CEIG - Spanish Computer Graphics Conference (2015) , pp. 1–9
Jorge Lopez-Moreno and Mateu Sbert (Editors)

Bidirectional Clustering for Scalable VPL-based Global Illumination

Adrian Jarabo Raul Buisan Diego Gutierrez

Universidad de Zaragoza

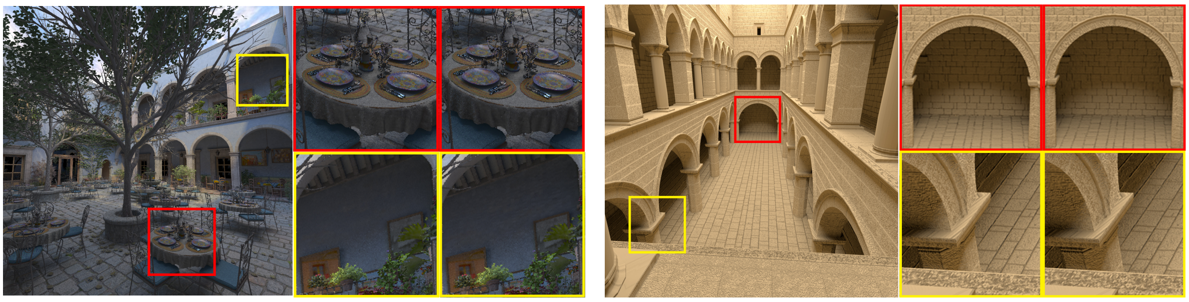

Figure 1: *Example renders of the* San Miguel *(left) and* Sponza *(right) scenes: Our scalable VPL-based bidirectional clustering method is capable of generating high-quality diffuse global illumination while leveraging the coherence between camera samples to accelerate computations. The insets show a comparison between our method (right) and the state of the art [Chr08] (left), which clusters only the light sources. Our method yields similar quality with a speed-up of up to ×2. Moreover, our bidirectional strategy scales even better as the image resolution increases*

Abstract

Virtual Point Lights (VPL) methods approximate global illumination (GI) in a scene by using a large number of virtual lights modeling the reflected radiance of a surface. These methods are efficient, and allow computing noise-free images significantly faster that other methods. However, they scale linearly with the number of virtual lights and with the number of pixels to be rendered. Previous approaches improve the scalability of the method by hierarchically evaluating the virtual lights, allowing sublinear performance with respect the lights being evaluated. In this work, we introduce a novel bidirectional clustering *approach, by hierarchically evaluating both the virtual lights* and *the shading points. This allows reusing radiance evaluation between pixels, and obtaining sublinear costs with respect to both lights and camera samples. We demonstrate significantly better performance than state-of-the-art VPL clustering methods with several examples, including high-resolution images, distributed effects, and rendering of light fields.*

1. Introduction

Efficient rendering of realistic illumination is still a challenging task: complex area light sources, environment lighting and indirect illumination are required to convey realistic looks in the scenes, making it necessary to compute the full global illumination in the scene. Many different algorithms have been proposed to compute global illumination [DBB06, KKG*14], each with their own pros and cons. From these algorithms, instant global illumination [Kel97, DKH*14] offers a remarkable convergence to a noiseless solution. It is based on approximating the illumination (both direct and indirect) using a set of Virtual Point Light (VPL) sources distributed across the scene, in effect transforming the full light transport into a many-lights problem, with overall cost proportional to the number of VPLs. However, in order to obtain good results in scenes with complex geometry, materials and light configurations, hundreds of thousands of VPLs are needed, which makes the algorithm impractical.

Several algorithms have been proposed to efficiently solve this many-lights problem, approximating the illumination by grouping individual VPLs into clusters of lights [WFA*05, Chr08]. This results into algorithms that scale well with the number of VPLs producing images with high-quality global illumination very efficiently. These methods hierarchically cluster the VPLs, so that the illumination can be computed in an adaptive manner. These algorithms, however, still need to be performed on a per-pixel basis and, although some extensions have been developed to support distributed effects (e.g. antialiasing or depth of field) [WABG06], still scale poorly with the number of pixels.

Unfortunately, the current trend is to increase the number of pixels being rendered, with HD resolutions being standard even in domestic images. Moreover, 3D displays are increasingly becoming widespread [MWA*13], and content in form of stereo-images or even light fields [LH96] for automultiscopic displays is therefore required. This means an explosion of the number of pixels that need to be rendered, for which current strategies simply do not scale well. New rendering algorithms that scale well with the number of samples (number of pixels times samples per pixel needed for distributed effects) have thus become a necessity.

In this work we propose a new method for rendering of global illumination using VPLs which is scalable with respect to both the number of lights and the number of shading points being rendered. Our work leverages the coherence between the camera samples to accelerate computations, by using a bidirectional adaptive evaluation of the illumination. We do that by hierarchically clustering and evaluating both the lights and the shading points, so that similar points receive similar contribution at coarse levels of illumination, while only fine scale details need to be refined. This strategy allows for a sublinear cost with respect to both the number of lights and shading points, making out technique very suitable for high-resolution images with distributed effects, excelling in multi-image representations such as stereo-images and light fields.

2. Related Work

Global illumination has been thoroughly studied, and a large number of algorithms have been proposed for realistic rendering, even in non-trivial domains different from the primal [DHS*05, JMM*14]. As such, we focus here on VPL-based scalable algorithms, and refer the reader to the excellent surveys by Dutre et al. [DBB06] and Krivanek et al. [KKG*14] for a wider overview of the state of the field.

