Eurographics Symposium on Rendering 2009
Hendrik P. A. Lensch and Peter-Pike Sloan
(Guest Editors)

Volume 28 (2009), Number 4

Hierarchical Image-Space Radiosity for Interactive Global Illumination

Greg Nichols[†], Jeremy Shopf[‡], and Chris Wyman[†]

[†]University of Iowa, USA [‡]AMD and Firaxis Games

Abstract

We introduce image-space radiosity *and a hierarchical variant as a method for interactively approximating diffuse indirect illumination in fully dynamic scenes. As oft observed, diffuse indirect illumination contains mainly low-frequency details that do not require independent computations at every pixel. Prior work leverages this to reduce computation costs by clustering and caching samples in world or object space. This often involves scene preprocessing, complex data structures for caching, or wasted computations outside the view frustum. We instead propose clustering computations in image space, allowing the use of cheap hardware mipmapping and implicit quadtrees to allow coarser illumination computations. We build on a recently introduced multiresolution splatting technique combined with an image-space lightcut algorithm to intelligently choose virtual point lights for an interactive, one-bounce instant radiosity solution. Intelligently selecting point lights from our reflective shadow map enables temporally coherent illumination similar to results using more than 4096 regularly-sampled VPLs.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—

1. Introduction

Global illumination presents a challenge for interactive rendering techniques, as reflected light from every surface potentially affects illumination at all other surfaces (as per the rendering equation [Kaj86]). Fortunately, diffuse indirect illumination generally varies in a low-frequency manner that allows simplifications without compromising perceived rendering quality. This allows reuse of illumination samples via caching schemes [GKBP05, GSHG98], sample clustering to reduce per-patch [HSA91, SAG94] and per-pixel [LZT*08, WFA*05] computations, and rendering indirect illumination at lower resolution than direct light [DS06, SGNS07].

Many of these techniques first appeared as acceleration techniques for batch rendering pipelines, where geometry positions are known in advance and data structure construction costs can be amortized over multiple frames. In interactive applications, though, users may move scene geometry arbitrarily, data structures may need to be built every frame, and time-to-image should remain under 30 milliseconds.

This paper introduces *image-space radiosity*, based on our observation that image-space techniques often have advantages over object and world-space techniques for interactive applications. In particular, image-space techniques divorce algorithmic and scene complexities and avoid wasting computations on off-screen portions of the scene. Because we aim for completely dynamic content, complex acceleration structures would be discarded each frame. Image-space techniques lend themselves to trivial quad-tree based hierarchical techniques that leverage hardware-accelerated mipmap creation and contain only relevant geometry (see Figure 1).

We use an instant radiosity [Kel97] approach that builds on recent multiresolution techniques [NW09, Wym08], reformulated for higher performance, to interactively render indirect illumination at an appropriate eye-space resolution. Combined with a technique similar to Lightcuts [WFA*05] and Light Pyramids [KO08], we propose a one-bounce radiosity algorithm where each fragment gathers illumination from a unique set of image-space light clusters selected to reduce fragment error below a user specified threshold. While we currently ignore visibility for indirect illumination, future work should enable incorporation of visibility approximations, such as imperfect shadow maps [RGK*08].

† E-mail: { gbnichol | cwyman }@cs.uiowa.edu
‡ E-mail: jshopf@gmail.com

Published by Blackwell Publishing, 9600 Garsington Road, Oxford OX4 2DQ, UK and 350 Main Street, Malden, MA 02148, USA.

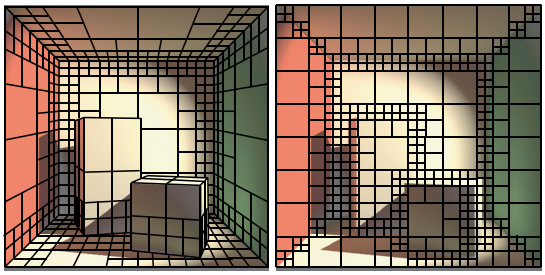


Figure 1: *Object-space patches (left) compared to image-space patches (right). Object-space patches remain constant through a simulation; image-space patches vary each frame and between eye and light space.*

2. Previous Work

Researchers have studied global illumination algorithms for decades, exploring techniques such as path tracing [Kaj86] and radiosity [GTGB84] to capture multi-bounce lighting. Over the years, improvements to both approaches have dramatically improved both quality and speed, but important differences remain. Path tracing techniques tend to introduce noise, especially with the sparse samples feasible in today's interactive applications. Radiosity-based techniques have difficulty with non-diffuse surfaces and often require a precomputed scene discretization. A discussion of offline rendering techniques is beyond the scope of this paper; we refer the reader to standard texts on the subject [DBB06, PH04].

Precomputed radiance transport. One interactive alternative precomputes light transport and compresses it using spherical harmonic [SKS02] or wavelet [KTHS06, NRH03] bases. While this enables dynamic environmental lighting, local lights [KAMJ05], moving geometry [IDYN07], and deformable objects [SGNS07] are complex to implement and do not easily scale to completely dynamic scenes.

Illumination caching. Another approach coarsely samples illumination using various sampling schemes [GKBP05, GSHG98, LZT*08]. Samples are stored in an acceleration structure and reused whenever possible. However, quality and speed depend on the sampling method and acceleration structure used. Scene modifications may fade in, as sample computation frequently occurs asynchronous to display.

Interactive ray-based techniques. Over the past decade, a number of interactive ray tracing techniques that support global illumination have emerged. Early techniques required massive supercomputers or PC clusters [WKB*02], but more recent approaches use graphics hardware to run interactively on commodity PCs [STK08, ZHWG08]. Unfortunately, the incoherent memory access patterns common in path tracing limit interactivity as scene sizes increase.

