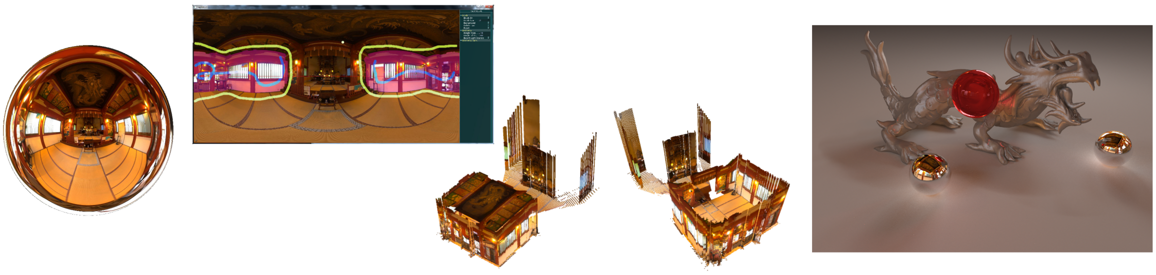
Pacific Graphics 2013
B. Levy, X. Tong, and KK. Yin
(Guest Editors)

Volume 32 (2013), Number 7

EnvyDepth: An Interface for Recovering Local Natural Illumination from Environment Maps

Francesco Banterle[1], Marco Callieri[1], Matteo Dellepiane[1], Massimiliano Corsini[1], Fabio Pellacini[2], and Roberto Scopigno[1]

[1]Visual Computing Laboratory, ISTI-CNR, Pisa, Italy, [2]Dipartimento di Informatica, Università La Sapienza, Roma, Italy



Abstract

In this paper, we present EnvyDepth*, an interface for recovering local illumination from a single HDR environment map. In* EnvyDepth*, the user quickly indicates strokes to mark regions of the environment map that should be grouped together in a single geometric primitive. From these annotated strokes,* EnvyDepth *uses edit propagation to create a detailed collection of virtual point lights that reproduce both the local and the distant lighting effects in the original scene. When compared to the sole use of the distant illumination, the added spatial information better reproduces a variety of local effects such as shadows, highlights and caustics. Without the effort needed to create precise scene reconstructions,* EnvyDepth *annotations take only tens of seconds to produce a plausible lighting without visible artifacts. This is easy to obtain even in the case of complex scenes, both indoors and outdoors. The generated lighting environments work well in a production pipeline since they are efficient to use and able to produce accurate renderings.*

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.3]: Illumination Estimation, Image-Based Lighting, Reflectance and Shading—Computer Graphics [I.3.7]: Three-Dimensional Graphics and Realism—

1. Introduction

Recent years have seen an increasing use of captured natural illumination in lighting synthetic scenes. In most cases, natural illumination is captured by HDR imaging of light probes. When lighting synthetic scenes, light probe data is used as distant illumination encoded as environment maps. The distant illumination approximation holds well when the captured scene is significantly larger than the re-lighted synthetic objects.

The distant illumination approximation is insufficient for smaller, common environments such as rooms or streets. In this case, one would like to have a reconstruction of the natural environment and its emitted radiance, in order to to capture local lighting properties. Unfortunately, such type of reconstruction is often impractical except for the simplest cases. This goes against the very reason for the use of natural illumination acquisition. Moreover, for most applications, such precise reconstruction is not necessary.

One way to mitigate this problem is to spatially sample illumination information by capturing light probes at different positions in the scene, and interpolating the resulting illumination [Tat05, UGLY08]. This process though is quite time consuming and still problematic since it remains unclear how to properly interpolate local lighting.

In this paper, we investigate a different approach, where the user performs a rough reconstruction of the environment. While this could also be done using imaging tools, such as Photoshop, the required workflow would be too cumbersome for a wide use. For this purpose, we propose *EnvyDepth*, an interface that supports the creation of the local natural illumination by manipulating an environment map. With *EnvyDepth*, the user creates rough strokes to mark regions of the environment map, and uses the generated selections to create geometric primitives. From these annotated strokes, *EnvyDepth* computes a *plausible* local reconstruction of the illumination environment, represented as a set of virtual point lights (VPLs) for the local component and as directional lights for the distant illumination. This reconstruction can be directly used to produce realistic renderings of the scene. With *EnvyDepth*, an expert user can perform a quick reconstruction in tens of seconds for a complex environment. We ran a user study to validate the usability of the system. This study showed untrained users feel comfortable with the system's results and they can perform a reconstruction in a few minutes.

2. Related Work

In this section, we firstly provide a detailed overview on the methods for acquiring real-world lighting. Secondly, we present methods and User Interfaces (UI) for generating depth maps from a single 2D image, because our system takes inspiration from those approaches. Finally, we present tools for editing real-world lighting.

2.1. Real-World Lighting Acquisition

The literature of image based lighting (IBL) is extensive, and it goes back to the the seminal works on environment mapping [BN76,MH84]. Debevec [Deb98] proposed a novel framework for acquiring real-world lighting, using environment maps assumed as distant lighting, and relighting virtual objects into photographs. This work was extended by Sato et al. [SSI99] introducing spatially-varying illumination by capturing the real-world illumination using two omni-directional cameras. Following this work, Corsini et al. [CCC08], contemporary with Korn et al. [KSvA*08], proposed a solution which extracts light sources from two mirrored spheres on a stereo rig by using spherical epipolar geometry. Unger et al. [UWH*03] proposed to capture spatially-varying illumination, *incident light fields* (ILF), by sampling the scene on a plane. This work was further improved by Unger et al. [UGLY08, UKL*13] by sampling every possible path in a 3D environment. Moreover, Unger et al. introduced novel methods for sampling, compressing, and editing ILFs.

