On the Use of Local Ray Termination for Efficiently Constructing Qualitative BSPs, BIHs and (S)BVHs

Matthias Moulin, Philip Dutré

Department of Computer Science, KU Leuven, Belgium

Dédié à la fille dont le sourire rayonnant et magique m'aidait à accomplir ce projet finalement.

Abstract

Acceleration data structures (ADSs) exploit spatial coherence by distributing a scene's geometric primitives into spatial groups, effectively reducing the cost of ray tracing queries. The most effective ADSs are hierarchical, adaptive tree structures such as BSPs, BIHs and (S)BVHs. The de facto standard cost metric for building these structures is the Surface Area Heuristic (SAH), which assumes a scene-exterior isotropic ray distribution of non-terminating rays. Despite its restrictive assumptions, the SAH remains competitive against many fundamentally different cost metrics targeting more common ray distributions. Our goal is not to radically change and replace the SAH, but to adapt it by introducing the concept of local ray termination in the context of voxel partitioning during the ADS construction and voxel traversal order during ADS traversal. We develop heuristics to approximate local ray termination efficiently without additional preprocessing or ray (sub)sampling. Our heuristics are used for approximating the visibility probabilities in the Ray Termination Surface Area Heuristic (RTSAH) for constructing BSPs, BIHs and (S)BVHs for accelerating closest-hit ray queries and for approximating the hit probabilities in the Shadow Ray Distribution Heuristic (SRDH) for constructing dedicated BVHs for accelerating any-hit ray queries. The main aim of our paper is to analyse the potential of including local ray termination into the SAH. The results indicate rendering performance close to the references (SAH and NodeSATO) on average due to small and/or compensating gains in the number of ray-triangle intersection tests and ADS node traversal steps. Furthermore, prerendering build times are higher for the RTSAH due to triangle clipping.

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Raytracing

1. Introduction

Ray tracing-based renderers trace rays through a virtual 3D scene, which need to be intersected with this scene's geometry to resolve visibility and which are used in shading calculations. *Acceleration data structures* (ADSs) such as BSPs, BIHs and (S)BVHs aim to reduce the cost of these ray tracing queries. The de facto standard cost metric for building these hierarchical, adaptive tree structures is the *Surface Area Heuristic* (SAH) [GS87, MB90].

We extend and generalize the SAH by taking the occlusion by the geometry contained in the voxel to partition into account, revising the SAH's restrictive assumption of infinitely long rays, irrespective of any intersections with intervening geometry that might terminate rays in an actual ray tracing algorithm. We approximate this *local ray termination* using various heuristics without requiring additional preprocessing or ray (sub)sampling, and use these approximations for evaluating:

- The *Ray Termination Surface Area Heuristic* (RTSAH) [IH11, MBD15] for constructing BSPs, BIHs and (S)BVHs for accelerating closest-hit ray queries;
- The *Shadow Ray Distribution Heuristic* (SRDH) [FLF12] for constructing BVHs for accelerating any-hit ray queries.

Our work provides a complete and exhaustive, theoretical and practical study of the inclusion of local ray termination in the SAH (in multiple dimensions: multiple types of ADSs, multiple types of ray queries, multiple approximation types, multiple approximations, etc.), which we believe is useful in isolation, in relation to the SAH and in related domains like collision detection.

2. Related Work

ADS types. The number of ray-geometric primitive (e.g., triangle) intersection tests, which is a crucial factor in the general performance of a ray tracing algorithm [App68, Whi80, Kaj86], can be kept within certain theoretical bounds by using appropriate ADSs. These (often hierarchical) structures exploit spatial coherence by distributing the scene's geometric primitives into spatial groups. If such a group is outside the range of a given ray, the geometric primitives contained in that group can be pruned (i.e. do not need to be

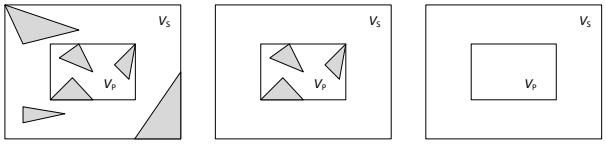
tested for intersection). Such ADSs can be subdivided into three categories:

1. *Spatial partitioning schemes* (recursively) subdivide a given space into spatially disjoint groups. This makes an efficient ray traversal possible at the expense of referencing geometric primitives multiple times.
2. *Object partitioning schemes* (recursively) subdivide a given set of geometric primitives into disjoint groups which tightly comprise their geometric primitives. Geometric primitives are referenced just once at the expense of a less efficient ray traversal in case of spatially overlapping groups.
3. *Hybrid partitioning schemes* combine both spatial and object partitioning schemes.

The most effective (i.e. largest reduction in intersection tests) ADSs are hierarchical, adaptive tree structures of which the leaf nodes reference the geometric primitives and the internal nodes contain spatial information (e.g., splitting plane position, bounding box) to cull the associated part of the scene. Generally, all these trees can be *n*-ary. In the remaining part of this paper, we will mainly focus on binary trees (due to the involved build heuristics) for 3D scenes where the bounding volumes associated with each node are axis-aligned bounding boxes (AABBs). For an extensive overview of ADSs, see Wald et al. [WMG*09].

Binary Space Partitions (BSPs) [BF79, Kap85, IWP08] and *Bounding Volume Hierarchies* (BVHs) [RW80] are the archetypical spatial and object partitioning scheme, respectively. *Spatial Kd-trees* (SKds) [OMR87, HHS06] a.k.a. *Bounded Kd-trees* (B-Kds) [WMS06] a.k.a. *Bounding Interval Hierarchies* (BIHs) [WK06] are similar to BVHs but only one out of six (instead of all six) planes of the child's bounding boxes, referred to as *voxels*, is tight. Child voxels are therefore characterized by their own splitting plane. BIHs trade-off the efficient spatial traversal of BSPs for the fixed and known in advance memory footprint of BVHs. Spatial Split Bounding Volume Hierarchies (SBVHs) [SFD09] allow spatial splits and thus the possibility of geometric primitives straddling the splitting plane in the BVH construction. Both BVH and BSP candidate partitions are considered. The BSP candidate partitions are made tight but constrained by the splitting plane to ensure a fair comparison with the same cost metric against BVH candidate partitions. Trees considering only such BSP candidate partitions are called GK-BVHs [PGDS09].

ADS build. The *Surface Area Heuristic* (SAH) is the de facto standard for generating qualitative tree ADSs. It is originally introduced for BVHs [GS87] and BSPs/kd-trees (axis-aligned BSPs a.k.a. rectilinear BSPs) [MB90] and is trivially extensible to other structures such as BIHs [HHS06, WMS06] and SBVHs [SFD09]. The trees are constructed top-down by greedily minimizing the cost of candidate partitions until some termination criterion is met. Each candidate child voxel is assigned a cost equal to the product of the cost of processing this child voxel and the probability of actually having to process this child voxel. In order to compute these probabilities for each candidate child voxel, the SAH assumes a scene-exterior isotropic ray distribution of infinitely long rays, irrespective of any intersections with intervening geometric primitives that might terminate rays in an actual ray tracing algorithm.



(a) global RT **(b)** local RT **(c)** no RT

Figure 1: ***(a)*** *Global ray termination considers all the geometry contained in the scene's voxel, V_S.* ***(b)*** *The RTSAH includes local ray termination which only considers the geometry contained in the voxel, V_P, to partition.* ***(c)*** *The SAH does not include ray termination.*

Several improvements have been made to the construction of BSPs with the SAH such as an efficient automatic termination criterion and clipping algorithm [HB02], favouring empty space cut-offs [HKRS02, WH06], and mail-boxing [HM08]. Furthermore, new build heuristics, targeting more common ray distributions, are introduced to obtain a more accurate cost model since the assumptions of the SAH do not hold well in practice: The *Scene-Interior Ray Origins Heuristic* (SIROH) [FFD09] assumes ray origins to be uniformly distributed inside the scene's volume. Havran and Bittner [HB99] consider the ray distribution for a single fixed ray origin or direction without taking the presence of geometry into account. Various blocking factors include *local ray termination* (i.e. considering only the scene's geometry inside the voxel to partition) [Hav00, RKJ96, MBD15, Mou15]. The *Voxel Visibility Heuristic* [CCI12] considers the non-uniform ray distribution, independent of the camera, by implicitly approximating *global ray termination* (i.e. considering all the scene's geometry) in a preprocess before constructing the ADS. The *Ray Distribution Heuristics* (RDHs) [BH09] subsample the actual ray distribution before building the ADS. Assuming significant frame-to-frame coherence, Vinkler et al. [VHS12] modify the SAH for BVHs by differentiating between potentially visible and invisible geometric primitives. Aila et al. [AKL13] introduce additional quality metrics to the SAH and use these to compare various (S)BVH build algorithms in the context of both scalar and SIMD execution of the ray tracing.

The difference between global, local and no ray termination is illustrated in Figure 1 in terms of the scene's geometry involved for partitioning a voxel. During the ADS construction, all the scene's geometry inside the voxel to partition is readily available. Querying the exterior geometry efficiently via the ADS itself is not guaranteed to be possible since the ADS is only partially constructed. Therefore, one must rely on an additional preconstructed ADS [CCI12] (e.g., ADS of earlier frame(s) [BH09]). Other build heuristics relying on scene's geometry outside the voxel to partition such as SIROH [FFD09] and the *Voxel Volume Heuristic* [WGS04] for constructing kd-tree photon maps, avoid sampling this geometry by coarse approximations involving volumes instead of surface areas.

The construction and/or (partial) updating of the ADSs is usually done during a prerendering phase. All the aforementioned build heuristics use a greedy divide-and-conquer approach. When these local heuristics are extended to global heuristics which consider all possible candidate partitions at all levels of the tree simultane-

ously, the problem of building the optimal tree becomes *NP*-hard. For BSPs constructed with the SAH, only splitting planes at the boundaries of the geometric primitives need to be considered due to the monotonically increasing/decreasing cost function between these planes. For BVHs, one typically partitions based on the geometric primitives' centroids [Wal07]. Wald and Havran [WH06] introduced a robust sweeping plane algorithm for doing this in $\mathcal{O}(|G_s| \log |G_s|)$ time for kd-trees, with $|G_s|$ the total number of geometric primitives to partition. This full sweeping plane approach is not applicable to general build heuristics and SBVHs. An alternative is a (parallel) binned algorithm [HKRS02, HMS06, PGpSS06, WH06, SSK07, SFD09] which optionally switches to the full sweeping plane approach or its candidate partitions, based on the number of geometric primitives to partition. Here, the latter differentiates between a full sweeping plane approach and a (adaptive) binned approach considering the candidate partitions of a full sweeping plane approach.

Several improvements are introduced for constructing BVHs: *Early Split Clipping* (ESC) [EG07] and *Edge Volume Heuristic* (EVH) [DK08] split the AABBs and geometric primitives, respectively, on a per-primitive basis in an additional preprocess. Ganestam and Doggett [GD16] rely on the SAH to decide on the splitting of geometric primitives in an additional preprocess (as opposed to splitting during the ADS construction itself [SFD09]). Stochastic search methods [NT03] and bottom-up clustering [WBKP08] are used for constructing BVHs. After construction, Kensler [Ken08] further reduces the total SAH cost via local adjustments (tree rotations). Bittner et al. [BHH13] remove expensive nodes and reinsert their child nodes.

ADS traversal. For *closest-hit* ray queries (e.g., camera, indirect), the closest intersection point with the scene's geometry needs to be found. Different traversal algorithms exist for tracing these rays in front-to-back order through kd-trees [HH11]. For BVHs, the voxel traversal order is typically decided heuristically [BH06].

Any-hit or *shadow* ray queries resolve occlusion and thus do not need to be traced in front-to-back order through the ADS. By changing the voxel traversal order, shadow ray traversal can be accelerated: The *Ray Termination Surface Area Heuristic* (RTSAH) [IH11] (based on [Hav00]) and the Surface Area Traversal Order (NodeSATO) [NM14] determine the traversal order of the child voxels for shadow rays in BVHs/BSPs constructed with the SAH by considering local ray termination and the child voxel's surface area, respectively. The *Shadow Ray Distribution Heuristic* (SRDH) [FLF12] is similar to the RDH but subsamples only the actual shadow ray distribution for constructing a dedicated BVH for shadow ray traversal. Ogaki and Derouet-Jourdan [ODJ15, ODJ16] also use this actual shadow ray distribution for sorting the child nodes for given *n*-ary BVHs.

Contribution. Our goal is not to radically change and replace the SAH, but to adapt it by introducing the concept of local ray termination in the context of voxel partitioning during the ADS construction and voxel traversal order during ADS traversal. Assuming a parent-exterior isotropic ray distribution (as for the SAH), including local ray termination in the SAH is a natural optimization and generalization affecting both the cost computation by taking (local) occlusion into account and the termination criterion.