Lightcuts [WFA*05] is one of the first scalable techniques for VPL-rendering, which reduces the computational cost by grouping the contribution of several similar VPLs into a single one. It hierarchically organizes the VPL into a light tree, where the internal nodes are clusters of lights, and selects adaptively the optimal cut in the tree, by minimizing the error upper bound until it is below a certain perceptual threshold. While scalable with the number of VPLs, lightcuts needs to find the optimal cut for each sample in the image, which makes it inefficient when dealing with distributed effects such as antialiasing or depth of field [CPC84]. To address this problem, *Multidimensional Lightcuts* (MDLC) [WABG06] extends lightcuts, allowing to handle these effects efficiently by grouping both VPLs and camera samples within a pixel. Later, Walter et al- [WKB12] extend the range of materials that can be handled in MDLC, by combining it with bidirectional path tracing [LW93, VG94] via multiple importance sampling (MIS) [VG95]. These techniques are however designed to efficiently integrate samples within a pixel, and therefore do not leverage computations between the rest of the image being rendered.

Similar in spirit to lightcuts, but simpler and more adapted to production environments, point-based techniques also reduce the complexity of global illumination computations by merging together several virtual lights into a single one when they are far away to the point being shaded [Bun05]. This approach is extended by Christensen [Chr08] in the PBGI algorithm, where the VPLs are stored in an octree and their clustered contribution is approximated by a spherical harmonics expansion, which is used to efficiently compute an approximated diffuse global illumination at each point. A similar approach is followed by Ritschel et al. [REG*09] and Holländer et al. [HREB11] for interactive global illumination. Due to its use in production, several works improve PBGI by reducing the memory footprint of the technique using compression [BB12], or designing out-of-core schemes improving the parallism of the technique in large-scale render farms [KTO11]. All these techniques compute the illumination at a single point without taking advantage of the coherence between different shading points to amortize the costs of traversing the shade tree. Wang et al. [WHB*13] handle this limitation by clustering points in image space and computing an initial cut on the lights tree for the cluster which is then refined for each shading point. While this allows to amortize the computations of the first cut for all points in a cluster, they only adapt the computations into two levels, without leveraging the possible similarities between clusters, or the coherence existing within each cluster. On the contrary, our work computes the cut adaptively for all the shading points at the same time, leveraging the light computations in all computed samples explicitly.

A different approach for scalable VPL-based indirect illumination bases on formulating the problem as a large matrix containing the relationship between lights and shading points [HPB07, HVAPB08], so that the final image is the sum along the pixels dimension. This allows using using low-rank matrix approximation to extract a small set of light clusters that then are used to compute the final image.

This method allows to amortize computations by sharing the light clusters in all the image, which makes it very efficient. Unfortunately, the light clusters are chosen globally for all the image, which results into locally-suboptimal clusters. Ou and Pellacini [OP11] handle this by performing the clustering into small subsets of the image, based on that local neighborhoods in pixel space will share a similar matrix structure; this allows to significantly improve the light clustering, but its quality is still dependent on the neighborhood size chosen, which similar to Wang et al.'s work [WHB*13] might result into suboptimal clusters.

3. Background

The outgoing radiance $L_o(x, \omega_\mathbf{o})$ from a point x with normal $\mathbf{n_x}$ towards a direction $\omega_\mathbf{o}$ is the sum of the emited and reflected radiace ($L_e(x, \omega_\mathbf{o})$ and $L_r(x, \omega_\mathbf{o})$ respectively), modeled as [Kaj86]:

$$L_o(x, \omega_\mathbf{o}) = L_e(x, \omega_\mathbf{o}) + L_r(x, \omega_\mathbf{o}), \quad (1)$$

$$L_r(x, \omega_\mathbf{o}) = \int_{\mathcal{H}^2} L_i(x, \omega_\mathbf{i}) f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o}) \langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+ \mathrm{d}\omega_\mathbf{i}, \quad (2)$$

where $\mathcal{H}^2$ is the hemisphere centered at $\mathbf{n_x}$, $L_i(x, \omega_\mathbf{i})$ is the incoming radiance at x in direction $\omega_\mathbf{i}$, $f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o})$ is the BRDF at x, and $\langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+$ models foreshortening. The incoming radiance L_i is itself recursive, since it comes from other positions of the scene, both as a directly emited radiance, or due to multiple interreflections of light. Therefore, $L_r(x, \omega_\mathbf{o})$ (2) can also be expressed as

$$L_r(x, \omega_\mathbf{o}) = \int_{\mathcal{S}} L_i(x, s) f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o}) G(x, s) V(x, s) \mathrm{d}s, \quad (3)$$

where $\mathcal{S}$ is the manifold of all surfaces in the scene[†], $V(x, s)$ is the visibility function between x and s, and $G(x, s)$ is the geometric attenuation modeled as

$$G(x, s) = \frac{\langle \mathbf{n_x}, \omega_\mathbf{i} \rangle^+ \langle \mathbf{n_s}, \omega_\mathbf{i} \rangle^+}{\|x - s\|^2}, \quad (4)$$

with $\mathbf{n_s}$ the normal at s. Note that in this point-to-point formulation, $\omega_\mathbf{i}$ is the direction from x to s.

In most cases it is impossible to solve $L_r(x, \omega_\mathbf{o})$ analyticaly, and stochastic techniques are usually used to compute it. An intuitive approximation of Equation (3) would be to model the light reflected at each point $s \in \mathcal{S}$ as a virtual point light [Kel97], with emission $I(s, x)$ equal to the energy reflected at s towards x. Therefore, if we precompute a sampled set of N virtual lights, then $L_r(x, \omega_\mathbf{o})$ can be approximated as

$$L_r(x, \omega_\mathbf{o}) \approx \sum_{j=1}^{N} I(s_j, x) f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o}) G(x, s_j) V(x, s_j). \quad (5)$$

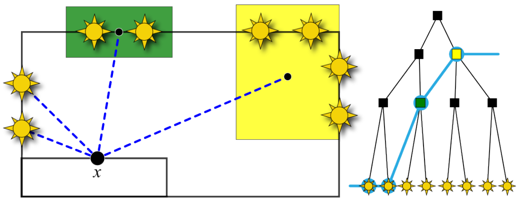

Figure 2: *Schematic overview of the rendering procedure in most adaptive clustered VPL methods [WFA*05, Chr08] in the scene shown on the left, illuminated by eight VPLs. These VPLs are stored in a tree structure (right), which is used in render time to shade point x, by selecting a tree that meets the given termination criteria. By using this structure, the number of light evaluations is reduced from eight to four.*

While this approximation leads to good results, a large number N of VPLs is needed to obtain an artifact-free image. Unfortunately, the algorithm scales linearly with N, making it to scale poorly with the number of lights, which in the end reduces the applicability of the method when rendering complex light transport, where hundreds of thousands of virtual lights are needed to obtain good results.

