Rendering Interior Cultural Heritage Scenes Using Image-based Shooting

Jassim Happa [†], Tom Bashford-Rogers, Kurt Debattista, and Alan Chalmers

International Digital Laboratory, University of Warwick, UK

Abstract

Rendering interior cultural heritage scenes using physically based rendering with outdoor environment maps is computationally expensive using ray tracing methods, and currently difficult for interactive applications without significant precomputation of lighting. In this paper, we present a novel approach to relight synthetic interior scenes by extending image-based lighting to generate fast high-quality interactive previews of these environments. Interior light probes are acquired from a real scene, then used to shoot light onto the virtual scene geometry to accelerate image synthesis by assuming the light sources shot act as the correct solution of light transport for that particular intersection point. We term this approach Image-Based Shooting. It is demonstrated in this paper with an approach inspired by Irradiance Cache Splatting. The methodology is well-suited for interior scenes in which light enters through narrow windows and doors, common at cultural heritage sites. Our implementation generates high-quality interactive preview renditions of these sites and can significantly aid documentation, 3D model validation and predictive rendering. The method can easily be integrated with existing cultural heritage reconstruction pipelines, especially ray tracing based renderers.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation -Display Algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism - Color, Shading, Shadowing, and Textures I.3.8 [Computer Graphics]: Applications

1. Introduction

The importance of correct use of lighting for Cultural Heritage (CH) reconstructions is a well-discussed topic in several CH-related research papers [HMD*10]. Global Illumination (GI) is essential to the high-fidelity rendering pipeline, allowing for accurate representation of these environments. Many methods to estimate and accelerate computation of sunlight are based on Image-based Lighting (IBL) [Deb98]; enabling fast image synthesis of objects lit by predominantly direct lighting. However, rendering interior synthetic environments using exterior IBL presents significant challenges due to the amount of indirect illumination often present in such scenes. The computational costs to correctly determine the contribution of outdoor illumination for interior scenes may be prohibitive due to the complexity of light transport simulation that GI methods traditionally require without tailoring the rendering method itself for the scene specifically. It is often not possible to acquire comprehensive information about light and material light reflectance within a sensitive CH site. Furthermore, adding arbitrary light sources may produce an incorrect visual appearance of the virtual scene. A fast way to reproduce location-faithful, natural exterior-to-interior lighting would be useful for CH researchers who wish to study and reconstruct interior real world locations.

In this paper, we propose a novel approach based on High Dynamic Range (HDR) environment-capture to relight interior virtual scenes and show its applications for CH. The correct lighting solution that already exists in environment maps is used to shoot information about light from the real scene onto the virtual scene to increase efficiency in the image synthesis pipeline. As irradiance information is shot onto the virtual scene using IBL light sources, we term this ap-

[†] e-mail: j.happa@warwick.ac.uk

proach Image-Based Shooting (IBS). The motivation behind this methodology is to provide a fast, high-quality, interactive preview of CH scenes before generating any offline rendering solutions (e.g. Path Tracing [Kaj86] or Photon Mapping [Jen96]). Multiple environment maps can be employed, and reconstruction (using interpolation) can approximate occluded areas. We show that this preview is similar to a photograph and Path Tracing. While IBS is not limited to a particular rendering method, we illustrate an adaptation of Irradiance Cache Splatting [GKBP05] in this paper.

A model of the Red Monastery; an Egyptian Coptic monastery [Bol06], has been selected to demonstrate IBS. Figure 1 shows an example of one of the environment maps used (in latitude/longitude format [BN76]).

Figure 1: *Example of one of the light probe used, in latitude/longitude format. Red Monastery 14:30, 21 Dec 2009.*

The remainder of the paper is organised as follows. Section 2 presents a summary of related work. In section 3 the core framework is described. A CPU implementation is presented in section 4, and a GPU version in section 5. Results are presented in section 6. A short discussion of the method is presented in section 7. Finally, we conclude the paper and describe possible future work in section 8.