Approximate indirect illumination. Substituting cheap approximations for accurate illumination often suffices, especially for applications where speed concerns trump realism. Ambient occlusion [ZIK98] replaces global illumination with a simple visibility heuristic. This approximation can be further simplified for interactive use [Bun05], and even coarse image-space occlusion approximations [BS09] give acceptable appearance.

Instant radiosity techniques. One popular interactive technique builds on instant radiosity [Kel97], which samples a subset of photon paths, placing *virtual point lights* (or VPLs) where these paths intersect the scene. These VPLs are used along with the actual light sources to illuminate the scene during a final render pass, in which both lights and VPLs contribute only direct illumination.

Recent work has greatly improved performance of instant radiosity-based techniques. Techniques based on reflective shadow maps (RSMs) [DS05] emit VPLs via rasterization from the light view, and render using either gathering [DS05] or scattering [DS06]. However, these techniques ignore visibility for indirect illumination.

Progressive and incremental variants [GKBP05, LSK*07] update visibility for a few VPLs each frame, which works well unless the scene changes quickly. Interleaved sampling [SIMP06] allows simultaneous coarse computation of multiple VPL contributions. Imperfect shadow maps [RGK*08] simultaneously sample visibility from 1024 lights. While this imperfectly approximates visibility, it likely suffices for most indirect illumination.

Still, instant radiosity methods often exhibit problems:

1. Speed scales linearly with number of VPLs.
2. Temporal incoherence from VPL variations.
3. Coarse illumination blurs high frequency details. While indirect light exhibits low frequencies, interactions with higher curvature surfaces introduce high frequencies.
4. Splatting-based techniques artificially limit VPL contributions to regions inside the splat; enlarging the splat size dramatically increases costs due to overdraw.

Our work addresses these issues by combining two ideas. Multiresolution splatting [NW09] significantly reduces overdraw by rendering illumination at varying frequencies, ranging from coarse 64^2 pixel clusters to individual pixels. Light clustering [WFA*05, KO08] allows intelligent selection of VPLs. Near a light cluster, its VPLs are used individually. Further away, a clustered contribution suffices. Thus, speed scales with the number of VPLs used *per-pixel* instead of the number used in the entire image.

3. Image-Space Radiosity

Our work builds on a recent multiresolution splatting technique [NW09], so first we briefly review that work. We then describe a stencil-based rendering scheme that improves performance of multiresolution splatting by nearly an order of magnitude. Finally we describe two light-clustering techniques that extend this to image-space radiosity. These methods cluster lights either on a per-frame basis or select VPL clusters individually for every multiresolution fragment.

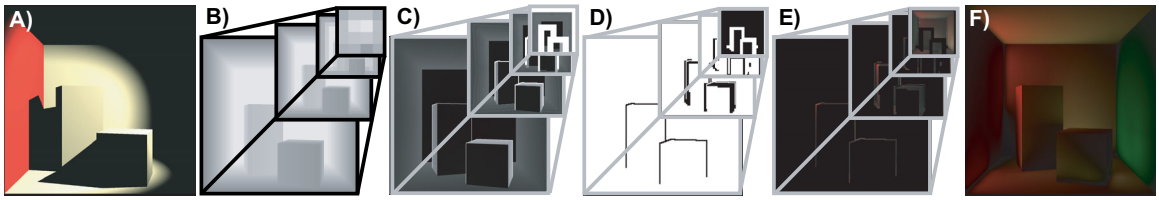


Figure 2: *Multiresolution illumination splatting begins by drawing (a) just the direct illumination, as seen from the eye. Creation of a (b) multiresolution depth map occurs simultaneous with creation of a (c) depth derivative max mipmap. Locations with high depth or normal discontinuities (found using a similar normal mipmap) require higher resolution detail. Our improved approach first computes a (d) multiresolution stencil buffer that stores relevant fragments at each level of detail, then gathers illumination from VPLs at all valid fragments. The resulting (e) multiresolution illumination buffer is upsampled, interpolated, and combined into the (f) final indirect illumination (brightened for display).*

3.1. Multiresolution Splatting of Indirect Illumination

Since diffuse indirect illumination generally varies slowly, coarsely sampling lighting in image-space often suffices. However, image-space discontinuities and normal variations introduce high frequencies missed by naive coarse sampling. Geometry-aware framebuffers [YSL08] handle discontinuities using bilateral filtering, but they still ignore high-frequency details introduced by normal variations.

Multiresolution splatting [NW09] proposes rendering illumination at multiple scales using an instant radiosity-based splatting scheme [DS06, Kel97]. In image-space regions where depth and normal variations introduce high-frequency details, illumination is finely sampled. As illumination variations decrease, rendering occurs at progressively coarser levels, creating a *multiresolution illumination buffer*. For each VPL, a multiresolution splat consisting of these samples is accumulated into the multiresolution buffer. This illumination buffer is then flattened via an additive upsampling and interpolation scheme to achieve indirect illumination at full-screen resolution. Figures 2 and 3 depict this process.

3.2. Stencil-Based Multiresolution Illumination

Prior multiresolution rendering techniques [KO08, NW09, Wym08] use a multipass geometry shader to refine a list of primitives and corresponding fragment locations. Unfortunately, outputting variable length computation results into temporary memory and GPU pass overhead leads to relatively poor performance, even though speeds exceed comparable single-resolution techniques.

We propose a stencil-based approach that vastly improves performance. Figure 3 provides a high level comparison between our approach and prior work [NW09]. We first replace the costly iterative refinement with a single pass that selects appropriate fragments in a stencil buffer. We then reformulate the final pass to gather indirect illumination rather than splatting illumination. This provides an immediate performance improvement, in addition to enabling the hierarchical VPL selection described in Section 3.5.

