On Average-case Complexity of Ray Tracing Algorithms

Gábor Márton and László Szirmay-Kalos

Department of Process Control
Technical University of Budapest
Budapest, Műegyetem rkp. 9/R, H-1111, Hungary
marton@seeger.fsz.bme.hu

Abstract

A theoretical framework for analyzing average-case time and storage complexity of ray tracing acceleration techniques is introduced by means of homogeneous spatial Poisson point processes. Then, as a demonstrative example of its application, the expected query time of the widely known technique based on a regular spatial grid is analyzed. Finally, an interpretation of the results is presented within the context of probability theory.

1 Introduction

Ray tracing [Whi80] is a very time-consuming image synthesis method but it can not be neglected due to its unique ability to model optical phenomena such as reflection or refraction of light. Since the overwhelming majority of running time is spent on ray-object intersection tests, extensive research work has been devoted to accelerate these operations. Two main directions of research seem to exist:

1. *Worst-case optimization.* The problem is known as the *ray shooting problem* within the framework of computational geometry: *given a set objects in the 3-space, preprocess them into a data structure so that the first object intersected by any query ray can be retrieved in optimal time.* This optimal time is probably $O(\log n)$, where n is the number of objects. Mark de Berg [dB92] has recently developed efficient ray shooting algorithms. His most general algorithm can shoot arbitrary rays into a set of arbitrary polyhedra with n edges altogether, with a query time of $O(\log n)$ and preprocessing time and storage of $O(n^{4+\varepsilon})$, where ε is a positive constant that can be made as small as desired. The overview of Schmitt *et al.* [SML88] refers to other algorithms with similar time and storage characteristics. Unfortunately, the complexity of the preprocessing and storage of these algorithms makes them not too attractive for practical use.

2. *Average-case optimization.* There are a number of techniques, on the other hand, which are more suitable for practical use. Arvo and Kirk presented a comprehensive overview of the basic techniques [AK89]. We can consider these algorithms as *heuristic methods* for two reasons. The first is that their approach is not based on complexity considerations, that is, the goal is not a worst-case optimization, but

rather to achieve a speed-up for the majority of situations. The second reason is that these algorithms really do not reduce the query time for the worst case, that is, the query time is $O(n)$. The achievement is that average-case analyzes show that they are better than that.

The rest of this article deals with the problem how a quantitative evaluation and comparison of the heuristic techniques can be carried out.

2 Theoretical Framework

The *average-case analysis* of an algorithm means that a probability distribution of the input objects is assumed and then usually the expected value of the relevant complexity measure (time or storage) is calculated.

2.1 Configuration Space and Expected Complexity

Let us assume that the algorithm processes the objects $o_1, \ldots, o_n, \ldots$ of the same type, that is $o_1, \ldots, o_n, \ldots \in \mathbf{O}$, where $\mathbf{O}$ is referred to as the *object space*. Even if the input objects are heterogeneous, this "uniform" object space can be constructed to be the space of all the possible object parameters (such as in the case of a ray tracer accepting a fixed domain of objects). Any given configuration of n input objects can then be described by an n-tuple: $(o_1, \ldots, o_n) \in \mathbf{O} \times \ldots \times \mathbf{O} = \mathbf{O}^n$, where $\mathbf{O}^n$ is referred to as the *configuration space*.

Let $t_n(o_1, \ldots, o_n)$ denote any complexity measure (time or storage) of the algorithm with respect to the objects $o_1, \ldots, o_n$. Its expected value, denoted by $\mathrm{E}[t(n)]$, taken over all the possible configurations is then calculated by the following integral:

$$\mathrm{E}[t(n)] = \int_{o_1 \in \mathbf{O}} \cdots \int_{o_n \in \mathbf{O}} t_n(o_1, \ldots, o_n)\, f_n(o_1, \ldots, o_n)\, do_1 \cdots do_n, \qquad (1)$$

where $f_n(o_1, \ldots, o_n)$ is the assumed joint probability density function. There are two crucial points: the first is the problem of what probability density should be assumed, the second is the calculation of the integral.

2.2 Attacking Ray Tracing

Let us have a given ray tracing acceleration technique (this will be the regular spatial grid in the next section), the problem is to perform an average-case analysis of its query time (other complexity measures such as preprocessing time and storage can be analyzed in a similar way). What assumptions should be made? In a demonstrative article such like this one, the majority of assumptions should aim at the goal that the calculations can be carried out at all. Let us see these simplifying assumptions in the following paragraphs.