Scalable VPL-based rendering The most common approach to reduce the cost of VPL-based algorithms is to cluster the virtual lights, approximating the contribution of the virtual lights in a cluster using a single super-VPL summing their contribution. This transforms Equation (5) in a sum of K clusters instead of N VPLs with $K \ll N$, effectively reducing the cost of the algorithm. While some techniques perform this clustering reduction by globally computing a set of clusters for the full image [HPB07, HVAPB08], here we introduce the basics of the methods that select the most appropriate set of clusters for a specific point x.

The general procedure of adaptive clustered VPL methods [WFA*05, WABG06, Chr08] consists of storing the full set of VPLs into a hierarchical structure, where the leaves of the structure are the actual virtual lights, and the internal nodes are the clusters. Thus, the idea is ordering the clusters with the upper levels of the tree being coarser representations of the illumination, while lower ones store finer details. Then, in render time, the tree is traversed top-down by refining the nodes that introduce more error for each particular shaded point x, until the error introduced is below a certain threshold, which results in a particular cut of the tree $\mathbb{C}$. Finally, the selected nodes $\tilde{c} \in \mathbb{C}$ of the tree are used to approximate the illumination as

$$L_r(x, \omega_\mathbf{o}) \approx \sum_{\tilde{c} \in \mathbb{C}} I_{\tilde{c}}(\tilde{c}, x) f_r(x, \omega_\mathbf{i}, \omega_\mathbf{o}) G(x, \tilde{c}) V(x, \tilde{c}), \quad (6)$$

where $I(\tilde{c}, x)$ is the sum of the radiance of all VPLs contained by $\tilde{c}$, and the geometric and visibility terms $G(x, \tilde{c})$ and $V(x, \tilde{c})$ respectively are evaluated with respect to a representative point within the cluster $\tilde{c}$. An example of the pro-

[†] Note that here we assume light transport between surfaces; In the presence of participating media $\mathcal{S}$ would not be restricted to surfaces, but to every differential point s in the scene.

cess is shown in Figure 2. This adaptively selection, based on hierarchically evaluating the light tree results into sublinear cost $O(\log N)$ with respect to the number of virtual lights N; however, this process is not amortized between several pixels, and thus needs to be repeated for each pixel, resulting into a total cost of $O(M \log N)$ with M the number of shaded pixels. This ignores the potential coherence of the incoming illumination between neighbor shading points x that cause that close-by points share similar cuts $\mathbb{C}$ in the light tree. In the following section we propose a new algorithm that leverages this similarity, which results into $O(\log(N+M))$ cost.

4. Bidirectional Clustering

As discussed in the previous section, while current VPL-based algorithms scale well with the number of lights, they scale poorly with the number of points being shaded, and do not leverage the local coherece between samples. In this section we propose a new algorithm that combines the hierarchical clustering and evaluation of both virtual lights approximating global illumination [WFA*05, Chr08], and the points being illuminated obtained from sampling the scene from the camera (*shading points*), with the goal to reduce the algorithm cost to $O(\log(N+M))$.

The key idea of the algorithm is to amortize the computations between different groups of shading points, by progressively refining the precision in both the shading points and the VPLs, in a similar way as previous methods refine the illumination alone for a single pixel. To do this, we first build two different hierarchies, one for VPLs $\mathfrak{T}_L$ and other for shading points $\mathfrak{T}_E$, where internal nodes $\tilde{c} \in \mathfrak{T}$ of these hierarchies cluster their respective subtree (finer levels). These two hierarchies are combined in render time, creating an additional hierarchy $\mathfrak{T}_{E,L}$ (Figure 3) where each node $\tilde{c}_{E,L} \in \mathfrak{T}_{E,L}$ is a pair of nodes of $\mathfrak{T}_L$ and $\mathfrak{T}_E$ so that $\tilde{c}_{E,L} = (\tilde{c}_L, \tilde{c}_E)$. Each pair $\tilde{c}_{E,L}$ models the contribution of the light node $\tilde{c}_L$ over the shading node $\tilde{c}_E$. Each node is then refined until the stop criteria is found. The result is a cut $\mathbb{C}_{E,L} \subset \mathfrak{T}_{E,L}$ where each node $\tilde{c}_{E,L} \in \mathbb{C}_{E,L}$ is used to compute the illumination at specific parts of the scene, where coarse-level illumination features are reused over a large set of the final image, and finer details are refined down in the hierarchy. This is similar in spirit to classic hierarchical radiosity schemes [SAG94, CLSS97], which also inspired previous scalable VPL-based algorithms.

The algorithm consists of two passes: in the first one the VPLs and shading points are sampled, and their respective trees $\mathfrak{T}_L$ and $\mathfrak{T}_E$ are created; in the second, the two trees are used to adaptively render the scene. In the following, we describe both steps in detail.

