Answering to 'What is the criterion?' questions using integral geometry tools.

A.Iones[*], S. Zhukov[**], A. Krupkin[**], M. Sbert[***]
[*]Advanced Productions Inc., Franklin Square, NY
[**]CREAT Studio (St.Petersburg, Russia)
Applied Mathematics Department
St.-Petersburg State Technical University (Russia),
[***] Institute of Computer Science and Applications
University of Girona (Spain)

iones@apigames.com
{zh,ak}@creatstudio.com www.creatstudio.com
mateu@ima.udg.es

Abstract

We present here in a unified way some pioneering work, together with new results, in computer graphics on the use of integral geometry tools to select efficient data structures to handle 3D scenes. The *integral* (or *average-case*) analysis enables to create 'good' data structures that significantly speed up the algorithms. This results in *provable, theoretically sound criteria* for appropriate data structures. The examples presented here are the selection of optimal oriented bounding boxes for ray shooting, frustum culling and collision detection, hierarchical bounding volume selection, polygon triangulation for faster rendering and global line distribution analysis for efficient Monte Carlo radiosity.

1. Introduction

Solving various problems in computer graphics one frequently needs to select a certain criterion to build necessary data structure. Examples of this are the criteria proposed to build 'well-fitting' oriented bounding boxes around 3D models for fast ray shooting or collision detection, or the selection of uniformly distributed line sets for Monte Carlo radiosity. Proper selection of such criteria based on the appropriate *integral* (or *average-case*) analysis of the structures enables to create 'good' data structures that significantly speed up the algorithms.

Quite often various *heuristic criteria* are used in such cases. In this paper we demonstrate that the use of Integral Geometry tools greatly helps to resolve the criterion selection problems. It allows developing an appropriate mathematical model that makes possible to statistically analyze the underlying data structures. This results in *provable, theoretically sound criteria* for appropriate data structures. Therefore, in contrast to heuristic techniques, one can construct data structures that are *provably good*. Integral geometry tools come in handy when it's necessary to analyze the performance of some data structure *on average* – this is unlike the computational geometry approach that usually concentrates on *worst-case* analysis. Nonetheless, once the criterion is established, computational geometry techniques are used to construct data structures that were proved to be good.

We should note that usually the *soundness* of the criteria (or, in other words, the soundness of the selected mathematical model) is usually valid under some reasonable assumptions such as the uniformity of ray distribution in space. Still, it seems more reasonable to build data structures basing on the criteria obtained within a strictly defined model than basing on heuristics.

To illustrate this approach we discuss, after presenting basic integral geometry concepts in Section 2, some typical problems that arise in quite different areas of computer graphics, namely:

- the selection of optimal oriented bounding boxes for ray shooting, frustum culling and collision detection, in Section 3;
- hierarchical bounding volume selection, in Section 4;
- polygon triangulation for faster rendering, Section 5;
- global line distribution analysis for efficient Monte Carlo radiosity in Section 6.

Finally we present in Section 7 our conclusions.

2. Some basics from integral geometry

We start this section with the discussion of the integral geometry results in 2D case. Their counterparts for 3D case are given in Sect. 2.4. A more in-depth introduction can be found in [CAZA98].

We will denote as ∂K the boundary of a convex body K. The length of the closed curve ∂K, defining the convex body K, is called the *perimeter* of body K. Perimeters will be denoted as L.

2.1 Support functions.

To explore the properties of the convex bodies and their boundaries, integral geometry uses the notion of *support functions*. Let K be a convex body and point $O \in K$ defines the coordinate system origin. Let $p(\varphi)$ be the distance from the origin O to the support line l of the body K, perpendicular to the direction φ (Figure 1).

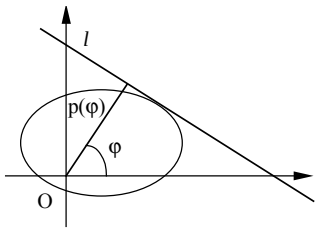

Figure 1. Support function of the convex body

Definition. Function $p(\varphi)$ is called the *support function* of the convex body K, related to the origin O.

In [SAN76] it is shown that the perimeter of the convex body K can be computed as follows:

$$L = \int_0^{2\pi} p(\varphi)d\varphi \,. \qquad (1)$$

Definition. The *breadth* $\Delta(\varphi)$ of the convex body K in the direction φ is the distance between two parallel support lines to K, that are perpendicular to the direction φ, such that the body K is in-between them (Figure 2).

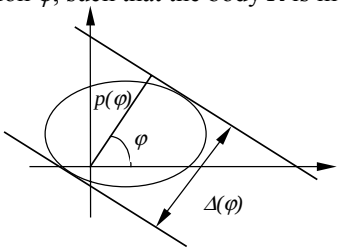

Figure 2. The breadth of the convex body K in the direction φ.

If $p(\varphi)$ is a support function of K, then $\Delta(\varphi) = p(\varphi) + p(\varphi+\pi)$. Hence, from (1) we deduce that for any convex body the following relation holds:

$$L = \int_0^{\pi} \Delta(\varphi)d\varphi \qquad (2)$$

Definition. *Diameter* D of body K is the greatest distance between any two points in K. It can be defined as well as the largest breadth of K: $D = \max_{\varphi}(\Delta(\varphi))$.

2.2 Densities of sets of bodies on the plane.

Let $C = (O, x, y)$ be an orthogonal coordinate system on the plane. Let K be a rigid body. From the integral geometry and geometric probability theory it is known that to determine the measure of the set of bodies congruent to the given body K (or the set of positions of body K), one should select the an arbitrary coordinate system $C' = (O', x', y')$ fixed in K and compute the integral of the differential form $dK = da \wedge db \wedge d\varphi$ [1] [SAN76][SOL78]. The integral is taken over the set of points (a, b) that define the origin of C' in the fixed coordinate system C and the directions φ defining the angle between the axes $O'x'$ and Ox (Figure 3). The differential form dK is called the *density* of the set of congruent bodies. Due to the invariance requirement under the group of rigid motions this density is unique up to constant factor [SAN76, chapter 6].

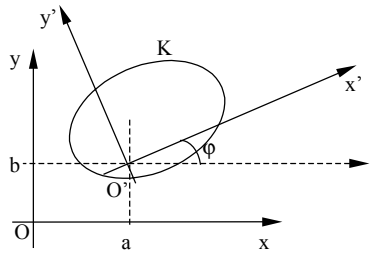

Figure 3.

2.3 Some important measures.

One of the problems that can be resolved using integral geometry tools is the problem of finding the measures of the sets of bodies. For example, the measure of the set of lines G intersecting a given convex body K is equal to the perimeter L of body K:

$$m(G;\, G \cap K \neq \varnothing) = L. \qquad (3)$$

[1] In this paper we use the rules of the exterior calculus [CART67]. Informally speaking, exterior product $dx \wedge dy$ is the area of the oriented differential element in the Cartesian coordinate system.

In terms of geometric probability, this can be interpreted as follows. Let K_1, K_2 be two convex bodies with the perimeters L_1, L_2; $K_1 \subset K_2$. For a given random line G, the probability p that line G intersects K_1 if it intersects K_2 is $p = L_1/L_2$. From here it is clear that the optimality criterion for line shooting is the minimization of the perimeter of bounding volume. Line shooting problem is similar to ray shooting, but there is a significant difference. Indeed, the measure of lines intersecting a body is finite, while the measure of rays is infinite. Therefore in the following sections we will use the asymptotic estimations to obtain the desired criteria.

2.4 Some results from integral geometry in 3D.

Equations (2) and (3) can be generalized in 3D (and in higher-dimension spaces) [BONN34, p. 187], [SAN76]. Let K be a convex body and F be its surface area. The measure of lines G intersecting K is computed as:

$$m(G;\ G \cap K \neq \varnothing) = F.$$

The generalization of equation (2) is the famous Cauchy's formula:

$$F = \frac{1}{\pi} \int_{S_2} F_n \, dn \qquad (4)$$

where S_2 – a unit sphere centered in some point within K, n – a normal to the unit sphere, F_n – the area of the projection of body K onto the plane with normal n.

3. Bounding volume optimality criteria for ray shooting and frustum culling

3.1 Bounding volumes

Bounding volumes are widely used in many computer graphics applications to speed up various intersection tests such as ray shooting, frustum culling and collision detection. Firstly non-hierarchical bounding volumes (bounding spheres) were suggested in [CLARK76] and were used in [WHIT80] to speed up ray tracing. Then, hierarchical bounding volumes were introduced [RUB80][WEGH84]. A (non-hierarchical) bounding volume encloses a given object and permits a simpler intersection check than the object itself. Similarly, hierarchical bounding volumes enclose the geometry of the associated object as well as all the object descendants in the hierarchy.

Bounding volumes are convex bodies usually having some simple geometry, such as spheres or boxes. Only if a ray (or viewing frustum, or other test body) intersects the bounding volume should the object itself (and object descendants in case of hierarchical bounding volumes) be checked for intersection. Despite a small computation overhead due to the additional tests with bounding volumes, the significantly decreased number of intersection tests with the objects results in a significant net gain in efficiency.

A number of bounding volume types have been proposed since their first exposition to the computer graphics community. They include spheres, axis-aligned and arbitrary oriented bounding boxes (AABB and OBB, respectively), 'slabs' (polyhedra formed by intersection of pairs of parallel planes) and so on.

An important issue is the selection of appropriate bounding volumes that are the 'best' for the given task or application. We discuss in Section 3 non-hierarchical bounding volumes and then proceed in Section 4 with the hierarchical case.

3.2 Bounding volume selection criteria

The following basic function is used to evaluate the performance of an algorithm working on bounding volume and object trees [WEGH84][ARVO][GOTT96]:

$$T = N_v * C_v + N_p * C_p,$$

where T is the total time spent on processing, N_v and N_p are the numbers of bounding volume and object tests, C_v and C_p are the costs of a single bounding volume and object intersection tests. Assuming N_v and C_p are fixed, we need to select a bounding volume that is both inexpensive, making C_v small, and tightly fitting, minimizing N_p. Therefore, one should analyze this trade-off. Simple bounding volumes such as spheres and axis-aligned bounding boxes (AABBs) have low C_v. However, being unable to fit scene objects tightly they cause the algorithm to process more objects.