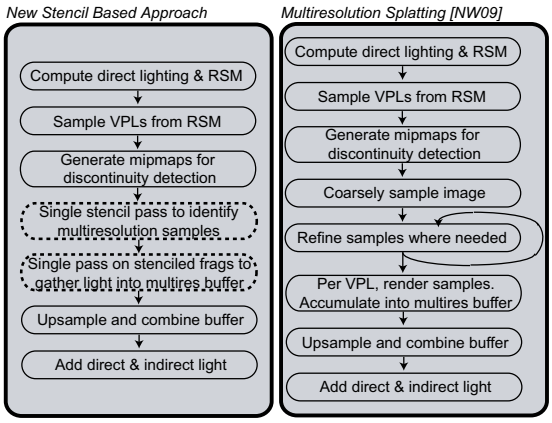


Figure 3: *Our stencil-based multiresolution illumination differs from prior work in two important ways. First, we avoid an iterative multi-pass refinement, costly on current GPUs. Second, we reformulate the final pass as a gather instead of a scatter, which proves more cache-friendly.*

3.2.1. One Pass Stencil Refinement

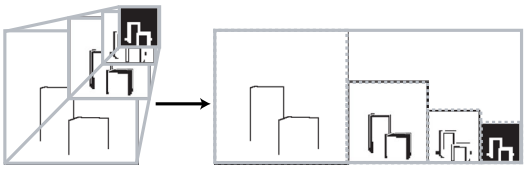
By observing that GPUs internally use hierarchical z-buffers [GKM93] that enable hardware-controlled early stencil culling, we can replace the software refinement loop depicted in Figure 3 with an equivalent, optimized technique that leverages this GPU hardware. Multiresolution splatting [NW09] subdivides image-space patches containing depth or normal discontinuities, finding the appropriate resolution for each piece of the image as seen in Figure 1. This process works as follows:

```
patches ← CoarseImageSampling();
for (i=1 to numRefinementPasses) do
  for all (p ∈ patches) do
    if ( NoDiscontinuity( p ) ) then
      continue;
    patches ← (patches − {p});
    patches ← (patches ∪ SubdivideIntoFour( p ) );
```

The resulting patch list is used as a vertex array identifying which fragments in the multiresolution buffer will accu-

mulate light. When splatting, this causes each patch to generate one vertex and one fragment for each VPL.

We simplify this by noting that each refinement pass does not truly depend on prior passes and can be parallelized. First, we flatten our multiresolution buffer into a 2D image:



Then we render a single full-screen quad over this flattened image to set a stencil bit for valid multiresolution fragments:

```
for all (fragments f ∈ image) do
  if ( ∀i, f ∉ MipmapLevel( i ) ) then
    continue; // Fragment not actually in multires buffer
  i ← CurrentMipmapLevel( f );
  if ( IsDiscontinuity( f, i ) ) then
    continue; // Patch not valid (needs subdivision)
  if ( NoDiscontinuity( f, i+1 ) ) then
    continue; // Coarser patch did not need subdivision
  SetStencil( f );
```

This approach observes multiresolution fragments are required only when they do not need subdivision, but corresponding next-coarser fragments do.

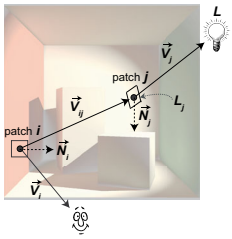
Once the appropriate stencil bits have been set, every multiresolution "splat" can be accomplished by drawing a single full-screen quad with stenciling enabled. Early stencil culling avoids generating fragments in sparsely populated regions of the illumination buffer, and we avoid explicitly creating and repeatedly processing the list of patches.

3.2.2. Gathering Illumination

In an instant radiosity-based algorithm, indirect illumination may either be scattered to pixels via splatting or gathered to pixels during a single pass over all pixels. If all VPLs affect all patches, as in our work, scattering and gathering give results of equal quality. Prior work where VPLs illuminate only a subset of the image [DS06] achieve better performance via splatting. More recent techniques achieved better performance via gathering [RGK*08]. We also found gathering to perform better.

To improve cache coherency during this gather, we first create a *VPL cache*, which stores a list of VPL positions, normals, and colors. This avoids fragments fetching VPLs from incoherent locations in the RSM and condenses all relevant data into a single texture.

When a fragment passes the stencil test, we gather contributions of all VPLs to the corresponding eye-space patch. To calculate the contribution from each light-space patch, we approximate the patch-to-patch form factor using Wallace et al.'s [WEH89] disk-to-point approximation:



L	Scene light source
E	Viewpoint
N_i	Surface normal at patch i
$\vec{V}_i$	Vector from i to the eye E
N_j	Surface normal at patch j
$\vec{V}_j$	Vector from j to the light L
L_j	VPL corresponding to patch j
$\vec{V}_{ij}$	Vector from patch i to patch j
$\vec{V}_{ji}$	Vector from patch j to patch i

Figure 4: *Notation used for image-space radiosity.*

$$F_{j\to i} = \frac{A_j(\vec{N}_i \cdot \vec{V}_{ij})(\vec{N}_j \cdot \vec{V}_{ji})}{\pi\|\vec{V}_{ij}\|^2 + A_j}, \tag{1}$$

where patches i and j are, respectively, the multiresolution fragment seen from the eye and the current VPL, L_j. See Figure 4 for a visual depiction of the area A_j, the normals $\vec{N}_i$ and $\vec{N}_j$ and the vectors $\vec{V}_i$, $\vec{V}_j$, $\vec{V}_{ij}$, and $\vec{V}_{ji}$. We do not consider visibility, so we omit the visibility from Equation 1.

Every VPL, L_j, corresponds to some block of pixels in the reflective shadow map. Each block of pixels, and thus each L_j, represents some solid angle ω_j. The total intensity I_j of L_j relates to the total intensity I of the light L as follows:

$$I_j = I\frac{\omega_j}{4\pi}. \tag{2}$$

Additionally, we can approximate area A_j based upon the solid angle ω_j and distance from L:

$$A_j \approx \omega_j\|V_j\|^2. \tag{3}$$

Finally, for ease of computation we make the approximation that all RSM texels have equal solid angle. For slightly reduced performance, this approximation can be eliminated. Each texel t thus represents solid angle ω_t:

$$\omega_t \approx \frac{\omega_{frustum}}{4\pi}\frac{1}{\mathrm{RSM}_{res}^2}. \tag{4}$$

As we use a 256^2 RSM with a 90° field-of-view, $\omega_t \approx 1/(6\times 256^2)$. The solid angle ω_j represented by each VPL depends linearly on the number of RSM texels it represents.