4.1. Tree construction

The input data for the algorithm are the set of stochastically sampled VPLs modeling the illumination in the scene and the set of shading points where we are computing the illumination. While VPLs model the indirect illumination, but can also be used to model in a unified manner the contribution from area or environment maps [WFA*05]. For each VPL we store its position and direction, together with its emission; in our implementation we consider that all VPLs are Lambertian, and thus we do not need to store any directionally-resolved radiance. The shading points, on the other hand, are computed by stochastically sampling from the camera, which allows handling distributed effects including antialiasing or depth of field [CPC84]. Similarly to VPLs, for each sampling point we store its position, normal and sampling probability, and additionally we include its BRDF, the incoming direction from the camera and a pointer to the pixel from which the sample was generated.

Once the VPLs and the shading points have been computed, they are introduced into two different hierarchies. Following Christensen [Chr08] we use an *octree*. Although other structures could have been used (e.g. binary trees built using bottom-up approaches [WFA*05]), we decide to use octrees due to their fast construction and traversal, which is important given the potentially large amount of elements to cluster. We build the octree using a top-down approach, recursively subdividing clusters in eight sub-nodes with identical size, until a minimum number of samples are stored in each node; we set this number to eight in all our tests.

For each node we store the total projected directionally-resolved area of the cluster $A_{\tilde{c}}(\omega_\mathbf{o})$, which we need in order to compute the stop criteria, as we discuss later in this section, and is the sum of the projected area $a_s(\omega_\mathbf{o}) = a_s \cdot \langle \mathbf{n_s}, \omega_\mathbf{i} \rangle^+$ of each VPLs in the cluster $\tilde{c}_L$ to direction $\omega_\mathbf{o}$ as $A_{\tilde{c}}(\omega_\mathbf{o}) = \sum a_s(\omega_\mathbf{o})$. In addition, for the nodes in the light tree $\mathfrak{T}_L$ we store the accumulated directional outgoing radiance $I_{\tilde{c}}$ of the cluster as the sum of the outgoing radiance of the node subtree. We store both the area of the cluster $A_{\tilde{c}}(\omega_\mathbf{o})$ and the outgoing radiance $I_{\tilde{c}}$ using spherical harmonics with nine coefficients (for radiance we store nine for each color channel), although more orders could be stored to handle non-Lambertian VPLs with higher illumination angular frequencies. Finally, for the nodes in the shading points tree $\mathfrak{T}_E$ we store the bounding cone of the clustered point's normals.

4.2. Rendering

In order to compute the illumination in the image we want to obtain the most appropriate cut in the implicitly combined hierarchy $\mathbb{C}_{E,L} \subset \mathfrak{T}_{E,L}$ of lights and shading points. We start from a cut consisting of a single node pairing the root of both trees, and then progressively refine it. In order to refine a node of the current cut $\tilde{c}_{E,L} \in \mathbb{C}_{E,L}$, we first evaluate if the node meets the stop criteria, and if it does we compute and store the contribution of the light cluster $\tilde{c}_L$ on the shading points cluster $\tilde{c}_E$.

On the other hand, if the node $\tilde{c}_{E,L}$ does not meet the stop

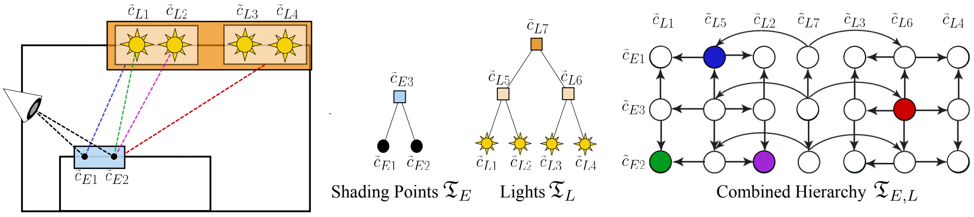
$\tilde{c}_{L1}$ $\tilde{c}_{L2}$ $\tilde{c}_{L3}$ $\tilde{c}_{L4}$
$\tilde{c}_{E1}$ $\tilde{c}_{E2}$
$\tilde{c}_{E3}$
$\tilde{c}_{L7}$
$\tilde{c}_{L5}$ $\tilde{c}_{L6}$
Shading Points $\mathfrak{T}_E$
Lights $\mathfrak{T}_L$
$\tilde{c}_{L1}$ $\tilde{c}_{L5}$ $\tilde{c}_{L2}$ $\tilde{c}_{L7}$ $\tilde{c}_{L3}$ $\tilde{c}_{L6}$ $\tilde{c}_{L4}$
$\tilde{c}_{E1}$
$\tilde{c}_{E3}$
$\tilde{c}_{E2}$
Combined Hierarchy $\mathfrak{T}_{E,L}$

Figure 3: *Schematic overview of our method: The scene on the left is rendered with two samples $\tilde{c}_{Ei}$ with $i = (1,2)$ and iluminated with four VPLs $\tilde{c}_{Lj}$, with $j = (1..4)$. We create two hierarchies, one for the shading points $\mathfrak{T}_E$ and other for the VPLs $\mathfrak{T}_L$ (middle); in render time, our technique traverses both trees creating an implicit hierarchy $\mathfrak{T}_{E,L}$ by combining internal nodes of $\mathfrak{T}_E$ and $\mathfrak{T}_L$ (right) until the termination criteria is found. The dashed lines represent final combined nodes in the tree, representing that e.g. the VPL cluster $\tilde{c}_{L6}$ shades the full cluster $\tilde{c}_{E3}$. Colored dots in $\mathfrak{T}_{E,L}$ represent these combinations in the combined hierarchy. Figure after [WABG06].*

criteria, then it is refined adding a new set of nodes in the current cut. Note that since we are using octrees, the combined node $\tilde{c}_{E,L}$ would potentially have 64 children (8 children of the light node $\tilde{c}_L$ times 8 children of the shading points node $\tilde{c}_E$); in order to avoid cut sizes growing too large, we instead refine only one of the base trees based on the heuristic described later in the section. The algorithm ends when all nodes in the current cut meet the stop criteria, and therefore the optimal cut has been found.