2.2. Depth Map Generation from 2D Content

The literature for depth map generation from 2D content includes many different approaches. In this section, we are focusing just on tools based on user inputs and thus we refer to Smolic et al. work [SKK*11] for a more in-depth survey. Recently, researchers have proposed several tools based on sketching interfaces, depth painting, and user inputs. These tools are typically designed for 2D-3D conversion for cinematic workflows. For example, Guttmann et al. [GWCO09] proposed a system for propagating depth annotation through videos. Similarly, Gerrits et al. [GDDA*11] introduced the propagation of painted depth values using an edge-aware interpolation and normals painting for taking care of slanted surfaces. Wang et al. [WLF*11] introduced *StereoBrush* a tool which is based on image warping and instant feedback. Finally, Liao et al. [LGYG12] proposed a semi-automatic tool for videos, which allows users to label depth relationship between objects/regions using strokes.

2.3. Editing Real-World Lighting

One of the first tools for creating environment maps is *Illumination Brush* by Okabe et al. [OMSI07]. In this tool, the user can specify the desired appearance of the model's surface using strokes with different brushes: one for diffuse lighting, and another one for specular effects. The environment lighting is estimated by solving a least square problem where strokes, projected onto the object surfaces, are used as constraints. Similarly to *Illumination Brush*, Kim et al. [KN09] proposed *LightShop* which includes light sources clustering.

A different tool, *EnvyLight*, for editing existing environment maps was proposed by Pellacini [Pel10]. The tool has an interactive interface in which users select lighting features on a rendering (such as diffuse gradients, highlights, and shadows) by sketching strokes. The user has two brushes: the first one for indicating if an area belongs to the feature, the second one if not. A selected feature can be edited with different operations such as: moving an highlight, increasing/decreasing the contrast of a diffuse gradient, increasing the contrast, or blurring shadows.

Recently, Karsch et al. [KHFH11] proposed a novel user-based tool for integrating synthetic objects into photographs in a few steps. Firstly, the user models the bounding geometry as a 3D cuboid and can add geometry by extruding 2D closed curves, including occluded surfaces. Secondly, the user defines the lighting by drawing polygons in the image; lighting parameters are computed by minimizing the SSD between the rendered image and the original one. The system estimates the albedo of scene's surfaces using a novel image decomposition. Finally, the user inserts synthetic objects in photographs in just a few minutes. This is effective and robust to challenging scenes.

Our proposed tool, *EnvyDepth*, as many tools mentioned

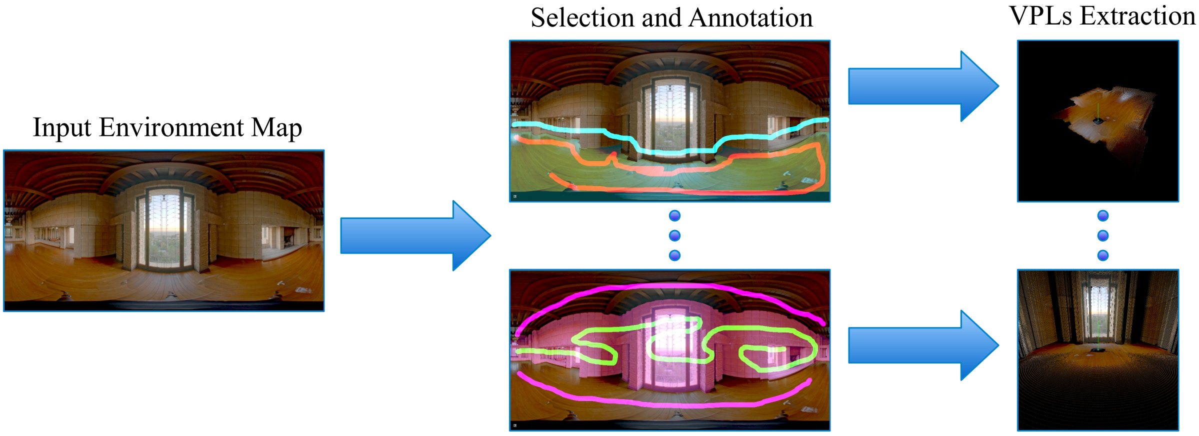


Figure 1: *The workflow of* EnvyDepth*: an input environment map is loaded. Then, the user selects different regions of it, and assigns a different geometric primitive. Geometry and associated lights sources are interactively extracted once parameters or new layers are created.* ***Top****: generation of a horizontal plane; main selection in red, and background selection in cyan.* ***Bottom****: generation of an extruded surface from a plane, in this case the previously generated plane; main selection in green, and background selection in light purple.*

before, is not meant for creating precise depth information for the lighting environment, but *plausible* depth information which is suitable for relighting purposes. Our system shares some similarities with the one of Horry et al. [HAA97] where some geometric constraint is provided by the user, e.g. the floor, to model the scene in an efficient and straightforward way. However, it can automatically infer the object boundaries as other approaches.

3. Rendering with Local Real-World Illumination

Illumination Decomposition. When real-world illumination is represented as distant lighting $L(\omega)$ [Deb98], the direct illumination at a point x can be computed as

$$L_o(x, n_x, \omega_o) = \int_\Omega L(\omega) V(x, \omega) f_r(\omega, \omega_o)(\omega \cdot n_x) d\omega \quad (1)$$

where n_x is the normal at x, Ω is the hemisphere around n_x, ω_o is the viewing direction, ω is the incoming lighting direction, L is the environment map radiance, V is the visibility term, and f_r is the BRDF. To capture local lighting effects, we propose to decompose the incoming lighting in two components: a local component $L^l(y)$ where the light has a spatial component y, and a distant illumination $L^d(\omega)$ for far-away illumination (e.g. the sky). The resulting direct illumination computation can be written as

$$L_o(x, n_x, \omega_o) = L_o^l(x, n_x, \omega_o) + L_o^d(x, n_x, \omega_o) = \quad (2)$$

$$\int_A L^l(y) V(x, \omega_{y,x}) f_r(\omega_{y,x}, \omega_o) G(x, y) dA(y) + \quad (3)$$

$$\int_\Omega L^d(\omega) V(x, \omega) f_r(\omega, \omega_o)(\omega \cdot n_x) d\omega \quad (4)$$

where L^l is the local component of real-world illumination, L^d the distant one, $G(x,y) = (\omega_{x,y} \cdot n_y)(\omega_{y,x} \cdot n_x)/||x-y||^2$ is the geometry term with $\omega_{x,y} = (x-y)/||x-y||^2$, and A is the surface of the scene. In this formulation, the contribution of L^d is computed just like the classical distant lighting approximation, but only for the parts of the real-world scene that are, in fact, distant.