2. Related Work

Several methods exist to generate fast renditions of real world objects, both with and without GI computation. These can include projective texturing [SKVW*92] or surface splatting [ZPVBG01]. Projective texturing is the inclusion of images that project colours onto surfaces, and is especially useful for shadow mapping techniques. Splatting can be implemented as an extension to projected texturing, (without any regard to illumination,) or to approximate GI. Radiance Cache Splatting [GKBP05] for instance is a technique inspired by irradiance caching [WRC88] that employs radiance and irradiance caching for splatting applications. Radiance cache splatting determines how records of illumination contribute to indirect lighting of visible points and splats data on the image plane.

The Irradiance Volume is a pre-processed volumetric representation of GI [GSHG98]. Irradiance is traditionally assumed to work for surfaces, however, the authors extend the definition for all points and directions in space represented in a grid structure. Complex GI effects can then be approximated using reconstruction from querying the Irradiance Volume. Several extensions to Irradiance Volumes exist. Kontkanen and Laine [KL06] introduced a sampling strategy for precomputing illumination in to a regular 3D grid. Aliasing is reduced for a small precomputation cost. Kaplanyan describes Light Propagation Volumes [Kap09] as an approach to GI inspired by several methods, including Irradiance Volumes and Reflective Shadow Maps [DS05].

Pacanowski et al. [PRL*08] present an approach to render detailed objects by using regular grids and a vector-based representation (3D and 2D textures) to capture low-frequency indirect light, and employ appearance-preserving simplification of geometry for interactive rendering.

A limitation of IBL is that the environment maps only acquire incident illumination at a particular point in space. This prevents accurate shadows as every object in the scene will be lit from the position of the light probe. Spatially varying IBL approaches such as Incident Light Fields [Ung09] and Stereo Light Probes [CCC08] provide more information about lighting in a scene by acquiring several light probes at various positions in the environment. A limitation here is the need for highly specialised equipment.

To our knowledge, only two pieces of work uses interior location-faithful illumination. The SIGGRAPH 1999 animation *Fiat Lux* by Debevec [Deb99] demonstrated the applications of identifying interior/exterior light areas to speed up efficiency of rendering interior scenes. By determining the entrance areas of exterior light and creating an average area light source from the HDR environment map photograph, a highly realistic virtual representation of St. Peter's Basilica in Rome was recreated using only a basic 3D model and a light probe from the real world location. While Debevec's approach to render Fiat Lux identifies exterior light sources and averages these as area light sources, our method uses as much information stored in the environment map as possible for relighting purposes.

Gibson et al. [GCHH04] presented an extension to IBL by combining augmented reality and IBL differential rendering. The approach creates a triangular illumination mesh used to shade virtual objects using Radiosity [GTGB84] to compute missing values of the mesh. An Irradiance Volume is used for computing real-to-virtual object shadows. The method is heavily reliant on a precomputation stage to build a hierarchy to encode light transport between illumination patches in the scene. Its primary application is generating appropriate shadows, giving the appearance that the local and distant scene are one and the same.

3. Image-Based Shooting - Framework

3.1. IBS Framework Overview

The general idea of IBS is to use a real world captured environment map in conjunction with laser scanned or 3D modelled interior geometry to render images with illumination that is both faithful to its real world counterpart and fast to compute. Traditional IBL uses environments maps that act as an infinite area light source to relight arbitrary objects. IBL is fast to compute for scenes with primarily direct illumination, however using this approach to relight interior scenes takes a long time to compute using physically based rendering methods as these environments will mostly contain indirect illumination. With IBS, we shoot as much information as possible contained within an indoor environment map by assuming the HDR photograph contains the correct solution of light for all visible surfaces from the position of the environment map, see Figure 2. This can speed up computation of GI for interior scenes significantly.

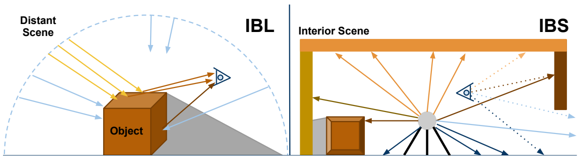


Figure 2: *Differences between IBL and IBS. Left: Traditional Image-based Lighting. Right: Image-based Shooting.*

3.2. Requirements

Three requirements are necessary for IBS. First, an interior environment map should be captured at a measured position in the real scene (we denote this position as p) and placed in the same position in the virtual scene. Second, a textured virtual 3D model that is similar to the real world scene is required (same proportional dimensions). This is already an essential component of most CH 3D reconstruction projects. Our method is robust enough to employ meshes that are modelled in 3D modelling software. Third, appropriate Bidirectional Reflectance Distribution Functions (BRDF) models should be captured or approximated. For our splatting-inspired implementation, a diffuse/lambertian BRDF approximation is used.