A number of bounding volume types capable to fit scene objects tightly have been proposed. Kay and Kajiya [KAY86] introduced 'slabs' and showed that intersection tests with them require little computation. Recently, Gottschalk et al. [GOTT96] pointed out that oriented bounding boxes (OBBs) are able to fit scene objects closely and used them to speed up collision detection tests.

Once the bounding volume type is selected, it is important to establish some criteria that will help to construct 'good' bounding volumes for the objects. Clearly, for simple bounding volumes such as spheres or axis-aligned boxes there is no choice – the smallest enclosing sphere or AABB is the best one[2]. More complex volumes, such as 'slabs' or OBBs, have more degrees of freedom, and need to be constructed appropriately to tightly fit the objects. One may think of different criteria for this "tight fit", such as minimizing the volume, surface area or Hausdorff distance of bounding volume. To establish the criteria, it is usually assumed that rays (or other appropriate objects such as viewing frustums) are effectively randomized, i.e. uniformly distributed in space.

Kay and Kajiya [KAY86] and other researchers [WATT92] give a list of desirable properties for any hierarchical bounding volume scheme. They heuristically claim that the volume of each bounding volume should be minimal, as well as the sum of the volumes of all bounding volumes in the hierarchy.

[2] Note that for hierarchical bounding volumes the smallest enclosing sphere is not always the best one – see Section 4.

Arvo and Kirk [ARVO] and Goldsmith and Salmon [GOLD87] investigated these criteria for hierarchical bounding volumes under the assumption that bounding volumes are nested in the hierarchy [ARVO][GOLD87]. It was shown that under the bounding volume nested-ness assumption the minimization of the bounding volume surface area is the optimality criterion for the ray shooting operation that minimizes the number of ray-object intersection tests.

In our opinion, this nested-ness assumption is too restrictive even for the simplest bounding volume types such as spheres or OBBs. This is illustrated on Figure 4. Here, under the nested bounding volume assumption, the smallest enclosing sphere cannot become a bounding volume since it does not belong to the parent bounding volume.

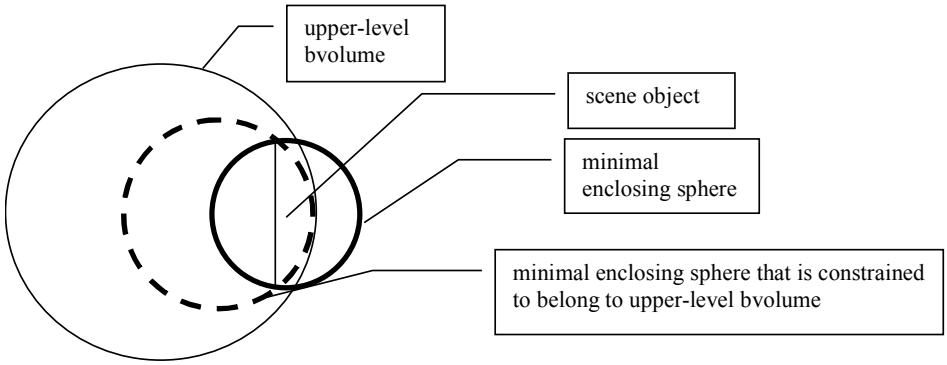

Figure 4. "Hierarchical nested-ness criterion" may yield bounding volumes that are not optimal.

Recently it was proved [ION98a,b] that surface area minimization criterion is valid for non-hierarchical bounding volumes for different operations such as ray shooting or frustum culling. A more complex criterion holds in case of hierarchical bounding volumes (Section 4). The proofs of these criteria were performed with the use of integral geometry tools. The essential ideas of the proofs are briefly summarized in subsequent sections to demonstrate the process of mathematical model selection and criterion obtaining using integral geometry tools.

3.3 Bounding volume optimality criterion for ray shooting from the integral geometry point of view

Let us discuss the bounding volume optimality criterion for ray shooting operation. Let K_0 be an object of the scene and K be its bounding volume. Note that we consider a single body and its bounding volume (without any hierarchy). Tracing a ray, the algorithm firstly tests for its intersection with the bounding volume K of an object K_0 (Figure 5). If the ray does not intersect the bounding volume, then it cannot intersect the enclosed object (ray G_1 on Figure 5). Otherwise, if the ray intersects K, it may or may not intersect K_0 (rays G_2 and G_3 on Figure 5). The latter case is the case of *improper intersection test* with the bounding volume. We assume a bounding volume is optimal if the *probability* of improper intersection for a given ray is minimal among other bounding volumes of this type. Alternatively, we may speak of the minimization of the *measure* of rays for which improper intersection happens: $p = \frac{m_{improper}}{m_{all}}$, where m_{all} and $m_{improper}$ are the measures of appropriate sets of rays.

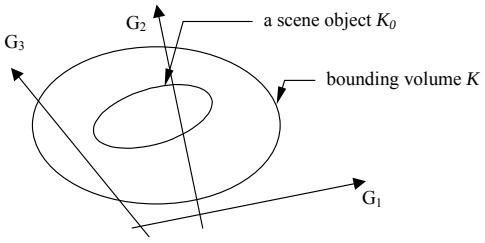

Figure 5. Ray tracing with bounding volumes. Ray G_3 intersects bounding volume K but does not intersect the enclosed body. Hence the result of intersection test with ray G_3 is improper.

The argument above is summarized in the following problem statement:

Problem. *"Bounding volume optimality criterion for ray shooting".*

Given: A scene object.

Build: Within a certain class of bounding volumes (e.g., oriented bounding boxes) find the one for which the measure of rays intersecting it but not intersecting the enclosed object is minimal.

3.4 Ray shooting in 2D

At the beginning, let us discuss the ray shooting problem in 2D. We denote the rays by symbol G. To obtain the criterion we assume that rays are uniformly distributed in space [ARVO]. The density of rays dG can be defined as follows (this density is unique up to a constant factor under the group of rigid motions):

$$dG = dx \wedge dy \wedge d\theta,$$

where *(x, y)* is the ray origin and θ is ray direction (Figure 6).

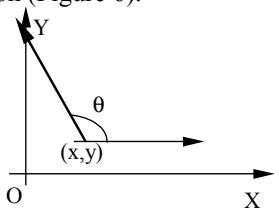

Figure 6. Rays on the plane

Let K_0 be a scene object, K be its bounding volume, $K_0 \subset K$.

The introduced above quality of a bounding volume with respect to ray shooting is quantified by the measure m of rays, intersecting K but not intersecting K_0:

$$m = m(G;\ G \cap K \neq \varnothing,\ G \cap K_0 = \varnothing).$$

The greater the measure m, the less effective the bounding volume is. The measure m can be computed as the integral from the density of rays dG over the appropriate set of rays:

$$m = m(G; G \cap K \neq \varnothing, G \cap K_0 = \varnothing) = \int_{G\cap K\neq\varnothing, G\cap K_0=\varnothing} dG = \int_{G\cap K\neq\varnothing, G\cap K_0=\varnothing} dx \wedge dy \wedge d\theta\ .$$

This integral taken over the whole plane *(x, y)* and directions θ is infinite. To compare the relative asymptotic effectiveness of the bounding volumes, we will firstly compute the measure $m(R)$ within some enclosing circle C_R with radius R and center in point O, $O \in K_0$:

$$m(R) = m(G; G \cap K \neq \varnothing, G \cap K_0 = \varnothing, (x,y) \in C_R) = \int_{G\cap K\neq\varnothing, G\cap K_0=\varnothing, (x,y)\in C_R} dG = \int_{G\cap K\neq\varnothing, G\cap K_0=\varnothing, (x,y)\in C_R} dx \wedge dy \wedge d\theta. \qquad (5)$$

In [ION98b] we proved the following theorem (this theorem was also mentioned implicitly in [GOLD87], but only with a heuristic argument):

Theorem 1. *("Estimation of measure m(R)"). Let K_1, K_2 be two bounding volumes with perimeters L_1 and L_2, respectively. Let $m_1(R)$ and $m_2(R)$ be integrals* (5)*, computed for K_1 and K_2. Then if $L_1 < L_2$, then $\exists\ R_0$ such that $\forall R > R_0$: $m_1(R) < m_2(R)$.*

Proof sketch. To prove the theorem we obtain the bounds for the measure $m(R)$. Firstly we fix the direction of view θ, and then integrate by x and y within the circle C_R. Let $\Delta(\theta)$ and $\Delta_0(\theta)$ be the breadths of the bodies K and K_0 in the direction θ.

Let us then denote as $m(R,\theta)$ the integral (5), taken by (x, y) within the circle C_R with the direction θ fixed:

$$m(R,\theta) = \int_{G(\theta)\cap K\neq\varnothing, G(\theta)\cap K_0=\varnothing, (x,y)\in C_R} dx \wedge dy\ .$$

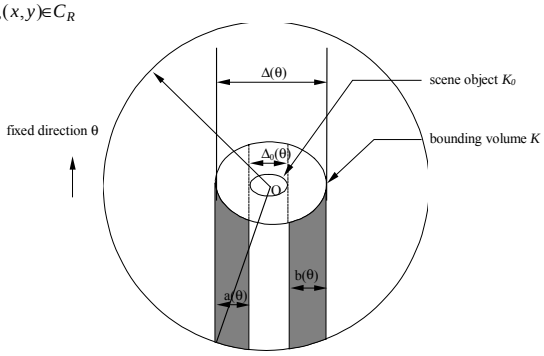

Figure 7. Estimating the measure $m(R,\theta)$

$m(R,\theta)$ is equal to the area of the two stripes grayed on Figure 7. It was shown in [ION98b] that $m(R,\theta) = (\Delta(\theta) - \Delta_0(\theta))*O(R)$. Integrating over θ, we obtain: $m(R) = (L-L_0)*O(R)$, from which the theorem follows. □

From this theorem we deduce that the less the perimeter of a bounding volume, the better its quality with respect to the ray shooting operation. Hence the convex hull of the body is the best bounding volume, and this result is quite intuitive. But it is clear that performing the test for ray shooting with the convex hull of the object is expensive. Usually, most applications use bounding volumes of certain type(s) (say, oriented bounding boxes or slabs), so to construct an optimal bounding volume of this type it is necessary to construct bounding volume having minimal perimeter.