We obtain the indirect color for each fragment, corresponding to an eye-space patch i, by summing over all VPLs:

$$C_i^{indirect} = \rho_i \sum_j I_j \rho_j F_{j\to i}, \tag{5}$$

where ρ_i and ρ_j are the diffuse colors of patches i and j.

After gathering illumination into the multiresolution illumination buffer, we produce a combined, interpolated indirect illumination image using the approach described by previous work [NW09]. We combine this with the direct illumination C_i^{direct} to produce the final result.

Figure 5: *Temporal coherence between two successive animation frames. From left to right: the brightened indirect illumination, frame-to-frame difference using RSMs with 256 VPLs, difference using image-space radiosity, and difference using hierarchical image-space radiosity.*

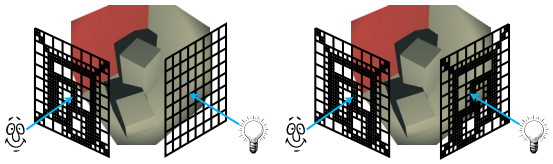


Figure 6: *(Left) Multiresolution splatting has temporal coherence issues when light samples jump suddenly from one surface to another. This is due to poor light-space patch selection. (Right) Using multiresolution VPLs lessens this problem, as VPLs near boundaries split into multiple, less important patches.*

3.3. Multiresolution VPL Clusters

This stencil-based technique renders indirect illumination significantly faster than prior work, but it still exhibits temporal incoherence. As geometry and lights move around the scene, the sampled VPLs can jump suddenly from one surface to another, as illustrated in Figure 5.

While this is clearly a sampling issue, it can also be viewed in another light (see Figure 6). Typically, VPLs are selected by quasi-Monte Carlo ray casting or sampling on a regular grid, without regard to image discontinuities that introduce high frequencies. Effectively, this means light-space patches may contain multiple surfaces! Thus, point sampling patches to find a single representative VPL can give significant variations from similar viewpoints.

We propose subdividing VPL patches, just as we divide patches in eye-space. Each light-space patch will cover a single surface, with energy varying by patch size. As a VPL approaches a discontinuity its patch splits into smaller pieces, each contributing less. On the other side of the boundary, these pieces recombine to form larger patches.

We describe two light-clustering techniques. The first creates a set of light-space patches and corresponding VPLs each frame. All contribute energy to all eye-space patches, just as basic radiosity [GTGB84] evaluates form factors between all patch pairs. The second approach realizes that VPLs may contribute little energy to distant patches, and thus clusters VPLs independently for each eye-space patch, reducing per-fragment computation.

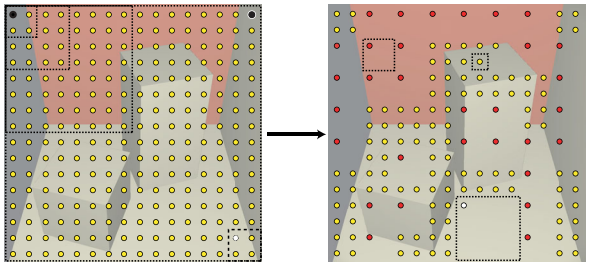


Figure 7: *(Left) Per-frame light patch construction starts from a dense RSM sampling, here shown at* 16^2*. Imagine an implicit mipmap on this sampling; all samples represent a texel at the finest resolution. Some, like the upper right point,* only *correspond to texels at the finest level. Others, like the white point, are valid at multiple levels. After implicitly clustering once, the white point represents the patch outlined by the dashed box. A few points, like the black sample, represent patches after numerous clustering stages, depicted by the dotted regions. A geometry shader performs our light clustering, testing each patch to identify any nearby discontinuities. If no nearby discontinuities are detected at any valid mipmap level, it is discarded. (Right) Here, the resulting VPLs are shown in yellow, red, or white, depicting regions after 0, 1, or 2 clustering stages. Three representative patches are displayed at varied resolutions. In our implementation, all clustering stages are computed in a single geometry shader pass.*

3.4. Basic Image-Space Radiosity

Our first clustering technique extends the stencil-based gather from Section 3.2. Instead of naively sampling the RSM to identify VPLs, we form VPLs by clustering RSM texels at various resolutions. Figure 7 depicts our clustering technique, which processes an initial set of finely sampled VPLs and discards those without nearby discontinuities. A geometry shader checks each VPL for nearby discontinuities at multiple resolutions to determine whether it is necessary or should be discarded. VPL clustering can be accomplished with a single geometry shader pass and does not create any VPLs, avoiding penalties from additional bandwidth or memory usage.

Discontinuity detection relies on light-space normal and depth max-min mipmaps, similar to those used during eye-space rendering. This produces a set of light-space patches of varying size: large patches in low-frequency areas, and smaller patches densely clustered around discontinuities.

After identifying our patches, corresponding VPLs are stored in the VPL cache. Each multiresolution eye-space fragment then gathers illumination as before, taking care to assign VPL solid angles ω_j based upon their hierarchy level, ensuring all contribute an appropriate amount of light.

3.5. Hierarchical Image-Space Radiosity

While partially addressing the undersampling depicted in Figure 6, this approach is not optimal. Although light patches may contain discontinuities that require a finer subdivision, it is not necessary to always *use* these finer patches. For distant eye-space fragments, coarser VPL sampling may introduce negligible error. Ideally, we would select the best VPL clusters for each fragment, using the finest light-space patches only where noticeable error otherwise occurs.