Our previous work [MBD15] introduced two practical blocking factor approximations based on rasterization (APOD) and the average projected surface area (APSA) of the geometric primitives, to build kd-trees with the RTSAH [IH11]. In this paper, we further analyse and build upon this work, and introduce various new heuristics to approximate local ray termination efficiently without requiring additional preprocessing or ray (sub)sampling as opposed to [Hav00, CCI12, BH09, VHS12, FLF12, ODJ15, ODJ16]. Our heuristics are used for approximating:

- The visibility probabilities in the RTSAH for constructing BSPs, BIHs and (S)BVHs for accelerating closest-hit ray queries;
- The hit probabilities in the SRDH for constructing dedicated BVHs for accelerating any-hit ray queries.

Moulin et al. [MBD15] only consider a parent-exterior isotropic ray distribution for closest-hit ray queries for constructing kd-trees (BSPs). We introduce a novel extension and generalization to BIHs using explicit form factors and a novel extension and generalization to BVHs and SBVHs using a Markov Transfer chain approach. We thoroughly analyse the parent-exterior isotropic ray distribution for both closest-hit and any-hit ray queries. Therefore, we can not only make connections between the SAH and RTSAH cost model, but between the (RT)SAH and SRDH cost model as well (see Section 3). Furthermore, we are the first (to our knowledge) to study and build dedicated tree ADSs for accelerating any-hit ray queries by using a cost function that does not require ray sampling.

Instead of only considering the surface area domain [MBD15], we include both the surface area (or AABB) and projected hemispherical domain in our visibility (or hit ability) heuristics. Furthermore, we define a generalized visibility and dual hit probability (see Section 4) connecting the high-level cost models with their low-level approximations. All our and [MBD15] approximations are derived from this generalized visibility probability, concretizing both the underlying assumptions and intuitions of these approximations (see Section 5).

3. Theory

ADSs exploit spatial coherence by distributing the scene's geometric primitives into spatial groups. To construct qualitative tree ADSs, splitting planes that minimize the per-ray cost (for some ray distribution) need to be chosen at each level of the tree construction. Given a position of the splitting plane, S, the left and right child partition are uniquely determined for spatial partitioning schemes. To assure this for object partitioning schemes as well, a criterion [KK86] is required to partition geometric primitives with regard to a fixed splitting plane. In practice, one typically partitions based on the geometric primitives' centroids (i.e. largest surface area contribution) [WK06, Wal07].

For computational efficiency, tree ADS construction is usually achieved by using appropriate (greedy) cost heuristics. We briefly recall the de facto standard SAH [GS87, MB90] for constructing ADSs (Subsection 3.1). Furthermore, we adapt the original RTSAH [IH11] for constructing BSPs, BIHs and (S)BVHs (Subsec-

Notation	Description
V_S, V_P, V_L, V_R	Scene, parent, left and right child voxel
B_i	AABB associated with voxel i
G_i	Set of geometric primitives contained in voxel i
SA_i	Surface area of voxel/polygon/plane i
$C_i(\{V_L, V_R\}(S, V_P))$	Cost function i for partitioning a parent voxel, V_P, into a left, V_L, and a right, V_R, child voxel with a splitting plane, S
p_L, p_R	Piercing probabilities of the SAH (3.2)–(3.3)
p_{jL}, p_{jR}, $p_{L\leftrightarrow R}$, $p_{j\epsilon}$, p_{fL}, p_{fR}, $p_{L\to R}$, $p_{R\to L}$	Piercing probabilities of the RTSAH (3.5)–(3.12)
$\mathcal{V}_{L\to R}$, $\mathcal{V}_{R\to L}$	Visibility probabilities of the RTSAH (3.16)–(3.17)
p_{vL}, p_{vR}	Combined piercing and visibility probabilities of the RTSAH (3.19)–(3.20)
$\mathcal{H}_L$, $\mathcal{H}_R$	Hit probabilities of the SRDH (3.24)–(3.25)
T^k	Transfer matrix of grid cell, k (4.13)
$\mathcal{F}_{\perp,j\to i}$, $\mathcal{F}_{\parallel,j\to i}$	Orthogonal and parallel plane-to-plane form factors (A.2)–(A.3)
A	Surface area domain
B	AABB area domain
$\Omega^\perp$	Projected hemispherical domain
$<A>$	Average projected surface area domain
$\mathcal{V}(A,G)$, $\mathcal{V}_i(A,G)$	Generalized visibility probability (4.1) and approximation i (see Section 5)
$\mathcal{V}_\perp(A,G)$	Orthogonal visibility probability (5.3)
$\mathcal{V}\Omega^\perp(A,G)$	Projected hemispherical visibility probability (5.10)
$\mathcal{H}(B,G)$, $\mathcal{H}_i(B,G)$	Generalized hit probability (4.2) and approximation i (see Section 5)
Pr(.)	Probability of an event
$\lvert . \rvert$	Size of a set/domain
$\vec{\omega}$, $\hat{\omega}$	Vector and unit length vector
$\widetilde{A}$	Discretization of the domain A
$\widetilde{G}$	Approximation of the set G

Table 1: *Symbols and operators used throughout this paper.*

Acronym	Description
ADS	Acceleration data structure
AABB	Axis-aligned bounding box
Acceleration data structures (see Section 2)	
BSP	Binary Space Partition
BIH	Bounding Interval Hierarchy
(S)BVH	(Spatial Split) Bounding Volume Hierarchy
Cost models and voxel traversal order heuristics	
SAH	Surface Area Heuristic (3.1)
RTSAH	Ray Termination SAH (3.4)
SATO	Surface Area Traversal Order
RDH	Ray Distribution Heuristic
SRDH	Shadow RDH (3.23)
Generalized visibility and hit probability approximations	
MC	Monte Carlo (5.1), (5.16)
APAD	All Points All Directions (5.2)
APOD	All Points One Direction (5.4)
EP(2)	Ellipsoid Parallel projection (2) (5.6)
APSA	Average Projected Surface Area (5.8), (5.19)
APE(2)	Average Projected Ellipsoid (2) (5.8), (5.19)
OPAD	One Point All Directions (5.11)
(WC)HAPSA	(Weighted Centroid) Hybrid APSA (5.14)
HAPE(2)	Hybrid APE (2) (5.14)
(WN(WC))HANPSA(2)	(Weighted Normal (Weighted Centroid)) Hybrid Average Normal Projected Surface Area (2) (5.15)

Table 2: *Frequently used acronyms throughout this paper.*

tion 3.2), and the original SRDH [FLF12] for constructing dedicated BVHs for shadow ray traversal without depending on additional preprocessing or ray (sub)sampling (Subsection 3.3). Table 1 summarizes our notation and Table 2 summarizes our frequently used acronyms.

3.1. SAH

The SAH cost function, $\mathcal{C}_{SAH}$ [GS87,MB90], estimates the average ray cost of partitioning a parent voxel, V_P, into a left, V_L, and right, V_R, child voxel with a splitting plane, S, as the cost of processing these child voxels, $|G_L|$ and $|G_R|$ (the number of geometric primitives contained in the left and right child voxel, respectively), times the probability of having to process these child voxels, p_L and p_R:

$$\mathcal{C}_{SAH}(\{V_L, V_R\}(S, V_P)) \coloneqq c + p_L|G_L| + p_R|G_R|. \tag{3.1}$$

Here, c is a constant expressing the relative cost of performing an intermediate node traversal versus a ray-geometric primitive intersection test. Assuming a scene-exterior isotropic ray distribution of non-terminating rays (and thus also a parent-exterior isotropic ray distribution), then the probabilities, p_L and p_R, are defined as:

$$p_L \coloneqq \Pr(V_L \text{ pierced} \,|\, V_P \text{ pierced}) = \frac{SA_L}{SA_P} \tag{3.2}$$

$$p_R \coloneqq \Pr(V_R \text{ pierced} \,|\, V_P \text{ pierced}) = \frac{SA_R}{SA_P}, \tag{3.3}$$

where SA_i is the surface area of V_i's bounding box. The chance of hitting a convex solid (e.g., AABB) given these assumptions is proportional to its average projected surface area, which for 3D convex solids is equal to 1/4 of their surface area [Cau41,Cau50,CS97]. Alternatively, the former equations can also be obtained via the form factors (a.k.a. configuration and view factors) used in the radiative heat transfer [Hot31] and radiosity literature [CWH93, SP94].

3.2. RTSAH

The RTSAH cost function, $\mathcal{C}_{RTSAH}$ [IH11, MBD15], extends the SAH by including local ray termination by means of the probabilities, $\mathcal{V}_{L\to R}$ and $\mathcal{V}_{R\to L}$. It differentiates, p_L and p_R, in rays that pierce just one child voxel, p_{jL} and p_{jR}, and rays that pierce both child voxels, $p_{L\leftrightarrow R}$. In the latter case, rays could first pierce the left (right) child voxel, $p_{L\to R}$ ($p_{R\to L}$), where p_{fL} (p_{fR}) denotes the probability of piercing the left (right) child voxel first, given that both child voxels are pierced:

$$\begin{aligned}\mathcal{C}_{RTSAH}(\{V_L, V_R\}(S, V_P)) \coloneqq c &+ (p_{jL} + p_{L\to R} + p_{R\to L}\mathcal{V}_{R\to L})\,|G_L| \\ &+ (p_{jR} + p_{R\to L} + p_{L\to R}\mathcal{V}_{L\to R})\,|G_R|.\end{aligned} \tag{3.4}$$

For completeness, $p_{j\epsilon}$ represents the fraction of rays piercing no child voxels, which can be nonzero for BIHs and (S)BVHs.

The *piercing probabilities* are defined as follows:

$$p_{jL} \coloneqq \Pr(\ V_L \wedge \neg V_R \text{ pierced} \,|\, V_P \text{ pierced}) \tag{3.5}$$

$$p_{jR} \coloneqq \Pr(\neg V_L \wedge \ V_R \text{ pierced} \,|\, V_P \text{ pierced}) \tag{3.6}$$

$$p_{L\leftrightarrow R} \coloneqq \Pr(\ V_L \wedge \ V_R \text{ pierced} \,|\, V_P \text{ pierced}) \tag{3.7}$$

$$p_{j\epsilon} \coloneqq \Pr(\neg V_L \wedge \neg V_R \text{ pierced} \,|\, V_P \text{ pierced}) \tag{3.8}$$

$$p_{fL} \coloneqq \Pr(V_L \text{ pierced first} \,|\, V_L \wedge V_R \text{ pierced} \,|\, V_P \text{ pierced}) \tag{3.9}$$

$$p_{fR} \coloneqq \Pr(V_R \text{ pierced first} \,|\, V_L \wedge V_R \text{ pierced} \,|\, V_P \text{ pierced}) \tag{3.10}$$

$$p_{L\to R} \coloneqq p_{L\leftrightarrow R} p_{fL} \tag{3.11}$$

$$p_{R\to L} \coloneqq p_{L\leftrightarrow R} p_{fR}. \tag{3.12}$$

Furthermore, these probabilities satisfy the following relations:

$$p_{jL} + p_{jR} + p_{L\leftrightarrow R} + p_{j\epsilon} = 1 \tag{3.13}$$

$$p_{jL} + p_{L\leftrightarrow R} = p_L \tag{3.14}$$

$$p_{jR} + p_{L\leftrightarrow R} = p_R. \tag{3.15}$$

Here, we only consider local ray termination (see Figure 1). Hence, the scene-exterior isotropic ray distribution also corresponds to a parent-exterior isotropic ray distribution for each voxel that needs

to be partitioned as is the case for the SAH. This implies that the same evaluation of p_{L} (3.2) and p_{R} (3.3) as for the SAH can be used for the RTSAH, and $p_{\mathrm{fL}} \equiv p_{\mathrm{fR}} \equiv 1/2$. Considering global ray termination is more complex since the geometry surrounding the voxel to partition is not readily available during ADS construction and no tractable closed form expressions of the form factors for computing and/or updating the piercing probabilities exist. Since the SAH has proven very well in practice, we do not want to radically change and replace, but rather adapt it. Therefore, only local ray termination is included.