Illumination Representation. In *EnvyDepth*, we represent the distant component L^d as an environment map, just as in standard IBL. For the local component L^l, a variety of representations might be possible, ranging from impostors to light fields. The chosen representation has to be flexible enough to support a variety of indoor and outdoor scenes as well as efficient to render.

We adopt a point-based representation for local environment lighting. We store the local illumination as a large set of Virtual Points Lights (VPLs). Each VPL i is defined by a light intensity, I_i; a position, y_i; and a normal, n_i. The local component of illumination can then be written as

$$L_o^l(x, n_x, \omega_o) = \sum_i I_i V(x, \omega_{y_i,x}) G(x, y_i) f_r(\omega_{y_i,x}, \omega_o) \quad (5)$$

We choose to use VPLs since they are well suited to capture intricate geometry and are supported by scalable offline rendering algorithms [WFA*05, HPB07] as well as by their interactive variants [RDGK12]. Their main drawback is that a big number of points is needed in the presence of highly glossy surfaces. This drawback is mitigated in our case since we do generate a large number of VPLs to capture geometric complexity. Furthermore, since any local reconstruction from a single environment map will have holes due to occlusion, any other representation will still be problematic for high gloss.

4. *EnvyDepth*: Recovering Local Real-World Illumination

In order to recover local illumination from an environment map, the user needs to decompose the original map into local and distant parts, and assign position and orientation to all VPLs. While this authoring could be performed manually using image-based tools (e.g. Photoshop), or by assisted image-based modeling inside 3D editing tools, this would be too cumbersome to be of practical use.

Workflow. In *EnvyDepth*, we propose to recover local illumination with an iterative select-and-annotate workflow supported by a sketch-based interface, illustrated in Figure 1. For each edit, the user selects regions of the environment map that should be reconstructed using a sketch-based metaphor [Pel10,WLF*11,GDDA*11]. Two types of strokes are placed directly on the environment map, to indicate regions that should be included or excluded from the selection. Edit propagation [AP08, XLJ*09] is then used to compute the selection from these imprecise strokes. Then, the user assigns a geometric primitive for each selection. Note that the system generates a partition of the original image.

Environment Map Visualization. In our interface, the user works directly on the environment map since this is familiar to users and commonly used in workflows that include image editing tools. We choose the latitude-longitude mapping for environment maps since it flattens the environment map in a rectangular area that is straightforward to visualize and simple to compute [RWP*10]. We take into account the fact that this mapping does not have equal area by pre-multiplying the image by the vertical cosine fall-off on loading. During the visualization, the environment map is tone mapped using the photographic tone operator [RSSF02] in order to display the entire dynamic range of the environment map.

4.1. Selection Algorithm

The selection algorithm takes as input the set of points I that should be included in the selection, and the set of points O that should be excluded from it. Then, it computes a soft selection mask S by edit propagation [AP08]. Figure 2 shows an example of selection and extraction.

Out of the many edit propagation methods presented in the literature, we choose the iterative joint bilateral filter, as formalized in [BCCS12], since for our cases of interest it is a fast approximation of [AP08] that is straightforward to compute interactively and provides a good edge-stopping behavior. We begin by setting the selection mask at each pixel i as

$$S_i^0 = \begin{cases} 1 & \text{for } i \in I \\ -1 & \text{for } i \in O \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

We then iteratively filter the selection S by applying the joint bilateral filter as

$$S_{\omega_i}^k = \frac{\sum_j S_{\omega_j}^{k-1} z_{ij}}{\sum_j z_{ij}} \tag{7}$$

$$z_{ij} = \exp(-\|\mathbf{L}_{\omega_i} - \mathbf{L}_{\omega_j}\|^2/\sigma_r)\exp(-\Delta(\omega_i, \omega_j)^2/\sigma_s) \tag{8}$$

where k is the current iteration, ω_i and ω_j are sampling direction of the environment map $\mathbf{L}$, Δ is the great-circle distance between two sampling points in the environment map. We apply this filter repeatedly for 200 iterations to provide a smooth transition [AP08]. We generate the final binary selection S by thresholding the soft selection with a small value $\varepsilon = 0.0001$, which we found to be numerically stable in our experiments.

4.2. Depth Values Generation

Fitting Geometric Primitives. Once a selection is determined, the artist indicates how to convert it to a collection of points and directional lights by assigning simple geometric primitives to the selection. We refer the reader to the video in the **additiona material** for an interactive editing session. *EnvyDepth* supports three basic primitives.

- *Planes*. In many cases, for the purpose of computing illumination, vertical or horizontal planes are a sufficiently accurate approximation of the selection. The most prominent cases are the ground plane in outdoor scenes, or floor and ceiling in indoor scenes (see Figure 1, top row, for an example of floor extracted as a horizontal plane). Planes can also be used to create simple impostors for horizontal or vertical elements that are not in the close proximity of the probe position (e.g. the tall buildings in Figure 3). More generally, a plane may be used to represent an entity that has no visible contact with an already extracted plane (see the next point). For this kind of elements, the user needs only to choose whether the plane is horizontal or vertical, and its distance from the probe center. When generating horizontal/vertical planes, the normal of the generated plane is computed as the mean direction in the region.
- *Extruded Geometry*. More complex entities may be created when an area of contact between an object and an area already assigned to a plane is visible in the input probe. In these cases, we can generate an extrusion from that plane. For example, the walls in Figure 1 (they have niches and protrusions) are correctly extruded from their outline on the plane extracted for the floor. In this case, no additional input to the selection is needed: using the adjacency of the selection with the reference plane, the position in space of the object's base can be derived from the contact points.
- *Distant Illumination*. Finally, a selection can be assigned to the distant illumination component. This includes cases such as the sky-dome, but also the objects that the user judges far away enough from the rendered object, such as