Our second clustering technique approaches this goal. When rendering fragments, we adaptively gather from light-space patches as coarse as 16^2 or as fine as the RSM resolution. We gather illumination for each fragment independently, with no assumed knowledge of which patches are required for any fragment.

Instead of creating a VPL cache each frame, we reorganize the full resolution RSM data into a *VPL tree*, a structure akin to the single texture, implicit mipmap depicted in Figure 7. The VPL tree encodes all necessary information for each VPL in a single flat texture, which is repeatedly accessed during when gathering illumination. Therefore, to facilitate cache coherence, we cluster VPLs representing coarser mipmap levels (i.e., similar to the clustering of Haar wavelet coefficients after applying a 2D wavelet transform). When gathering illumination, we start with the 16^2 coarsest light patches and adaptively subdivide whenever the error introduced by clustering exceeds a user-specified threshold.

3.5.1. VPL Tree Traversal

To determine when to subdivide patches, we must quantify image-space error. To do this, we find a computationally cheap, conservative bound to patch j's contribution to the fragment i, only refining light patches when this contribution exceeds a user threshold. First, from Equations 1 and 5, we know the contribution $C_{j\to i}$ from j to i:

$$C_{j\to i} = \rho_i \rho_j I_j \left(\frac{A_j (\vec{N}_i \cdot \vec{V}_{ij})(\vec{N}_j \cdot \vec{V}_{ji})}{\pi \|\vec{V}_{ij}\|^2 + A_j} \right). \tag{6}$$

Observing that $(\vec{N}_i \cdot \vec{V}_{ij})$, $(\vec{N}_j \cdot \vec{V}_{ji})$, ρ_i, and ρ_j always remain less than 1, we bound $C_{j\to i}$ by replacing them. We then apply Equations 2, 3, and 4 to rephrase this bound only in terms of the constant RSM resolution and the distances $\|\vec{V}_j\|$ from light to VPL and $\|\vec{V}_{ij}\|$ from i to j:

$$C_{j\to i} \le \frac{2I\|V_j\|^2}{3 \cdot \mathrm{RSM}_{res}^2 \|\vec{V}_{ij}\|^2 + 2\|V_j\|^2} \equiv \mathbb{B}(C_{j\to i}). \tag{7}$$

We refine patch j whenever the bound $\mathbb{B}(C_{j\to i})$ exceeds a user threshold τ. By reorganizing further, we identify a simpler per-frame constant T such that refinement occurs when:

$$\|\vec{V}_{ij}\|^2 \le \frac{2(I-\tau)}{3\tau \cdot \mathrm{RSM}_{res}^2} \|V_j\|^2 \equiv T\|V_j\|^2, \tag{8}$$

Multiresolution Stenciled Gathering	Cornell Box	Feline Scene	Sponza Atrium	Indoor Garden
Direct Light	1.8 ms	5.8 ms	2.8 ms	2.8 ms
Generate Reflective Shadow Map	0.9 ms	2.3 ms	1.4 ms	1.6 ms
Create Min-Max Mipmap	0.7 ms	0.7 ms	0.7 ms	0.7 ms
Multires Stencil Setup	0.7 ms	0.7 ms	0.7 ms	0.7 ms
Multires Light Gather	2.5 ms	3.5 ms	3.0 ms	6.6 ms
Upsample Indirect Illumination	1.6 ms	1.7 ms	1.3 ms	1.3 ms
Total	8.2 ms	14.7 ms	9.9 ms	13.7 ms

Multiresolution Splatting [NW09]	Cornell Box	Feline Scene	Sponza Atrium	Indoor Garden
Direct Light	1.8 ms	5.8 ms	2.8 ms	2.8 ms
Generate Reflective Shadow Map	0.9 ms	2.3 ms	1.4 ms	1.6 ms
Create Min-Max Mipmap	0.7 ms	0.7 ms	0.7 ms	0.7 ms
Geometry Shader Refinement	1.3 ms	1.7 ms	1.3 ms	1.6 ms
Multires Light Splatting	45.8 ms	67.0 ms	58.1 ms	130.0 ms
Upsample Indirect Illumination	1.6 ms	1.7 ms	1.3 ms	1.3 ms
Total	52.1 ms	79.2 ms	65.6 ms	138.0 ms

Table 1: *Comparison of per-frame costs for our new stencil-based multiresolution gathering and multiresolution splatting [NW09] using 256 VPLs. The "direct light" stage includes all other rendering overhead.*

allowing evaluation of each patch's refinement criteria with a single texture lookup, multiply, and comparison.

Point sampling the RSM to find $\|V_j\|^2$ may mistakenly end refinement too soon, especially on surfaces seen obliquely from the light. To avoid the resulting artifacts, we sample $\|V_j\|^2$ from a min-max mipmap and use the maximum depth over the entire patch as $\|V_j\|^2$.

4. Results and Discussion

Our implementation uses OpenGL and GLSL, with performance measured on a dual-core 3GHz Pentium 4 with a GeForce GTX 280. All images and results in this paper were generated at a resolution of 1024^2. Because we use a geometry shader only to discard geometry, which can be also done via clipping, older Shader Model 3 hardware can also run our techniques.

Figure 8 compares static results from our hierarchical technique, our stencil-based multiresolution gather, a full-resolution RSM gather, and a splatting-based RSM approach [DS06] that limits VPL influence to nearby fragments to reduce overdraw. Here, the threshold is chosen to generate the same number of fragments per splat as our multiresolution approach.

4.1. Stencil Refinement

Table 1 demonstrates the large performance gains achieved with our stencil refinement, as compared to splatting using a geometry shader for iterative patch refinement. Note the constant speed of the "stencil setup" phase, which identifies the multiresolution fragments. Comparatively, the iterative refinement of prior work varies in performance based upon the specified subdivision thresholds and the number of refined fragments.