The *visibility probabilities* are defined as:

$$\mathcal{V}_{\mathrm{L}\to\mathrm{R}} := \Pr(\text{no hit in } V_{\mathrm{L}} \,|\, V_{\mathrm{L}} \text{ pierced first} \,|\, V_{\mathrm{L}} \wedge V_{\mathrm{R}} \text{ pierced} \,|\, V_{\mathrm{P}} \text{ pierced}) \quad (3.16)$$

$$\mathcal{V}_{\mathrm{R}\to\mathrm{L}} := \Pr(\text{no hit in } V_{\mathrm{R}} \,|\, V_{\mathrm{R}} \text{ pierced first} \,|\, V_{\mathrm{L}} \wedge V_{\mathrm{R}} \text{ pierced} \,|\, V_{\mathrm{P}} \text{ pierced}). \quad (3.17)$$

Note that if $\mathcal{V}_{\mathrm{L}\to\mathrm{R}} = \mathcal{V}_{\mathrm{R}\to\mathrm{L}} = 1$, then $\mathcal{C}_{\mathrm{RTSAH}} = \mathcal{C}_{\mathrm{SAH}}$ and if $\mathcal{V}_{\mathrm{L}\to\mathrm{R}} = \mathcal{V}_{\mathrm{R}\to\mathrm{L}} = 0$, then $\mathcal{C}_{\mathrm{RTSAH}} = \mathcal{C}_{\mathrm{SAH}} - 1/2\; p_{\mathrm{L}\leftrightarrow\mathrm{R}}\,(|G_{\mathrm{L}}| + |G_{\mathrm{R}}|)$ for BSPs.

The RTSAH [IH11] is originally designed for determining the voxel traversal order in BSPs and BVHs for tracing shadow rays. The visibility of non-empty leaf nodes is assumed to be completely opaque and the visibility of intermediate nodes is recursively obtained from its child nodes via a separate pass after finishing the ADS construction. In order to guide the ADS build with the RTSAH, the visibility probabilities need to be evaluated during the construction process itself. For these probabilities, no tractable closed form expression exist and ray (sub)sampling is expensive. Therefore, we introduce in Section 5 different visibility probability approximations which trade off accuracy for efficiency.

Depending on the type of ADS, it can be computationally expensive to compute all probabilities involved in Equation (3.4) explicitly. Therefore, we also define the RTSAH in its implicit form which is very similar to the SAH (3.1):

$$\mathcal{C}_{\mathrm{RTSAH}}(\{V_{\mathrm{L}}, V_{\mathrm{R}}\}(S, V_{\mathrm{P}})) = \mathrm{c} + p_{\mathrm{vL}}|G_{\mathrm{L}}| + p_{\mathrm{vR}}|G_{\mathrm{R}}|. \quad (3.18)$$

Here, the *combined piercing and visibility probabilities*, p_{vL} and p_{vR}, express the probability of a ray traversing the left and right child voxel, respectively, implicitly taking local ray termination into account:

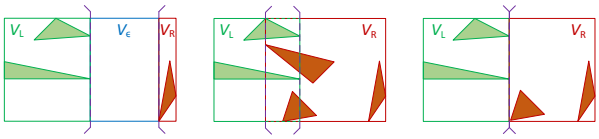
$$p_{\mathrm{vL}} := \Pr(V_{\mathrm{L}} \text{ traversed} \,|\, V_{\mathrm{P}} \text{ pierced}) \quad (3.19)$$

$$p_{\mathrm{vR}} := \Pr(V_{\mathrm{R}} \text{ traversed} \,|\, V_{\mathrm{P}} \text{ pierced}). \quad (3.20)$$

Note that both the SAH (3.1) and the RTSAH in its implicit form (3.18) can be trivially extended to construct n-ary ADS trees, which is not the case for the RTSAH in its explicit form (3.4).

RTSAH for BSPs. In BSPs, the spatial union of both child voxels is equal to the parent voxel. So every ray piercing the parent voxel, has to pierce at least one child voxel, $p_{\mathrm{j\epsilon}} \equiv 0$, and all piercing probabilities can be trivially deduced from Equations (3.13)–(3.15). For $\mathcal{V}_{\mathrm{L}\to\mathrm{R}}$ ($\mathcal{V}_{\mathrm{R}\to\mathrm{L}}$), one needs to consider only the clipped geometry to the left (right) of the splitting plane contained inside the parent voxel, since geometric primitives are not necessarily fully contained in a single child and/or the parent voxel.

RTSAH for BIHs. We differentiate between three possible configuration types of BIH candidate partitions (see Figure 2): both child voxels do not overlap (type I), overlap strictly (type II), or overlap but not strictly (type III) (i.e. similar to a BSP). Obtaining the piercing probabilities in case of overlap (types II and III) is analogous to BSPs since $p_{\mathrm{j\epsilon}} = 0$. In case of no overlap, $p_{\mathrm{j\epsilon}}$ can be expressed in terms of the form factors between the planes of V_ϵ's AABB, which comprises the empty volume between both child voxels (see the Appendix):



(a) Type I **(b)** Type II **(c)** Type III

Figure 2: *Three possible configuration types of BIH candidate partitions: both child voxels do* ***(a)*** *not overlap (type I),* ***(b)*** *overlap strictly (type II), or* ***(c)*** *overlap but not strictly (i.e. touch) (type III).*

$$p_{\mathrm{j\epsilon}} = 2\Big(\big(\mathcal{F}_{\parallel,\text{xy-plane}\to\text{xy-plane}} + 2\mathcal{F}_{\perp,\text{xy-plane}\to\text{xz-plane}}\big)\, p_{\text{xy-plane}} + \big(\mathcal{F}_{\parallel,\text{xz-plane}\to\text{xz-plane}} + 2\mathcal{F}_{\perp,\text{xz-plane}\to\text{xy-plane}}\big)\, p_{\text{xz-plane}}\Big). \quad (3.21)$$

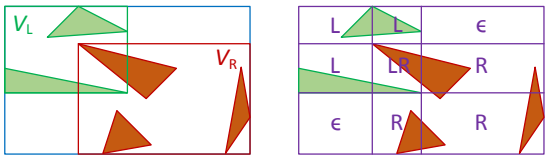
Here, we assume both splits are performed along the x-axis (analogous for the y- and z-axis) and the probabilities, $p_{ij\text{-plane}}$ with $ij \in \{\mathrm{xy}, \mathrm{xz}, \mathrm{yz}\}$, are defined and evaluated as:

$$p_{ij\text{-plane}} := \Pr(ij\text{-plane pierced from the outside} \,|\, V_{\mathrm{P}} \text{ pierced}) = \frac{\mathrm{SA}_{ij\text{-plane}}}{\mathrm{SA}_{\mathrm{P}}}. \quad (3.22)$$

Note that $\mathrm{SA}_{ij\text{-plane}}$ refers to the surface area of one side only. Both sides constitute a convex solid, but only half of the rays are considered (i.e. piercing from the outside). For $\mathcal{V}_{\mathrm{L}\to\mathrm{R}}$ ($\mathcal{V}_{\mathrm{R}\to\mathrm{L}}$), one needs to consider only the clipped geometry to the left (right) of the right (left) child voxel's associated splitting plane. The geometry contained in the overlapping volume does not contribute to the local ray termination. If a ray is not terminated before reaching this volume while traversing one child voxel, the other child voxel always needs to be traversed as well.

RTSAH for (S)BVHs. (S)BVH candidate partitions could appear in numerous configuration types due to the tightness of the six planes of both child voxels as opposed to BSP (zero tight planes in general) and BIH (one tight plane in general) candidate partitions. Furthermore, we have no explicit splitting planes as opposed to BSPs and BIHs. Enumerating these configurations and handling each one separately as for the BIH is cumbersome. Instead, we construct an irregular grid containing our candidate partition as shown in Figure 3. The grid resolution varies between $1\times1\times1$ and $3\times3\times3$ grid cells.

Each grid cell is assigned a 6×6 transfer matrix, T, expressing the transmission of ray distributions from its incoming to its outgoing planes. The entries, T_{ij}, thus correspond to the actual form factors taking the geometry contained in the parent voxel into account while assuming a parent-exterior isotropic ray distribution. For all grid planes coinciding with the parent voxel, the incoming ray fractions (from the outside) can be evaluated with Equation (3.22). A



(a) BVH candidate partition **(b)** Irregular grid

Figure 3: ***(a)*** *BVH candidate partition with the left and right child voxel's planes tight in all directions.* ***(b)*** *Irregular grid containing the BVH candidate partition with grid cells tagged as belonging to the left child voxel (L), right child voxel (R), both child voxels (LR) or no child voxel (ϵ).*

Markov chain [Mar13, Mar06, Win04] can be used to transmit the incoming ray fractions over the grid according to the transfer matrices of the grid cells until no non-terminated rays remain in the grid. During this process, the accumulated fraction of rays entering our left and right child voxel is tracked ($p_{\nu L}$ (3.19) and $p_{\nu R}$ (3.20)). Finally, the RTSAH cost (3.18) of the given candidate partition can be evaluated. Note that this procedure is generally applicable to all kinds of ADSs including BSPs and BIHs.

3.3. SRDH

When only constant kernels (*left-child-first* or *right-child-first*) are considered for each candidate partition, the SRDH [FLF12] cost function, $\mathcal{C}_{\text{SRDH}}$, assuming a parent-exterior isotropic ray distribution, is defined for BVHs as:

$$\mathcal{C}_{\text{SRDH}}(\{V_L,V_R\}(S,V_P)) := \max\big(p_L\mathcal{H}_L(c_{\text{AABB}}+p_R|G_R|),\; p_R\mathcal{H}_R(c_{\text{AABB}}+p_L|G_L|)\big). \quad (3.23)$$

Here, the hit probabilities, $\mathcal{H}_L$ and $\mathcal{H}_R$, define the fraction of shadow rays hitting some geometric primitive inside the left and right child voxel, respectively:

$$\mathcal{H}_L := \Pr\big(\text{hit in } G_L \,|\, V_L \text{ pierced} \,|\, V_P \text{ pierced}\big) \quad (3.24)$$

$$\mathcal{H}_R := \Pr\big(\text{hit in } G_R \,|\, V_R \text{ pierced} \,|\, V_P \text{ pierced}\big). \quad (3.25)$$

G_L and G_R are the sets of geometric primitives contained in V_L and V_R, respectively. c_{AABB} is the relative cost of testing an AABB [WBMS05] versus a geometric primitive for intersection. Candidate partitions with the highest cost are favoured and their constant traversal kernel is chosen accordingly when applied. Furthermore, we also consider SRDH′, a derivative of SRDH assuming worst-case behaviour ($p_L \equiv p_R \equiv 1$):

$$\mathcal{C}'_{\text{SRDH}}(\{V_L,V_R\}(S,V_P)) := \max\big(\mathcal{H}_L(c_{\text{AABB}}+|G_R|),\; \mathcal{H}_R(c_{\text{AABB}}+|G_L|)\big). \quad (3.26)$$

4. Local Ray Termination

In this section, we generalize the (plane) visibility probabilities, $\mathcal{V}_{L\to R}$ (3.16) and $\mathcal{V}_{R\to L}$ (3.17), and the dual hit probabilities, $\mathcal{H}_L$ (3.24) and $\mathcal{H}_R$ (3.25), (Subsection 4.1). Next, we describe how our generalized visibility probability can be used for constructing BSPs (Subsection 4.2), BIHs (Subsection 4.3) and (S)BVHs (Subsection 4.4) with the RTSAH, and how our generalized hit probability can be used for constructing BVHs (Subsection 4.5) with the SRDH.

4.1. Generalized Visibility and Hit Probability

The *generalized visibility probability*, $\mathcal{V}$, over a surface area domain, A, taking a set of geometric primitives, G, into account, is defined as:

$$\mathcal{V}(A,G) := 1 - \frac{1}{|A|\pi}\int_A\int_{\Omega^\perp}\mathcal{H}(\text{ray}(x,\widehat{\omega}),G)\,\mathrm{d}A_x\,\mathrm{d}\Omega^\perp_{\widehat{\omega}}, \quad (4.1)$$

where 'ray' represents a ray composed of a possible endpoint (on the surface area domain, A) and a unit length direction (parametrized in the projected hemisphere of incoming directions, $\Omega^\perp$, at the endpoint, x), and $\mathcal{H}$ is defined as the binary hit function in terms of a ray and a set of geometric primitives (i.e. evaluates to 1 if the ray hits at least one of the geometric primitives, 0 otherwise). Note that if all geometric primitives are double-sided (i.e. no back-face or front-face culling), the direction of the rays can be inverted and the possible endpoints become the actual origins of the rays: if a ray is (not) terminated before reaching A, an inverted, outgoing ray will (not) terminate as well (either).

The *generalized hit probability*, $\mathcal{H}$, over an AABB area domain, B, (i.e. the union of six surface area domains), taking a set of geometric primitives, G, into account, is defined as:

$$\mathcal{H}(B,G) := \frac{1}{|B|\pi}\int_B\int_{\Omega^\perp}\mathcal{H}(\text{ray}(x,\widehat{\omega}),G)\,\mathrm{d}B_x\,\mathrm{d}\Omega^\perp_{\widehat{\omega}}. \quad (4.2)$$

In Section 5, we will propose computationally efficient heuristics to approximate these generalized visibility and hit probability.