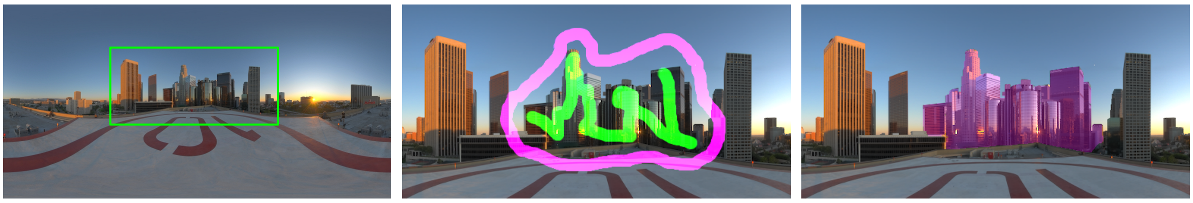


Figure 2: *An example of selection. On left: the* LA Helipad *environment map; where the green box is the zoomed area. In the middle: a zoom of the enviroment map the input strokes are visualized; the light green stroke represents I while the light purple stroke represents O. On right: the output mask S^M.*

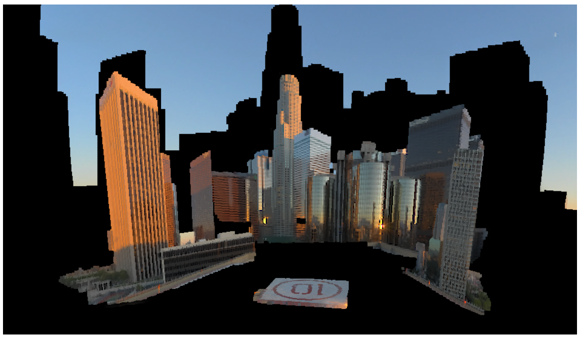


Figure 3: *A view of the extracted geometry from the* LA Helipad *environment map (see Figure 2(left)), where black is background. Skyscrapers are modeled using vertical planes.*

the distant small buildings in the city environment shown in Figure 2(left).

Once primitives are determined, they are converted into VPLs. For each pixel of the selected region in the area-equalized environment map, a VPL is created. We assign a position and a normal to each VPL via ray primitive intersections [PH10], and an intensity by fetching the environment map. We obtain the ray direction for a pixel by

$$\begin{aligned} (\phi,\theta) &= \left(2\pi\frac{x}{w},\pi\frac{y}{h}\right) \\ \omega(\phi,\theta) &= [\cos\phi\sin\theta,\sin\phi\sin\theta,\cos\theta]^T \end{aligned} \tag{9}$$

where (x,y) are the pixel coordinates, w and h are respectively the width and height of the environment map.

Discussion. While these primitives are quite simple, they allow to cope with the majority of situations found in environment maps; at the same time, they are very simple to handle for users. Instead of providing more complex primitive, in *EnvyDepth* we prefer to ask the user to reduce complex situations into smaller elements. For example, a non-flat ground can be generated using multiple horizontal plane elements at different heights.

Energy Conservation. Our system creates a partition of the environment map, i.e. a pixel is assigned to only a single primitive. This means that energy is not added during the editing process. However, our system converts an area light at infinite distance, or a collection of directional light sources, into VPLs. This leads to a different final energy value, and this needs to be taken into account. The energy values of the VPLs have to be scaled to preserve the total energy. The first attempt was to find a global solution by minimizing the energy mismatch between these two representations. However, the amount of samples needed to obtain a coherent solution was too large and made this approach impractical. Therefore, we opted for the following approximation; we computed the exact scaling factors, ϕ_i (where i is the i-th VPL), at $x_o = [0,0,0]^T$ with normal $n_o = [0,1,0]^T$ for a diffuse BRDF without occlusions. In this case, the scaling factors can be easily computed. If we define the total energy at (x_o,n_o), E_c, computed by directional light sources as

$$E_c(x_o,n_o) = \frac{1}{\pi}\sum_{i=1}^{M}(\omega_i\cdot n_o)L(\omega_i) \tag{10}$$

where M is the number of samples in the environment map; and we define the total energy, E_v, for VPLs (with unknown scaling factors) as

$$E_v(x_o,n_o) = \frac{1}{\pi}\sum_{i=1}^{M}G(x_o,y_i)I_i\phi_i \tag{11}$$

we can now independently compensate each i-th VPL for the corresponding i-th directional light. We obtain that

$$\phi_i = \|x_o - y_i\|^2 \tag{12}$$

and this leads to $E_c = E_v$ at x_o with normal n_o.

5. Results

Testing Environment. We tested *EnvyDepth* on an Intel iCore7 2.8 Ghz with 4 GB of RAM with an NVIDIA 480GTX. We rendered our scenes at the resolution of 910 × 512 using a multi-threaded Montecarlo pathtracer using around 5000 paths per pixel. We chose this renderer since it produces reference solutions without further approximations. Rendering times can be greatly speed up by using a dedicated many-lights algorithm [OP11].

Timings. *EnvyDepth* is an interactive interface, and the image processing operations are accelerated using OpenGL. The computation of the mask from a selection takes less than 0.4 seconds using environment maps with a 1600 × 800 resolution; this resolution was used in all our examples.

Our renderings took between 0.2s to 0.9s (Figure 4) for each sampling pass (in which a path per pixel is evaluated), note that the renderer has no modern optimizations for ray-primitive intersections [WMG*09], but it is only parallelized via multi-threading.

Comparisons with distant-light illumination. Comparing our local representation of the lighting with the standard distant-light approximation used in image-based lighting (IBL) [Deb98], it is possible to observe how lighting effects like shadows, reflections and caustics do change significantly across the scene, better conforming to the actual position of illuminants.