Criterion. The optimal bounding volume of a certain type with respect to ray shooting operation is a bounding

volume with minimal perimeter.

Remark. The assumption that rays are uniformly distributed in space is crucial in the proof of this theorem. To compare the relative asymptotic effectiveness of bounding volumes instead of taking the circle C_R one could take any other enclosing body K_e, $K_0 \subset K \subset K_e$. Then, instead of obtaining the asymptotic bound with $R \to +\infty$ it would be possible to obtain the asymptotic bound with some other type of "expansion" of the body K_e by considering the measure $m(K_e)$ instead of measure $m(R)$. It is important to note that with this approach both the shape of the enclosing body and the way of "expansion" are crucial. Let's discuss this informally. For example, let K_e be a rectangle, and during the expansion one of its sides extends infinitely (so K_e becomes a stripe). Then the optimality criterion would be minimization of the breadth of the bounding volume in the direction parallel to the infinitely extending side of the rectangle K_e – and this is the criterion for the cases where exists a preferred direction of rays.

Note that in most actual computer graphics applications that use ray shooting (e.g., in ray tracing or radiosity) it is needed to trace rays more or less uniformly distributed in the space, that corresponds to the selected model of the uniformly expanding circle.

3.5 Frustum culling from the integral geometry point of view.

The problems of optimal bounding volume selection for frustum culling and ray shooting are similar. For frustum culling, the bounding volume of an object is qualified against the viewing frustum (we assume that perspective projection is used – the case of orthogonal projection is a much easier one). If bounding volume is qualified as being inside or outside the frustum, then the enclosed object is also inside or outside. Otherwise, if the bounding volume properly intersects the frustum, it may still happen that the enclosed object is actually entirely inside or outside the frustum. These two cases are the cases of *improper intersection test* (Figure 8). In these cases the object is passed to the renderer for per-primitive (per-polygon) processing with expensive (but actually unnecessary) per-primitive clipping.

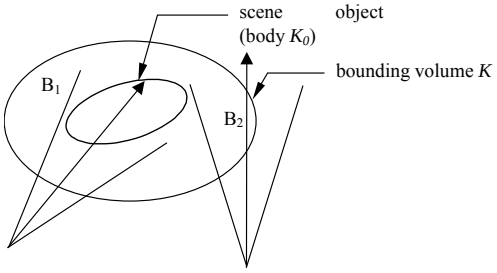

Figure 8. Two cases of improper intersection tests with viewing frustum. Enclosing volume K is qualified by frustums B_1 and B_2 as intersecting, while object K_0 is wholly inside B_1 and wholly outside B_2.

Therefore, an optimal bounding volume is the bounding volume for which the *probability* of improper intersection test with a frustum (or, in other words, the *measure* of frustums for which improper intersection happens) is minimal. This is summarized in the following problem statement:

Problem. *"Bounding volume optimality criterion for frustum culling"*.

Given: A scene object.

Build: Among a certain class of bounding volumes (e.g., oriented bounding boxes) find the one for which the measure of viewing frustums for which improper qualification happens is minimal.

3.6 Bounding volume optimality criterion for frustum culling in 2D.

We now continue with the discussion of 2D case. To obtain the criterion we assume that the object can be viewed from an arbitrary viewpoint, so we assume that viewing frusta are uniformly distributed in space.

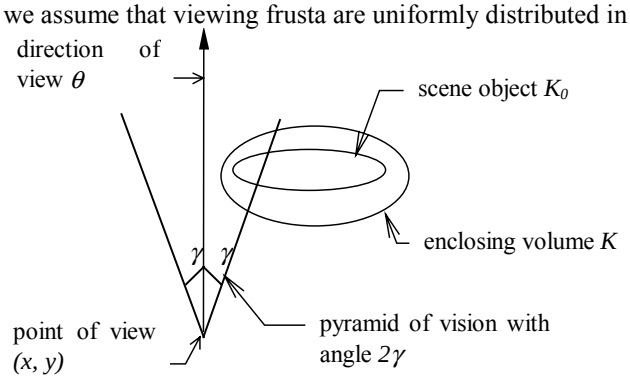

Figure 9. Viewing frustum on the plane.

We denote viewing frustum by symbol B. A viewing frustum with the fixed angle can be specified by its orientation θ and the origin of its apex (x,y) (Figure 9). Then the (unique up to a constant factor under the group of rigid motions) density of viewing frusta dB can be defined as follows:

$$dB = dx \wedge dy \wedge d\theta.$$

Let K_0 be a scene object, K be its bounding volume, $K_0 \subset K$.

The introduced above quality of bounding volumes with respect to frustum culling is quantified by the measure m of frusta for which improper qualification happens, that is, when the enclosing volume intersects the viewing frustum, but the enclosed scene object actually lies wholly inside or outside:

$$m = m_o(B;\ B \cap K \neq \varnothing,\ B \cap K_0 = \varnothing) + m_i(B;\ B \cap K \neq \varnothing,\ K \not\subset B,\ K_0 \subset B).$$

The greater the measure m, the less effective the bounding volume is. The measure m can be computed as the integral of the density of frusta dB over the appropriate sets:

$$m = m_i + m_o = \int_{B\cap K\neq\varnothing, B\cap K_0=\varnothing} dB + \int_{B\cap K\neq\varnothing, K\not\subset B, K_0\subset B} dB = \int_{B\cap K\neq\varnothing, B\cap K_0=\varnothing} dx\wedge dy\wedge d\theta + \int_{B\cap K\neq\varnothing, K\not\subset B, K_0\subset B} dx\wedge dy\wedge d\theta\ .$$

This integral taken over the whole plane (x, y) and all directions θ is infinite. To compare the relative asymptotic effectiveness of the bounding volumes, we will firstly compute the measure $m(R)$ within some enclosing circle C_R with radius R and center in point O, $O \in K_0$:

$$m(R) = m_o(R) + m_i(R) = \int_{B\cap K\neq\varnothing, B\cap K_0=\varnothing, (x,y)\in C_R} dx\wedge dy\wedge d\theta + \int_{B\cap K\neq\varnothing, K\not\subset B, K_0\subset B, (x,y)\in C_R} dx\wedge dy\wedge d\theta\ . \quad (6)$$

Theorem 2. *("Estimation of measure m(R)"). Let K_1, K_2 be two bounding volumes with the perimeters L_1 and L_2, respectively. Let $m_1(R)$ and $m_2(R)$ be the integrals* (6)*, computed for K_1 and K_2. Then if $L_1 < L_2$, then $\exists\, R_0$ such that $\forall R > R_0$: $m_1(R) < m_2(R)$.*

Proof sketch. To prove the theorem we will obtain the estimations for the measure $m(R)$. Similarly to Theorem 1, we firstly fix the direction of view θ, and then integrate by x and y within the circle C_R (i.e. the apex of the frustum belongs to C_R). Let $\Delta(\theta)$ and $\Delta_0(\theta)$ be the breadths of the bodies K and K_0 in the direction θ.

Let us then denote as $m(R,\theta)$ the integral (6), taken by (x, y) within the circle C_R with the direction θ fixed:

$$m(R,\ \theta) = \int_{B(\theta)\cap K\neq\varnothing, B(\theta)\cap K_0=\varnothing, (x,y)\in C_R} dx\wedge dy\wedge d\theta + \int_{B(\theta)\cap K\neq\varnothing, K\not\subset B(\theta), K_0\subset B(\theta), (x,y)\in C_R} dx\wedge dy\wedge d\theta\ .$$

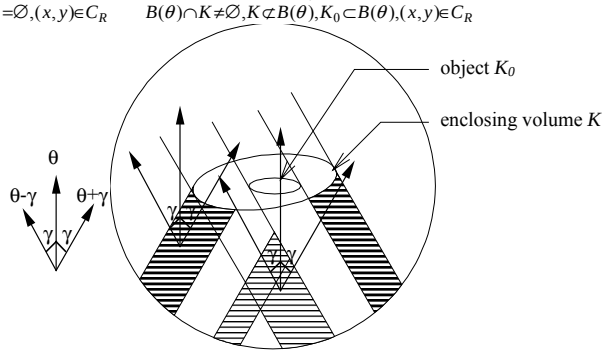

Figure 10. Improper qualifications by viewing frustum oriented in the direction θ happen when the frustum apex lies in the grayed areas. In light-grayed stripes the object is wholly inside and in dark-grayed stripes the object is wholly outside. In both cases the enclosing volume is improperly qualified as intersecting the frustum.

Note that $m(R,\theta)$ is equal to the area of the four grayed stripes (Figure 10). Two internal stripes correspond to measure $m_i(R,\ \theta)$ (light-grayed), while two external stripes correspond to measure $m_o(R,\ \theta)$ (dark grayed). To obtain the bounds for the measure $m(R,\ \theta)$, let us consider the stripes corresponding to the measure $m(R,\ \theta+\pi)$ (Figure 11.a). The stripes corresponding to $m_o(R,\ \theta+\pi)$ and $m_i(R,\ \theta+\pi)$ are located on the extents of the stripes corresponding to $m_i(R,\ \theta)$ and $m_o(R,\ \theta)$ (Figure 11.a).

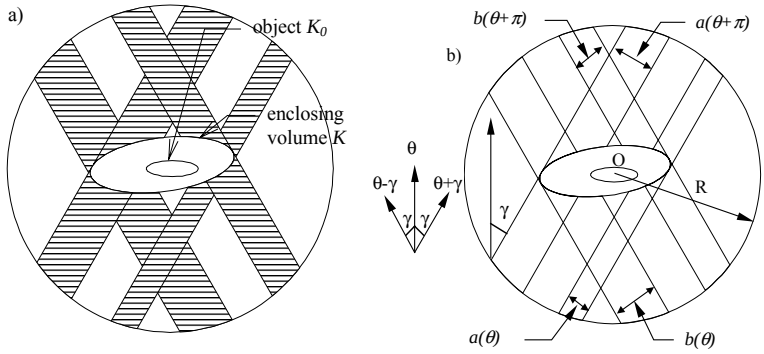

Figure 11. Areas corresponding to $m(R,\ \theta) + m(R,\ \theta+\pi)$.