4.2. RTSAH for BSPs

For constructing BSPs with the RTSAH, our visibility probabilities, $\mathcal{V}_{L\to R}$ (3.16) and $\mathcal{V}_{R\to L}$ (3.17), are expressed as follows in terms of our generalized visibility probability, $\mathcal{V}$ (4.1):

$$\mathcal{V}_{L\to R} = \mathcal{V}\big(S^-,\text{clip}(B_L,G_P)\big) = \mathcal{V}\big(S^-,\text{clip}(B_L,G_L)\big) \quad (4.3)$$

$$\mathcal{V}_{R\to L} = \mathcal{V}\big(S^+,\text{clip}(B_R,G_P)\big) = \mathcal{V}\big(S^+,\text{clip}(B_R,G_R)\big), \quad (4.4)$$

where S^- (S^+) represents the surface area domain corresponding to the negative (positive) side of the splitting plane, S; B_i the AABB and G_i the set of geometric primitives associated with V_i for $i \in \{P,L,R\}$. 'clip' represents the clipping of a set of geometric primitives against an AABB. Instead of clipping against an AABB, the rays themselves can alternatively be constrained against that AABB.

4.3. RTSAH for BIHs

For constructing BIHs with the RTSAH, our visibility probabilities, $\mathcal{V}_{L\to R}$ (3.16) and $\mathcal{V}_{R\to L}$ (3.17), are expressed as follows in terms of our generalized visibility probability, $\mathcal{V}$ (4.1), in case of non-strict overlap (i.e. $S_L \equiv S_R \equiv S$; type III see Figure 2):

$$\mathcal{V}_{L\to R} = \mathcal{V}\big(S^-,G_L\big) \quad (4.5)$$

$$\mathcal{V}_{R\to L} = \mathcal{V}\big(S^+,G_R\big), \quad (4.6)$$

in case of strict overlap (type II see Figure 2):

$$\mathcal{V}_{\mathrm{L}\to\mathrm{R}}=\mathcal{V}\left(S_{\mathrm{R}}^{-},\mathrm{clip}(B_{\mathrm{P}}\setminus B_{\mathrm{R}},G_{\mathrm{L}})\right) \tag{4.7}$$

$$\mathcal{V}_{\mathrm{R}\to\mathrm{L}}=\mathcal{V}\left(S_{\mathrm{L}}^{+},\mathrm{clip}(B_{\mathrm{P}}\setminus B_{\mathrm{L}},G_{\mathrm{R}})\right), \tag{4.8}$$

and in case of no overlap (type I see Figure 2):

$$\mathcal{V}_{\mathrm{L}\to\mathrm{R}}=1-\frac{1}{|S_{\mathrm{R}}^{-}|}\int_{S_{\mathrm{R}}^{-}}\frac{1}{|\Omega_{\mathrm{L}}^{\perp}(x)|}\int_{\Omega_{\mathrm{L}}^{\perp}(x)}\mathcal{H}(\mathrm{ray}(x,\widehat{\omega}),G_{\mathrm{L}})\,\mathrm{d}A_x\,\mathrm{d}\Omega_{\widehat{\omega}}^{\perp} \tag{4.9}$$

$$\mathcal{V}_{\mathrm{R}\to\mathrm{L}}=1-\frac{1}{|S_{\mathrm{L}}^{+}|}\int_{S_{\mathrm{L}}^{+}}\frac{1}{|\Omega_{\mathrm{R}}^{\perp}(x)|}\int_{\Omega_{\mathrm{R}}^{\perp}(x)}\mathcal{H}(\mathrm{ray}(x,\widehat{\omega}),G_{\mathrm{R}})\,\mathrm{d}A_x\,\mathrm{d}\Omega_{\widehat{\omega}}^{\perp}. \tag{4.10}$$

Here, $\Omega_{\mathrm{L}}^{\perp}(x)$ ($\Omega_{\mathrm{R}}^{\perp}(x)$) is defined as the subdomain of $\Omega^{\perp}$ that restricts the possible ray directions for a possible endpoint, x, to those piercing B_{L} (B_{R}) (i.e. rays that pierce both S_{L}^{-} and S_{R}^{-} (S_{L}^{+} and S_{R}^{+})). For our non-ray-tracing based approximations in case of no overlap, we approximate our visibility probabilities, $\mathcal{V}_{\mathrm{L}\to\mathrm{R}}$ (3.16) and $\mathcal{V}_{\mathrm{R}\to\mathrm{L}}$ (3.17), in terms of our generalized visibility probability, $\mathcal{V}$ (4.1), as:

$$\mathcal{V}_{\mathrm{L}\to\mathrm{R}}\approx\mathcal{V}\left(S_{\mathrm{R}}^{-},G_{\mathrm{L}}\right) \tag{4.11}$$

$$\mathcal{V}_{\mathrm{R}\to\mathrm{L}}\approx\mathcal{V}\left(S_{\mathrm{L}}^{+},G_{\mathrm{R}}\right). \tag{4.12}$$

4.4. RTSAH for (S)BVHs

For (S)BVHs, we have no explicit splitting planes as opposed to BSPs and BIHs. As explained in the previous section, we can encapsulate our candidate partition in an irregular grid. Each grid cell, k, is assigned a 6×6 transfer matrix, T^k, expressing the transmission of incoming to outgoing ray distributions onto the planes of its AABB, B^k. Assuming a parent-exterior isotropic ray distribution, the entries, T_{ij}^k, correspond to the actual form factors taking the geometry, G_{P}, contained in the parent voxel, V_{P}, into account. We approximate the entries, T_{ij}^k, by splitting the actual form factors in a plane visibility, $\mathcal{V}$ (4.1), and a plane-to-plane form factor (A.2)–(A.3) taking no geometry into account:

$$T_{ij}^k\approx\begin{cases}0 & \text{if } i=j\\ \mathcal{V}\left(S_j^k,\mathrm{clip}\left(B^k,G_{\mathrm{P}}\right)\right)\mathcal{F}_{\parallel,j\to i} & \text{if } S_i^k\parallel S_j^k\\ \mathcal{V}\left(S_j^k,\mathrm{clip}\left(B^k,G_{\mathrm{P}}\right)\right)\mathcal{F}_{\perp,j\to i} & \text{if } S_i^k\perp S_j^k\end{cases}. \tag{4.13}$$

Here, S_j^k is one of the six inner planes of B^k : $\{\mathrm{yz}_{\min}^{+},\mathrm{yz}_{\max}^{-},\mathrm{xz}_{\min}^{+},\mathrm{xz}_{\max}^{-},\mathrm{xy}_{\min}^{+},\mathrm{xy}_{\max}^{-}\}$. G_{P} can be further reduced to G_{L}, G_{R} or $\emptyset$ based on the grid cell tag (see Figure 3b), if and only if G_{L} and G_{R} are completely known (which is not the case while restricting spatial split attempts in SBVHs [SFD09]). Furthermore, note that there are at most 9 (out of 36) unique plane-to-plane form factors for each grid cell.

If no geometry is contained in V_{P}, equation (4.13) is exact. After transmitting the incoming ray distributions over the grid according to the transfer matrices until no non-terminated ray distributions remain in the grid, the obtained $p_{\mathrm{vL}}\equiv p_{\mathrm{L}}$ and $p_{\mathrm{vR}}\equiv p_{\mathrm{R}}$. In the presence of geometry, the parent-exterior isotropic ray distribution is only valid for the exterior planes of the grid and is not maintained on the interior planes. To resolve this, one can trace (computationally expensive) individual rays, whereas we trace approximate ray distributions through the grid. Note that if we enumerate all (S)BVH configuration types, some (S)BVH configuration types can be resolved exactly via a similar form factor approach as for BIHs.

The transmission itself can be executed iteratively by keeping track of the current outgoing and incremental incoming ray distributions on each plane. Alternatively, a sparse linear system of equations can be solved by combining the transfer matrices for each grid cell into a single higher dimensional transfer matrix for the whole grid.

As an extension, the incoming ray distributions can be transmitted over a (ir)regular grid of the whole scene. Here, the grid resolution must be larger than $3\times3\times3$, introducing lots of extra interior planes. Furthermore, the transmission process is only executed once (versus once for every candidate partition), making the tracing of individual rays more feasible. When candidate splitting planes are only considered at the interior grid planes for constructing kd-trees, this approach boils down to a binned RDH [BH09]. Furthermore, both the surface area and projected hemispherical domain can be discretized to represent the non-uniform ray distribution more accurately in the presence of geometry and reflections using higher dimensional $n\times n$ transfer matrices for each grid cell. In the limit $n\to\infty$, these ray distributions become Dirac delta distributions (i.e. separate rays). The advantage of this ray distribution transmission approach over ordinary ray sampling, is the decoupling of the scene and camera (i.e. advantageous in an animation context): for a given arbitrary camera ray distribution, the invariant incoming ray distributions on all grid planes can be obtained using the transfer matrix of the whole grid.

4.5. SRDH for BVHs

For constructing dedicated BVHs for tracing shadow rays with the SRDH, our hit probabilities, $\mathcal{H}_{\mathrm{L}}$ (3.24) and $\mathcal{H}_{\mathrm{R}}$ (3.25), are expressed as follows in terms of our generalized hit probability, $\mathcal{H}$ (4.2):

$$\mathcal{H}_{\mathrm{L}}=\mathcal{H}\left(B_{\mathrm{L}},\mathrm{clip}(B_{\mathrm{L}},G_{\mathrm{L}})\right)=\mathcal{H}\left(B_{\mathrm{L}},G_{\mathrm{L}}\right) \tag{4.14}$$

$$\mathcal{H}_{\mathrm{R}}=\mathcal{H}\left(B_{\mathrm{R}},\mathrm{clip}(B_{\mathrm{R}},G_{\mathrm{R}})\right)=\mathcal{H}\left(B_{\mathrm{R}},G_{\mathrm{R}}\right). \tag{4.15}$$

5. Local Ray Termination Approximations

In order to obtain a practical binned RTSAH build algorithm, the splitting cost (3.4) (BSPs, BIHs) or (3.18) (BSPs, BIHs, (S)BVHs) needs to be evaluated for a finite set of candidate partitions for every voxel, V_{P}, to partition in $\mathcal{O}(|G_{\mathrm{P}}|)$ time. For the SRDH, a practical sweeping plane build algorithm can be used, requiring the splitting cost (3.23) (BVHs) to be updated incrementally in $\mathcal{O}(1)$ time (independent of the number of geometric primitives to partition) while sweeping the splitting plane.

The piercing probabilities (p_{jL} (3.5), p_{jR} (3.6), $p_{\mathrm{L}\leftrightarrow\mathrm{R}}$ (3.7) and $p_{\mathrm{j}\in}$ (3.8)) can be trivially computed for BSPs and BIHs (see Section 3), assuming a parent-exterior isotropic ray distribution. For evaluating the generalized visibility, $\mathcal{V}$ (4.1), and hit, $\mathcal{H}$ (4.2), probabilities, no tractable closed form expression can be found due to the dependence on the geometrical distribution inside the voxel to partition.

In this section, we propose computationally efficient heuristics

to approximate the generalized visibility probability, $\mathcal{V}$ (4.1), considering both the surface area domain, A, and projected hemispherical domain, $\Omega^{\perp}$, (Subsection 5.1), only the surface area domain, A, (Subsection 5.2), and only the projected hemispherical domain, $\Omega^{\perp}$, (Subsection 5.3). Furthermore, we propose computationally efficient heuristics to approximate the generalized hit probability, $\mathcal{H}$ (4.2), considering both the AABB area domain, B, and projected hemispherical domain, $\Omega^{\perp}$ (Subsection 5.4).

5.1. Surface Area Domain+Projected Hemispherical Domain

A *Monte Carlo* (MC) evaluation, $\mathcal{V}_{\text{MC}}$, of $\mathcal{V}$ (4.1) with N samples is defined as:

$$\mathcal{V}_{\text{MC}}(A,G) \coloneqq 1-(1/N)\sum_{(x,\widehat{\omega})\sim p(A)}^{N}\mathcal{H}(\text{ray}(x,\widehat{\omega}),G)\,, \tag{5.1}$$

where $\mathrm{d}p\,(A) \coloneqq \frac{\mathrm{d}A_x \mathrm{d}\Omega^{\perp}_{\widehat{\omega}}}{|A|\pi}$ represents the probability density function of the incoming isotropic ray distribution through A.