Termination criteria The main termination criteria used on a node $\tilde{c}_{E,L} = (\tilde{c}_L, \tilde{c}_E)$ is based on the projected solid angle of the light or shading point cluster that form $\tilde{c}_{E,L}$ over the other. The intuition is that smaller or far-away clusters have smaller projected solid angle, and therefore the introduced error is lower. For example, if the projected solid angle $\Omega(\tilde{c}_L, \tilde{c}_E)$ from a the light node into the shading points node is smaller than a user-specified threshold ε_L then the criteria is met. We compute the solid angle as $\Omega(\tilde{c}_L, \tilde{c}_E) = \frac{A_{\tilde{c}_L}(\omega_\mathbf{o})}{d^2}$, where $\omega_\mathbf{o}$ is the direction from the center of $\tilde{c}_L$ to the center of $\tilde{c}_E$, d is the distance between both centers, and $A_{\tilde{c}_L}(\omega_\mathbf{o})$ is the projected area of $\tilde{c}_L$ stored in spherical harmonics.

While this heuristic works well in general, it produces some artifacts due to the errors introduced by encoding the area in spherical harmonics (see Figure 4). To overcome this, we impose an additional condition: that the projected solid angle of the volume bounding the cluster $\tilde{\Omega}(\tilde{c}_L, \tilde{c}_E)$ is smaller than a threshold $\tilde{\varepsilon}_L$, such as $\varepsilon_L \leq \tilde{\varepsilon}_L$. This additional threshold is a weaker termination condition. Note that $\tilde{\Omega}(\tilde{c}_L, \tilde{c}_E) = \frac{\pi r^2}{d^2}$ is constant in all directions, and only depends on the radius of the bounding sphere of the cluster r and the distance between clusters d.

While these two conditions work analogously for the projected solid angle $\Omega(\tilde{c}_E, \tilde{c}_L)$ from the shading points node to the light node, we found that the stronger constraint ε_E does not improve the results significantly; we therefore only use the weak constraint $\tilde{\Omega}(\tilde{c}_E, \tilde{c}_L) \leq \tilde{\varepsilon}_E$, which allows us to avoid the storage of the area via spherical harmonics encoding in $\mathfrak{T}_E$.

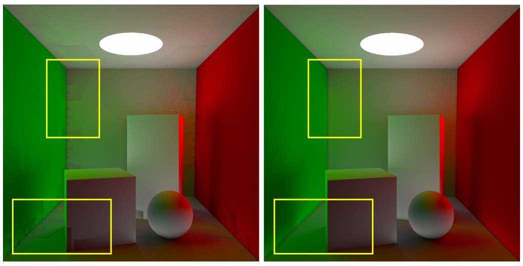

Figure 4: *The use of the weaker termination $\tilde{\Omega}(\tilde{c}_L, \tilde{c}_E)$ (right) allows reducing the artifacts introduced by the strict criteria $\Omega(\tilde{c}_L, \tilde{c}_E)$ due to the use of an approximation based on spherical harmonics encoding to store the projected area $A_{\tilde{c}_L}(\omega_\mathbf{o})$, and that is visible in the form of banding (left).*

Finally, we introduce an additional condition for the shading points, imposing that if the angle $\theta_{\tilde{c}_E}$ of the cone bounding the normals inside the cluster is wider than a threshold α then the cluster should be refined. This allows us to eliminate artifacts happening in clusters with strong normal variations (e.g. corners or surfaces with high curvature).

Refinement heuristic If the current node $\tilde{c}_{E,L} = (\tilde{c}_L, \tilde{c}_E)$ does not meet the termination criteria, then it needs to be refined. However, as discussed before, the naïve approach of adding into the cut all combinations between the children nodes of $\tilde{c}_L$ and $\tilde{c}_E$ would be too inefficient. Instead, we refine only one of the subtrees, while leaving the node of the other tree. In order to select which tree $\mathfrak{T}_L$ or $\mathfrak{T}_E$ should be refined, we base on one heuristic for each tree h_L and h_E respectively, so that we select refining the light tree if $h_L > h_E$,

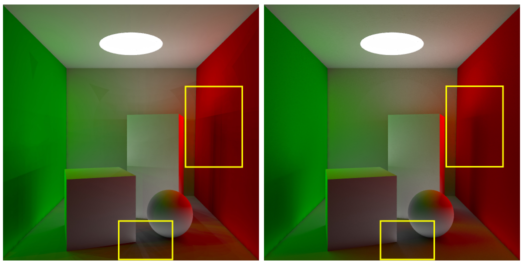

Figure 5: *Sampling the visibility $V(\tilde{c}_E,\tilde{c}_L)$ using a deterministic shadow connection between the center of the clusters introduce noticeable artifacts in the form of banding (left), due to correlation between the samples, which introduces strong discontinuities in the Fourier spectra of the signal [RAMN12]. By using stochastic sampling in the solid angle $\Omega(\tilde{c}_L,\tilde{c}_E)$ as viewed from $\tilde{c}_E$ we remove these artifacts (right), without introducing neither significant noise.*

and the shading points tree otherwise, with

$$h_L = \max\left(\frac{\Omega(\tilde{c}_L,\tilde{c}_E)}{\varepsilon_L}, \frac{\tilde{\Omega}(\tilde{c}_L,\tilde{c}_E)}{\tilde{\varepsilon}_L}\right), \quad (7)$$

$$h_E = \begin{cases} \infty & \text{if } \theta_{\tilde{c}_E} \geq \alpha \\ \frac{\tilde{\Omega}(\tilde{c}_E,\tilde{c}_L)}{\tilde{\varepsilon}_E} & \text{if } \theta_{\tilde{c}_E} < \alpha \end{cases}. \quad (8)$$

Computing the illumination in a node In order to compute the illumination from $\tilde{c}_L$ to $\tilde{c}_E$ we need to evaluate the term inside the sumation in Equation (6). Evaluating this term is straight forward when both $\tilde{c}_L$ and $\tilde{c}_E$ are a leaf in the tree, since it comes down to the standard shadow connection between the VPL and the shading point being illuminated. In our case, in order to avoid artifacts produced by the weak singularity due to the inverse squared distance in $G(\tilde{c}_L,\tilde{c}_E)$ (4), specially visible in the corners and in non-Lambertian surfaces [KFB10], we transform the VPLs into *virtual spherical lights* [HKWB09], which diffuses the contribution of a VPL into its local neighborhood, effectively removing these artifacts.