Let us denote the breadths of the stripes by $a(\theta)$, $b(\theta)$, $a(\theta+\pi)$, $b(\theta+\pi)$, as on the Figure 11.b. The following relations clearly hold:

$a(\theta) + a(\theta+\pi) = \Delta(\theta+\gamma) - \Delta_0(\theta+\gamma)$.
$b(\theta) + b(\theta+\pi) = \Delta(\theta-\gamma) - \Delta_0(\theta-\gamma)$.

Like in the proof of Theorem 1, it can be shown that $m(R, \theta) + m(R, \theta+\pi) = (\Delta(\theta) - \Delta_0(\theta))*O(R)$. Integrating over θ, obtain: $m(R)=(L-L_0)*O(R)$, from which the theorem follows. □

Similarly to ray shooting problem, from this theorem we deduce that the same optimality criterion holds for frustum culling as well:

Criterion. The optimal bounding volume of a certain type with respect to view frustum qualification is a bounding volume with minimal perimeter.

3.7 Optimality criteria in 3D

All the optimality criteria obtained for the 2D case can be generalized into 3D. In this section we discuss the validity of optimality criterion in 3D for ray shooting operation; a similar generalization can be done for viewing frustum culling operation as well.

Let us discuss again the results obtained in Theorems 1 and 2. Let C_r and C_R be two concentric circles, $r < R$. The key to the proofs of both theorems 1 and 2 is the fact that the area of the part of the stripe B lying within the circle C_R and originating in circle C_r is equal to $a*O(R)$, where a is the breadth of the stripe B (Figure 12.a, stripe B is grayed).

A similar result can be obtained for 3D case. Let S_r and S_R be two concentric spheres, $r < R$. Let A be infinite prismatic body obtained by the translation of some bounded plane body – an analogue of the stripe in 2D (which is obtained by translation of a line segment). In 3D, a body similar to the part of the stripe B is the body C, $C = A \cap S_R - S_r$. Clearly, the volume of the body C is equal to $f*O(R)$, where f is the area of the orthogonal section of the body A (Figure 12.b, body C is grayed).

To obtain bounding volume optimality criterion for ray shooting in 3D let us consider the object K_0 and its bounding volume K. As in 2D case, enclose both bodies into two concentric spheres S_r and S_R with radii r and R, $r < R$. Then for the fixed direction θ the measure $m(R, \theta)$ of the set of rays, intersecting body K but not intersecting body K_0 is equal to $(F(\theta) - F_0(\theta))*O(R)$, where $F(\theta)$, $F_0(\theta)$ are the areas of the projection of bodies K and K_0 onto the plane with normal in the direction θ. Integrating over all directions θ and employing Cauchy's formula (4), obtain: $m(R) = (F-F_0)*O(R)$, where F, F_0 are the surface areas of bodies K and K_0. Therefore, optimality criterion for ray shooting in 3D is the minimization of the surface area of the bounding volume - QED.

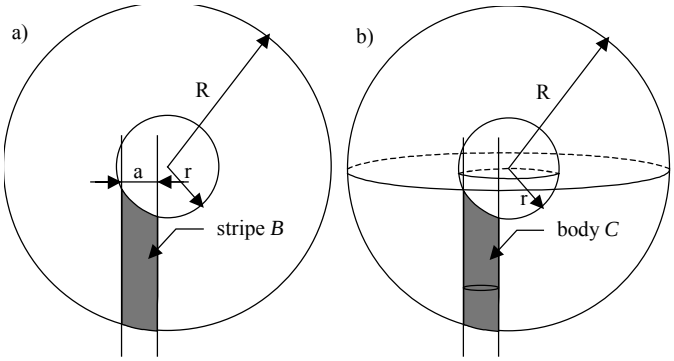

*Figure 12. The breadth of the grayed stripe B is equal to a*O(R), where a is the breadth of the stripe (left). The volume of the grayed space C is equal to f*O(R), where f is the area of the orthogonal section (right).*

3.8 Algorithms

Many authors have considered the problem of finding minimal enclosing figures. In 2D, Klee and Laskowsky [KLEE83] proposed an $O(N\log^2 N)$ algorithm for finding minimum area circumscribing triangle for a convex polygon with N sides. O'Rourke et al. [O'ROUR86] improved it to $O(N)$. Freeman and Shapira [FREE75] presented an $O(N^2)$ algorithm for finding minimal area circumscribing rectangle around a convex polygon, while Toussaint [TOUS83] showed how to reduce its complexity to $O(N)$ using the method of rotating calipers. Similar results were obtained for the problem of finding minimal perimeter enclosing rectangle [ION98a,b] for convex polygons and arbitrary point sets; the algorithms run in $O(N)$ and $O(N\log N)$ time, respectively. It can be shown that $O(N\log N)$ is the worst-case lower bound for minimal perimeter/area rectangle circumscription problem for arbitrary point sets [LEE86a,b][ION98b].

In 3D, O'Rourke [O'ROUR85] presented a tricky $O(N^3)$ algorithm for finding minimal volume enclosing box (OBB), and other authors [KLEE86] have generalized some of these results into higher dimensions. In [ION98a,b] we studied in detail the problem of minimal perimeter rectangle circumscription in 2D and presented an exact as well as a number of approximate algorithms for finding minimal surface area enclosing box in 3D.

4. Hierarchical bounding volume criteria

Most authors studying hierarchical bounding volume construction problem [ARVO][GOLD][WATT92] assume that

bounding volumes are nested through the hierarchy. This leads to minimal surface area optimality criterion, which holds as well for non-hierarchical bounding volumes. As we showed above (Section 3.2), this assumption is rather restrictive. In this section, we discuss the optimality criteria for bounding volume hierarchies. At the current stage of research, these criteria seem to be mostly theoretical due to the complexity of the involved computations. Because of this, we present the results for ray shooting in 2D case only – this is enough to demonstrate the technique and the use of integral geometry tools.

4.1 2-level hierarchies

Let K be a scene object, K_1 be a bounding volume for the parent object of K, and K_2 be a bounding volume of K. Clearly, $K_1 \cap K_2 \neq \varnothing$, $K \subset K_1 \cap K_2$. In [SAN76] it is shown that if K_1, K_2 are two convex bodies, $K_1 \cap K_2 \neq \varnothing$, $K \subset K_1 \cap K_2$, then the measure m of the set of lines G intersecting both bodies K_1 and K_2 but not intersecting K is computed as:

$$m(G;\ G \cap K_1 \neq \varnothing,\ G \cap K_2 \neq \varnothing,\ G \cap K = \varnothing) = L_1 + L_2 - L_{CH} - L, \qquad (7)$$

where L_{CH} is the perimeter of the convex hull of $K_1 \cup K_2$.

Therefore, bounding volume optimality criterion accounting for the presence of the bounding volume on the upper level for line tracing operation is the minimization of equation (7). Note that in (7) we used the measure of lines, not rays. But the results of the preceding sections indicate that the measures of lines and measures of rays are similar.

Let us consider the following example. Let K be a triangle with the base length equal to $2l$, and the two-level hierarchy is composed as on Figure 13. Let the bounding volumes be circles and assume that bounding volume K_1 on the upper level of hierarchy is a circle standing on the base of the triangle K as on diameter. Building the bounding volume for body K not accounting for K_1 gives a circle $K_2 = K_1$.

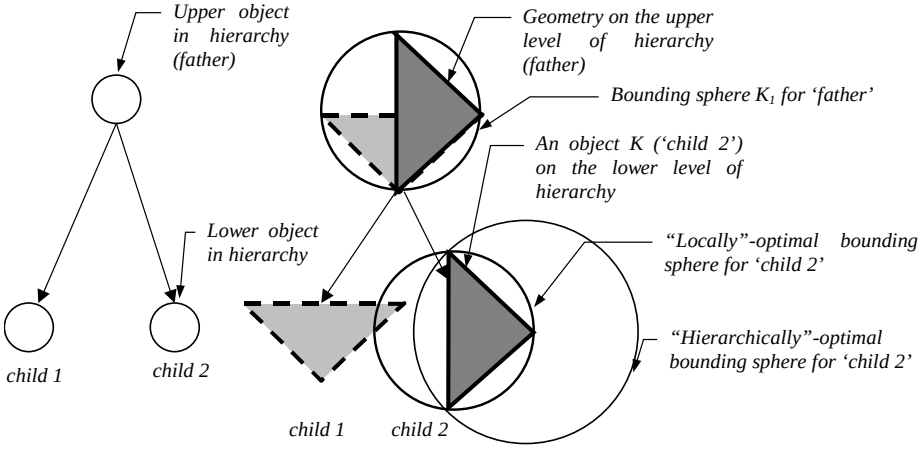

Figure 13. Two-level hierarchy used to illustrate hierarchical optimality criterion.

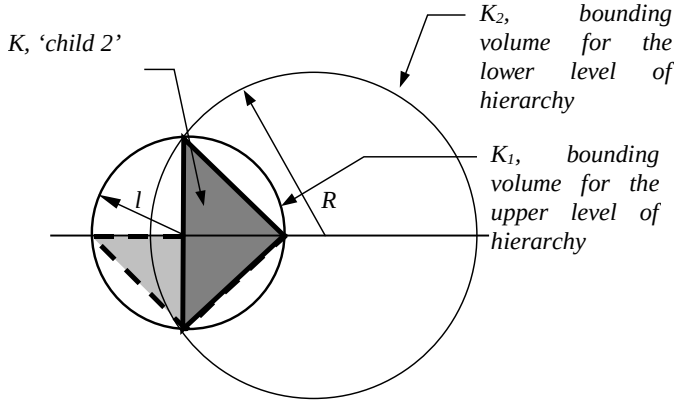

Figure 14. Hierarchically-optimal bounding volume for the lower level object is larger than the locally-optimal bounding volume.