The regular discretization of both the surface area domain, $\overline{A}$, and projected hemispherical domain, $\overline{\Omega}^{\perp}$, is called our *All Points All Directions* (APAD) approximation [MBD15] and is defined as:

$$\mathcal{V}_{\text{APAD}}(A,G) \coloneqq 1-\frac{1}{|\overline{A}||\overline{\Omega}^{\perp}|}\sum_{x\in\overline{A}}\sum_{\widehat{\omega}\in\overline{\Omega}^{\perp}}\mathcal{H}(\text{ray}(x,\widehat{\omega}),G)\,. \tag{5.2}$$

Both approximations (5.1) and (5.2) involve the actual tracing of rays, making them impractical due to the (general) absence of an ADS during the ADS construction. Even if a preconstructed ADS is available, the total number of rays to trace to reach convergence for every candidate partition would be extremely high while ignoring the purpose of an ADS in the first place.

5.2. Surface Area Domain

The *orthogonal visibility probability*, $\mathcal{V}_{\perp}$, over a surface area domain, A, taking a set of geometric primitives, G, into account, is defined as:

$$\mathcal{V}_{\perp}(A,G) \coloneqq 1-\frac{1}{|A|}\int_{A}\mathcal{H}(\text{ray}(x,\widehat{\omega}_{A}),G)\,\mathrm{d}A_x, \tag{5.3}$$

where $\widehat{\omega}_{A}$ is the unit length direction orthogonal (or any other fixed unit length direction parametrized in $\Omega^{\perp}$) on A. Instead of approximating $\mathcal{V}$ (4.1) directly, we can assume $\mathcal{V}\approx\mathcal{V}_{\perp}$ by neglecting the hemispherical dependence and thus use $\mathcal{V}_{\perp}$ (5.3).

The regular discretization of the surface area domain, $\overline{A}$, is called our *All Points One Direction* (APOD) approximation [MBD15] and is defined as:

$$\mathcal{V}_{\text{APOD}}(A,G) \coloneqq 1-\frac{1}{|\overline{A}|}\sum_{x\in\overline{A}}\mathcal{H}(\text{ray}(x,\widehat{\omega}_{A}),G)\,. \tag{5.4}$$

The APOD approximation corresponds to an orthogonal projection of G onto A and can be evaluated via rasterization [MBD15].

Instead of using the original set of geometric primitives, G, we can use an approximation, $\widetilde{G}$:

$$\mathcal{V}_{\perp}(A,G) \approx 1-\frac{1}{|A|}\int_{A}\mathcal{H}\Big(\text{ray}(x,\widehat{\omega}_{A}),\widetilde{G}\Big)\,\mathrm{d}A_x. \tag{5.5}$$

Our *Ellipsoid Parallel projection* (EP) approximates the geometry, G, as an ellipsoid, $\widetilde{G}$, centred at the mean geometry vertex with axes equal to the standard deviation of the geometry vertices. Our *Ellipsoid Parallel projection 2* (EP2) approximates the geometry, G, as an ellipsoid, $\widetilde{G}$, centred at the median geometry vertex with axes equal to the median absolute deviation (MAD) of the geometry vertices multiplied by a constant, k. Assuming that the vertices are distributed normally, $k\approx 1.4826$ [Rup11] and relates the MAD to the standard deviation. MAD is a robust measure (i.e. less prone to outliers) as opposed to directly computing the standard deviation. The EP and EP2 approximations are defined as:

$$\mathcal{V}_{\text{EP(2)}}(A,G) \coloneqq 1-\min\left(\frac{\text{SA}_{\widetilde{G}}}{\text{SA}_{A}},1\right), \tag{5.6}$$

where SA represents the surface area; $|A|\equiv\text{SA}_{A}$ and $\text{SA}_{\widetilde{G}}$ corresponds to the surface area of the ellipse obtained after orthogonally projecting the ellipsoid, $\widetilde{G}$, onto the splitting plane, A. The min operator ensures that the visibility probability lies in $[0,1]$ which is not guaranteed due to the coarse approximations.

Cauchy [Cau41,Cau50,CS97] showed that the average projected surface area of a 3D convex solid is 1/4 of its surface area. Assuming the geometry, G, constitutes such a single convex solid, $\widetilde{G}$, we can approximate the orthogonal visibility probability by integrating over the average projected surface area domain, $<A>$, instead of the splitting plane's surface area domain, A:

$$\mathcal{V}_{\perp}(A,G) \approx 1-\frac{1}{|A|}\int_{<A>}\mathcal{H}\Big(\text{ray}(x,\widehat{\omega}_{A}),\widetilde{G}\Big)\,\mathrm{d}A_x. \tag{5.7}$$

Our *Average Projected Surface Area* (APSA) [MBD15] (as opposed to APOD (5.4)), our *Average Projected Ellipsoid* (APE) (as opposed to EP (5.6)) and our *Average Projected Ellipsoid 2* (APE2) (as opposed to EP2 (5.6)) approximations are defined as:

$$\mathcal{V}_{\text{APSA|APE(2)}}(A,G) \coloneqq 1-\min\left(\frac{1/4\ \text{SA}_{\widetilde{G}}}{\text{SA}_{A}},1\right), \tag{5.8}$$

where $\text{SA}_{\widetilde{G}}$ corresponds to the sum of the surface areas of the geometric primitives in G for the APSA approximation and the surface area of the ellipsoid for the APE and APE2 approximations. Since no elementary formula exist for the surface area of an ellipsoid, $\widetilde{G}$, we use the following expression which yields an error of at most 1.061% for $p\approx 1.6075$ [TM01]:

$$\text{SA}_{\widetilde{G}} \approx 4\pi\left(\frac{a^p b^p + a^p c^p + b^p c^p}{3}\right)^{1/p}, \tag{5.9}$$

where a, b and c are the lengths of the three semi-principal axes of the ellipsoid.

5.3. Projected Hemispherical Domain

The *projected hemispherical visibility probability*, $\mathcal{V}_{\Omega^{\perp}}$, over a surface area domain, A, taking a set of geometric primitives, G, into account, is defined as:

$$\mathcal{V}_{\Omega^{\perp}}(A,G) \coloneqq 1-\frac{1}{\pi}\int_{\Omega^{\perp}}\mathcal{H}(\text{ray}(x_{A},\widehat{\omega}),G)\,\mathrm{d}\Omega^{\perp}_{\widehat{\omega}}, \tag{5.10}$$

where x_{A} is the center (or any other fixed) position in A. Instead of approximating $\mathcal{V}$ (4.1) directly, we can assume $\mathcal{V}\approx\mathcal{V}_{\Omega^{\perp}}$ by neglecting the surface area domain dependence and thus use $\mathcal{V}_{\Omega^{\perp}}$ (5.10).

The regular discretization of the projected hemispherical domain, $\overline{A}$, is called our *One Point All Directions* (OPAD) approximation [MBD15] and is defined as:

$$\mathcal{V}_{\text{OPAD}}(A,G) \coloneqq 1 - \frac{1}{|\overline{\Omega}^{\perp}|} \sum_{\widehat{\omega}\in\overline{\Omega}^{\perp}} \mathcal{H}(\text{ray}(x_A,\widehat{\omega}),G). \tag{5.11}$$

The OPAD approximation corresponds to a hemispherical projection of G onto $\Omega^{\perp}$, which equates to the projected solid angle subtended by G measured from x_A.

Instead of using the original set of geometric primitives, G, we can use an approximation $\widetilde{G}$:

$$\mathcal{V}_{\Omega^{\perp}}(A,G) \approx 1 - \frac{1}{\pi}\int_{\Omega^{\perp}} \mathcal{H}\left(\text{ray}(x_A,\widehat{\omega}),\widetilde{G}\right) \mathrm{d}\Omega^{\perp}_{\widehat{\omega}}. \tag{5.12}$$

Even if the geometry, G, constitutes a single convex solid, $\widetilde{G}$, the APSA approximation does not include the projected hemispherical domain. Therefore, geometric primitives will be assumed to have the same blocking behaviour independent of the distance to the splitting plane. A finite discretization of the infinitesimal projected solid angle, $\mathrm{d}\Omega^{\perp}$, subtended by a (double-sided) polygon, p, with surface area, SA_p, and surface normal, $\widehat{N}_p$, measured from the normal, $\widehat{N}_A$, at some point, x_A, in A, is given by:

$$\Delta\Omega^{\perp} = \frac{\text{SA}_p \left|\widehat{D}\cdot\widehat{N}_p\right|}{\left(\vec{D}\cdot\vec{D}\right)\left(\widehat{D}\cdot\widehat{N}_A\right)}. \tag{5.13}$$

Here, $\vec{D}$ is the distance vector from x_A to the center of gravity of the polygon, p.

Our *Hybrid APSA* (HAPSA) and *Weighted Centroid HAPSA* (WCHAPSA) approximations extend the APSA approximation, our *Hybrid APE* (HAPE) approximation extends our APE approximation, and our *Hybrid APE2* (HAPE2) approximation extends our APE2 approximation by considering the projected solid angle subtended by the average projected surface area, $1/4\ \text{SA}_{\widetilde{G}}$, with center of gravity positioned in $\vec{\widetilde{G}}_{\text{c}}$ measured from the normal, $\widehat{N}_A$, at the center of the splitting plane, $\vec{A}_{\text{c}}$, as:

$$\mathcal{V}_{\text{(WC)HAPSA|HAPE(2)}}(A,G) \coloneqq 1 - \min\left(\frac{1/4\ \text{SA}_{\widetilde{G}}\left(\widehat{D}\cdot\widehat{N}_A\right)}{\pi\left(\vec{D}\cdot\vec{D}\right)}, 1\right), \tag{5.14}$$

with distance vector, $\vec{D} \coloneqq \vec{\widetilde{G}}_{\text{c}} - \vec{A}_{\text{c}}$. To compute the average geometry vertex (i.e. center of gravity), $\vec{\widetilde{G}}_{\text{c}}$, the HAPSA approximation gives equal weight to all geometric primitives in G whereas the WCHAPSA approximation gives weights based on the surface area of the geometric primitives in G. For axis-aligned splitting planes, the dot product, $\widehat{D}\cdot\widehat{N}_A$, reduces to a single multiplication. When the denominator is zero, the center of gravity coincides with the center of the splitting plane. In this special case, (WC)HAPSA reduces to APSA, HAPE to APE, and HAPE2 to APE2.

Our *Hybrid Average Normal Projected Surface Area* (HANPSA), *Weighted Normal HANPSA* (WNHANPSA) and *Weighted Normal Weighted Centroid HANPSA* (WNWCHANPSA) approximations take a different scaling factor into account based on the average normal:

$$\mathcal{V}_{\text{(WN(WC))HANPSA(2)}}(A,G) \coloneqq 1 - \min\left(\frac{\text{SA}_{\widetilde{G}}\left|\widehat{D}\cdot\widehat{N}_{\widetilde{G}}\right|\left(\widehat{D}\cdot\widehat{N}_A\right)}{\pi\left(\vec{D}\cdot\vec{D}\right)}, 1\right), \tag{5.15}$$

where $\widehat{N}_{\widetilde{G}}$ is the average surface normal in G. The HANPSA approximation uses equal weights for computing the average normal and centroid. The WNHANPSA approximation uses weights based on the surface area of the geometric primitives in G for computing the average normal and uses equal weights for computing the average centroid. The WNWCHANPSA approximation uses weights based on the surface area of the geometric primitives in G for computing the average normal and centroid. In case of a single convex or concave solid, the average normal is zero. Therefore, we switch to the HAPSA approximation if the Euclidean length of the summed normals is less than one (very coarse indication of concavity/convexity).

Only geometric primitives with a normal pointing away from the splitting plane, A, (i.e. $\widehat{N}_p\cdot\widehat{N}_A \geq 0$) need to be considered. We call the latter our *HANPSA2*, *WNHANPSA2* and *WNWCHANPSA2* approximations (see Equation (5.15) as well).

5.4. AABB Area Domain+Projected Hemispherical Domain

A *Monte Carlo* (MC) evaluation, $\mathcal{H}_{\text{MC}}$, of $\mathcal{H}$ (4.2) with N samples is defined as:

$$\mathcal{H}_{\text{MC}}(B,G) \coloneqq (1/N)\sum_{(x,\widehat{\omega})\sim p(B)}^{N} \mathcal{H}(\text{ray}(x,\widehat{\omega}),G), \tag{5.16}$$

where $\mathrm{d}p(B) \coloneqq \frac{\mathrm{d}B_x \mathrm{d}\Omega^{\perp}_{\widehat{\omega}}}{|B|\pi}$ represents the probability density function of the incoming isotropic ray distribution through B. This approximation is impractical to evaluate since it involves the actual tracing of rays.