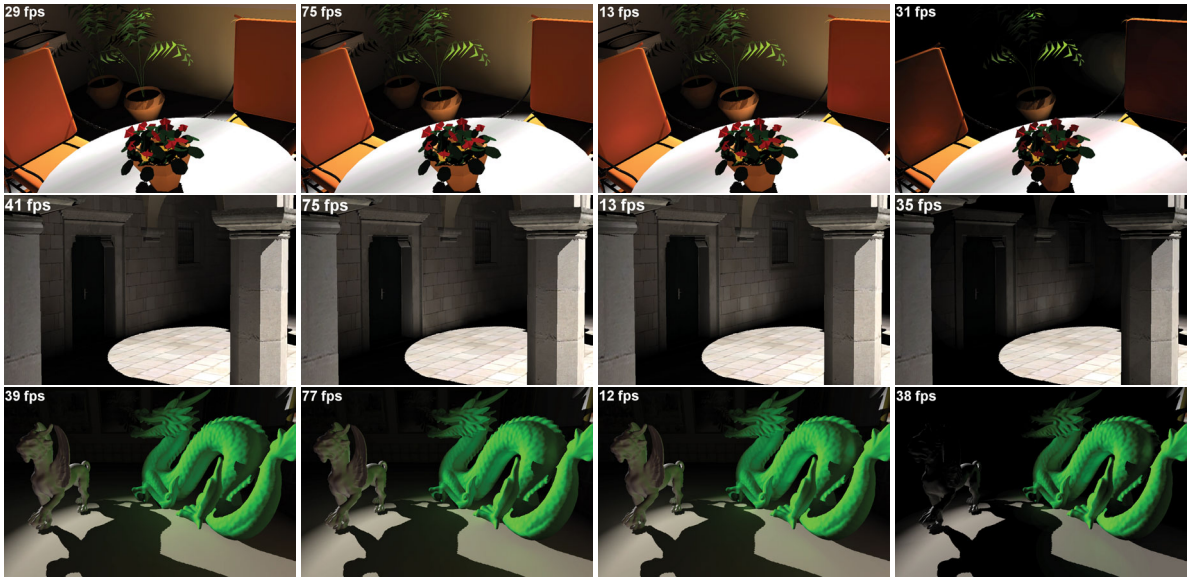


Figure 8: *Scenes depicted with four rendering techniques (left to right): hierarchical image space radiosity, our stencil-based multiresolution gather, RSMs using full-screen splats, and RSM splatting with reduced overdraw (i.e., VPL influence limited to nearby fragments). Note that static images do not convey the vastly improved temporal coherence enabled by our approach.*

The big performance win comes from the stenciled gathering process. Multiresolution splatting typically generates around 50,000 multiresolution fragments for a 1024^2 image. For each VPL, all these fragments are processed by both a vertex and fragment shader and scattered incoherently into a multiresolution buffer. This is clearly the most expensive stage. Reformulating as a gather roughly halved this stage's cost, and leveraging the hierarchical z-buffer for early stencil culling avoided the incoherent scatter of fragments into the buffer. Combined, these techniques reduce the cost of the indirect illumination by over an order of magnitude.

As with multiresolution splatting, performance depends more on eye-space *visual complexity* than on scene complexity. For instance, the feline scene contains more geometry than the garden, but high frequencies introduced by the plants require finer eye-space patches, increasing costs.

Figure 9 illustrates the error introduced solely by multiresolution eye-space patches as a function of the depth discontinuity required to refine fragments. At a depth threshold of zero, indirect light is gathered at full resolution, yielding zero RMS error. Increasing this threshold increases error as fragments are clustered together. The same principle applies to the threshold used to detect normal discontinuities; these thresholds can be used to adjust performance or image quality at the expense of the other. In our tests, at thresholds that yield reasonable performance, error ranges from 2-3%.

4.2. Light Hierarchies

Just as Table 1 demonstrates a performance dependence on eye-space visual complexity, clustering VPLs adds a performance sensitivity to light-space visual complexity. Figure 10 shows framerate variations in the feline scene as the light moves. As visual complexity changes, so does the required number of VPLs, leading to fluctuations in performance.

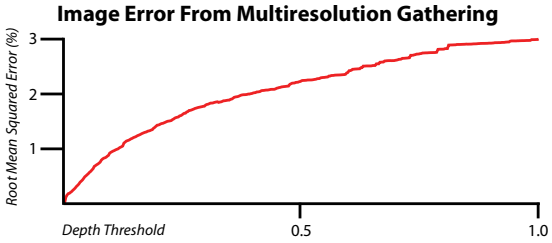


Figure 9: *Root mean squared error between multiresolution gathering and gathering illumination at every eye-space pixel. Both use one million VPLs (i.e., a 1024^2 RSM) to help differentiate between error caused by clustered eye-space patches and clustered light-space patches (light patch error is depicted in Figure 13).*

Performance characteristics for image space radiosity and our hierarchical variant are displayed in Figures 11 and 12. These figures start with 1024 coarse VPLs and subdivide in important areas. Performance tops out around 50 fps, when gathering into the illumination buffer is no longer the bottleneck; fixed cost passes including VPL tree creation or per-frame patch construction then limit performance.