2.2.1 Image Sphere and Average Rays

The query time, t_n, depends not only on the configuration of the input objects but also on the spatial location of the ray origin and the direction of the ray: $t_n = t_n(p, \delta, o_1, \ldots, o_n)$, where $p \in \mathbf{E}^3$ is the ray origin and $\delta \in \Delta^2$ is the ray direction (Δ^2 denotes the direction space that is the surface of the origin centered unit sphere).

Considering only the *pixel rays* first, that is those rays which originate from the eye position, the corresponding average query time can be calculated as:

$$\overline{\mathrm{E}}\,[t(n)] = \frac{1}{R^2}\sum_{k=1}^{R^2}\int_{o_1\in\mathbf{O}}\cdots\int_{o_n\in\mathbf{O}} t_n(0,\delta_k,o_1,\ldots,o_n)\,f_n(o_1,\ldots,o_n)\,do_1\cdots do_n,$$

where the eye position is assumed to be in the origin and δ_k denotes the direction of the kth ray, and R^2 is the resolution of the image.

For the sake of mathematical treatability, assume that the image is created on the surface of an eye centered sphere instead of a planar image rectangle. Increasing the resolution of the image beyond all limits ($R^2 \to \infty$) with keeping the area of the pixels equal yields

$$\overline{\mathrm{E}}\,[t(n)] = \frac{1}{4\pi}\int_{\delta\in\Delta^2}\int_{o_1\in\mathbf{O}}\cdots\int_{o_n\in\mathbf{O}} t_n(0,\delta,o_1,\ldots,o_n)\,f_n(o_1,\ldots,o_n)\,do_1\cdots do_n\,d\delta. \qquad (2)$$

A crucial question is how the above argument can be extended to handle *child rays* (reflected and refracted rays), but the answer to this question is definitely beyond the scope of this paper. Thus we will restrict the analysis to pixel rays.

2.2.2 Poisson Spheres

Without having any preliminary knowledge about the distribution of the objects, the most straightforward is to assume *uniform distribution* (which also implies that the objects are independently distributed). Extending the analysis to other distributions is possible but would probably cause more difficulties.

The object scene is assumed to consist of spheres. Extending the analysis to other object types is possible again at the expense of more mathematical difficulties. First the simplest case, when the spheres have equal radii, will be examined in the next section, and then the analysis will be extended to spheres of various radii.

Assuming that the sphere centers are distributed in the interior of a finite volume would cause extra difficulties when evaluating the integral (2) of the expected query time because the integration would have to be "stopped" at the boundary surface of the volume. That is, since geometric probabilities are usually calculated from associated volume extents, these associated volumes whould have to be "clipped" against the boundary of the containing volume during the integration.

In order to avoid this problem, the sphere centers are assumed to be produced by a *homogeneous spatial Poisson point process.* Such a process can be obtained from uniform distribution in the following way. Assume that n points, $p_1,\ldots,p_n$ are uniformly distributed in the interior of a set $X \subset \mathbf{E}^3$, that is, the probability of the event that a given point p_k is contained by a set $A \subseteq X$ is $\Pr\{p_k \in A\} = |A|/|X|$ $(1 \le k \le n)$, where $|\cdot|$ denotes volume. Allowing then both $n, |X| \to \infty$ in such a way that $n/|X| \to \rho$, the limiting stochastic point process is called the homogeneous Poisson point process of *intensity* ρ [San76], [Mil70]. The probability of the event that a given set $A \subset \mathbf{E}^3$ contains exactly k *particles* is:

$$p(k) = \frac{(\rho\,|A|)^k}{k!}\,e^{-\rho|A|} \quad (k \ge 0). \qquad (3)$$

Let a sphere be centered around each particle of the point process. These spheres are called *Poisson spheres.* The radii of the spheres are characterized by further probability density

functions. The problem is now to give the expected query time of a given acceleration technique if the object scene is a set of Poisson spheres. The calculated expected time, $\overline{\mathrm{E}}[t]$, will reflect the limiting behavior of the query time with respect to a "very large number" of spheres: $\overline{\mathrm{E}}[t] = \lim_{n\to\infty} \overline{\mathrm{E}}[t(n)]$.