On the other hand, when at least one of the nodes $\tilde{c}_L$ or $\tilde{c}_E$ is an internal node of the tree (a cluster) is more challenging, since the illumination is no longer computed point-to-point, and instead several shading points are being illuminated at the same time from a light cluster. Since computing the radiance reflected from each point of the cluster $\tilde{c}_E$ to the camera using a single value might result in severe artifacts, we instead compute the incoming radiance $L_i(\tilde{c}_E,\tilde{c}_L)$ at the cluster. This allows us to compute the final reflected radiance on a per leaf basis, which reduces significantly any possible banding. We compute $L_i(\tilde{c}_E,\tilde{c}_L)$ as:

$$L_i(\tilde{c}_E,\tilde{c}_L) \approx I_{\tilde{c}}(\tilde{c}_E,\tilde{c}_L)\Omega(\tilde{c}_L,\tilde{c}_E)V(\tilde{c}_E,\tilde{c}_L), \quad (9)$$

where $I_{\tilde{c}}(\tilde{c}_E,\tilde{c}_L)$ is the sum of the direction-resolved emission of the cluster's VPLs, and $\Omega(\tilde{c}_L,\tilde{c}_E)$ accounts for the foreshortening and the inverse squared distance of the geometric term. All these values are precomputed in the tree construction process. On the other hand, the visibility term $V(\tilde{c}_E,\tilde{c}_L)$ must be computed in run time. This value is, as opposed to point-to-point samples, not binary, since the clusters might be partially occluded. A simple approach of computing the visbility deterministicaly at a fixed sampling point results into banding artifacts due to the correlation between pixel samples [RAMN12]; instead, we randomly sample the visibility along the projected solid angle $\Omega(\tilde{c}_L,\tilde{c}_E)$ of cluster $\tilde{c}_L$ viewed from $\tilde{c}_E$. This introduces a little variance, but it is practically imperceptible, and it removes the banding artifacts (Figure 5). This directionally-dependent incoming energy is stored in the cluster of shading points using spherical harmonics, and propagated at the end to the leafs for reconstructing the final image.

5. Results

We have implemented our technique on top of PBRT [PH10]. Images are rendered using one-bounce indirect illumination, with the VPLs generated by uniformly sampling the surfaces in the scene and computing the radiance from the light sources to each point, although the method would work similarly with multiple-bounces global illumination computed using particle tracing [Kel97]. We compare our method against a variant of the original PBGI [Chr08] using the same visibility computations as the described in Section 4; we choose this comparison since PBGI is widely used in production, and our work tries to improve its scalability with respect to the number of pixels. All tests have been performed on a Intel i7 4790k at 4 GHz with 16 GB of main memory. Timings can be found in Table 1.

As shown in Figures 1 and 6 (left), our method is capable of rendering large scenes with complex illumination and geometry featuring multi-scale details and bump-mapping, with speed-ups up to x2 with respect to light clustering alone. This gaining on perfomance increases with the numberof shading points (e.g. higher resolution images), as shown in Figure 7; while for low resolution images our method introduces the cost of traversing the hierarchy of shading points, which makes it more inefficient than PBGI for low-resolution images, as the image increases the importance of the fixed costs is reduced. This makes our method very suitable to render high-resolution images with several distributed samples per pixel. This is the case of the rendering of multiview content such as stereo images or light fields, as exampled in Figure 6 (right), where our technique excels on taking advantage of the large amount of redundancy on the shading points.

6. Discussion & Future Work

In this work we propose a new method for scalable diffuse global illumination based on VPLs based on a bidirectional

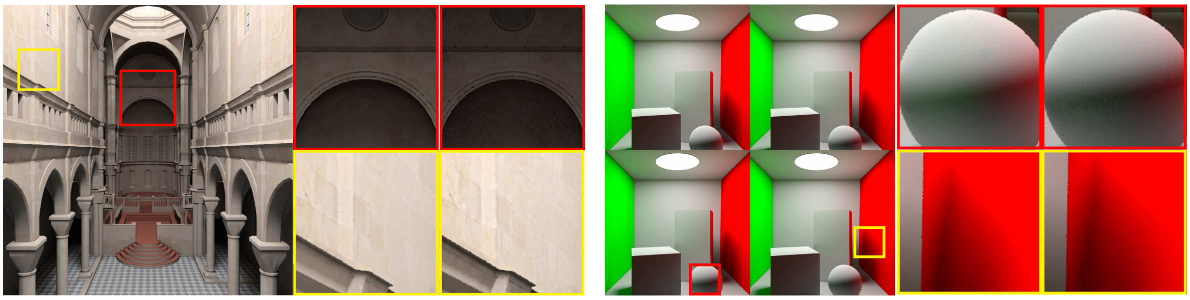

Figure 6: *Renders of the* Sibenik *(left) and* Light Field *(right) scenes. The insets show a comparison between our method (right) and PBGI [Chr08] (left), with our method computing the images about 77% (*Sibenik*) and 39% (*Light Field*) faster than PBGI. Our method is specially good at rendering light fields, given their large number of pixels and the intrinsic coherence between them.*