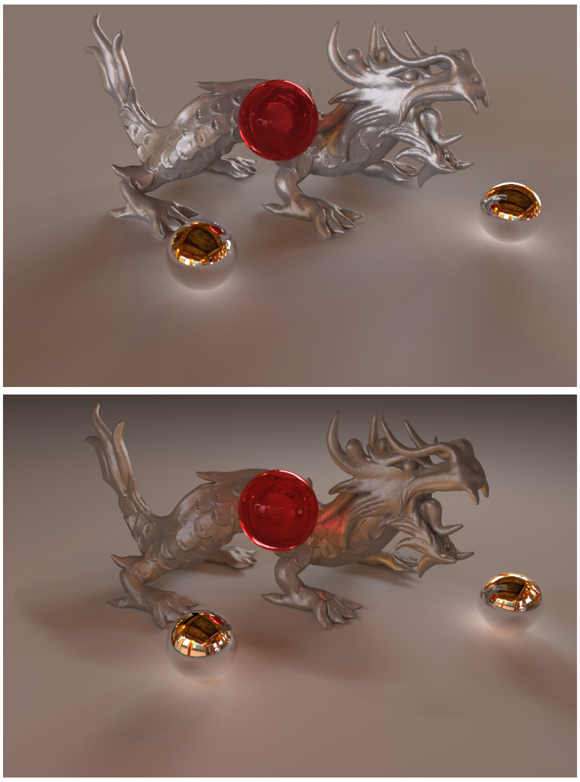


Figure 4: *Two renderings of a scene rendered using the* Ueno *environment map (see Teaser). On top, the standard IBL rendering; on bottom, our method. The position, shape and details of shadows, reflections and caustics are, by using our method, more correlated to the actual spatial position of the radiant elements, with respect to the distant-light approximation of standard IBL.*

In Figure 5, we show three renderings of objects in the same scene (the Ennis environment map) in order to focus on the improvements of each individual lighting effect. The Gargoyle models with a diffuse material in front of the main window of the environment (first row) better account for the spatial variation in the shadows, which are not reproducible with IBL.
In the second row, three Armadillos with a glossy BRDF are shown. In this case, the glossy reflections rendered using our method are smaller. This is because the subtended specular angle is narrower due to the fact that the main light (the window) is nearer to the objects. Moreover, they are correctly oriented toward the position of the window, while in the IBL rendering they all come from a single "frontal" direction.
In the third row, four mirrored spheres and a refractive one in front of the main window account for full specular effects. In this case, reflections elicit the spatial nature of the reconstructed environment, allowing the user to perceive the position of the spheres in the environment just looking at the reflections. The generated caustics show the strong contribution of the main window which is close to the refractive sphere, and its shape is clearly visible in the caustics. This is not possible using IBL, and this greatly affects the realism of image. In this case, images were tone mapped with Reinhard et al.'s operator [RSSF02] (using the same parameters for all images) in order to properly display caustics.

In Figure 4, we show a more comprehensive scene setup, rendered inside the Ueno environment map: on top, the rendering using IBL, on bottom, our representation. Again, the shadows are more spatially coherent using our representation, respect to IBL. The glossy surface of the dragon appears more uniformly lit in IBL, and while it shows the differently illuminated and colored areas inside the scene in our representation. Finally, in the caustics rendered using our method appear extremely detailed, to the point of correctly showing the bars of the windows generating them.

Comparisons with ground truth. In order to assess the correctness of the proposed method, we compared the results of the proposed lighting representation to a synthetic ground truth.
We started from two 3D scenes (a simple room and Sponza), and used them to generate HDR environment maps. We then proceeded in creating our lighting representation starting from these environment maps. Finally, we rendered some objects inside the two scenes in three different ways: 1) as a ground truth, producing a rendering of the 3D scene plus the added objects, 2) rendering the added objects using our method, and then adding them to a rendered image of the background, and 3) rendering the added objects using the basic IBL method and then adding them to a rendered image of the background. These comparisons are presented in Figure 6. It can be elicited how both the diffuse and specular behavior of the objects, when lighted using our method, are closer to the ground truth in terms of lighting intensity, dominant illumination color and geometrical correctness.

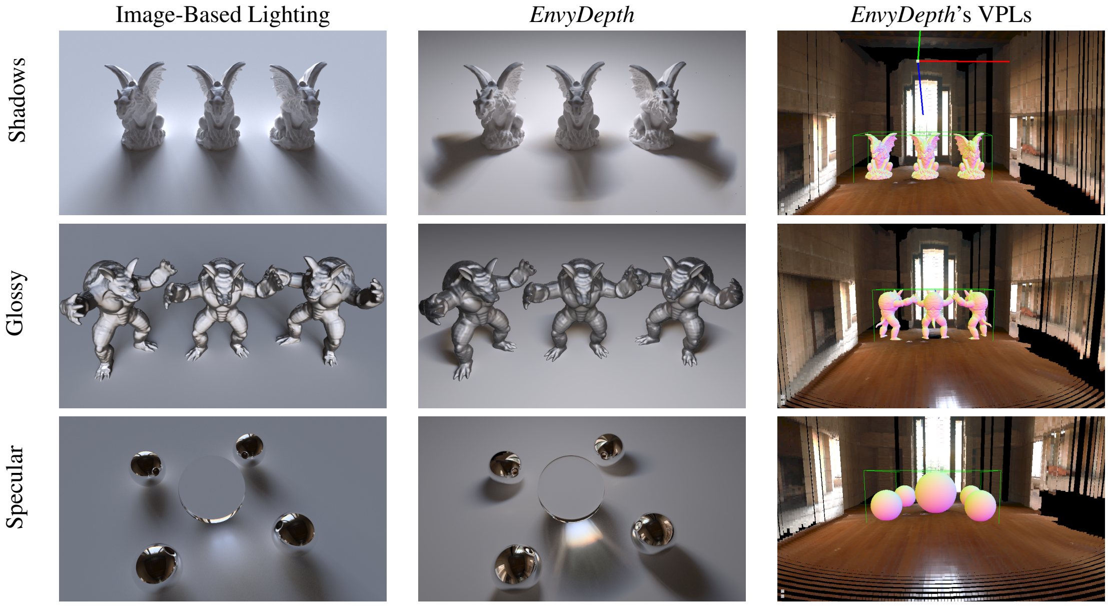


Figure 5: *A comparison of our method with image-based lighting [Deb98] using the* Ennis *environment map. In this comparison, we tested for different spatial effects such as: local shadows (Shadows), glossy BRDF (Glossy), and full specular effects (Specular).*