3 Analysis of the Regular Grid

This acceleration method falls into the category of *object space subdivision* methods [AK89]. The object space is partitioned into congruent cubes called *cells*, and a list containing object references is associated with each cell. The list associated with a given cell contains a reference to a given object if they have non-empty intersection. These lists are built in a *preprocessing phase*. If a ray is to be tested against the objects later in the *tracing phase*, first its origin is located in the cell structure and then the cells pierced by the ray are visited one after each other until an intersection with any of the objects on the associated lists is found. The location of the ray origin in the regular structure is straightforward, and visiting the cells can also be efficiently implemented by using a 3-dimensional DDA line generator [FTK86].

The query time t consists of two main components: the number of objects tested, N_I, and the number of steps to a new cell, N_S. Then the query time is $t = N_\mathrm{I} T_\mathrm{I} + N_\mathrm{S} T_\mathrm{S}$, where T_I and T_S are the times required by a ray-object test and a cell step, respectively (the cost of locating the ray origin is omitted). Only the number of ray-object intersection tests will be examined here, a more extensive analysis of this and other techniques can be found in one of the authors' dissertation [Már95].

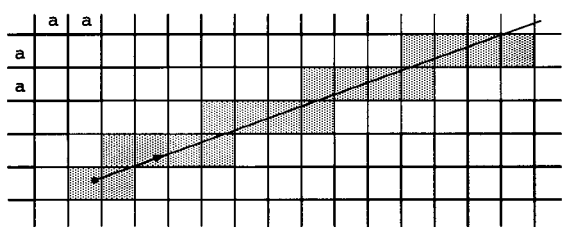

Figure 1: Embedding of the ray

A ray-object test has to be performed obviously only on those Poisson spheres which are "close enough" to the ray; more precisely: which intersect one of the cells visited by the ray. These cells are shaded in figure 1. The set of these cells will be referred to as the *embedding of the ray* further on. Let us first see the case of spheres having the same radius r.

3.1 Distribution of the Intersection Parameter

Let t^* denote the distance of the *first* intersection point along the ray from its origin. The probability density function of t^*, denoted by $f_{t^*}(\tau)$, will be necessary later. Let us first determine the probability distribution function of t^*: $F_{t^*}(\tau) = \Pr\{t^* < \tau\}$; and then the required density function can be obtained by differentiation. The event that the first intersection point is further than a given distance τ is equivalent with the event that no sphere intersects the first segment of length τ of the ray. The territory where no sphere center must fall in this case is painted by light gray in figure 2. That is, this volume must

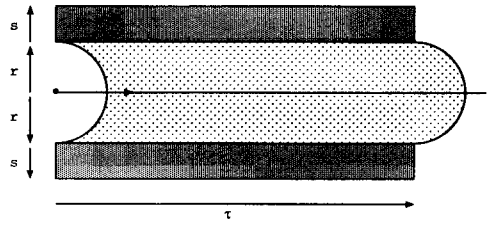

Figure 2: Approximation of event probabilities

be *empty*. Note that "first intersected" object here means the first one not containing the origin of the ray. This simplification does not significantly influence the results but significantly simplifies the expressions obtained. The volume of the painted territory is $r^2\pi\tau$, hence the probability of the event $t^* \geq \tau$ is:

$$\Pr\{t^* \geq \tau\} = e^{-\rho r^2 \pi \tau}, \tag{4}$$

according to equation 3. The distribution function of t^* is then $F_{t^*}(\tau) = 1 - e^{-\rho r^2 \pi \tau}$, from which the probability density function of t^* is obtained by differentiation:

$$f_{t^*}(\tau) = \rho r^2 \pi \, e^{-\rho r^2 \pi \tau}. \tag{5}$$

The spheres intersecting the embedding of the ray are tested one after each other until an intersection is found. The center of the spheres intersecting the embedding of the ray are contained by a volume containing the embedding, too. Let this volume be called the *range of the ray*.