3.3. Splatting-IBS Overview

Splatting-IBS is comprised of three main components: *Shooting*, *Constructing an Irradiance Volume* and *Rendering*. The first step consists of shooting light from environment maps (multiple maps may be used to improve scene coverage) to form irradiance records. These records are used during pre-processing to build an Irradiance Volume for reconstruction of light in areas that are occluded in the light probe. During rendering, we use raycasting to obtain information about the irradiance records to generate each frame. Areas that are not covered by these records are relit using neighbouring/local records stored in the Irradiance Volume.

4. CPU Splatting-IBS Implementation

4.1. Shooting

At the start of the shooting stage, irradiance records are generated at surfaces visible from the environment map(s). As the environment map contains the solution to light transport in parts of the scene, this can be directly used to generate irradiance records, see Figure 3.

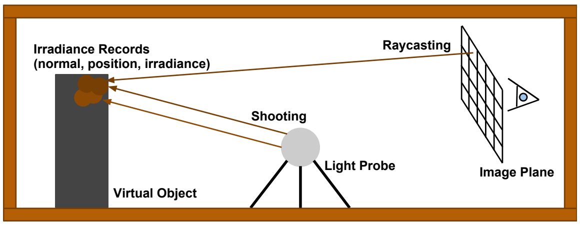


Figure 3: *IBS adaptation of Irradiance Cache Splatting. Each record contains a position, normal and irradiance which are generated during the shooting stage.*

Records are generated as a pre-process through a ray casting phase. For each pixel in the environment map, the nearest surface has to be determined in order to place a record on the geometry. The spherical coordinates of the pixel in the environment map are converted to Cartesian coordinates, and a ray starting at p is created in this direction. At the closest point of intersection of this ray, we place an irradiance record. However, as the radiance I_e is shot, but only the irradiance I_s is required, it is necessary to divide by the approximated diffuse surface reflectance ρ_a:

$$I_s = I_e / \rho_a$$

Once this process has been repeated for all the pixels in the light probe, the points are stored in a k-d tree to accelerate rendering. Each k-d tree node stores ray hit position (X), the normal at the intersection point (N_S), irradiance (I_s) and radius (R). This radius specifies the size of each splat. Currently a fixed, scene dependant size is used, however, this can be automated based on the distance from p to X. Increasing image resolution and shooting more lights (with a smaller radius) will increase accuracy of lighting for the scene at the expense of increasing memory usage. 1024×512 resolution HDR images were used in this paper, making a total of $524,288$ splats being shot per environment map per scene. Only shooting splats results in images with areas that are not covered by the values in the environment map. For these areas a default value, e.g. black can generate results similar to Figure 4.

Figure 4: *Visual appearance of the Red Monastery. Left: Albedo (no lighting). Right: Shooting with no reconstruction of light.*

4.1.1. Multiple Environment Maps

By including *m* number of environment maps in the nearby areas of the first light probe, it is possible to cover greater areas of an interior scene. Pillars, walls, large physical objects, and buildings with several rooms may limit the current viewpoint (of the light probe), see Figure 5. This can be referred to as viewpoint inaccessible occlusion. Viewpoint inaccessible occlusion can be combated with use of several environment maps. However, this means that overlapping data will occur. Occasionally, overlapping areas may not produce the same splat value (especially at times of the year when the sun moves fast as capturing light probe data may take several minutes). These splats can be averaged to form a better representation of the overlapping splats. This is assuming that lighting conditions between each HDR image have not drastically changed (e.g. the sun going behind a cloud).