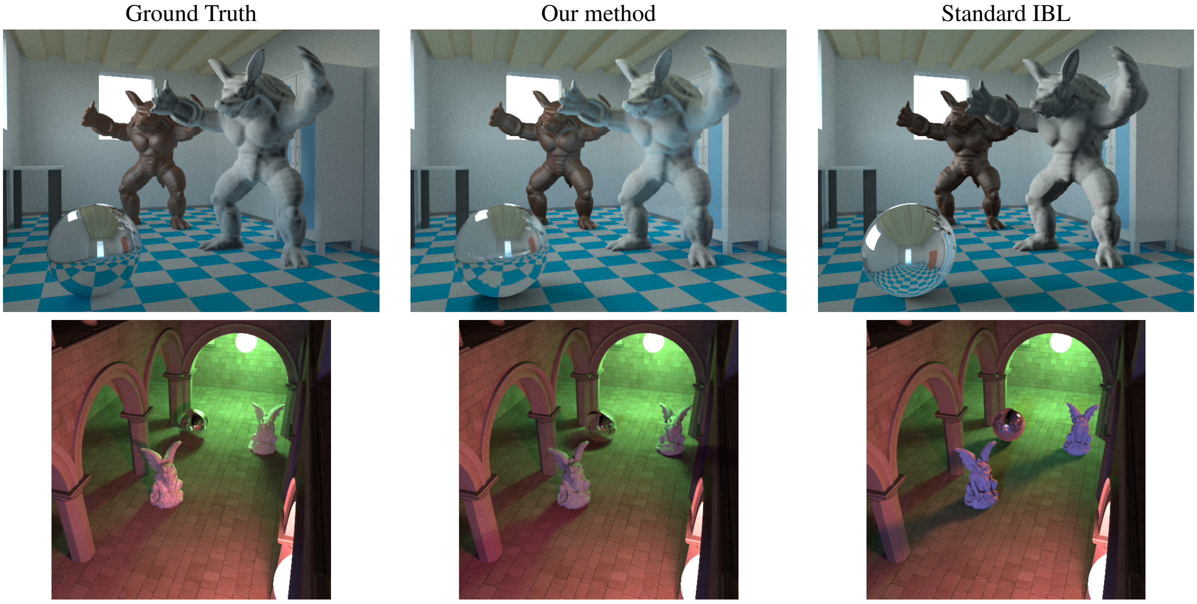


Figure 6: *A comparison with the ground truth. First column: ground truth, rendered from a full 3D scene. Second column: our method, inserted objects are rendered using our lighting representation and layered over a background. Third column: simple IBL, inserted objects are rendered using IBL and layered over a background. Top row (simple room): the reflections on the sphere are more accurate, and the lighting on the farther armadillo is much more coherent with ground truth. Bottom row (Sponza): the reflections on the sphere are more accurate, and the dominant illumination color takes into account the actual proximity of the objects to the light sources (the blue light opposite to the green one, on the other end of the courtyard, does not affect much the objects in the ground truth rendering. This is preserved by our method, while in the simple IBL, this is lost, since all the environment pixels have the same "distance" from the scene).*

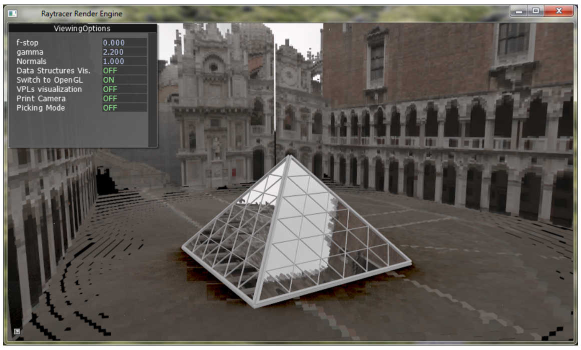
Comparisons with State-of-the-Art. We then compared the results of our method with two state-of-the-art approaches where a spatial definition of the lights in the scene is known. The goal of this comparison was to show that, by using our method, it is possible to obtain similar results to existing approaches that reconstruct the spatial nature of local illumination, but with a more flexible and easy-to-use approach.

Figure 7 shows the results of our method w.r.t. an image which was used in the Stereo Light Probe paper [CCC08]. As sketched in Section 2.3, Stereo Light Probe did the acquisition of the environment using a stereo rig which included two mirrored spheres, while our spatial description was modeled in a few minutes using one of the original environment maps. Our method is able to reconstruct correctly both the position of main illuminants and the diffuse contribution from the environment (e.g. the table surface), matching both the original photo and the original rendering.

In Figure 8, our method is compared with Karsch et al.'s [KHFH11] approach. Their method relies on an explicit reconstruction of the scene (see Section 2.3), starting from the photograph for the photographic composition. While this approach may work also without a complete environment map, the radiant areas not framed in the starting photo cannot be reconstructed and do not affect the rendering. This results in an incomplete illumination. In the test, we started from the full environment map of the scene, provided by the authors. While our result was able to match well both the starting photo and the rendering, it was not possible to exactly replicate the materials used: this can be noted in a different index of refraction in the glass and in a slightly different subsurface scattering in the egg.

Photographic composition preview. In order to compute a photographic composition (with synthetic element added to a real-world scene), it is often required to have both an environment map and a photograph that has been geometrically registered with respect to the environment and the 3D scene. Since, by using *EnvyDepth*, we build a three-dimensional representation of the environment, it can be also used when composing the scene to be rendered, and to have a fast but spatially accurate preview of the final rendering.

In Figure 9, we show a Louvre-style reflective pyramid placed inside a historical square in Venice. While still a preview of a more refined rendering, the geometric entities generated from our method are more than enough to display the position of the inserted object inside the three-dimensional structure of the square, evaluate the camera placement and, more important, have a realistic preview of the reflections of the square geometry over the inserted object surface. These reflections are more accurate, due to the spatial nature of the extracted VPLs, than using standard IBL rendering. Moreover, they may effectively help to evaluate the placement of a new architectural element in a urban setup.