Let us consider the family of circles $K_2(R)$ centered on the line perpendicular to the base of triangle K, circles $K_2(R)$ are the bounding volumes for K: $K \subset K_2(R)$ (Figure 14). Clearly, $R \geq l$. Explicit computation delivers the following equation for measure m (7) as a function of R (with l as a parameter):

$$m(R) = 2(R-l)\arcsin\left(\frac{l-R}{\sqrt{R^2-l^2}}\right) + \pi(l+R) - 2\sqrt{2}\sqrt{R^2-l^2}\sqrt{\frac{l}{l+R}}. \qquad (8)$$

The following asymptotics hold for the function $m(R)$:

$\lim_{R\to+\infty} m(R) = \lim_{R\to l+0} m(R) = 2\pi l - 4l = L_1 - L.$

Function *m(R)* has a single minimum; a schematic plot of this function is shown on Figure 15. The less the function *m(R)* the better the bounding circle is. Analyzing function *m(R)* we come up to an amazing conclusion: any bounding circle, not equal to K_1 (locally optimal), is better in a hierarchical sense. Hierarchically optimal bounding circle (corresponding to the minimum of *m(R)*) is approximately 40% better than the locally optimal circle K_1.

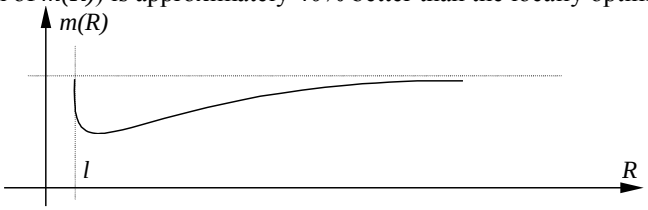

Figure 15. A schematic plot of the function m(R). This function illustrates the 'quality' of bounding circles $K_2(R)$ in a sense described above. The less the function m(R) the better bounding circle is.

This simple example proves that the use of hierarchically optimal bounding volumes may significantly improve the overall quality of tests with bounding volume.

The intuition that lies behind hierarchical bounding volumes (and especially behind the given example) is quite simple. Indeed, in our example minimal bounding sphere for child object equals the one of the parent object. Since it is clearly useless to do the test with the same bounding volume twice (firstly for the parent object and then for the child), we should select another bounding volume (sphere in our case) for the child. Obviously, this sphere is not the minimal enclosing sphere, but it is better for ray shooting. This situation happens in a general case as well.

4.2 *n*-level hierarchies.

Extending the results described above, one can obtain *n-level* hierarchical bounding volume optimality criterion. Let *K* be a scene object, $\{K_i\}$ be bounding volumes for the parent objects of *K* in the hierarchy. Clearly, $K \subset \cap K_i$. Using the algorithm for computing the measures of Buffon sets of lines on the plane [AMB82], we obtain that the measure of lines intersecting all K_i, but not intersecting *K*, is:

$$m(G: \forall i\ G \cap K_i \neq \varnothing, G \cap K = \varnothing) = \Sigma l_j - \Sigma t_k - L, \qquad (9)$$

where:

l_i – arclengths of arc ∂K_i such that in all points of the arc the tangent intersects all K_i;

t_i – lengths of segments of such common tangents for K_i, K_j, that intersect all K_i;

L – perimeter of *K*.

The variables used in this formula are illustrated on Figure 16. The expression (9) is the optimality criterion for *n*-level hierarchies.

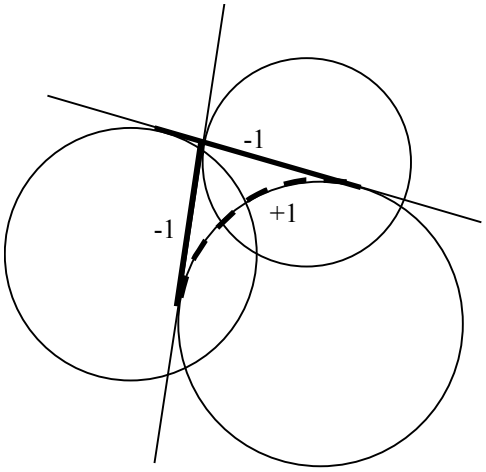

Figure 16. Illustration of tangents and arcs used in n-level hierarchical criterion (9) (n = 3). Bold line segments correspond to t_k, dashed bold arc corresponds to l_i. "+1"/"-1" indicate the sign with which respective terms are used in (9).

4.3 Analysis

Unlike the problem of non-hierarchical bounding volume construction where it is possible to construct optimal bounding volumes for the class of, for example, oriented bounding boxes in 3D, the problem of *n*-level optimal bounding volume construction is much harder even in 2D for the simplest class of bounding volumes – circles. Using an incremental algorithm one may build optimal 2-level bounding circles on the plane in $O(N^2)$ time (where *N* is the maximum number of points in an object in the hierarchy). As for other classes of bounding volumes and for the 3D case there are no algorithms available. The general *n*-level case is even harder – there are no known algorithms even for circles on the plane.

This lack of algorithms makes the results on multi-level bounding volumes mostly theoretic. In spite of this, from the authors' point of view they are important to better understand the problems related to bounding volume construction.

Still, there is a practical application of the n-level optimality criteria. Constructing bounding volume hierarchies, one may use the technique described in [GOLD87], where the object tree is constructed incrementally basing on the estimated complexity of the tree after the insertion of an object. To evaluate the complexity, Goldsmith and Salmon used heuristic formulae obtained for nested bounding volume hierarchies. More proper formulae can be obtained with the use of n-level optimality criteria. The detailed discussion of these issues is beyond the scope of this paper.

5. Polygon triangulation to speed up rendering

Polygon triangulation problem has been thoroughly addressed in the field of Computational Geometry [O'ROUR93][BERG97].

Let us discuss now the problem of polygon triangulation in the context of computer graphics. To perform rendering, any polygon in the scene is usually triangulated at the preprocessing. The performance of a computer graphics application frequently depends on the performance of its triangle rasterizer, which, in its turn, depends on the number of pixels drawn per frame as well as on the number of rasterized triangles. Clearly, the number of drawn pixels does not depend on the polygon triangulation. On the other hand, it turns out that it is possible to triangulate any given polygon in such a way that *on average* the number of triangles passed to rasterizer is minimal among all possible triangulations. So our goal is to find a criterion for such a triangulation. Note, that we are averaging on all possible camera positions and directions. As usual, we assume that the scene can be observed from an arbitrary viewpoint, so we assume that cameras are uniformly distributed in space.

5.1 Integral geometry point of view

Let P be a given polygon, T be its triangulation, k be the number of triangles in T (k is the same for all possible triangulations), and C be a given camera that specifies the viewpoint V_O, image plane q and viewport – a rectangle Q. Note that camera is uniquely defined by Q. For the sake of simplicity, we state the results for orthogonal projection type – their extension for the case of perspective projection is rather complicated. We assume that P is a planar polygon in 3D and that P lies completely beyond the image plane (Figure 17). The projection of P onto the image plane q is another polygon P'. Since P is not clipped by near clipping plane, the projection of triangulation T onto q naturally induces a triangulation T' of P', the triangles of which correspond one-to-one to triangles of T. Only the triangles of T' that intersect viewport Q are passed to rasterizer. Our goal is to ensure that *on average* the number of triangles passed to rasterizer is minimal. Therefore, we come up with the following problem statement:

Problem. "*Polygon triangulation to speed up rendering*".

Given: A polygon P.

Build: Find a triangulation T of P such that on average the number of visible triangles is minimal among all triangulations.

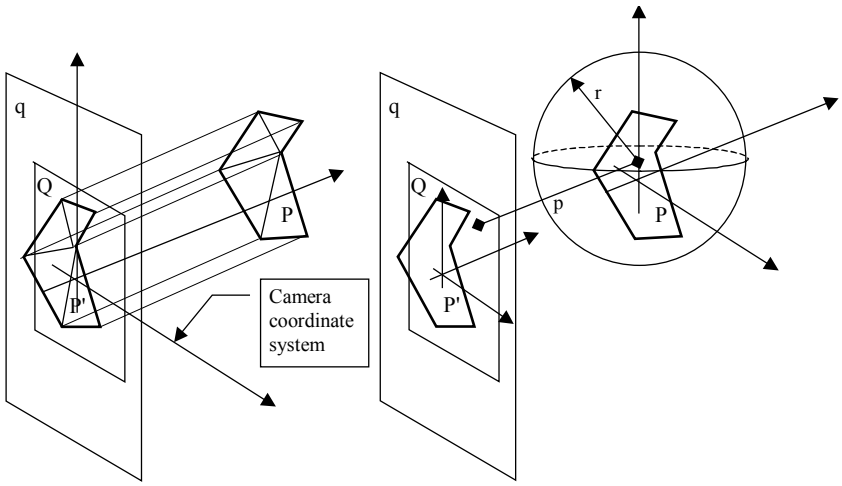

Figure 17. Illustration of the notation used in this section.

Let us assume that the coordinate system origin O lies on P, and P is enclosed into a sphere S_r centered in O with radius r. Obviously, if the distance from O to image plane q is greater than r, then P is not clipped by camera near plane. Let $X(C)$ denote the number of triangles of T that are visible for a given camera C. Let $X_j(C)$ be equal to *1* if *j-th* triangle is visible for a given camera C and *0* otherwise.

In computer graphics, it is usually assumed that polygon P can be viewed from an arbitrary viewpoint, i.e. camera C can be arbitrarily placed in the space. To establish the criterion, we compute the average number n of triangles visible in camera C for all cameras positioned in $S_R \setminus S_r$, where S_R is a sphere centered in O with radius R. Camera C can be specified by the position of the camera origin (the center of viewport Q) and the direction of camera viewing axis (normal to Q). By averaging the number of visible triangles over the domain $S_R \setminus S_r$ we are using the assumption that cameras are uniformly distributed in space.