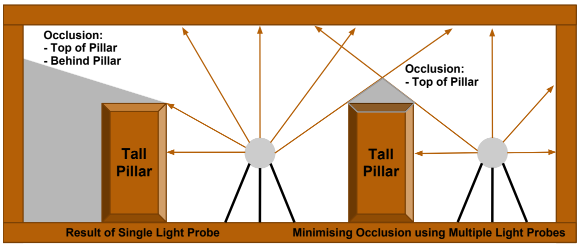


Figure 5: *Shadow areas show the two types of occlusion that occur with IBS. Physically inaccessible occlusion happens in areas that are unreachable for practical reasons (e.g. too tall, such as on top of a pillar). Viewpoint inaccessible occlusion (e.g. behind the pillar) occur in sections that can be seen using a different real world position.*

4.2. Constructing the Irradiance Volume

In IBS, the need for reconstruction of light arises in areas where light information from light probes does not reach. Physically inaccessible (or considerably difficult-to-access) sections may not warrant the additional environment map data capture, because these areas are not likely to ever be seen under normal viewing circumstances (e.g. very narrow cracks in walls or on top of tall pillars), see Figure 5. Furthermore, some sections may be physically impossible to access for light probe data capture. As these areas cannot be relit using only information from environment maps, light information for these parts needs to be reconstructed from existing data.

Applying an average value estimate for the entire scene may leave reconstructed areas too bright, especially if the environment map contains both exterior and interior lighting, such as Figure 1. A median value may end up appearing too dark for the brighter sections of the scene (e.g. areas actually hit by direct sunlight). It is therefore important to preserve local information of the scene as much as possible in order to reconstruct as accurate lighting for physically inaccessible occlusion.

During the pre-process, an Irradiance Volume [GSHG98] is constructed. A 3D grid is placed over the scene, with each cell representing the approximate average irradiance within the cell. This is constructed in a three step process. The first accumulates the irradiance from each hit from the shooting stage into the relevant grid cell. This is similar to the injection stage of Light Propagation Volumes [Kap09], however irradiance is injected instead of radiance. Values in each full cell are then averaged.

In the second step, we iteratively fill the empty cells in the Irradiance Volume via a nearest neighbour search. At each iteration, the 26 surrounding cells are inspected to see if they have an irradiance value assigned. If cells are found to contain irradiance, then the empty cell is assigned the average of the surrounding cells, else the cell is skipped this iteration.

Once the entire grid is full, the final step is carried out. As each cell is made up of an average of values, discontinuities may be caused by interruptions of light from one grid cell

Figure 6: *Steps of constructing and using the Irradiance Volume. Left to Right: Step 1) Accumulation of irradiance from the shooting stage and averaging cells. Step 2) Nearest Neighbour search filling. Step 3) Filtering and blurring. 4) Interpolation (during rendering) creates the final output images.*

to the next. This gives the impression of a grid-like structure present in the scene. It is therefore beneficial to filter and blur the values generated based on a distance from each cell's centre, see Figure 6. A higher subdivision of the scene leads to a better approximation, however, this can lead to a considerable growth of pre-processing necessary, assuming a uniform grid.

4.3. Rendering

During rendering, both the irradiance records and the Irradiance Volume are used to light the scene. In a ray tracing based renderer, this can be carried out with minimal ray casting. For each pixel, a ray is shot, and the nearest intersection P_S is located. When shading this point, the irradiance has to be computed. If possible, information should be used from the environment map, i.e. the irradiance records. These are calculated from a traversal of the irradiance record k-d tree. Irradiance records are considered suitable for shading provided two conditions:

1. $P_S - X < R$, where X is the irradiance record position and R is its radius.
2. $N_S \cdot N > 0$, where N_S is the normal at the shading point, and N is the normal assigned to the irradiance record.

These checks guarantee spatial and angular proximity of the irradiance record to the point being shaded. This value is then multiplied by the diffuse surface reflectance (ρ_a) at P_S in order to calculate outgoing radiance to the eye e:

$$L(P_S \rightarrow e) = \rho_a \, I_e$$

This enables the use of lighting from the environment map, while preserving acquired texture details. In areas which are not covered by irradiance records, the Irradiance Volume is used. This calculates the cell in the volume for which P_S is located, and interpolates the irradiance values from the cell corners, and again multiples by diffuse surface reflectance. Figure 6 illustrates the visual appearance after each stage of processing and the final render.