3.2 Approximation of the Embedding by a Cylinder

Assume temporarily that the embedding of the ray is a ray centered cylinder of radius s. That is, the chain of embedding cells is approximated by a cylinder. Accept furthermore that the range of the ray is then a cylinder of radius $s + r$ with the same axis, as shown by figure 2. The error caused by these simplifications will be estimated later. The volume content of the range of a ray segment of length τ is now:

$$V_{\mathrm{R}}(\tau) = (s + r)^2 \pi \tau. \tag{6}$$

Figure 2 also shows that under the *condition* $t^* = \tau$ the number of tested spheres is one minus the number of sphere centers falling into the territory painted by dark gray, since the center of the last (successfully) tested sphere does not fall there. The volume of this territory is:

$$V(\tau) = \left((s + r)^2 - r^2\right) \pi \tau.$$

Under the condition $t^* = \tau$, the conditional probability of the event that exactly k number of sphere centers fall into this territory, that is $N_{\mathrm{I}} = k + 1$ is:

$$\Pr\{N_{\mathrm{I}} = k + 1 \mid t^* = \tau\} = \frac{(\rho V(\tau))^k}{k!} e^{-\rho V(\tau)}. \tag{7}$$

The probability of the event $N_{\mathrm{I}} = k + 1$ can be calculated from this by

$$\Pr\{N_{\mathrm{I}} = k + 1\} = \int_0^\infty \Pr\{N_{\mathrm{I}} = k + 1 \mid t^* = \tau\} \, f_{t^*}(\tau) d\tau. \tag{8}$$

Substituting the expression (5) of the probability density function $f_{t^*}(\tau)$ and performing the integration yields

$$\Pr\{N_{\mathrm{I}} = k+1\} = \left(1 - \left(\frac{r}{s+r}\right)^2\right)^k \left(\frac{r}{s+r}\right)^2.$$

This is no else but the well known formula of the *geometric distribution* [KK75] with the event probability $p = (r/(s+r))^2$, hence the expected value is:

$$\mathrm{E}[N_{\mathrm{I}}] = \left(\frac{s+r}{r}\right)^2, \tag{9}$$

at least under the assumption that the embedding of the ray is well approximated by a cylinder of radius s. The question is now how the value of s should be chosen.

3.3 The Approximation Error

Note that the following errors were made when approximating the range of the ray by a cylinder of radius $s+r$ and using its volume (6) in the previous argument:

1. the actual shape of the embedding is not a cylinder but rather a chain of cells joined by their faces to each other,
2. even if its actual shape were a cylinder, the shape of the range of the ray would be badly determined, since the "terminating hemispheres" are omitted, that is, a smaller volume is used,
3. it was implicitly assumed during the argument that the spheres are taken exactly in the order of their distance along the ray during the intersection calculation; this is not completely true, however, since the object references on the list associated with a given cell can be in any order.

Let us now overview the effect of these errors.

Mistakes 1 and 2 will be handled together: first an upper and then a lower estimation will be given for the value of s in expression 9.

Note that any point of a given embedding cell is not further from the ray than the length of the cell diagonal. This implies that the embedding cells are contained by a ray centered cylinder of radius $a\sqrt{3}$, where a is the width of the cells, that is: $s \leq a\sqrt{3}$.

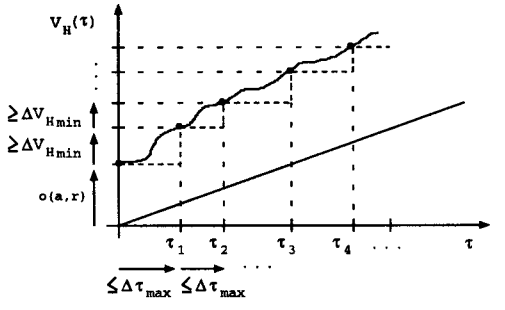

Figure 3: Lower estimation of the ray range

In order to get a lower estimation, not the radius of the embedding but the volume of the range itself will be considered. Even if the spheres are tested in their order along the

ray (error 3), the volume of the embedding is a non-linear (although monoton increasing) function of τ, as shown by figure 3 (curved line). The starting value of the function is not zero, since all the spheres intersecting the first cell (containing the ray origin) must be tested, independently from how long the ray segment contained by this cell is. This starting value is equal to the volume of the r-offset of a cube of width a, denoted by $o(a,r)$. Its volume is: $|o(a,r)| = a^3 + 6a^2r + 3ar^2\pi + 4r^3\pi/3$. In the worst case, the function increases according to the stair line in the figure. (Note that this is actually the case since stepping from one cell into the next one increases the volume in this way. Note also that both the τ and the V_R values are not equally spaced.) On the one hand, the difference of the τ values between two stair steps, $\Delta\tau$, can not exceed $a\sqrt{3}$ (the ray is now incident with the diagonal of the cell). On the other hand, the difference of the V_R values between two steps, ΔV_R, is at least a^3 (counting only those sphere centers which fall into the cell itself). Hence a lower estimation of the function $V_R(\tau)$ is the following:

$$V_R(\tau) \geq \frac{\min \Delta V_R}{\max \Delta \tau}\tau \geq \frac{a^2}{\sqrt{3}}\tau.$$

Taking the assumed expression (6) of $V_R(\tau)$ into consideration yields $s \geq a/\sqrt{\sqrt{3}\,\pi} - r$.