The hit probability can be trivially approximated based on a weighted sum of our plane visibility probability approximations applied to the six faces of the AABB, B, as [RKJ96, Mou15]:

$$\mathcal{H}(B,G) \approx 1 - \frac{\sum_{i=1}^{6}|A_i|\,\mathcal{V}(A_i,G)}{\sum_{i=1}^{6}|A_i|}. \tag{5.17}$$

Note that our APSA|EP(2)|APE(2) approximations evaluate to the same value for parallel surface areas. This is also the case for all visibility probability approximations considering only the surface area domain in the presence of double-sided geometry. This means that only three out of six faces of the AABB need to be considered.

Instead of using the original set of geometric primitives, G, we can use an approximation, $\widetilde{G}$:

$$\mathcal{H}(B,G) \approx \frac{1}{|B|\pi}\int_B\int_{\Omega^{\perp}} \mathcal{H}\left(\text{ray}(x,\widehat{\omega}),\widetilde{G}\right) \mathrm{d}B_x\,\mathrm{d}\Omega^{\perp}_{\widehat{\omega}}. \tag{5.18}$$

This naturally leads to a specific extension of our average projected surface area based approximations (APSA|APE(2)) with regard to an AABB (convex solid) instead of a splitting plane:

$$\mathcal{H}_{\text{APSA|APE(2)}}(B,G) \coloneqq \min\left(\frac{\text{SA}_{\widetilde{G}}}{\text{SA}_B}, 1\right). \tag{5.19}$$

6. Results

Our build heuristics and build heuristic framework is implemented in `pbrt-v2` [PH10]. We chose some common test scenes of difficult but realistic geometric complexity, ranging between 100 k and 1500 k triangles, for which three interesting camera viewpoints are selected (see Table 3). For each test scene and viewpoint an ambient occlusion image is rendered at a resolution of 512×512 pixels using 512 primary (camera) rays per pixel and one shadow (ambient) occlusion ray per primary ray. The length of these ambient occlusion rays is not set to a fixed, finite, maximum distance (i.e. cameras fully enclosed by geometry will result in completely black images). For our RTSAH approximations, we use a fixed number of 256 equidistant bins for each of the three primary axes at each level of the tree construction, resulting in a time complexity of $\mathcal{O}(|G_P|)$ for each voxel V_P to partition and a total time complexity of $\mathcal{O}(|G_S| \log |G_S|)$ for the ADS construction. If $|G_P| < 256$ for BIHs, BVHs and the BVH contribution of SBVHs, we switch from equidistant bins to all adaptive bins (superset). For our SRDH approximations, we use a sweeping plane algorithm for the BVH construction (except for approximations involving a median or MAD), since no geometric primitive clipping is required as opposed to our RTSAH approximations.

Table 4 summarizes the gains in the number of ray-triangle intersection tests and ADS node traversal steps, and the gains in rendering and build times, for kd-trees, BIHs and (S)BVHs built with our RTSAH approximations and BVHs built with our SRDH approximations. We only tabulated our best RTSAH and SRDH approximations. Our SRDH APE(2) approximations result in very deep and skewed trees with multiple heavily populated leaf nodes at the maximum tree depth. Similarly, all the BVHs constructed with SRDH′ APSA|APE(2) result in very skewed trees with one heavily populated leaf node at the maximum tree depth due to the clamping of the AABB hit probabilities (5.19) between zero and one. We refer the interested reader to our supplementary material for a more detailed and in depth comparison of our different RTSAH and SRDH approximations and the ground truth (for some small artificial scenes), using both a directional and uniform ray distribution.

7. Discussion

The average gains both in the number of ray-triangle intersection tests and in the number of node traversal steps for ADSs built with the RTSAH are close to the SAH. Due to the impact of the less-conditional piercing probabilities (which do not take occlusion into account and are thus part of both the RTSAH and SAH cost functions) and the importance of the positions of the cost function's extrema as opposed to the cost function errors, the SAH is still competitive in selecting optimal candidate partitions close to those selected by the RTSAH. Depending on the scene, however, large positive gains can be noticed since the RTSAH exploits the occlusion between voxels resulting in smaller costs compared to the SAH, leading to different split decisions and allowing further refinement of ADS tree branches. On the other hand, consistently deviating from the SAH's worst-case assumptions can lead refinement too far, negatively impacting memory usage and performance. Furthermore, although the inclusion of ray termination in the SAH is a natural generalization of the SAH, assuming a parent-exterior isotropic ray distribution does not hold well in practice and using a greedy divide-and-conquer approach is not guaranteed to behave well for the RTSAH as is the case for the SAH, which explains the noticeable negative gains. ADS construction can get stuck in local minima of the global RTSAH cost function. Moreover, the obtained ADSs are not guaranteed to be close to the global optimal ADS (of which the construction is *NP*-hard) according to the global RTSAH cost metric. The largest drawback, however, for using the RTSAH is the build time. Using ray sampling with a preconstructed ADS can take up to multiple weeks. Our RTSAH approximations do not use ray sampling but triangle clipping operations which can still take up to several hours on a single thread (in `pbrt-v2` [PH10]) for large numbers of geometric primitives (e.g., duplication of references to geometric primitives in BSPs and SBVHs).

Our SRDH and SRDH′ approximations for constructing BVHs do not require ray sampling nor triangle clipping, resulting in similar build times as with the SAH (with the difference of having two separate ADSs instead of one). For our SRDH ASPA and SRDH′ SA, the number of ray-triangle intersection tests is reduced on average by 11 and 22%, respectively, compared to NodeSATO at the expense of an average increase in the number of ADS node traversal steps (which includes a ray-AABB intersection test for BVHs) by 86 and 71%, respectively. The latter becomes the determining factor in the rendering time, which is worse compared to NodeSATO.

8. Conclusions and Further Work

We extended and generalized the SAH by including local ray termination. We presented various new heuristics to approximate local ray termination that can be used to construct BSPs, BIHs and (S)BVHs with the RTSAH for accelerating closest-hit ray queries and to construct BVHs with the SRDH for accelerating any-hit ray queries. The overall impact of the RTSAH on the rendering time is in general small due to the small differences in the number of ray-triangle intersection tests and ADS node traversal steps. Furthermore, gains in ray-triangle intersection tests are often compensated by losses in ADS node traversal steps, resulting in rendering performance close to the SAH. On the other hand, ADS build times can take up to several hours due to triangle clipping (which is still a magnitude faster than ray sampling). Furthermore, ADS construction with the RTSAH is more sensitive to the underlying assumption of a parent-exterior isotropic ray distribution and the use of a greedy divide-and-conquer partitioning approach than the SAH. BVHs constructed with the SRDH require no clipping and can be built as fast as with the SAH. The ADSs constructed with the SRDH result in less ray-triangle intersection tests compared to NodeSATO, but perform worse due to more ADS node traversal steps (including ray-AABB intersection tests) on average.

We conclude that local ray termination (heuristically approximated or ground truth) is not of great practical use in the context of ADSs. Our results and conclusions are, however, not a plea for the use of local ray termination, but should rather be considered as a further analysis and understanding of the SAH.

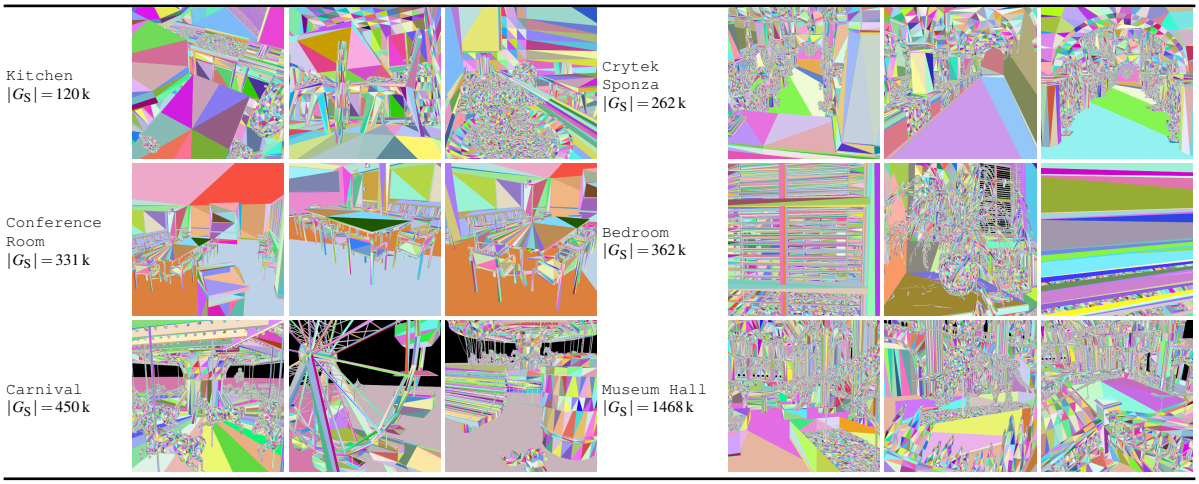


Table 3: *Our test scenes with their number of (unique) geometric primitives and chosen viewpoints visualized by hashing and rendering the identifiers of the geometric primitives.*

Further research. Considering less geometric primitives during the ADS construction allows for more precise approximations of the actual ray distribution, local or global ray termination without incurring higher build times. Shevtsov et al. [SSK07] for instance only consider every *n*th primitive. Instead, we want to investigate two possible strategies:

1. Geometric primitives can be clustered based on centroid, bounds and normal information. One representative geometric primitive is selected or created for each cluster. Next, the ADS is constructed for the original geometry, but the cost metric operates only on the cluster representatives. Depending on the cluster size, the clusters themselves can be further subdivided to obtain smaller and less populated leaf nodes. This approach generalizes Bonsai [GBDAM15] which first clusters the geometric primitives in mini-trees and then constructs an ADS on top of these mini-trees.
2. Scene geometry (or specific scene objects) can be simplified efficiently via a voxelization based approach. Next, the ADS is constructed for the original geometry, but the cost metric operates only on the simplified geometry. Since we only intend to use the simplified geometry for constructing the ADS, the simplification algorithm can be more aggressive and less conservative compared to previous work in this area (e.g., [SSLL14]).

Finally, we want to investigate splits in object partitioning schemes that are not only based on the centroids of the geometric primitives, since the latter constitute only a small subset of all possible candidate partitions.

Acknowledgments

We would like to thank Naiwen Xie and especially Roald Frederickx for many fruitful discussions, and the anonymous reviewers for their feedback and suggestions.

The `Bedroom` [Bir06] scene is modelled by David Vacek and designed by David Tousek. The `Kitchen` [Bir06] scene is modelled by Jeremy Birn. The `Carnival` [Bir06] scene is modelled by Dan Konieczka. The `Museum Hall` [Bir06] scene is modelled by Alvaro Luna Bautista and Joel Andersdon. The `Conference Room` [McG17] scene is created by Anat Grynberg and Greg Ward. The `Crytek Sponza` [McG17] scene is created by Frank Meinl. pbrt-v2 [PH10] is courtesy of Matt Pharr and Greg Humphreys.

Matthias Moulin is a predoctoral fellow of the Research Foundation–Flanders (FWO).

This is a post-peer-review, pre-copyedit version of an article published in The Visual Computer. The final authenticated version is available online at: https://doi.org/10.1007/s00371-018-1575-x. The online version of this article contains supplementary material, which is available to authorized users.

Conflict of Interest

The authors declare that they have no conflict of interest.

Appendix A: Form Factors

The (patch-to-patch) *form factor*, $\mathcal{F}_{i\to j}$, between two surface area domains, A_i and A_j, describes the fraction of *energy* (i.e. irradiance in the context of radiosity algorithms or rays in the context of ray termination) leaving A_i reaching A_j:

$$\mathcal{F}_{i\to j} := \frac{1}{A_i}\int_{A_i}\int_{A_j} \mathcal{V}(x_i,x_j)\,\frac{\left(\widehat{x_i x_j}\cdot\widehat{N}_{x_i}\right)\left(\widehat{x_j x_i}\cdot\widehat{N}_{x_j}\right)}{\pi r^2_{x_i x_j}}\,\mathrm{d}A_i\,\mathrm{d}A_j, \quad \text{(A.1)}$$

where x_i and x_j belong to the surface area domain, A_i and A_j, respectively, $\mathcal{V}(x_i,x_j)$ is the binary visibility function (i.e. evaluates to 1 if x_i and x_j are mutually visible, 0 otherwise), $\widehat{N}_{x_i}$ and $\widehat{N}_{x_j}$ are the surface normals in x_i and x_j, respectively, and $r_{x_i x_j}$ is the distance between x_i and x_j. Due to the visibility component, no tractable closed form expression can be found handling all geometrical configurations between both surface area domains, A_i and A_j. For the remainder we assume the absence of such geometrical configurations (i.e. $\mathcal{V}(x_i,x_j)\equiv 1$).