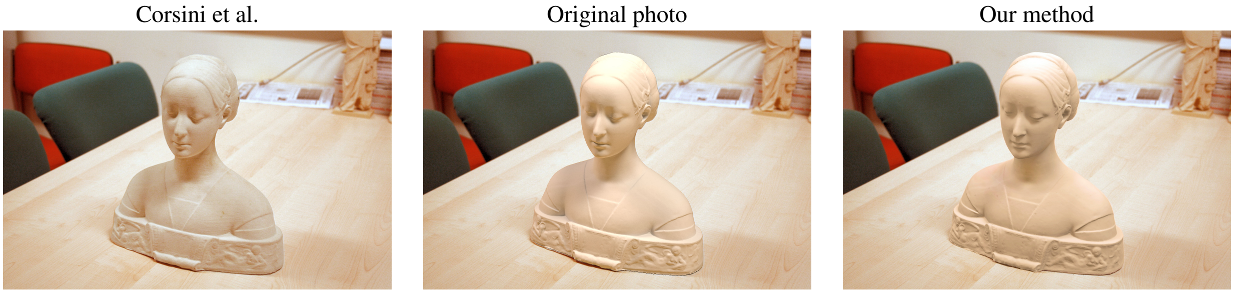
f-stop 0.000
gamma 2.200
Normals 1.000
Data Structures Vis. OFF
Switch to OpenGL ON
VPLs visualization OFF
Print Camera OFF
Picking Mode OFF

Figure 9: *A preview of a photographic composition. A new reflective architectural element has been inserted inside the scene extracted from the Doge environment map (see additional material). The spatial structure reconstructed by our method allows to preview the arrangement of the scene, and to produce geometrically-correct reflections.*

User Study. In order to assess the simplicity of use and the effectiveness of the *EnvyDepth*, we carried out a user study with 6 participants. The users had different backgrounds: experts in computer graphics, professional 3D modelers, and archaeologists with experience in modeling and rendering. The self-rating evaluation approach was followed. Firstly, each subject was instructed on the tool's goal and a video tutorial was shown to him/her (this video was 9 minutes long). Then, the subject was asked to create two illumination environments, one indoor (see Figure 1) and one outdoor (see Figure 2), starting from the corresponding environment map. While the first environment required on average 15-20 minutes, the second one required 5-10 minutes. This was a relatively short time given the first usage of the tool. It was also suggestive that the second environment map required half the time of the first to be generated. Finally, the subject had to evaluate her/his experience with the tool by answering a set of questions giving scores between 0 and 10; the obtained results were encouraging:

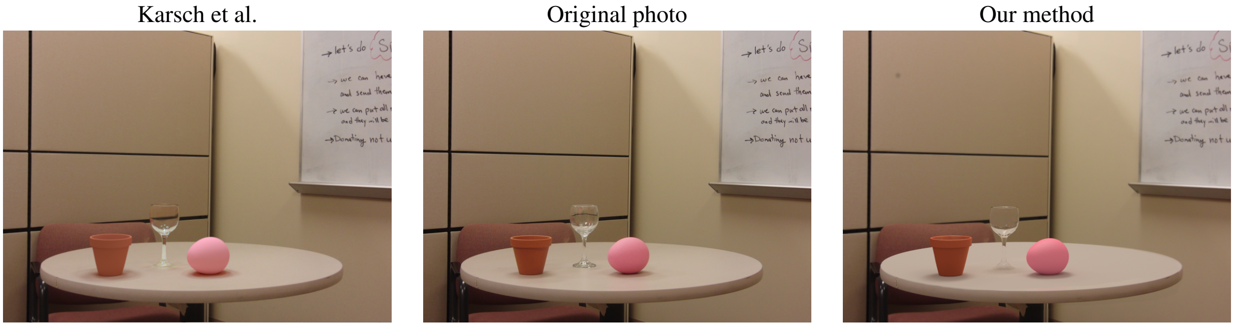
- *simplicity of use for the tool*: how simple was to use the provided primitives and the sketch-based interface. Scores: mean 7.5, min 6, max 9.
- *satisfaction of the results obtained*: how close was the result to the user idea of a well-modeled environment. Scores: mean 7.8, min 6, max 10.
- *tool's suitability in a production pipeline*: how useful, in the user opinion, could this tool be in a production pipeline. Scores: mean 8.7, min 8, max 10.

Limitations. The main limitation of the *EnvyDepth* is the set of possible primitives that in a few cases does not allow to model the scene in an accurate way. If several objects with peculiar features (i.e. different heights w.r.t. the ground, peculiar curved shapes) are present in the scene, it could be very hard to obtain a realistic representation. However, the key component for generating a coherent lighting is to correctly model the main light sources, which can be modeled



Corsini et al. Original photo Our method

Figure 7: *A comparison with the Corsini et al.'s method [CCC08] against a real-world scene: on the left side a rendered image of the scene using Corsini et al.'s method. Note that our method matches both the photograph and the Corsini et al.'s method, without the need of an acquisition device.*



Karsch et al. Original photo Our method

Figure 8: *A comparison with the Karsch et al.'s method [KHFH11] against a real-world scene: on the left side a rendered image of the scene using Karsch et al.'s method. Note that our method matches both the photograph and the Karsch et al.'s method, without the need of a cumbersome modeling step.*

in most of cases using the system primitives.

Therefore, even if a complex scene cannot be properly modeled, the impact on the final rendering introducing the simplifications can be minor, as shown in the curved Amsterdam road example provided in the **additional material**.

Another issue is holes can be present in a reconstruction, because occluded geometry may be present in the capture; e.g. see Figure 3. This may be not an issue when geometry is distant as in Figure 3.

Finally, an ideal case is to have a fully diffuse environment, to avoid issues with reflective objects when varying the point of view from the capturing point. This issue as shown in Figure 3 is not critical for distant geometry. Due to the fact we use the reconstruction only during shading, not for hit points, this is a neglectable issue.

6. Conclusions

We presented *EnvyDepth*, an interface to author local illumination from an environment map. The spatial representation of lighting is obtained with an intuitive sketching interface, that gives the possibility to easily model different types of simple primitives, in order to deal with heterogeneous (i.e. indoor and outdoor) scenes. The modeling of a scene takes only tens of seconds (for an expert user), and the user can also modify the scene to explore different visual effects for the same scene. We believe that for spatial lighting *EnvyDepth* could replace the more cumbersome methods of fully reconstructing complex geometry, or fully sampling the real-world scene. In particular, this would improve applications such as architectural simulation, visual effects, and computer games.