Figure	Scene	# Points	# VPLs	PBGI	Our
1	*San Miguel*	$1024^2 \times 2$	8M	10678	8299
1	*Sponza*	$2048^2 \times 2$	5M	15134	7690
6	*Sibenik*	$1024^2 \times 2$	1M	4905	3769
6	*Light Field*	$4096^2 \times 2$	0.1M	12936	5056
8	*Bunny*	$2048^2 \times 2$	5M	3515	5858

Table 1: *Comparison of the cost (in seconds) between PBGI [Chr08] and our technique for the render examples shown in the paper. The number of points are specified in number of pixels times the number of samples per pixel.*

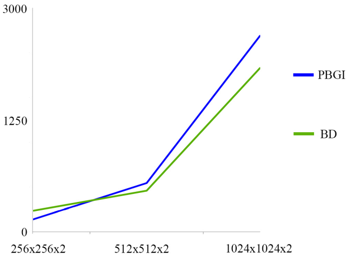

Figure 7: *Comparison of the cost (in seconds) between PBGI with $\varepsilon_L = 0.01$ and our technique with $\varepsilon_L = 0.01$, $\tilde{\varepsilon}_E = 0.0001$ and $\alpha = 0.1$. In both cases, $\tilde{\varepsilon}_L = 0.1$ is used for varying number of shading points. The test is performed in the* Cornell Box *scene shown in Figure 4 with 100K VPLs.*

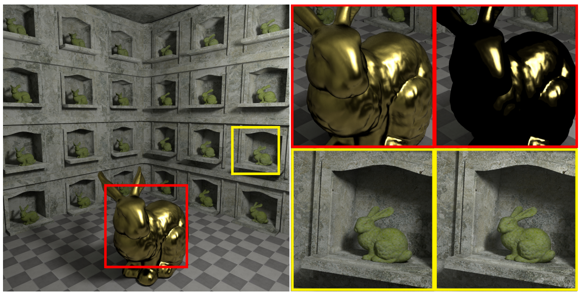

Figure 8: *Our bidirectional technique (top-right inset) assumes diffuse light transport, which might result into energy losses when rendering glossy materials, which can be handled by other scalable VPL-based techniques (e.g. PBGI, top-left inset).*

clustering of both the VPLs and the shading points. As opposed to previous works [WABG06, WHB*13], our technique works with the full set of samples in the scene, which allows leveraging the spatial coherence between the shading points. Our technique allows sublinear times in both the number of VPLs and shading points, which makes it a very good choice for rendering high-resolution images with distributed effects, and is specially suitable for multiview images such as stereo images or light fields. In addition, given that our technique generalizes previous work on scalable VPL-based global illumination, it can benefit from particular improvements to make it more efficient in memory [BB12], or to work in out-of-core schemes [KTO11] in large scale distributed environments.

In the presence of curved surfaces, some quantization in the spatial domain may be visible for low numbers of VPLs. A more sophisticated strategy [KCLU07] can be used to significantly improve the reconstruction in this high-dimensional domain, as already shown in other problems with high dimensionality [JMG11].

Our proposed bidirectional cluster assumes diffuse interreflections, as shown in Figure 8. This can be extended generating new diffuse shading points by sampling the glossy reflections, as in Stochastic Progressive Photon Mapping [HJ09]; while this would be gracefully handled by our system, it would potentially introduce some noise in the final image and increase the cost of the algorithm. Warping the spherical domain of the point samples to match better the specular lobes [REG*09], together with more sophisticated clustering algorithms, is a promising solution for this problem. However, note that the directional frequency of the interreflections would be limited to moderately glossy, given the limitations of the VPLs handling this light paths [KFB10].

Finally, our work uses a simple terminating criteria, fundamentally based on the solid angle; while this condition is intuitive for a user, it requires a very low threshold to avoid artifacts in localized points in the scene, which is suboptimal in large parts of the scene. Including a more sophisticated criteria, based on cluster's contribution in the final image, or using an adaptive boundary condition leveraging low-level [FPSG01, DBD*07] and high-level [RFWB07, HČA*12, JVS*12] perceptual knowledge are promising avenues for improvement.

Acknowledgements

We thank the reviewers for their insightful comments and suggestions. This research has been partially funded by Spanish Ministry of Science and Technology (project LIGHTSLICE), the BBVA Foundation, and a gift from Adobe. Diego Gutierrez is additionally supported by a Google Faculty Research Award.

References

[BB12] Buchholz B., Boubekeur T.: Quantized point-based global illumination. *Computer Graphics Forum 31*, 4 (2012). 2, 7

[Bun05] Bunnell M.: Dynamic ambient occlusion and indirect lighting. *GPU Gems 2* (2005). 2

[Chr08] Christensen P.: *Point-based approximate color bleeding*. Tech. Rep. 08-01, Pixar, 2008. 1, 2, 3, 4, 6, 7

[CLSS97] Christensen P. H., Lischinski D., Stollnitz E. J., Salesin D. H.: Clustering for glossy global illumination. *ACM Trans. Graph. 16*, 1 (1997). 4

[CPC84] Cook R. L., Porter T., Carpenter L.: Distributed ray tracing. In *Proceedings of SIGGRAPH'84* (1984). 2, 4

[DBB06] Dutre P., Bala K., Bekaert P.: *Advanced global illumination*. AK Petes, 2006. 1, 2

[DBD*07] Drettakis G., Bonneel N., Dachsbacher C., Lefebvre S., Schwarz M., Viaud-Delmon I.: An interactive perceptual rendering pipeline using contrast and spatial masking. In *Proceedings of EGSR'07* (2007). 8

[DHS*05] Durand F., Holzschuch N., Soler C., Chan E., Sillion F. X.: A frequency analysis of light transport. *ACM Trans. Graph. 24*, 3 (2005). 2

[DKH*14] Dachsbacher C., Křivánek J., Hašan M., Arbree A., Walter B., Novák J.: Scalable realistic rendering with many-light methods. *Computer Graphics Forum 33*, 1 (2014). 1