The average number $n(R)$ can be computed as follows:

$$n(R) = \frac{\int_{S_R \setminus S_r} X(C)dC}{\int_{S_R \setminus S_r} dC} = \frac{\int_{S_R \setminus S_r} \sum_{j=1}^{k} X_j(C)dC}{\int_{S_R \setminus S_r} dC} = \frac{\sum_{j=1}^{k} \int_{S_R \setminus S_r} X_j(C)dC}{\int_{S_R \setminus S_r} dC}. \quad (10)$$

The unique up to a constant factor density of cameras dC can be defined as follows:
$dC = \sin\theta\, dp \wedge d\theta \wedge d\phi \wedge dQ$,
where:
(θ,ϕ) – the angles defining the direction of camera C;
p – the distance from O to image plane q;
dQ – the density of camera viewports Q on image plane q.

Let $I_j = \int_{S_R \setminus S_r} X_j(C)dC$. To compute the integrals I_j in (10), we initially fix camera direction (θ,ϕ) and distance p from the origin to image plane q. For any given j and fixed (θ,ϕ) and p, integral $I_j(\theta,\phi,p) = \int_{\text{by all } Q \text{ on plane } (\theta,\phi,p)} X_j(C(\theta,\phi,p))dQ$ is the measure of rectangles Q that intersect triangle T'_j: $I_j(\theta,\phi,p) = m(Q;\, Q \cap T'_j \neq \varnothing)$. In [SAN76], it is proved that this measure is as follows:

$m(Q;\, Q \cap T'_j \neq \varnothing) = 2\pi(F_Q + F_{T'j}) + L_Q + L_{T'j}$,
where F and L are area and perimeter of Q and T'_j.

Therefore, $I(\theta,\phi,p) = \Sigma I_j(\theta,\phi,p) = 2\pi(kF_Q + F_{P'}) + kL_Q + 2\Sigma a_i \cos\delta(a_i,\theta,\phi) + \Sigma b_i \cos\delta(b_i,\theta,\phi)$,
where:
$F_{P'}$ – area of polygon P', $F_{P'} = F_P * \cos\delta(\theta,\phi)$, where $\delta(\theta,\phi)$ is the angle between plane q and plane of P;
F_Q, L_Q – area and perimeter of viewport rectangle Q;
a_i – the lengths of diagonals of P used in triangulation T;
b_i – the lengths of sides of P (all of them are used in any triangulation of P);
$\delta(x,\theta,\phi)$ – the angles between the diagonals or sides of P and the selected image plane $q = (\theta,\phi,p)$;
$a_i\cos\delta$, $b_i\cos\delta$ are the lengths of diagonals and sides of T projected onto q.

Integrating $I(\theta,\phi,p)$ by (θ,ϕ,p), obtain:

$$I = \left((2\pi kF_Q + kL_Q)\int_{(\theta,\phi)} \sin\theta\, d\theta\, d\phi + F_P \int_{(\theta,\phi)} \cos\delta(\theta,\phi)\sin\theta\, d\theta\, d\phi + 2\sum a_i \int_{(\theta,\phi)} \cos\delta(a_i,\theta,\phi)\sin\theta\, d\theta\, d\phi + \sum b_i \int_{(\theta,\phi)} \cos\delta(b_i,\theta,\phi)\sin\theta\, d\theta\, d\phi\right) * (R-r) =$$
$$\left((2\pi kF_Q + kL_Q) * 4\pi + F_P * 4\pi + 2\pi^2 * \sum a_i + \pi^2 * \sum b_i\right) * (R-r).$$

Note that $L_P = \Sigma a_i$, $W_T = \Sigma b_I$ is the weight of triangulation T, i.e. the length of all diagonals in T.
Therefore, $I = (C + W_T) * (R-r)$, where C is some constant.

Therefore, to minimize the average number of visible triangles $n(R)$ we have to minimize the *weight of triangulation* T – this is the desired criterion.

5.2 Analysis and algorithms

The problem of minimal weight triangulation has been studied in the field of computational geometry. Minimal weight triangulation of a simple polygon (without holes) can be constructed using a dynamic programming algorithm in $O(N^3)$ time [KLIN80]. The degree of the polynomial goes up by one for every hole, so for polygons with k holes the time complexity is $O(N^{3+k})$.

6. Global line distribution analysis for efficient Monte Carlo radiosity, IBR and other applications

The interest for simulating a global density of lines in the sense of Integral Geometry arosed first in Radiosity [SBER97], where *global* lines have been used both for Form Factor computation and Random Walk simulation. Recently, the interest has extended to the Image Based Rendering (IBR) field. In IBR radiance is parameterized through lines (it is constant along a line). Thus we need lines (as in the Radiosity case) that cover the scene. Global lines have also been used in [FEIX99] and [CAZA98] for an analysis of scene complexity. We will review here different density generations. For

their use in Radiosity within the quasi-Monte Carlo context see [CAST98b].

6.1 Global lines and Radiosity.

Monte Carlo generation of global lines is used in Radiosity to simulate the Form Factor density of lines. The use of quasi-Monte Carlo has been explored in [CAST98,CAST98b]. Now we will review the Form Factor density. Suppose the scene is discretized into polygons or *patches* (not necessarily planar). The Form Factor between patches *i* and *j* is then defined as the integral

$$F_{ij} = \frac{1}{\pi A_i} \int_{A_i} \int_{A_j} \frac{\cos\theta_i \cos\theta_j}{r^2} V_{ij} dA_i dA_j \qquad (11)$$

where V_{ij} = Visibility function: 1 if dA_i and dA_j are mutually visible, and 0 otherwise. The visibility computation is the main source of cost in the Form Factor computation. Taking into account the equivalency $d\varpi = \frac{\cos\theta_i}{r^2} dA_j$, we can give a second form for the Form Factor integral:

$$F_{ij} = \frac{1}{\pi A_i} \int_{\Omega} \int_{A_i} V_{ij}(\theta,\phi,x) \cos\theta dA_i d\varpi = \frac{1}{\pi A_i} \int_{\Omega} \int_{A_i} V_{ij}(\theta,\phi,x) \cos\theta \sin\theta dA_i d\theta d\phi \qquad (12)$$

where the integral is taken over the patch *i* and the hemisphere of directions above the patch, and V_{ij} is 0 or 1 according to whether the patch *j* is visible or not from area differential dA_i with center *x* and in direction (θ,ϕ) (see figure 18).

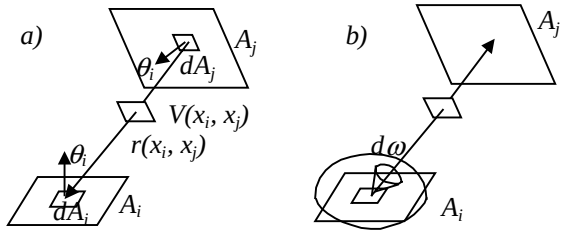

Figure 18. Geometry for the Form Factor as area integral (a) and hemisphere integral (b).

This second integral can be given a simple geometrical interpretation. Suppose that we cast lines from patch *i* with probability density function $f(\theta,\phi,x) = \frac{\cos\theta \sin\theta}{\pi A_i}$. Sampling this probability is easy. We only have to take an uniformly distributed point over patch *i*, and weight an uniformly distributed direction ω by *cos*θ. This amounts to taking *(θ,ϕ,x)* = *(arccos* $\sqrt{R_1}$ *, 2πR₂, , x)*, where R_1 and R_2 are random values from a uniform distribution in the interval [0,1]. Then, using Monte Carlo integration we have:

$$\begin{aligned} F_{ij} &\approx \frac{1}{N} \sum_{k=1}^{N} \left(\frac{1}{\pi A_i} \right) \frac{V_{ij}(\theta_k,\phi_k,x_k) \cos\theta_k \sin\theta_k}{f(\theta_k,\phi_k,x_k)} \\ &= \frac{1}{N} \sum_{k=1}^{N} \left(\frac{1}{\pi A_i} \right) \frac{V_{ij}(\theta_k,\phi_k,x_k) \cos\theta_k \sin\theta_k}{\frac{\cos\theta_k \sin\theta_k}{\pi A_i}} \\ &= \frac{1}{N} \sum_{k=1}^{N} V_{ij}(\theta_k,\phi_k,x_k), \end{aligned}$$

that is, the number of hits on patch *j* divided by the total number of trials. So the Form Factor F_{ij} between patch *i* and patch *j* may be interpreted as *the probability of a line that exiting patch i lands in patch j*. From that we have that the Form Factor F_{ij} may be estimated as the relation given by r_{ij}/r_i, where r_i is the number of lines that exit patch *i* and r_{ij} is the number of these lines which also have patch *j* as nearest intersected patch (see figure 19a).
It must be remarked here that this is the only density that leads to this interpretation. Using other densities we must *weight* each line. As an example, suppose we take directions uniformly distributed on the hemisphere. This means taking as pdf

$f_2(\theta,\phi,x) = \dfrac{\sin\theta}{2\pi A_i}$. The value of F_{ij} can then be estimated as:

$$F_{ij} \approx \frac{1}{N}\sum_{k=1}^{N}\left(\frac{1}{\pi A_i}\right)\frac{V_{ij}(\theta_k,\phi_k,x_k)\cos\theta_k \sin\theta_k}{f_2(\theta_k,\phi_k,x_k)}$$

$$= \frac{1}{N}\sum_{k=1}^{N}\left(\frac{1}{\pi A_i}\right)\frac{V_{ij}(\theta_k,\phi_k,x_k)\cos\theta_k \sin\theta_k}{\frac{\sin\theta_k}{2\pi A_i}}$$

$$= \frac{2}{N}\sum_{k=1}^{N} V_{ij}(\theta_k,\phi_k,x_k)\cos\theta_k \ ,$$

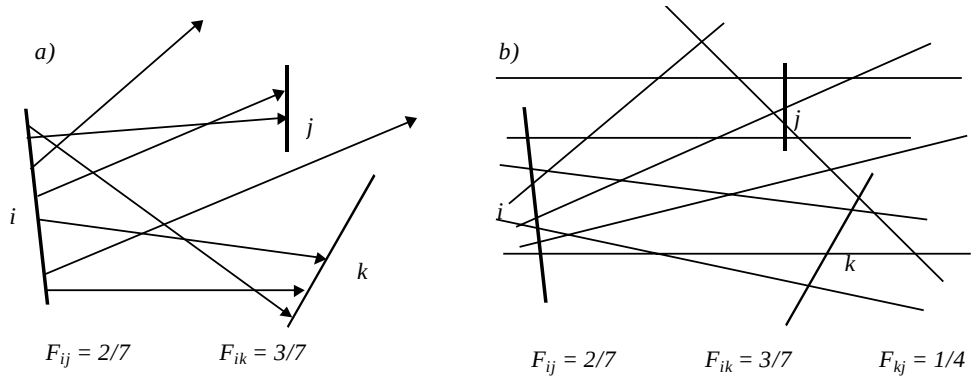

Figure 19. (a) Form Factors with local lines from patch i. (b) Global density to compute Form Factors from all patches.