	Kd-tree											
Build heuristic	Average case				Best case				Worst case			
	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B
SAH (SP)	-14.43%	7.84%	-0.1%	0.1	6.74%	18.89%	3.6%	0.1	-53.54%	-0.72%	-13.6%	0.2
SAH	Reference				Reference				Reference			
RTSAH SA	-6.78%	-7.95%	-5.2%	162.6	21.18%	9.60%	2.7%	118.6	-37.21%	-25.66%	-19.5%	248.2
RTSAH APSA	0.45%	-1.67%	-0.4%	94.1	11.94%	14.19%	5.6%	74.2	-10.02%	-12.30%	-4.2%	141.8
RTSAH HANPSA2	-3.20%	-5.73%	-4.0%	107.3	3.60%	13.35%	3.4%	80.6	-15.45%	-17.93%	-8.7%	166.4
RTSAH WNHANPSA2	-1.57%	-5.73%	-3.6%	91.4	5.91%	16.04%	4.4%	67.4	-8.98%	-36.96%	-11.4%	124.3
RTSAH WNWCHANPSA2	-3.08%	-6.43%	-4.3%	92.3	3.75%	18.17%	1.3%	68.4	-15.87%	-47.95%	-16.5%	124.6
RTSAH EP	-0.36%	-5.42%	-2.0%	87.2	15.05%	4.06%	5.3%	68.3	-13.97%	-22.43%	-9.6%	113.1
RTSAH APE	0.22%	-3.94%	-1.5%	91.9	18.58%	6.65%	2.0%	76.2	-21.26%	-18.95%	-3.9%	107.8
	BIH											
Build heuristic	Average case				Best case				Worst case			
	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B
SAH (SP)	Reference				Reference				Reference			
SAH	-0.33%	1.54%	0.5%	2.3	8.22%	18.12%	9.9%	2.0	-7.80%	-9.07%	-9.0%	2.7
RTSAH SA	3.33%	-4.15%	-4.7%	369.5	98.42%	12.87%	5.5%	301.5	-40.26%	-17.87%	-12.8%	413.0
RTSAH APSA	1.09%	-6.03%	-4.9%	179.9	88.38%	20.95%	8.3%	149.0	-39.33%	-44.22%	-27.3%	204.0
RTSAH HANPSA2	-2.81%	-17.30%	-12.4%	183.7	102.95%	5.37%	1.0%	155.5	-47.58%	-40.30%	-25.3%	200.7
RTSAH WNHANPSA2	-3.55%	-18.35%	-13.9%	185.0	108.36%	6.37%	1.2%	159.0	-47.05%	-48.71%	-36.5%	199.3
RTSAH WNWCHANPSA2	-6.18%	-17.85%	-12.7%	186.0	120.97%	13.23%	7.1%	161.8	-50.67%	-60.06%	-40.3%	205.6
RTSAH EP	-13.54%	-8.01%	-5.2%	172.9	89.27%	15.88%	10.4%	139.5	-59.89%	-43.57%	-23.9%	189.7
RTSAH APE	-7.80%	-9.73%	-5.4%	174.8	115.82%	9.74%	9.4%	137.5	-61.54%	-44.59%	-25.3%	209.1
	BVH											
Build heuristic	Average case				Best case				Worst case			
	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B
SAH (SP)	Reference				Reference				Reference			
SAH	0.22%	2.97%	0.8%	10.4	2.10%	24.03%	7.2%	8.5	-3.61%	-6.95%	-5.6%	11.7
RTSAH APSA	0.75%	1.12%	0.5%	962.4	3.32%	23.29%	8.5%	762.5	-1.48%	-11.08%	-6.5%	1163.3
RTSAH HANPSA2	0.76%	2.08%	-0.7%	999.2	3.24%	27.46%	7.6%	803.0	-1.80%	-13.94%	-8.0%	1202.1
RTSAH WNHANPSA2	0.52%	1.51%	-0.3%	996.1	3.32%	22.04%	5.5%	783.5	-2.19%	-14.66%	-6.5%	1200.1
RTSAH WNWCHANPSA2	0.38%	0.45%	-0.9%	991.1	2.96%	22.01%	4.7%	780.5	-3.07%	-12.66%	-9.2%	1181.1
RTSAH EP	-0.15%	0.46%	0.8%	936.4	12.14%	15.14%	11.0%	715.5	-14.99%	-11.28%	-5.1%	1171.9
RTSAH APE	0.27%	-0.57%	0.6%	931.7	5.34%	13.55%	15.8%	723.5	-7.58%	-6.21%	-4.7%	1154.1
	SBVH											
Build heuristic	Average case				Best case				Worst case			
	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B	ΔI_c	ΔT_c	ΔR	B
SAH	Reference				Reference				Reference			
RTSAH APSA	-5.67%	-5.16%	-3.4%	9.7	7.67%	5.34%	2.8%	7.0	-52.64%	-23.75%	-14.6%	13.2
RTSAH HANPSA2	1.43%	-10.10%	-5.0%	10.3	35.22%	10.12%	-0.7%	6.9	-32.36%	-38.05%	-9.3%	15.3
RTSAH WNHANPSA2	3.05%	-11.01%	-5.1%	10.1	38.84%	9.41%	-1.9%	7.1	-17.47%	-44.67%	-10.8%	14.6
RTSAH WNWCHANPSA2	0.53%	-2.82%	-2.6%	10.0	35.49%	14.59%	1.4%	6.8	-18.87%	-17.98%	-10.5%	14.8
RTSAH EP	-9.74%	-11.43%	-4.7%	8.3	19.53%	3.23%	-0.8%	5.9	-30.35%	-32.28%	-11.4%	14.1
RTSAH APE	4.59%	-9.90%	-3.4%	8.9	38.14%	6.58%	4.9%	6.8	-6.69%	-24.81%	-11.0%	14.3
	BVH											
Build heuristic	Average case				Best case				Worst case			
	ΔI_s	ΔT_s	ΔR	B	ΔI_s	ΔT_s	ΔR	B	ΔI_s	ΔT_s	ΔR	B
SAH + NodeSATO	Reference				Reference				Reference			
SRDH APSA	11.09%	-86.24%	-59.4%	1.6	60.54%	-72.39%	-60.6%	1.5	-21.89%	-99.06%	-61.5%	1.7
SRDH′ SA	22.36%	-71.01%	-39.8%	1.3	112.83%	-40.33%	-41.2%	1.0	-23.65%	-96.09%	-42.6%	1.5

Table 4: *The gains in the number of ray-triangle intersection tests, ΔI_c, and in the number of ADS node traversal steps, ΔT_c, to perform for primary (camera) rays, the gain in the rendering CPU time, ΔR, and the number of times the build CPU time is increased, B, for kd-trees, BIHs, (S)BVHs, constructed with a binned RTSAH build versus a sweeping plane SAH build (SAH SP) and binned SAH build (SAH). The gains in the number of ray-triangle intersection tests, ΔI_s, and in the number of ADS node traversal steps, ΔT_s, to perform for shadow (ambient) rays, the gain in the rendering CPU time, ΔR, and the number of times the build CPU time is increased, B, for BVHs, constructed with a sweeping plane SRDH build versus a sweeping plane SAH build with NodeSATO. Our SA approximation assumes that all hit probabilities linearly depend on the surface area sum of the geometry contained in a child voxel (see our supplementary material).*

The form factor between two finite, orthogonal rectangles sharing one complete edge with length l, $\mathcal{F}_{\perp,1\to 2}$, and the form factor between two finite, parallel and identical rectangles separated by a distance l, $\mathcal{F}_{\parallel,1\to 2}$, can be evaluated via the following tractable closed form expressions [Hot31]:

$$\mathcal{F}_{\perp,1\to 2} = \frac{1}{\pi W}\left\{W\operatorname{atan}\left(\frac{1}{W}\right) + H\operatorname{atan}\left(\frac{1}{H}\right) - \sqrt{H^2+W^2}\operatorname{atan}\left(\frac{1}{\sqrt{H^2+W^2}}\right)\right.$$
$$\left. + \frac{1}{4}\ln\left[\left[\frac{(1+W^2)(1+H^2)}{1+W^2+H^2}\right]\left[\frac{W^2(1+W^2+H^2)}{(1+W^2)(W^2+H^2)}\right]^{W^2}\left[\frac{H^2(1+W^2+H^2)}{(1+H^2)(W^2+H^2)}\right]^{H^2}\right]\right\} \quad \text{(A.2)}$$

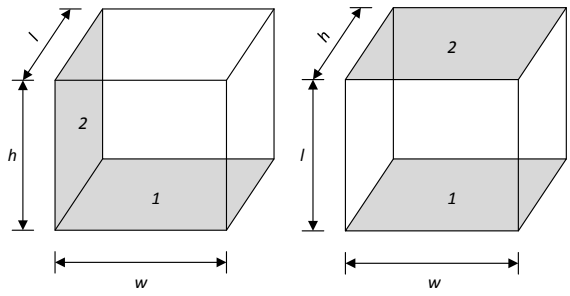
$$\mathcal{F}_{\parallel,1\to 2} = \frac{2}{\pi W H}\left\{\ln\left[\sqrt{\frac{(1+W^2)(1+H^2)}{1+W^2+H^2}}\right] - W\operatorname{atan}(W) - H\operatorname{atan}(H)\right.$$
$$\left. + W\sqrt{1+H^2}\operatorname{atan}\left(\frac{W}{\sqrt{1+H^2}}\right) + H\sqrt{1+W^2}\operatorname{atan}\left(\frac{H}{\sqrt{1+W^2}}\right)\right\}. \quad \text{(A.3)}$$

Here, w, h and l are defined as shown in Figure 4, and the dimensionless variables are defined as: $H := h/l$ and $W := w/l$.

References

[AKL13] AILA T., KARRAS T., LAINE S.: On Quality Metrics of Bounding Volume Hierarchies. In *Proceedings of the 5th High-Performance Graphics Conference* (New York, NY, USA, 2013), HPG '13, ACM, pp. 101–107. 2

[App68] APPEL A.: Some Techniques for Shading Machine Renderings



(a) $\mathcal{F}_{\perp,1\to2}$ **(b)** $\mathcal{F}_{\parallel,1\to2}$

Figure 4: *The two types of form factors between the rectangular planes of a bounding box:* ***(a)*** *the form factor between two finite, orthogonal rectangular planes sharing one complete edge with length l,* $\mathcal{F}_{\perp,1\to2}$*,* ***(b)*** *the form factor between two finite, parallel and identical rectangular planes separated by a distance l,* $\mathcal{F}_{\parallel,1\to2}$*.*

of Solids. In *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference* (New York, NY, USA, 1968), AFIPS '68 (Spring), ACM, pp. 37–45. 1

[BF79] BENTLEY J. L., FRIEDMAN J. H.: Data Structures for Range Searching. *ACM Comput. Surv. 11*, 4 (Dec 1979), 397–409. 2

[BH06] BOULOS S., HAINES E.: Ray-Box Sorting. *Ray Tracing News 19*, 1 (Sep 2006). 3

[BH09] BITTNER J., HAVRAN V.: RDH: Ray Distribution Heuristics for Construction of Spatial Data Structures. In *Proceedings of the 25th Spring Conference on Computer Graphics* (New York, NY, USA, 2009), SCCG '09, ACM, pp. 51–58. 2, 3, 7

[BHH13] BITTNER J., HAPALA M., HAVRAN V.: Fast Insertion-Based Optimization of Bounding Volume Hierarchies. *Computer Graphics Forum 32*, 1 (2013), 85–100. 3

[Bir06] BIRN J.: Lighting Challenges, 2006. `http://3drender.com/challenges`. 11

[Cau41] CAUCHY A.-L.: Note sur divers théorèmes relatifs à la rectification des courbes et à la quadrature des surfaces. *Comptes Rendus de l'Académie des Sciences 13* (Dec 1841), 1060–1066. 4, 8

[Cau50] CAUCHY A.-L.: Mémoire sur la rectification des courbes et la quadrature des surfaces courbes. *Mémoires de l'Académie des Sciences 22* (1850), 3–13. 4, 8

[CCI12] CHOI B., CHANG B., IHM I.: Construction of Efficient Kd-trees for Static Scenes Using Voxel-Visibility Heuristic. *Computers & Graphics 36*, 1 (2012), 38–48. 2, 3

[CS97] CAZALS F., SBERT M.: *Some Integral Geometry Tools to Estimate the Complexity of 3D Scenes*. Research Report RR-3204, INRIA, 1997. 4, 8

[CWH93] COHEN M. F., WALLACE J., HANRAHAN P.: *Radiosity and Realistic Image Synthesis*. Academic Press Professional, Inc., San Diego, CA, USA, 1993. 4