Figure 13 explores the error introduced by hierarchical image space radiosity as a function of the user-specified error threshold. To distinguish error from our light- and eye-

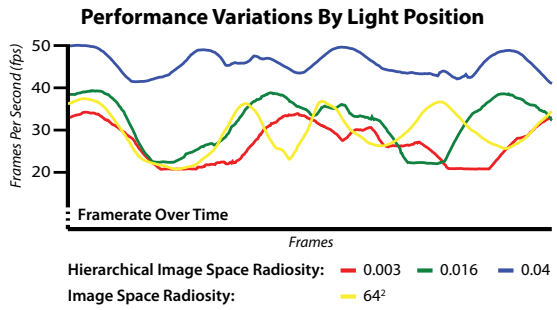


Figure 10: *Framerate varies as a light moves through the feline scene. Compared are image space radiosity with a maximum VPL subdivision to* 64^2 *and hierarchical image-space radiosity with three different error thresholds.*

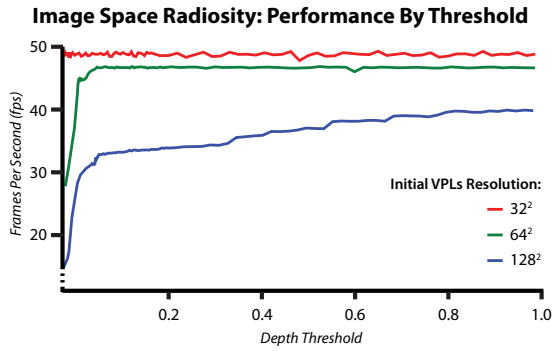


Figure 11: *Performance of image space radiosity in the feline scene, with different initial VPL resolutions. Lower depth thresholds introduce more light patches.*

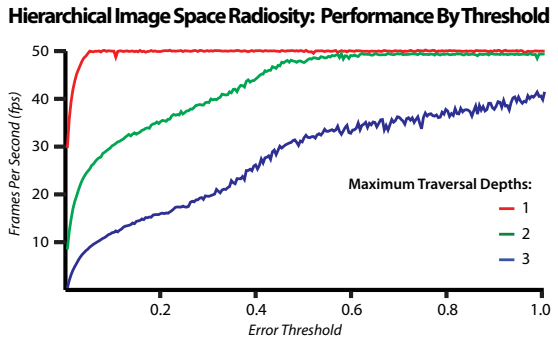


Figure 12: *Performance of hierarchical image space radiosity in the feline scene, using different maximal traversal depths. As we allow additional subdivisions, lower error thresholds introduce more light patches.*

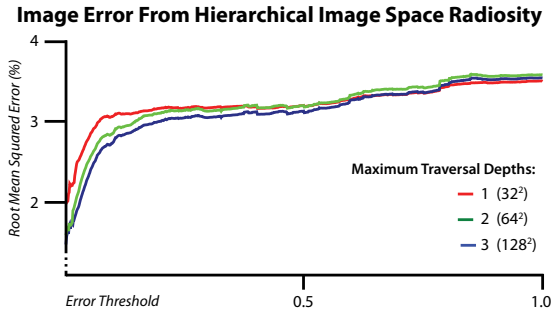


Figure 13: *Root mean squared error between hierarchical image space radiosity (with different maximal traversal levels) and our multiresolution gathering using one million VPLs (i.e., from a* 1024^2 *RSM).*

space hierarchies, the baseline image uses our multiresolution gathering with one million VPLs; even with multiresolution gathering, this requires more than 10 seconds. A threshold of zero forces a full traversal of the VPL tree, though due to reduced VPL sampling (32^2–128^2) error is not completely eliminated. At higher thresholds, fewer VPLs contribute illumination, and error increases to 3–4% for thresholds with acceptable performance.

Combined with the error demonstrated in Figure 9, we get interactive performance with a 5–6% RMS error compared to a baseline gather of a million VPLs at every pixel in eye-space. Considering the half hour computation for the baseline and the temporal coherence maintained by our method, we believe this is an acceptable trade-off.

5. Conclusions and Future Work

We presented image-space radiosity and a hierarchical variant, techniques that build upon our improved one-pass stenciled multiresolution gathering for indirect illumination. These techniques dramatically reduce the cost of approximating one-bounce indirect illumination using reflective-shadow map based instant radiosity. We achieve this by gathering indirect light at varying resolutions in image space, depending on proximity to discontinuities. When gathering illumination for each image space patch, we proposed two light-space clustering techniques to reduce the number of VPLs used per patch; one creates VPL clusters once per frame, the other selects VPL clusters based upon a per-fragment error metric and a user specified error threshold.

Interesting future directions include accounting for indirect light visibility, exploring other importance metrics (such as artist direction) for sampling, application to non-diffuse materials, and approximation of multi-bounce illumination.

Acknowledgments

This work was partially supported by a University of Iowa Graduate College Summer Fellowship and DARPA grant HR0011-09-1-0027. The authors would like to acknowledge generous hardware donations by both NVIDIA and AMD. 3D models were provided by Google's 3D Warehouse, the

Stanford repository, the AIM@SHAPE repository, and the 3D Meshes Research Database from the INRIA Gamma Group. Additional thanks for feedback on the text from Sabarish Babu and the anonymous reviewers.

References

[BS09] BAVOIL L., SAINZ M.: *ShaderX7*. Charles River Media, March 2009, ch. Image-Space Horizon-Based Ambient Occlusion, pp. 425–444. 2

[Bun05] BUNNELL M.: *GPU Gems 2*. Addison-Wesley, 2005, ch. Dynamic Ambient Occlusion and Indirect Lighting, pp. 223–233. 2

[DBB06] DUTRE P., BEKAERT P., BALA K.: *Advanced Global Illumination, 2nd Edition*. AK Peters, 2006. 2

[DS05] DACHSBACHER C., STAMMINGER M.: Reflective shadow maps. In *Proceedings of the Symposium on Interactive 3D Graphics and Games* (2005), pp. 203–208. 2

[DS06] DACHSBACHER C., STAMMINGER M.: Splatting indirect illumination. In *Proceedings of the Symposium on Interactive 3D Graphics and Games* (2006), pp. 93–100. 1, 2, 3, 4, 6

[GKBP05] GAUTRON P., KŘIVÁNEK J., BOUATOUCH K., PATTANAIK S.: Radiance cache splatting: a gpu-friendly global illumination algorithm. In *Proceedings of the Eurographics Symposium on Rendering* (2005), pp. 55–64. 1, 2

[GKM93] GREENE N., KASS M., MILLER G.: Hierarchical z-buffer visibility. In *Proceedings of SIGGRAPH* (1993), pp. 231–238. 3