From the above results, we see that the most natural Form Factor generating density of lines on a patch *i* is $dG = \cos\theta \sin\theta d\theta d\phi dA_i = \cos\theta d\omega dA_i$. But this is the same density submitted by an IG global density of lines on patch *i* [SANT76]. Thus a global density can be used to generate the required densities on *all* patches of the scene (see figure 19b). In this way, we use all the intersections of a line with the scene, instead of just using the nearest one with the local lines. Just now when a patch is not planar the value for r_i in the Form Factor estimation r_{ij}/r_i has to be substituted by the number of intersections of the global lines with the patch (which reverts to number of lines for a planar patch). In fact, πA_i, the denominator in the Form Factor definition (11), is the IG measure of the total number of intersections of the global density with patch *i*, and the integral in (11,12) is the IG measure of lines intersecting both patches *i* and *j* without being occluded.

The meaning of Form Factors as a visibility measure between patches is thus clear. Visibility directions should be homogeneous and isotropic all over the scene, a quality fulfilled by the uniform global line density, and the Form Factors measure this visibility. The visibility complexity defined in [FEIX99] is based upon this quality.

As we can simulate with global lines the Form Factor density, so we can use them to build a Random Walk simulation to compute radiosity [SBER97,SBER96] (see figure 20).

A discussion on the efficiency of the techniques using both local and global lines can be found in [SBER97].

Now we are confronted with the following problem:

Problem. *"Uniform line density generation"*.

Given: A scene S.

Build: Find an efficient uniform line generation covering scene S.

6.2 Line density generation

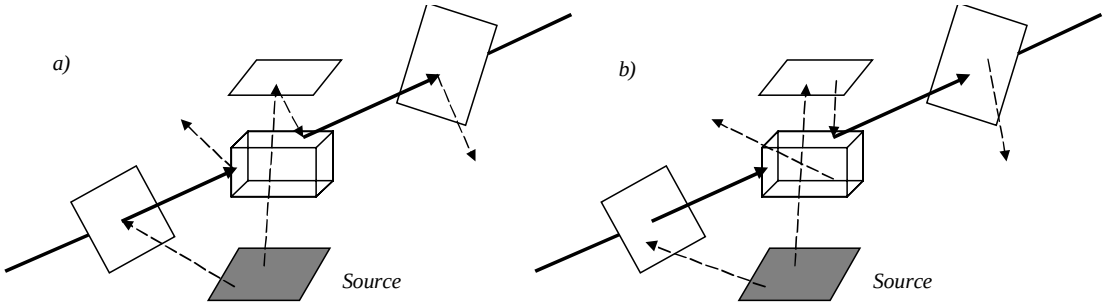

Figure 20. (a) Random walk simulation with global lines. A global line *(the thick continuous one) makes two paths advance at once. Considering bidirectionality of the global lines, two other paths will also advance in the reverse*

direction of the line. (b) Stressing the fact that the exit point on each patch is at random.

As we have seen in the previous section, the line density is tightly related to the Form Factors, and thus to the integrand of formulae (11) and (12). Following the discussion in [CAST98b] we will examine next several alternatives to generate a global line density.

- **Two random points on a sphere surface**

In [SANT76] it is shown that the density of global lines intersecting a convex body *K* (that is, density of chords) is given by $\frac{\cos\theta\cos\theta'}{r^2}d\sigma d\sigma'$, where θ, θ' are the angles of the intersecting line with the normals in the intersecting points and $d\sigma$, $d\sigma'$ are area differentials in the same points and *r* is the chord. This density, for a sphere, becomes simply (save a constant factor) $d\sigma d\sigma'$ (see figure 21a).

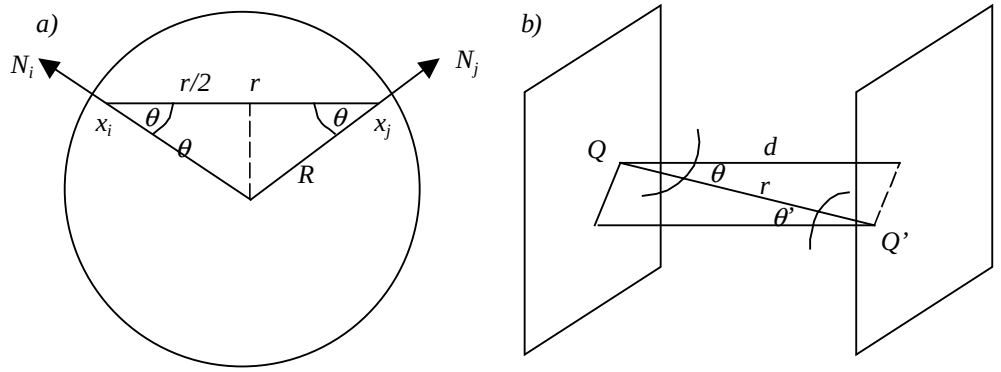

Figure 21. a) Geometry for two points on sphere line generation. (b) Geometry for incorrect line generation.

This means that taking pairs of random points in the sphere surface we obtain a global uniform density of lines (see figure 22a). But observe that this is only valid for a sphere, i.e., taking pairs of points on the surface of a convex body does not result in a uniform density (except of course for the sphere). To see how one can deviate from the uniform density, let us consider taking pairs of points on opposite faces of an orthogonal prism [LEVO96]. The correct density should be proportional to $\frac{\cos\theta\cos\theta'}{r^2}d\sigma d\sigma'$, the one taken is proportional to $d\sigma d\sigma'$. The ratio of the densities is proportional to $\frac{\cos\theta\cos\theta'}{r^2}$. Now, observe figure 21b, from this figure $cos\theta = cos\theta' = d/r$, thus the ratio is proportional to $1/r^4$. This means for instance that for twice the distance we cast 2^4 more lines than necessary. In other words, much more lines are cast in oblique directions than in orthogonal ones.

Observe also that the sphere density is equivalent to taking a single point on the sphere and a uniform direction from this point, weighted according to the cosinus of the angle θ between the radius at this point and the tangent plane (this is the same to say between the tangent plane and the normal to the plane). Thus, taking simply a uniformly distributed direction does not result in uniform density.

The sphere density is described in [SOL78], and was first used for Radiosity in [SBER93] and in IBR in [CAMA98]. Interestingly, this uniform density generation has not a counterpart in 2D. This is, taking pairs of points uniformly distributed on a circumference does not provide a uniform density within the circumference [SOL78].

- **Random direction and point in main circle**

This is the generation that appears in classic IG books [REYP51,SANT76,SOL78] and it has been used in IBR by Camahort et al. [CAMA98]. (See figure 22c). It is obviously equivalent to selecting a tangent plane (thus a point in the sphere), and a point in the projection of the sphere on the plane.

- **Bundles of parallel lines**

If one limits randomness in the previous defined density to selecting the tangent plane and taking the points on the plane on a regular grid (see figure 22d), one obtains bundles of parallel lines. This density has been used in the Radiosity context in [SBER97,SBER96,NEUM95,SZIR98], and in IBR in [CAMA98,LICHI98]. A point to remark with this density is that the expected number of lines crossing a planar polygon with area *A* given *N* bundles of lines is $2AN/\Delta$, where Δ is the area of the grid cell. Another point is that the grid cell can be any parallelogram. Also, to avoid aliasing, the origin of the grid can be jittered [SBER97]. The advantage of using bundles of parallel lines is that fast projection algorithms as the

z-buffer or painter algorithm can be used.

- **Lines from the walls of a convex bounding box**

Using the two random points on a sphere density has the hindrance that the lines not crossing the scene will be wasted. This can be specially inefficient for "flat" scenes as for instance a building's floor. We see next how to overcome this difficulty.

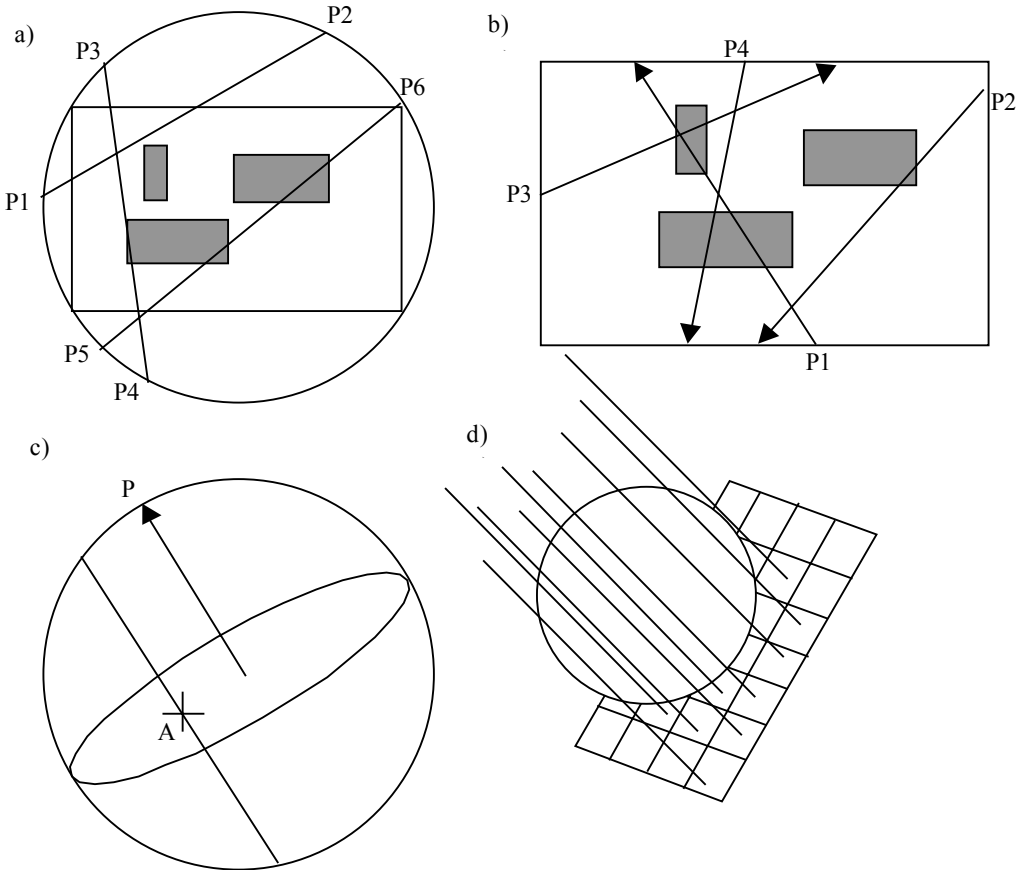

Figure 22. Different ways to simulate uniform global lines. (a) Two random points on the surface of the sphere. (b) Local *lines are cast from the walls of convex bounding box. (c) Random direction defining a disc by the bounding sphere center, and random point in this disc. (d) Tangent plane by a random point on the sphere surface and bundle of parallel lines perpendicular to this plane.*