[FPSG01] Ferwerda J. A., Pattanaik S., Shirley P., Greenberg D. P.: A model of visual masking for computer graphics. In *Proceedings of SIGGRAPH'01* (2001). 8

[HČA*12] Herzog R., Čadík M., Aydčin T. O., Kim K. I., Myszkowski K., Seidel H.-P.: Norm: No-reference image quality metric for realistic image synthesis. *Computer Graphics Forum 31*, 2pt3 (2012). 8

[HJ09] Hachisuka T., Jensen H. W.: Stochastic progressive photon mapping. *ACM Trans. Graph. 28*, 5 (2009). 7

[HKWB09] Hašan M., Křivánek J., Walter B., Bala K.: Virtual spherical lights for many-light rendering of glossy scenes. *ACM Trans. Graph. 28*, 5 (2009). 6

[HPB07] Hašan M., Pellacini F., Bala K.: Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. 26*, 3 (2007). 2, 3

[HREB11] Holländer M., Ritschel T., Eisemann E., Boubekeur T.: Manylods: Parallel many-view level-of-detail selection for real-time global illumination. *Computer Graphics Forum 30*, 4 (2011). 2

[HVAPB08] Hašan M., Velázquez-Armendáriz E., Pellacini F., Bala K.: Tensor clustering for rendering many-light animations. *Computer Graphics Forum 27*, 4 (2008). 2, 3

[JMG11] Jarabo A., Masia B., Gutierrez D.: Efficient propagation of light field edits. In *Proc. of SIACG 2011* (2011). 7

[JMM*14] Jarabo A., Marco J., Muñoz A., Buisan R., Jarosz W., Gutierrez D.: A framework for transient rendering. *ACM Trans. Graph. 33*, 6 (2014). 2

[JVS*12] Jarabo A., Van Eyck T., Sundstedt V., Bala K., Gutierrez D., O'Sullivan C.: Crowd light: Evaluating the perceived fidelity of illuminated dynamic scenes. *Computer Graphics Forum 31*, 2 (2012). 8

[Kaj86] Kajiya J. T.: The rendering equation. In *Proceedings of SIGGRAPH'86* (1986). 3

[KCLU07] Kopf J., Cohen M. F., Lischinski D., Uyttendaele M.: Joint bilateral upsampling. *ACM Trans. Graph. 26* (2007). 7

[Kel97] Keller A.: Instant radiosity. In *Proceedings of SIGGRAPH'97* (1997). 1, 3, 6

[KFB10] Křivánek J., Ferwerda J. A., Bala K.: Effects of global illumination approximations on material appearance. *ACM Trans. Graph. 29*, 4 (2010). 6, 8

[KKG*14] Křivánek J., Keller A., Georgiev I., Kaplanyan A., Fajardo M., Meyer M., Nahmias J.-D., Karlík O., Canada J.: Recent advances in light transport simulation: Some theory and a lot of practice. In *ACM SIGGRAPH 2014 Courses* (2014). 1, 2

[KTO11] Kontkanen J., Tabellion E., Overbeck R. S.: Coherent out-of-core point-based global illumination. *Computer Graphics Forum 30*, 4 (2011). 2, 7

[LH96] Levoy M., Hanrahan P.: Light field rendering. In *Proceedings of SIGGRAPH'96* (1996), ACM, pp. 31–42. 2

[LW93] Lafortune E. P., Willems Y. D.: Bi-directional path tracing. In *Proceedings of CompuGraphics* (1993), vol. 93, pp. 145–153. 2

[MWA*13] Masia B., Wetzstein G., Aliaga C., Raskar R., Gutierrez D.: Display adaptive 3d content remapping. *Computers & Graphics 37*, 8 (2013). 2

[OP11] Ou J., Pellacini F.: Lightslice: matrix slice sampling for the many-lights problem. *ACM Trans. Graph. 30*, 6 (2011), 179. 3

[PH10] Pharr M., Humphreys G.: *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2010. 6

[RAMN12] Ramamoorthi R., Anderson J., Meyer M., Nowrouzezahrai D.: A theory of monte carlo visibility sampling. *ACM Trans. Graph. 31*, 5 (2012). 6

[REG*09] RITSCHEL T., ENGELHARDT T., GROSCH T., SEIDEL H.-P., KAUTZ J., DACHSBACHER C.: Micro-rendering for scalable, parallel final gathering. *ACM Trans. Graph. 28*, 5 (2009). 2, 7

[RFWB07] RAMANARAYANAN G., FERWERDA J., WALTER B., BALA K.: Visual equivalence: towards a new standard for image fidelity. *ACM Trans. Graph. 26*, 3 (2007). 8

[SAG94] SMITS B., ARVO J., GREENBERG D.: A clustering algorithm for radiosity in complex environments. In *Proceedings of SIGGRAPH '94* (1994). 4

[VG94] VEACH E., GUIBAS L.: Bidirectional estimators for light transport. In *Proceedings of EGWR'94*. 1994. 2

[VG95] VEACH E., GUIBAS L. J.: Optimally combining sampling techniques for monte carlo rendering. In *Proceedings of SIGGRAPH'95* (1995). 2

[WABG06] WALTER B., ARBREE A., BALA K., GREENBERG D. P.: Multidimensional lightcuts. *ACM Trans. Graph. 25*, 3 (2006). 2, 3, 5, 7

[WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: a scalable approach to illumination. *ACM Trans. Graph. 24*, 3 (2005). 2, 3, 4

[WHB*13] WANG B., HUANG J., BUCHHOLZ B., MENG X., BOUBEKEUR T.: Factorized point based global illumination. *Computer Graphics Forum 32*, 4 (2013). 2, 3, 7

[WKB12] WALTER B., KHUNGURN P., BALA K.: Bidirectional lightcuts. *ACM Trans. Graph. 31*, 4 (2012). 2