Finally, due to error 3, it was not considered that even if an intersection is found during ray-object tests, the rest of spheres on the list associated with the actual cell should also be tested, since the spheres on the list do not appear in their order along the ray. The effect of this error can be compensated by adding a constant to the expected value (9). This constant is at least zero and at most the expected number of spheres intersecting a cell (its value is $\rho|o(a,r)|$). Our results are summarized in the following statement:

result 3.1 *Assume that the object scene consists of Poisson spheres of intensity ρ and radius r. Using a regular grid of cubes of width a, the expected number of ray-object tests with respect to an* arbitrary *ray is characterized by the following expression:*

$$\mathrm{E}[N_I] = \left(\frac{s+r}{r}\right)^2 + c, \qquad \textit{where} \quad a\sqrt{1/\sqrt{3\pi}} - r \leq s \leq a\sqrt{3} \quad \textit{and} \quad 0 \leq c \leq \rho|o(a,r)|.$$

Note that, according to the statement (and accepting that $\mathrm{E}[N_S]$ is also a constant), the *expected* query time turns out to be *superoptimal*: $\mathrm{E}[t(n)] = O(1)$ although the best achievable *worst-case* complexity is $t(n) = O(\log n)$.

3.4 Extending to Spheres of Various Radii

The most general extension of the theory to spheres of non-equal radii would be based on the assumption of different distribution of each individual sphere radius. Since it seems too complicated mathematically, a simplified extension will be discussed here.

Assume that each sphere of the scene falls into one of n classes. The spheres falling into the same class, say class i ($1 \leq i \leq n$), have the *same radius*, R_i, the probability density function of which is denoted by $f_{R_i}(r)$. Let the ratio of the number of spheres in class i and the total number of spheres be α_i, satisfying the obvious constraint $\alpha_1 + \ldots + \alpha_n = 1$.

Based on this setup, first the *conditional expectation* $\mathrm{E}[N_I \mid R_1 = r_1, \ldots, R_n = r_n]$ can be calculated and then the final result can be obtained by

$$\mathrm{E}[N_I] = \int_0^\infty \cdots \int_0^\infty \mathrm{E}[N_I \mid R_1 = r_1, \ldots, R_n = r_n]\, f_{R_1}(r_1) \cdots f_{R_n}(r_n)\, dr_1 \cdots dr_n$$

The calculation of the conditional expectation goes very similarly to that of the expectation with equal radii. It is only sketched here, putting the emphasis onto the differences.

Note first that the centres of the spheres falling into class i form a Poisson point process of intensity $\alpha_i\rho$, and the distribution of the spheres falling into different classes is statistically independent. Thus, equation 4 becomes

$$\Pr\{t^* \geq \tau\} = e^{-\alpha_1\rho r_1^2\pi\tau} \ldots e^{-\alpha_n\rho r_n^2\pi\tau},$$

and hence the probability density function of the intersection parameter t^* is now

$$f_{t^*}(\tau) = \rho(\alpha_1 r_1^2 + \ldots + \alpha_n r_n^2)\pi\, e^{-\rho(\alpha_1 r_1^2 + \ldots + \alpha_n r_n^2)\pi\tau}.$$

The event that exactly k number of spheres are tested before the first intersection found, that is the event $N_{\mathrm{I}} = k+1$, is the sum of all the possible events where k_1 spheres of class 1, k_2 spheres of class 2, ... , k_n spheres of class n are refused before the first successful test, the summation being taken over all the n-tuples $k_1, \ldots, k_n$ with $k_1 + \ldots + k_n = k$. Equation 7 hence becomes

$$\Pr\{N_{\mathrm{I}} = k+1 \mid t^* = \tau\} = \frac{(\rho(\alpha_1 V_1(\tau) + \ldots + \alpha_n V_n(\tau)))^k}{k!} e^{-\rho(\alpha_1 V_1(\tau) + \ldots + \alpha_n V_n(\tau))},$$

where

$$V_i(\tau) = \left((s_i + r_i)^2 - r_i^2\right)\pi\tau \quad (1 \leq i \leq n).$$