Using the fact that the differential of solid angle around $d\sigma$ can be written as $d\varpi = \frac{\cos\theta'}{r^2} d\sigma'$, we can transform the chord density into $\cos\theta d\varpi$. This means taking a random point on the surface of the convex bounding box and a cosinus weighted uniformly distributed direction (see figure 22b). This density generation was first described in [SBER97]. It is useful as we can use the same bounding box of the scene to generate the lines. If the (convex) bounding box can be tightly adjusted to the scene, very few lines will be wasted. Observe that this is not the same as taking pairs of uniformly distributed random points on the bounding box surface, which is incorrect as seen above.

7. Conclusions

We have presented in this paper the study of several computer graphics problems from the point of view of integral geometry. The examples presented here are the selection of optimal oriented bounding boxes for ray shooting, frustum culling and collision detection, hierarchical bounding volume selection, polygon triangulation for faster rendering and global line distribution analysis for efficient Monte Carlo radiosity. Previous results together with new ones have been presented in an unified way. The *integral* (or *average-case*) analysis enables to create 'good' data structures that significantly speed up the algorithms. This results in *provable, theoretically sound criteria*, in contraposition to *heuristic* ones, for appropriate data structures.

Acknowledgements
Andrei Iones, Sergei Zhukov and Anton Krupkin wish to acknowledge the invaluable support of this research by the administration of CREAT Studio who always considered it important – even in view of the upcoming production deadlines.
Mateu Sbert has been funded with grant number TIC98-0856-C03-02 from the Spanish Government.

References
[AGGA85] A. Aggarwal, J. S. Chang, C. K. Yap: “Minimal area circumscribing polygons”, The Visual Computer, 1:112-117
[AMB82] Ambartzumian R.V.: Combinatorial Integral Geometry, Wiley, NY, 1982
[ARVO89] J. Arvo, D. Kirk: "A Survey of Ray Tracing Acceleration Techniques", in "An Introduction to ray tracing", ed. A. Glassner, Academic Press, 1989, pp. 206-217
[BERG97] M. de Berg et al.: "Computational Geometry. Algorithms and applications", Springer-Verlag, 1997
[BONN34] T. Bonnesen, W. Fenchel: “Theorie der convexen Körper”, Ergeb. Math., Berlin: Springer, 1934
[CAMA98] E. Camahort, A. Lerios and D. Fussell: “Uniformly Sampled Light Fields”, Rendering Techniques'98, pp.117--130, Springer--New-York, 1998.
[CART67] H. Cartan: “Formes differentelles”, Paris, Hermann, 1967
[CAST98] F. Castro, R. Martínez, M. Sbert: "Quasi Monte Carlo and extended first shot improvement to the multi-path method for radiosity", Proceedings of SCCG'98, Budmerice, Slovakia, April 1998.
[CAST98b] F. Castro, M. Sbert: Application of quasi-Monte Carlo sampling to the multi-path method for radiosity", 3rd International Conference on Monte Carlo and quasi-Monte Carlo Methods in Scientific Computing, Claremont, USA, June 1998 (to appear in Springer series Lecture Notes in Computational Science and Engineering, Springer Berlin, 1999).
[CAZA98] F.Cazals, M.Sbert: Some Integral Geometry Tools to estimate the complexity of 3D scenes, INRIA Research Report RR-3204.
[CLARK76] J. H. Clark: “Hierarchical Geometric Models for Visible Surface Determination”, Comm. ACM, 19(10), 547-554, 1976
[DORI83] D. Dori and M. Ben-Bassat: “Circumscribing a Convex Polygon by a Polygon of Fewer Sides with Minimal Area Addition”, Computer Vision, Graphics and Image Processing, 24:131-159
[FEIX99] M.Feixas, E.Acebo, P.Bekaert and M.Sbert, “An Information Theory Framework for the analysis of scene complexity”, Eurographics’99 Proc., Milan, 1999.
[FREE75] H. Freeman and R. Shapira: “Determining the minimum-area encasing rectangle for an arbitrary closed curve”, Comm. ACM, Vol. 18, July 1975, pp. 409-413
[GOLD87] J. Goldsmith, J. Salmon: "Automatic creation of object hierarchies for ray tracing", IEEE CG&A, 7(5), pp. 14-20 (1987).
[GOTT96] Gottschalk et al.: “OBBTree: A Hierarchical Structure for Rapid Interference Detection”, Proc. of SIGGRAPH’96 Conference, pp. 171-180.
[ION98a] A. Iones, S. Zhukov, A. Krupkin: “Building optimal bounding volumes for real-time 3D visualization”, Machine Graphics & Vision, Vol.7 (Proc. GKiPO’98, Warsaw, Poland), 1998
[ION98b] A. Iones, S. Zhukov, A. Krupkin: “On Optimality of OBBs for Visibility Tests for Frustum Culling, Ray Shooting and Collision Detection”, in Proc. of CGI’98, Hannover, Germany, 1998, pp. 256-264
[KAY86] T. L. Kay, J. T. Kajiya: “Ray tracing complex scenes”, Computer Graphics’86, 20(4), 269-278 (Proc. SIGGRAPH’86)
[KLEE83] V. Klee and M. Laskowski: “Finding the Smallest Triangles Containing a Given Convex Polygon”, J. of Algorithms, 6:359-375
[KLEE86] V. Klee: “Facet-centroids and Volume Minimization”, Studia Scientiarium Mathematicarum Hungarica 21:143-147
[KLIN80] Klincsek G. T.: "Minimal triangulations of polygonal domains", Discrete Math. J., v. 9, 1980, pp. 121-123.
[LEE86a] D. T. Lee “Geometric Location Problems and Their Complexity” Proc. of 12 Intl. Symposium on Math. Foundation of Computer Science, Bratislava, Czechoslovakia, 1986
[LEE86b] D. T. Lee, Y. F. Wu “Geometric Complexity of Some Location Problems”, Algorithmica, 1: 193-211, 1986
[LEVO96] M. Levoy and P. Hanrahan: "Light Field Rendering",
Proceedings of Siggraph 96, pp.31--42, LA, August 1996
[LICHI98] Dani Lischinski and Ari Rappoport: "Image-Based Rendering for Non-Diffuse Synthetic Scenes", Rendering Techniques'98, Springer--New-York, 1998.
[NEUM95] Laszlo Neumann: "Monte Carlo Radiosity", Computing, 55(1):23--42, 1995
[O’ROUR85] J.O’Rourke: “Finding minimal enclosing boxes”, Internat. J. Comput. Inform. Sci., 14:183-199, 1985
[O’ROUR86] J. O’Rourke et al.: “An Optimal Algorithm for Finding Minimal Enclosing Triangles”, J. of Algorithms, 7:258-269
[O’ROUR93] J. O’Rourke: “Computational Geometry in C”, Cambridge University Press, 1993
[REYP51] J.Rey-Pastor and Luis Santaló Sors, Geometría Integral, Espasa-Calpe, Madrid, 1951
[RUB80] S. M. Rubin, T. Whitted: “A Three-dimensional Representation for Fast Rendering of Complex Scenes”, Computer Graphics 14(3), 110-116 (Proc. of SIGGRAPH’80)
[SANT76] L. Santalo: “Integral geometry and geometric probability”, Addison-Wesley, 1976
[SBER93] Mateu Sbert: "An Integral Geometry Based Method for Fast Form-Factor
Computation", Computer Graphics Forum, Vol 12 N. 3, pp.409--420 , 1993, (Eurographics'93)

[SBER96] Mateu Sbert, Xavier Pueyo, Laszlo Neumann, and Werner Purgathofer: "Global multipath Monte Carlo algorithms for radiosity", The Visual Computer, 12(2):47--61, 1996.
[SBER97] Mateu Sbert: The use of global random directions to compute radiosity.
Global Monte Carlo methods. Ph.D. thesis, Universitat Politècnica de Catalunya, Barcelona, March 1997. Available in http://ima.udg.es/~mateu
[SOL78] H. Solomon: “Geometric probability”, SIAM-CBMS 28, 1978
[SZIR98] Laszlo Szirmay-Kalos and Werner Purgathofer: "Global ray-bundle tracing with hardware acceleration", Rendering Techniques'98, pp.247--258, Springer--New-York, 1998.
[TOUS83] G. Toussaint: “Solving Geometric Problems with the ‘Rotating Calipers’”, Proc. of IEEE MELECON’83, Athens, Greece, 1983
[WATT92] A. Watt, M. Watt: “Advanced Animation and Rendering Techniques. Theory and Practice”. ACM Press, NY 1992
[WEGH84] H. Weghorst et al.: “Improved computational methods for ray tracing”, ACM Transactions on Graphics, pp. 52-69, 1984
[WHIT80] T. Whitted: “An Improved Illumination Model for Shaded Display”, Comm. ACM, 26(6):342-349