5. GPU Extension

The core concepts of the first two stages (shooting and constructing an Irradiance Volume) from the CPU implementation remains largely the same for our GPU version, however, during rendering, the abilities of the GPU are taken into account to decrease rendering times substantially.

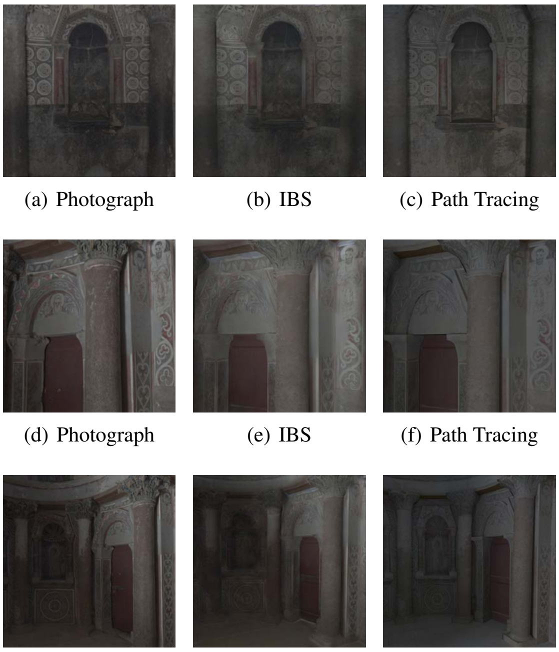
When using the GPU, the pre-processing stage is slightly different from the ray tracing based version. Instead of storing the irradiance records in a k-d tree, splats (quads) are generated in world space, and stored in a buffer. Each splat consists of the irradiance, positions of the quad vertices, and splat normal. The positions of the quad are generated so that they are aligned with the tangent plane at the surface intersection point during the shooting phase. These points are placed in a square a distance $\sqrt{2R^2}$ (R is the radius assigned to the irradiance record) from the intersection point. The Irradiance Volume is also stored as a 3D texture. The actual rendering consists of three steps:

1. Create a G-Buffer containing positions, normals and diffuse surface reflectance.
2. Splat irradiance records to the screen.
3. For any area which splats do not cover, interpolate irradiance from the 3D texture.

The first step creates a G-Buffer. World space positions, normals and diffuse surface reflectance are stored in the G-Buffer, all represented by 32 bits per colour channel floating point render targets. When splatting the irradiance records, the splats are rasterised to an intermediate render target. A texture look-up based on the screen coordinates of the fragment being shaded retrieves the world space position and normal of the fragment. These are used to determine whether the fragment should be shaded in the same way as the ray tracing method. Each fragment which passes this test is combined with the diffuse surface reflectance and rendered. In the final step, any fragments which have not been shaded are assigned a value from the Irradiance Volume via a 3D texture look-up and are combined with the diffuse surface reflectance to produce a final image.

6. Results

Figure 7 shows a comparison of: HDR photographs, IBS and Path Tracing. The path traced images were generated using exterior IBL environment maps captured outside Red Monastery, at the same time of day as the IBS interior light probes were captured. Lambertian BRDFs were employed as the light reflectance model for the entire environment. The path traced images were each rendered over 80,000 samples per pixel on a 24 quad-core machine cluster, each core running at 2.6GHz.



(a) Photograph (b) IBS (c) Path Tracing

(d) Photograph (e) IBS (f) Path Tracing

(g) Photograph (h) IBS (i) Path Tracing

Figure 7: *Comparing Photographs, IBS and Path Tracing*

The Frames-Per-Second (FPS) count for the ray tracing rendering ranges between 1.2 to 1.9 FPS on an Intel Core2 Extreme Q6850 (4 core @ 3GHz) processor at resolutions of 512×512 with an average of 1.6 FPS. Our renderer is multi-threaded to run on all 4 cores. On the GPU the framerate ranges between 40 to 60 FPS on an Nvidia 8800GTX at higher resolutions of 1024×768 with an average of 55 FPS. Modern GPUs are especially useful for computing first bounce GI [GKBP05], making our GPU implementation perform better than the CPU version. For the CPU implementation, an acceleration structure such as a k-d tree is essential to achieve interactive rates. Our motivation for a CPU implementation is for practical reasons: a non-computer graphics audience may use computers that hold integrated, or lower-end GPUs. As desktop parallelism quickly increases to more and more cores, the splatting-IBS method will scale accordingly. From a developer and usability standpoint, it may be simpler to relate to one piece of hardware, the CPU. Being able to determine whether IBS is possible at interactive rates on the CPU, is therefore useful to know in the context CH rendering.

