}$.

The same reasoning shows that the average overlap number of any endpoint in an arrangement of n arcs is $(n-1)/2$: if endpoint i is chosen and we consider the relation of i with any pair (r,s) with $r,s \neq i$, we see that i lies in the interior of the arc with endpoints (r,s) for exactly half the arrangements that include this pair.

4. CONCLUSIONS

The analysis in Section 2 is of interest for computer graphics algorithms dealing with objects' projections along lines. The results in Section 3 may be the first step toward an average case analysis of the NDBG-based algorithm for computing assembly sequences in the simple case of polygons in the plane moved with infinite translations. Although this particular assembly sequencing problem might appear quite restrictive, it is actually one of the few for which it is reasonable to come up with an implementation for, so that any precise analysis would be of interest.

We remark that, from the study of the combinatorial structure of arrangements presented in this paper, it is easy to randomly generate such arrangements in order to test and validate geometric software. An algorithm to do this might go as follows.

Assume we have an array t of integers, of length $2n$, and two functions: $\mathrm{swap}(t, i, j)$, which swaps the contents of slots i and j in t, and $\mathrm{random}(k)$, which returns an integer in the range $1 \,..\, k$. The algorithm returns the endpoint b_i and e_i, for $i \in (1 \,..\, k)$, of the arrangement being generated.

$$
\begin{aligned}
&\text{for } i \in (1\,..\,2n) \text{ do}\\
&\quad t[i] \leftarrow i;\\
&\text{for } i \in (1\,..\,n) \text{ do}\\
&\quad p \leftarrow t[\mathrm{random}(2n{+}2{-}2i)];\\
&\quad \mathrm{swap}(t,\, 2n{+}2{-}2i,\, p);\\
&\quad q \leftarrow t[\mathrm{random}(2n{+}1{-}2i)];\\
&\quad \mathrm{swap}(t,\, 2n{+}1{-}2i,\, q);\\
&\quad b_i \leftarrow \inf(p, q);\\
&\quad e_i \leftarrow \sup(p, q);
\end{aligned}
$$

Many interesting issues remain open, in particular the calculation of higher moments for the statistics presented here. It would be interesting to find two-dimensional analogs for the results presented here; the work done so far in this direction deals with arrangements of lines in the plane, but not line segments [Edelsbrunner 1986].

ACKNOWLEDGMENTS

This work received a financial support from Matra Datavision for a joint project on assembly planning with the Stanford University Robotics Laboratory.

The author wishes to thank Danny Halperin and Jean-Marc Schlenker for insightful comments, the reviewer of this paper for important pointers to the bibliography, and Stéphane Rivière and Ram Ramkumar for rereading the paper.

The author also thanks Don Greenberg of the Cornell University program of Computer Graphics for the kitchen model in Section 1.

REFERENCES

[Cazals and Sbert 1997] F. Cazals and M. Sbert, "Some integral geometry tools to estimate the complexity of 3D scenes", Technical report, INRIA, 1997.

[Cazals et al. 1995] F. Cazals, G. Drettakis, and C. Puech, "Filtering, clustering and hierarchy construction: a new solution for ray-tracing complex scenes", *Computer Graphics Forum* **14**:3 (1995).

[Comtet 1974] L. Comtet, *Advanced combinatorics*, D. Reidel, Dordrecht and Boston, 1974. Translated from the French by J. W. Nienhuys.

[Edelsbrunner 1986] H. Edelsbrunner, *Algorithms in Combinatorial Geometry*, Springer, 1986.

[Foley et al. 1990] J. Foley et al., *Computer Graphics: Principles and Practice*, 2nd ed., Addison Wesley, 1990.

[Knuth 1973] D. E. Knuth, *The art of computer programming*, Addison-Wesley, 1973.

[Latombe 1991] J. C. Latombe, *Robot motion planning*, Kluwer, 1991.

[Latombe et al. 1996] J. Latombe, R. H. Wilson, and F. Cazals, "Assembly sequencing with toleranced parts", *Computer-Aided Design* (1996). To appear.

[Natarajan 1988] B. K. Natarajan, "On planning assemblies", pp. 299–308 in *Proc. Fourth Annual ACM Symp. Comput. Geom.* (Urbana/Champaign, 1988), Assoc. Computing Machinery (ACM), New York, 1988.

[Riordan 1975] J. Riordan, "The distribution of crossings of chords joining pairs of $2n$ points on a circle", *Math. Comp.* **29** (1975), 215–222.

[Sloane and Plouffe 1995] N. J. A. Sloane and S. Plouffe, *The encyclopedia of integer sequences*, Academic Press, 1995.

[Touchard 1950] J. Touchard, "Contribution à l'étude du problème des timbres poste", *Canadian Journal of Mathematics* **2** (1950), 385–398.

[Wilson and Latombe 1994] R. H. Wilson and J.-C. Latombe, "Geometric reasoning about mechanical assembly", *Artificial Intelligence* **71** (1994), 371–396.

[Yaglom and Yaglom 1964] A. M. Yaglom and I. M. Yaglom, *Challenging mathematical problems with elementary solutions*, Holden-Day, 1964.

[Zimmermann 1994] P. Zimmermann, "Gaia: a package for the random generation of combinatorial structures", *Maple Technical Newsletter* **1**:1 (1994), 1–9.

Frédéric Cazals, iMAGIS-IMAG, BP 53, 38041 Grenoble cedex 09, France (Frederic.Cazals@imag.fr)

Received July 17, 1995; accepted in revised form September 14, 1996