[GSHG98] GREGER G., SHIRLEY P., HUBBARD P., GREENBERG D.: The irradiance volume. *IEEE Computer Graphics & Applications 18*, 2 (March-April 1998), 32–43. 1, 2

[GTGB84] GORAL C., TORRANCE K., GREENBERG D., BATTAILE B.: Modelling the interaction of light between diffuse surfaces. In *Proceedings of SIGGRAPH* (1984), pp. 213–222. 2, 5

[HSA91] HANRAHAN P., SALZMAN D., AUPPERLE L.: A rapid hierarchical radiosity algorithm. In *Proceedings of SIGGRAPH* (1991), pp. 197–206. 1

[IDYN07] IWASAKI K., DOBASHI Y., YOSHIMOTO F., NISHITA T.: Precomputed radiance transfer for dynamic scenes taking into account light interreflection. In *Proceedings of the Eurographics Symposium on Rendering* (2007), pp. 35–44. 2

[Kaj86] KAJIYA J.: The rendering equation. In *Proc. ACM SIGGRAPH* (1986), pp. 143–150. 1, 2

[KAMJ05] KRISTENSEN A. W., AKENINE-MÖLLER T., JENSEN H. W.: Precomputed local radiance transfer for real-time lighting design. *ACM Transactions on Graphics 24*, 3 (2005), 1208–1215. 2

[Kel97] KELLER A.: Instant radiosity. In *Proceedings of SIGGRAPH* (1997), pp. 49–54. 1, 2, 3

[KO08] KI H., OH K.: A gpu-based light hierarchy for real-time approximate illumination. *Visual Computer 24*, 7–9 (2008), 649–658. 1, 2, 3

[KTHS06] KONTKANEN J., TURQUIN E., HOLZSCHUCH N., SILLION F.: Wavelet radiance transport for interactive indirect lighting. In *Proceedings of the Eurographics Symposium on Rendering* (2006), pp. 161–171. 2

[LSK*07] LAINE S., SARANSAARI H., KONTKANEN J., LEHTINEN J., ALIA T.: Incremental instant radiosity for real-time indirect illumination. In *Proceedings of the Eurographics Symposium on Rendering* (2007), pp. 277–286. 2

[LZT*08] LEHTINEN J., ZWICKER M., TURQUIN E., KONTKANEN J., DURAND F., SILLION F., AILA T.: A meshless hierarchical representation for light transport. *ACM Transactions on Graphics 27*, 3 (2008), 1–9. 1, 2

[NRH03] NG R., RAMAMOORTHI R., HANRAHAN P.: All-frequency shadows using non-linear wavelet lighting approximation. *ACM Transactions on Graphics 22*, 3 (2003), 376–381. 2

[NW09] NICHOLS G., WYMAN C.: Multiresolution splatting for indirect illumination. In *Proceedings of the ACM Symposium on Interactive 3D Graphics and Games* (2009), pp. 83–90. 1, 2, 3, 4, 6

[PH04] PHARR M., HUMPHREYS G.: *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2004. 2

[RGK*08] RITSCHEL T., GROSCH T., KIM M., SEIDEL H.-P., DACHSBACHER C., KAUTZ J.: Imperfect shadow maps for efficient computation of indirect illumination. *ACM Transactions on Graphics 27*, 5 (2008), 1–8. 1, 2, 4

[SAG94] SMITS B., ARVO J., GREENBERG D.: A clustering algorithm for radiosity in complex environments. In *Proc. ACM SIGGRAPH* (1994), pp. 435–442. 1

[SGNS07] SLOAN P.-P., GOVINDARAJU N., NOWROUZEZAHRAI D., SNYDER J.: Image-based proxy accumulation for real-time soft global illumination. In *Proceedings of Pacific Graphics* (2007), pp. 97–105. 1, 2

[SIMP06] SEGOVIA B., IEHL J.-C., MITANCHEY R., PÉROCHE B.: Non-interleaved deferred shading of interleaved sample patterns. In *Proceedings of the Symposium on Graphics Hardware* (2006), pp. 53–60. 2

[SKS02] SLOAN P.-P., KAUTZ J., SNYDER J.: Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Transactions on Graphics 21*, 3 (2002), 527–536. 2

[STK08] SCHMITZ A., TAVENRATH M., KOBBELT L.: Interactive global illumination for deformable geometry in cuda. *Computer Graphics Forum 27*, 7 (2008), 1979–1986. 2

[WEH89] WALLACE J., ELMQUIST K., HAINES E.: A ray tracing algorithm for progressive radiosity. In *Proceedings of SIGGRAPH* (1989), pp. 335–344. 4

[WFA*05] WALTER B., FERNANDEZ S., ARBREE A., BALA K., DONIKIAN M., GREENBERG D. P.: Lightcuts: a scalable approach to illumination. *ACM Transactions on Graphics 24*, 3 (2005), 1098–1107. 1, 2

[WKB*02] WALD I., KOLLIG T., BENTHIN C., KELLER A., SLUSALLEK P.: Interactive global illumination using fast ray tracing. In *Proceedings of the Eurographics Rendering Workshop* (2002), pp. 15–24. 2

[Wym08] WYMAN C.: Hierarchical caustic maps. In *Proc. ACM Symp. Interactive 3D Graphics* (2008), pp. 163–171. 1, 3

[YSL08] YANG L., SANDER P. V., LAWRENCE J.: Geometry-aware framebuffer level of detail. *Computer Graphics Forum 27*, 4 (2008), 1183–1188. 3

[ZHWG08] ZHOU K., HOU Q., WANG R., GUO B.: Real-time kd-tree construction on graphics hardware. *ACM Transactions on Graphics (to appear)* (2008). 2

[ZIK98] ZHUKOV S., IONES A., KRONIN G.: An ambient light illumination model. In *Proceedings of the Eurographics Rendering Workshop* (June 1998), pp. 45–56. 2