It is important to note that this implementation of IBS is not aimed to compete with path tracing, photon mapping or any of its extensions. However, simply relighting a real world scene with (exterior) IBL acquired near the site and diffuse BRDFs may not be good enough to generate results similar to a photograph, even using offline, unbiased rendering solutions. IBL uses illumination at a particular point in space, in this case; outside the building. While the light transport computation is accurate, the goniometric representation of the environment may also not be accurate enough without physical measurements from the scene. Gonioreflectometers, especially portable ones, are not easily available today. As appropriate modelling materials can be a difficult task in CH today, CH material representations are often approximated to a lambertian BRDF or analytical models such as Ward, Lafortune, Blinn-Phong or Cook-Torrance. While a diffuse approximation in (for instance) Path Tracing may be sufficient for directly lit scenes, this is likely to not be the case for scenes predominantly lit by indirect illumination. We believe our method can therefore complement the predictive rendering pipeline [GTS*97] in the pursuit of approximating better BRDFs heuristically through output images that are based on lighting captured at CH sites.

7. Discussion

IBS delivers fast, perceptually good previews of real world scenes, and is useful for CH research, reconstruction, documentation and rendering. The approach is not strictly limited to a splatting-IBL extension, and can be used to give an idea about the distribution of light and visual appearance for a scene even by simply displaying irradiance records.

IBS keeps illumination information separate from geometry and material. The fast feedback delivered by IBS enables users to relight a site with illumination at different times of the day or year. IBS can thus allow for laser scanned models to be scanned once, then any light condition from the site can be applied by only capturing an HDR photograph. A standard D-SLR camera and a fish-eye lens [SJW*04] or a mirrored sphere [Deb98] are considerably cheaper in comparison, both in time and resources required, with less post-processing of acquired data. HDR panoramic camera equipment [Sph02] can also be employed for faster photography acquisition of such scenes. By only swapping the environment maps in the scene for different ones, it is possible to illustrate how changes in lighting drastically alters the visual perception of a scene. Figure 8 illustrates an example of this difference in the Red Monastery.

As a documentation approach, IBS can be employed to validate a 3D model created by a 3D modeller or determine inaccuracies in laser scanned or procedurally generated environments with a single (or very few) HDR photographs.

This is achieved by highlighting problematic areas in the existing model such as texture maps or geometry that need to be corrected through image editing or mesh processing as these will normally stand out as an anomalies in the IBS renders. This is especially useful in accurate documentation of CH environments, see Figure 9. The model in this paper for instance was created using the 3D modelling tool Maya.

(a) Morning, 10:00 (b) Afternoon, 14:30

Figure 8: *Red Monastery lit with different lighting conditions. Left: At 10:00 Right: Same day at 14:30.*

Figure 9: *Examples of where IBS is useful for 3D model accuracy assessment. Two problems became apparent when comparing the virtual model (bottom) to a photograph (top). Firstly, the floor texture coordinates are incorrect, leaving the floor bricks to face the wrong direction. Secondly, the brick tile on the left hand side of the photograph is not modelled, showing up as incorrect bright splats.*

Spatial IBL methods have shown improvements over traditional IBL by extrapolating spatial information about light sources. The Irradiance Volume takes advantage of the fact that for most interior scenes, indirect lighting will be similar to neighbouring illumination. Coarse shadow details remain in the scene from environment maps as we are able to obtain crude spatial information about light from a single light probe, see Figure 10.