[DK08] DAMMERTZ H., KELLER A.: The Edge Volume Heuristic - Robust Triangle Subdivision for Improved BVH Performance. In *IEEE Symposium on Interactive Ray Tracing 2008* (Aug 2008), pp. 155–158. 3

[EG07] ERNST M., GREINER G.: Early Split Clipping for Bounding Volume Hierarchies. In *IEEE Symposium on Interactive Ray Tracing 2007* (Sept 2007), pp. 73–78. 3

[FFD09] FABIANOWSKI B., FOWLER C., DINGLIANA J.: A Cost Metric for Scene-Interior Ray Origins. In *Eurographics 2009 - Short Papers* (2009), Alliez P., Magnor M., (Eds.), The Eurographics Association. 2

[FLF12] FELTMAN N., LEE M., FATAHALIAN K.: SRDH: Specializing BVH Construction and Traversal Order Using Representative Shadow Ray Sets. In *Proceedings of the Fourth ACM SIGGRAPH / Eurographics Conference on High-Performance Graphics* (Aire-la-Ville, Switzerland, Switzerland, 2012), EGGH-HPG'12, Eurographics Association, pp. 49–55. 1, 3, 4, 6

[GBDAM15] GANESTAM P., BARRINGER R., DOGGETT M., AKENINE-MÖLLER T.: Bonsai: Rapid Bounding Volume Hierarchy Generation using Mini Trees. *Journal of Computer Graphics Techniques (JCGT) 4*, 3 (September 2015), 23–42. 11

[GD16] GANESTAM P., , DOGGETT M.: SAH guided spatial split partitioning for fast BVH construction. *Computer Graphics Forum 35*, 2 (2016). 3

[GS87] GOLDSMITH J., SALMON J.: Automatic Creation of Object Hierarchies for Ray Tracing. *IEEE Computer Graphics and Applications 7*, 5 (May 1987), 14–20. 1, 2, 3, 4

[Hav00] HAVRAN V.: *Heuristic Ray Shooting Algorithms*. PhD thesis, Faculty of Electrical Engineering, Czech Technical University, Prague, Nov 2000. 2, 3

[HB99] HAVRAN V., BITTNER J.: Rectilinear BSP Trees for Preferred Ray Sets. *Proceedings of Summer School of Computer Graphics 99* (1999), 171–179. 2

[HB02] HAVRAN V., BITTNER J.: On Improving Kd-Trees for Ray Shooting. In *Proceedings of Winter School of Computer Graphics 2002 Conference* (2002), pp. 209–217. 2

[HH11] HAPALA M., HAVRAN V.: Review: Kd-tree Traversal Algorithms for Ray Tracing. *Computer Graphics Forum 30*, 1 (2011), 199–213. 3

[HHS06] HAVRAN V., HERZOG R., SEIDEL H.-P.: On the Fast Construction of Spatial Hierarchies for Ray Tracing. In *IEEE Symposium on Interactive Ray Tracing 2006* (Sept 2006), pp. 71–80. 2

[HKRS02] HURLEY J. T., KAPUSTIN A., RESHETOV A., SOUPIKOV A.: Fast Ray Tracing for Modern General Purpose CPU. In *Proceedings of GraphiCon* (2002). 2, 3

[HM08] HUNT W. A., MARK W. R.: Ray-specialized acceleration structures for ray tracing. In *2008 IEEE Symposium on Interactive Ray Tracing* (Aug 2008), pp. 3–10. 2

[HMS06] HUNT W. A., MARK W. R., STOLL G.: Fast kd-tree Construction with an Adaptive Error-Bounded Heuristic. In *IEEE Symposium on Interactive Ray Tracing 2006* (Sept 2006), pp. 81–88. 3

[Hot31] HOTTEL H. C.: Radiant Heat Transmission between Surfaces Separated by Non-Absorbing Media. *Trans. ASME 53* (1931), 265–273. 4, 12

[IH11] IZE T., HANSEN C.: RTSAH Traversal Order for Occlusion Rays. *Computer Graphics Forum 30*, 2 (2011), 297–305. 1, 3, 4, 5

[IWP08] IZE T., WALD I., PARKER S. G.: Ray tracing with the BSP tree. In *IEEE Symposium on Interactive Ray Tracing 2008* (Aug 2008), pp. 159–166. 2

[Kaj86] KAJIYA J. T.: The Rendering Equation. *SIGGRAPH Comput. Graph. 20*, 4 (Aug 1986), 143–150. 1

[Kap85] KAPLAN M. R.: The Use of Spatial Coherence in Ray Tracing. *ACM SIGGRAPH Course Notes 11* (1985). 2

[Ken08] KENSLER A.: Tree Rotations for Improving Bounding Volume Hierarchies. In *IEEE Symposium on Interactive Ray Tracing 2008* (Aug 2008), pp. 73–76. 3

[KK86] KAY T. L., KAJIYA J. T.: Ray Tracing Complex Scenes. *SIGGRAPH Comput. Graph. 20*, 4 (Aug. 1986), 269–278. 3

[Mar13] MARKOV A. A.: Primer statisticheskogo issledovanija nad tekstom 'Evgenija Onegina' illjustrirujuschij svjaz' ispytanij v tsep. *Izvestija Imp. Akademii nauk (Bulletin of the Imperial Academy of Sciences of St. Petersburg) 7*, 3 (1913), 153–162. 6

[Mar06] MARKOV A. A.: An example of statistical investigation of the text 'Eugene Onegin' concerning the connection of samples in chains. *Science in Context 19*, 4 (Nov 2006), 591–600. 6

[MB90] MACDONALD D. J., BOOTH K. S.: Heuristics for Ray Tracing Using Space Subdivision. *Vis. Comput. 6*, 3 (May 1990), 153–166. 1, 2, 3, 4

[MBD15] MOULIN M., BILLEN N., DUTRÉ P.: Efficient Visibility Heuristics for Kd-Trees Using the RTSAH. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations* (June 2015), Lehtinen J., Nowrouzezahrai D., (Eds.), The Eurographics Association, pp. 31–39. 1, 2, 3, 4, 8, 9

[McG17] MCGUIRE M.: Computer Graphics Archive, July 2017. `https://casual-effects.com/data`. 11

[Mou15] MOULIN M.: *Hybrid Kd-trees for Photon Mapping and Accelerating Ray Tracing*. Master's thesis, Department of Computer Science, KU Leuven, Belgium, 2015. 2, 9

[NM14] NAH J.-H., MANOCHA D.: SATO: Surface Area Traversal Order for Shadow Ray Tracing. *Computer Graphics Forum 33*, 6 (2014), 167–177. 3

[NT03] NG K., TRIFONOV B.: Automatic Bounding Volume Hierarchy Generation Using Stochastic Search Methods. In *Proceedings Mini-Workshop on Stochastic Search Algorithms* (2003). 3

[ODJ15] OGAKI S., DEROUET-JOURDAN A.: MBVH Child Node Sorting for Fast Occlusion Test. In *Eurographics Symposium on Rendering - Experimental Ideas & Implementations* (June 2015), Lehtinen J., Nowrouzezahrai D., (Eds.), The Eurographics Association, pp. 57–64. 3

[ODJ16] OGAKI S., DEROUET-JOURDAN A.: An N-ary BVH Child Node Sorting Technique for Occlusion Tests. *Journal of Computer Graphics Techniques (JCGT) 5*, 2 (June 2016), 22–37. 3

[OMR87] OOI B. C., MCDONELL K. J., RON S.-D.: Spatial Kd-Tree: An Indexing Mechanism for Spatial Databases. In *Proceedings of the IEEE COMPSAC Conference* (1987). 2

[PGDS09] POPOV S., GEORGIEV I., DIMOV R., SLUSALLEK P.: Object Partitioning Considered Harmful: Space Subdivision for BVHs. In *Proceedings of the Conference on High Performance Graphics 2009* (New York, NY, USA, 2009), HPG '09, ACM, pp. 15–22. 2

[PGpSS06] POPOV S., GUNTHER J., P. SEIDEL H., SLUSALLEK P.: Experiences with Streaming Construction of SAH Kd-Trees. In *2006 IEEE Symposium on Interactive Ray Tracing* (Sept 2006), pp. 89–94. 3

[PH10] PHARR M., HUMPHREYS G.: *Physically Based Rendering, Second Edition: From Theory To Implementation*, 2nd ed. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. 10, 11

[RKJ96] REINHARD E., KOK A. J. F., JANSEN F. W.: *Cost Prediction in Ray Tracing*. Tech. rep., University of Bristol, Bristol, UK, UK, 1996. 2, 9

[Rup11] RUPPERT D.: *Statistics and Data Analysis for Financial Engineering*, 1st ed. Springer, San Francisco, CA, USA, 2011. 8

[RW80] RUBIN S. M., WHITTED T.: A 3-dimensional Representation for Fast Rendering of Complex Scenes. *SIGGRAPH Comput. Graph. 14*, 3 (Jul 1980), 110–116. 2

[SFD09] STICH M., FRIEDRICH H., DIETRICH A.: Spatial Splits in Bounding Volume Hierarchies. In *Proceedings of the Conference on High Performance Graphics 2009* (New York, NY, USA, 2009), HPG '09, ACM, pp. 7–13. 2, 3, 7

[SP94] SILLION F. X., PUECH C.: *Radiosity and Global Illumination*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1994. 4

[SSK07] SHEVTSOV M., SOUPIKOV A., KAPUSTIN A.: Highly Parallel Fast KD-tree Construction for Interactive Ray Tracing of Dynamic Scenes. *Computer Graphics Forum 26*, 3 (2007), 395–404. 3, 11

[SSLL14] SILVENNOINEN A., SARANSAARI H., LAINE S., LEHTINEN J.: Occluder Simplification Using Planar Sections. *Computer Graphics Forum 33*, 1 (2014), 235–245. 11

[TM01] THOMSEN K., MICHON G. P.: Surface Area of an Ellipsoid. *Final Answers* (2001). 8

[VHS12] VINKLER M., HAVRAN V., SOCHOR J.: Technical Section: Visibility Driven BVH Build Up Algorithm for Ray Tracing. *Comput. Graph. 36*, 4 (Jun 2012), 283–296. 2, 3

[Wal07] WALD I.: On Fast Construction of SAH-based Bounding Volume Hierarchies. In *Proceedings of the 2007 IEEE Symposium on Interactive Ray Tracing* (Washington, DC, USA, 2007), RT '07, IEEE Computer Society, pp. 33–40. 3

[WBKP08] WALTER B., BALA K., KULKARNI M., PINGALI K.: Fast Agglomerative Clustering for Rendering. In *IEEE Symposium on Interactive Ray Tracing 2008* (Aug 2008), pp. 81–86. 3

[WBMS05] WILLIAMS A., BARRUS S., MORLEY R. K., SHIRLEY P.: An Efficient and Robust Ray-box Intersection Algorithm. In *ACM SIGGRAPH 2005 Courses* (New York, NY, USA, 2005), SIGGRAPH '05, ACM. 6

[WGS04] WALD I., GÜNTHER J., SLUSALLEK P.: Balancing Considered Harmful - Faster Photon Mapping using the Voxel Volume Heuristic. *Computer Graphics Forum 23*, 3 (2004), 595–603. 2

[WH06] WALD I., HAVRAN V.: On Building Fast Kd-Trees for Ray Tracing, and on Doing That in O(N log N). In *IEEE Symposium on Interactive Ray Tracing 2006* (Sept 2006), pp. 61–69. 2, 3

[Whi80] WHITTED T.: An Improved Illumination Model for Shaded Display. *Communications of the ACM 23*, 6 (June 1980), 343–349. 1

[Win04] WINSTON W. L.: *Operations Research: Applications and Algorithms*, 4th ed. Brooks/Cole, Cengage Learning, Belmont, CA, USA, 2004. 6

[WK06] WÄCHTER C., KELLER A.: Instant Ray Tracing: The Bounding Interval Hierarchy. In *Proceedings of the 17th Eurographics Conference on Rendering Techniques* (Aire-la-Ville, Switzerland, Switzerland, 2006), EGSR '06, Eurographics Association, pp. 139–149. 2, 3

[WMG*09] WALD I., MARK W. R., GÜNTHER J., BOULOS S., IZE T., HUNT W., PARKER S. G., SHIRLEY P.: State of the Art in Ray Tracing Animated Scenes. *Computer Graphics Forum 28*, 6 (2009), 1691–1722. 2

[WMS06] WOOP S., MARMITT G., SLUSALLEK P.: B-Kd Trees for Hardware Accelerated Ray Tracing of Dynamic Scenes. In *Proceedings of the 21st ACM SIGGRAPH/EUROGRAPHICS Symposium on Graphics Hardware* (New York, NY, USA, 2006), GH '06, ACM, pp. 67–77. 2