Performing finally the integration (8) yields

$$\mathrm{E}[N_{\mathrm{I}} \mid R_1 = r_1, \ldots, R_n = r_n] = \frac{\alpha_1(s_1 + r_1)^2 + \ldots + \alpha_n(s_n + r_n)^2}{\alpha_1 r_1^2 + \ldots + \alpha_n r_n^2}.$$

The approximation radii $s_1, \ldots, s_n$ can be estimated in the same way as earlier. The result is formulated as follows:

result 3.2 *Assume that the object scene consists of Poisson spheres of intensity ρ and with various radii. Each sphere falls into one of n equivalence classes, the proportion of the number of spheres falling into the different classes being characterized by the positive constants $\alpha_1, \ldots, \alpha_n$ ($\alpha_1 + \ldots + \alpha_n = 1$). The spheres in class i have the* same radius, *R_i, the probability density function of which is $f_{R_i}(r_i)$ ($1 \leq i \leq n$). Then, using a regular grid of cubes of width a, the expected number of ray-object tests with respect to an* arbitrary *ray is characterized by the following expression:*

$$\mathrm{E}[N_{\mathrm{I}}] = \int_0^\infty \cdots \int_0^\infty \left\{ \frac{\alpha_1(s_1 + r_1)^2 + \ldots + \alpha_n(s_n + r_n)^2}{\alpha_1 r_1^2 + \ldots + \alpha_n r_n^2} + c \right\} f_{R_1}(r_1) \cdots f_{R_n}(r_n)\, dr_1 \cdots dr_n,$$

where

$$a\sqrt{1/\sqrt{3}\pi} - r_i \leq s_i \leq a\sqrt{3} \quad \textit{and} \quad 0 \leq c \leq \rho(\alpha_1 |o(a, r_1)| + \ldots + \alpha_n |o(a, r_n)|).$$

Note that this analysis also shows that the expected query time of the method is super-optimal.

4 Interpretation of the Results

Considering the results of the previous section, one might ask how well the average-case complexity characterizes the algorithm. Thinking about practical applications, a possible question is: given n number of objects, what can we state about the probability $p(n, c)$ of the event that the actual query time is within a constant factor c to the value calculated in the previous section? Although this question can not be answered without information about the variance of the query time, an interesting statement, similar to the weak law of large numbers, will be proven here:

proposition 4.1 *An algorithm is given which processes the objects $o_1, \ldots, o_n, \ldots$ of the same type. Let $t_n(o_1, \ldots, o_n)$ be a complexity measure (time or storage) of the algorithm, the expectation of which, $\mathrm{E}[t(n)]$, has been calculated according to expression 1. If the complexity classification of $\mathrm{E}[t(n)]$ gives*

$$\mathrm{E}[t(n)] = O(G(n)),$$

then it implies that for any function $H(n) \neq G(n)$ and positive constant $c > 0$:

$$\lim_{n\to\infty} \Pr\{t_n(o_1, \ldots, o_n) \geq c\,H(n)\} = 0,$$

provided that the objects are selected according to the same density function $f_n(o_1, \ldots, o_n)$.

Note that, according to the statement, the *measured* complexity of the algorithm with respect to *any* random sequence of input objects *tends* to fall into the class of the expected complexity even if the worst-case complexity falls into a higher class (as in the case of the examined acceleration technique).

The above statement will now be proven indirectly. Assuming that the statement is false, we expect that it leads to a contradiction, namely to $\mathrm{E}[t(n)] \neq O(G(n))$. According to the definition of the limit, the statement itself can be written in the following form:

$$\forall c > 0,\ H(n) \neq O(G(n)),\ \varepsilon > 0\ \exists N > 0\ \forall n > N\ \left[\Pr\{t_n(o_1, \ldots, o_n) \geq c\,H(n)\} < \varepsilon\right].$$

If the statement is false, then it is equivalent with the following statement:

$$\exists c > 0,\ H(n) \neq O(G(n)),\ \varepsilon > 0\ \forall N > 0\ \exists n > N\ \left[\Pr\{t_n(o_1, \ldots, o_n) \geq c\,H(n)\} \geq \varepsilon\right]. \tag{10}$$