7.1. Obtaining Optimal Positions of Environment Maps

The placement of a light probe in the real world, and subsequently, the virtual environment essentially becomes an

Figure 10: *Reconstruction of light from the grid only. Coarse shadows are preserved, but finer shadows are gone. The greater amount of grid cells allows for greater accuracy.*

important component of data capture to achieve best possible results. It can be regarded as an Art Gallery Theorem problem [O'r08]. The Art Gallery problem is a well-known visibility problem from computational geometry that is commonly solved for both visibility checks and mesh triangulation purposes.

In an IBS context, it is possible to employ the same theorem to determine the minimum number of environment maps necessary in order to optimise coverage of the scene. The guards (or point lights) from the theorem are replaced by environment maps (that have a visibility span of 360 degrees). In practical terms, the solving of the problem becomes somewhat cumbersome because of geometry complexity often found at CH sites. Essentially, m (number of environment maps) depends on time available and practicality of data capture while still obtaining the best possible coverage of the scene with these restrictions in mind. This can be simplified to a top-down 2D problem for each floor and room of a building. An example of efficient light probe positioning is shown with the Red Monastery in mind, see Figure 11.

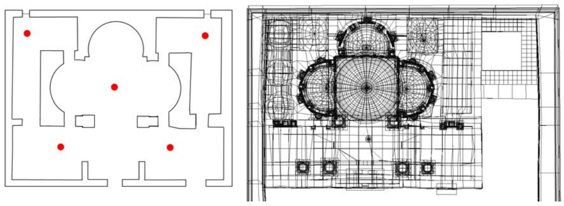


Figure 11: *Determining visibility of light probes, red dots indicate position of light probes. Left: Five environment maps cover the best range. Right: The scene in wireframe, showing the amount of simplification to the problem of determining best positions of light probes.*

7.2. Current limitations

Our method works very well for static scenes, but in dynamic scenes, GI computation will be incorrect; unless the illumination changes made from the dynamic object in the scene is calculated after splatting based on the Irradiance Volume. By identifying where the majority of the light is coming from (using multiple light probes), it should be possible to scale illumination of the scene if a dynamic object occludes these areas. While not physically based, the dynamic method could be sufficient to give a realistic impression of indirect occlusion. Video light probes may also solve this problem assuming the dynamic object is captured in the real scene and simulated accurately in its virtual counterpart. Another limitation is that our method currently only assumes a diffuse scene, leading to incorrect representation of specular and transparent objects in the environment.

8. Conclusion and Future Work

In this paper we have presented a novel approach to IBL that involves shooting light for synthetic interior scenes based on environment maps captured inside their real world counterpart. We demonstrated this with a splatting-inspired approach and extended it to deal with problematic cases such as occlusion, making this a good method to generate fast previews of CH virtual environments with appropriate, scene-captured lighting. The technique can be integrated into interactive ray tracers in a straightforward fashion. Its intent is not to replace high-fidelity rendering algorithms, but complement them allowing for fast image previews with GI. As the method is based on scene-specific, natural lighting, it can also serve to help documenting the visual appearance inside a CH building, and validate the accuracy geometric and colour attributes of 3D models in the scene.

Further work is needed to investigate how exterior changes in lighting (over the span of a day) can affect interior illumination. We believe this can be done by interpolating light data between various light probes captured at several intervals, or by using a real-time light probe for example from an HDR video camera [UGOJ04], using recently presented hardware [HACC10]. The current implementation employs manual positioning of the environment maps. An automated approach of determining optimal positions of light probes in the virtual scene, based on comparing albedo values to those in the environment map would reduce time spent fine tuning its positioning in virtual space. Combining Splatting-IBS and the use of Virtual Point Lights (VPLs) (instead of grid reconstruction) may be a potential direction for the future allowing for additional objects to be rendered in the scene with accurate shadows.

Acknowledgements

Our thanks go to the American Research Center in Egypt (ARCE), Elizabeth Bolman for permission to use the Red Monastery model, to Agnieszka Szymanska and Laurel D. Hackley for help on site, and Elmedin Selmanovic and Carlin Yuen for the Monastery model. All renders and photographs have been brightened for viewing purposes.

References

[BN76] BLINN J. F., NEWELL M. E.: Texture and reflection in computer generated images. *Communications of the ACM* (1976).

[Bol06] BOLMAN E.: Late antique aesthetics, chromophobia and the red monastery, sohag, egypt. *Eastern Christian Art* (2006).

[CCC08] CORSINI M., CALLIERI M., CIGNONI P.: Stereo light probe. *Computer Graphics Forum* (2008).

[Deb98] DEBEVEC P.: Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *SIGGRAPH* (1998).

[Deb99] DEBEVEC P.: Image-based modeling, rendering, and lighting in Fiat Lux. In *ACM SIGGRAPH Sketches* (1999).

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

[GCHH04] GIBSON S., COOK J., HOWARD T., HUBBOLD R.: Illumination Capture and Rendering for Augmented Reality. In *Eurographics Symposium on Rendering* (2004).

[GKBP05] GAUTRON P., KŘIVÁNEK J., BOUATOUCH K., PATTANAIK S.: Radiance cache splatting: A GPU-friendly global illumination algorithm. In *Eurographics* (2005).

[GSHG98] GREGER G., SHIRLEY P., HUBBARD P., GREENBERG D.: The irradiance volume. *IEEE Computer Graphics and Applications* (1998).

[GTGB84] GORAL C. M., TORRANCE K. E., GREENBERG D. P., BATTAILE B.: Modeling the interaction of light between diffuse surfaces. *SIGGRAPH* (1984).

[GTS*97] GREENBERG D. P., TORRANCE K., SHIRLEY P., ARVO J., FERWERDA J. A., PATTANAIK S., LAFORTUNE E., WALTER B., FOO S.-C., TRUMBORE B.: A framework for realistic image synthesis. In *SIGGRAPH* (1997).

[HACC10] HAPPA J., ARTUSI A., CZANNER S., CHALMERS A.: High Dynamic Range Video for Cultural Heritage Documentation and Experimental Archaeology. In *VAST* (2010).

[HMD*10] HAPPA J., MUDGE M., DEBATTISTA K., ARTUSI A., GONÇALVES A., CHALMERS A.: Illuminating the Past: State of the Art. *Virtual Reality* (2010).

[Jen96] JENSEN H. W.: Global illumination using photon maps. In *Eurographics* (1996).

[Kaj86] KAJIYA J. T.: The rendering equation. In *SIGGRAPH* (1986).

[Kap09] KAPLANYAN A.: Light propagation volumes in cryengine 3. *SIGGRAPH Course Notes-Advances in Real-Time Rendering in 3D Graphics and Games* (2009).

[KL06] KONTKANEN J., LAINE S.: Sampling precomputed volumetric lighting. *Journal of Graphics Tools* (2006).

[O'r08] O'ROURKE J.: *Computational geometry in C*, 2 ed. Cambridge University Press, (2008).

[PRL*08] PACANOWSKI R., RAYNAUD M., LACOSTE J., GRANIER X., REUTER P., SCHLICK C., POULIN P.: Compact Structures for Interactive Global Illumination on Large Cultural Objects. In *VAST: Short and Project Paper* (2008).

[SJW*04] STUMPFEL J., JONES A., WENGER A., TCHOU C., HAWKINS T., DEBEVEC P.: Direct HDR capture of the sun and sky. In *AFRIGRAPH* (2004).

[SKVW*92] SEGAL M., KOROBKIN C., VAN WIDENFELT R., FORAN J., HAEBERLI P.: Fast shadows and lighting effects using texture mapping. In *SIGGRAPH* (1992).

[Sph02] SPHERON: SpheroCam HDR, website. http://www.spheron.com/, (2002).

[UGOJ04] UNGER J., GUSTAVSON S., OLLILA M., JOHANNESSON M.: A real time light probe. In *Eurographics, Short Papers and Interactive Demos* (2004).

[Ung09] UNGER J.: *Incident Light Fields*. PhD thesis, Linköping University, (2009).

[WRC88] WARD G. J., RUBINSTEIN F. M., CLEAR R. D.: A ray tracing solution for diffuse interreflection. In *SIGGRAPH* (1988).

[ZPVBG01] ZWICKER M., PFISTER H., VAN BAAR J., GROSS M.: Surface splatting. In *SIGGRAPH* (2001).