Let $O'_n(\varepsilon) \subseteq \mathbf{O} \times \cdots \times \mathbf{O} = \mathbf{O}^n$ be that part of the configuration space for which $t_n(o_1, \ldots, o_n) \geq c\,H(n)$ holds:

$$O'_n(\varepsilon) = \{(o_1, \ldots, o_n) \in \mathbf{O}^n \mid t_n(o_1, \ldots, o_n) \geq c\,H(n)\}. \tag{11}$$

Consider now the equation (1) of the expected measure. A lower estimation of its value is naturally

$$\mathrm{E}[t(n)] \geq \int\limits_{O'_n(\varepsilon)} t_n(o_1, \ldots, o_n)\, f_n(o_1, \ldots, o_n)\, do_1 \cdots do_n.$$

Combining it with the definition of $O'_n(\varepsilon)$ (11) yields

$$\mathrm{E}[t(n)] \geq \int\limits_{O'_n(\varepsilon)} c\,H(n)\, f_n(o_1, \ldots, o_n)\, do_1 \cdots do_n = c\,H(n) \Pr\{t_n(o_1, \ldots, o_n) \geq c\,H(n)\}.$$

Then, according to the assumption (10):

$$\exists c > 0,\ H(n) \neq O(G(n)),\ \varepsilon > 0\ \forall N > 0\ \exists n > N\ [\,\mathrm{E}[t(n)] \geq \varepsilon\, c\, H(n)\,]\,. \qquad (12)$$

Now let $c, H(n), \varepsilon$ be any such triple, and let $n_1 < n_2 < \ldots$ be any sequence for which

$$\mathrm{E}[t(n_k)] \geq \varepsilon\, c\, H(n_k) \quad (k = 1, 2, \ldots)$$

holds. The above statement (12) implies the existence of at least one such *infinite* sequence. Let us, finally, define the function $h(n)$ in the following way:

$$h(n) = \begin{cases} \varepsilon\, c\, H(n_k), & \text{if } n = n_k \text{ for some } k > 0; \\ 0, & \text{otherwise} \end{cases}$$

Then, on the one hand $h(n) \neq O(G(n))$ since $H(n) \neq O(G(n))$, and on the other hand $\mathrm{E}[t(n)] \geq h(n)$. These inequalities imply $\mathrm{E}[t(n)] \neq O(G(n))$, which is a contradiction (q.e.d.).

The question whether the statement of the proposition can be extended (or in what circumstances it can be extended) to $\Pr\{t_n(o_1, \ldots, o_n) = O(G(n))\} = 1$, similarly to the strong law of large numbers, remains an open question.

References

[AK89] James Arvo and David Kirk. A survey of ray tracing acceleration techniques. In Andrew S. Glassner, editor, *An Introduction to Ray Tracing*, pages 201–262. Academic Press, London, 1989.

[dB92] Mark de Berg. *Efficient Algorithms for Ray Shooting and Hidden Surface Removal.* PhD thesis, Rijksuniversiteit te Utrecht, Nederlands, 1992.

[FTK86] Akira Fujimoto, Tanaka Takayuki, and Iwata Kansei. Arts: Accelerated ray-tracing system. *IEEE Computer Graphics and Applications*, 6(4):16–26, 1986.

[KK75] Granino A. Korn and Theresa M. Korn. *Mathematical Handbook for Scientists and Engineers.* McGraw-Hill, 1975.

[Már95] Gábor Márton. *Stochastic Analysis of Ray Tracing Algorithms.* PhD thesis, Department of Process Control, Budapest University of Technology, Budapest, Hungary, 1995. to appear, in Hungarian.

[Mil70] Roger E. Miles. On the homogeneous planar poisson point process. *Mathematical Biosciences*, (6):85–127, 1970.

[San76] Luis A. Santaló. *Integral Geometry and Geometric Probability.* Addison-Wesley, Reading, Massachusetts, 1976.

[SML88] Alfred Schmitt, Heinrich Müller, and Wolfgang Leister. Ray tracing algorithms — theory and practice. In R. A. Earnshow, editor, *Theoretical Foundations of Computer Graphics and CAD*, pages 997–1030. Springer-Verlag, Berlin Heidelberg, 1988. NATO ASI Series, Vol. F40.

[Whi80] Turner Whitted. An improved illumination model for shaded display. *Communications of the ACM*, 23(6):343–349, 1980.