Some possible future improvements include: a set of operators for editing and tuning the already created primitives; strategies to fill, in the VPL representation, the parts of the scene which are occluded in the environment map; a method for re-sampling and densifying the scene representation to improve photographic composition.

Acknowledgement. We greatly thank Kevin Karsch for providing data for our comparisons. Paul Debevec, Christian Bloch (www.hdrlabs.com), and Gob Groothuis for their environment maps. Stanford 3D Repository for the Armadillo model, and Marko Dabrovic for the Sponza model. We also thank the anonymous reviewers for their feedback and suggestions to improve the paper. This work was founded by the EC IST IP project “3D-COFORM” (IST-2008-231809), and it was partially supported by the European COST Action IC-1005.

References

[AP08] An X., Pellacini F.: Appprop: all-pairs appearance-space edit propagation. *ACM Trans. Graph. 27*, 3 (Aug. 2008), 40:1–40:9. 4

[BCCS12] Banterle F., Corsini M., Cignoni P., Scopigno R.: A low-memory, straightforward and fast bilateral filter through subsampling in spatial domain. *Computer Graphics Forum 31*, 1 (February 2012), 19–32. 4

[BN76] Blinn J. F., Newell M. E.: Texture and reflection in computer generated images. In *SIGGRAPH '76: Proceedings of the 3rd annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1976), ACM, pp. 266–266. 2

[CCC08] Corsini M., Callieri M., Cignoni P.: Stereo light probe. *Computer Graphics Forum 27*, 2 (2008), 291–300. 2, 8, 9

[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 '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1998), ACM Press, pp. 189–198. 2, 3, 6, 7

[GDDA*11] Gerrits M., De Decker B., Ancuti C., Haber T., Ancuti C., Mertens T., Bekaert P.: Stroke-based creation of depth maps. In *Multimedia and Expo (ICME), 2011 IEEE International Conference on* (july 2011), pp. 1–6. 2, 4

[GWCO09] Guttmann M., Wolf L., Cohen-Or D.: Semi-automatic stereo extraction from video footage. In *Computer Vision, 2009 IEEE 12th International Conference on* (Oct. 2009), pp. 136–142. 2

[HAA97] Horry Y., Anjyo K.-I., Arai K.: Tour into the picture: using a spidery mesh interface to make animation from a single image. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques* (New York, NY, USA, 1997), SIGGRAPH '97, ACM Press/Addison-Wesley Publishing Co., pp. 225–232. 3

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

[KHFH11] Karsch K., Hedau V., Forsyth D., Hoiem D.: Rendering synthetic objects into legacy photographs. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (New York, NY, USA, 2011), SA '11, ACM, pp. 157:1–157:12. 2, 8, 9

[KN09] Kim Y., Noh J.: Lightshop: An interactive lighting system incorporating the 2d image editing paradigm. In *Proceedings of the 5th International Symposium on Advances in Visual Computing: Part II* (Berlin, Heidelberg, 2009), ISVC '09, Springer-Verlag, pp. 59–70. 2

[KSvA*08] Korn M., Stange M., von Arb A., Blum L., Kreil M., Kunze K.-J., Anhenn J. A., Wallrath T., Grosch T.: Interactive augmentation of live images using a hdr stereo camera. *Journal of Virtual Reality and Broadcasting 4*, 12 (January 2008). 2

[LGYG12] Liao M., Gao J., Yang R., Gong M.: Video stereolization: Combining motion analysis with user interaction. *Visualization and Computer Graphics, IEEE Transactions on 18*, 7 (july 2012), 1079–1088. 2

[MH84] Miller G., Hoffman C. R.: Illumination and reflection maps: Simulated objects in simulated and real environments. In *Siggraph '84 Advanced Computer Graphics Animation seminar note* (New York, NY, USA, July 1984), ACM Press. 2

[OMSI07] Okabe M., Matsushita Y., Shen L., Igarashi T.: Illumination brush: Interactive design of all-frequency lighting. In *15th Pacific Conference on Computer Graphics and Applications (PG2007)* (29 2007-nov. 2 2007), pp. 171 –180. 2

[OP11] Ou J., Pellacini F.: Lightslice: matrix slice sampling for the many-lights problem. *ACM Trans. Graph. 30*, 6 (Dec. 2011), 179:1–179:8. 5

[Pel10] Pellacini F.: envylight: an interface for editing natural illumination. In *ACM SIGGRAPH 2010 papers* (New York, NY, USA, 2010), SIGGRAPH '10, ACM, pp. 34:1–34:8. 2, 4

[PH10] Pharr M., Humphreys G.: *Physically Based Rendering: From Theory to Implementation (2nd Edition)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. 5

[RDGK12] Ritschel T., Dachsbacher C., Grosch T., Kautz J.: The state of the art in interactive global illumination. *Comput. Graph. Forum 31*, 1 (Feb. 2012), 160–188. 3

[RSSF02] Reinhard E., Stark M., Shirley P., Ferwerda J.: Photographic tone reproduction for digital images. *ACM Trans. Graph. 21*, 3 (2002), 267–276. 4, 6

[RWP*10] Reinhard E., Ward G., Pattanaik S., Debevec P., Heidrich W., Myszkowski K.: *High Dynamic Range Imaging, Second Edition: Acquisition, Display, and Image-Based Lighting (The Morgan Kaufmann Series in Computer Graphics)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2010. 4

[SKK*11] Smolic A., Kauff P., Knorr S., Hornung A., Kunter M., Müller M., Lang M.: Three-Dimensional Video Postproduction and Processing. *Proceedings of the IEEE 99*, 4 (April 2011), 607–625. 2

[SSI99] Sato I., Sato Y., Ikeuchi K.: Acquiring a radiance distribution to superimpose virtual objects onto a real scene. *IEEE Transactions on Visualization and Computer Graphics 5*, 1 (/1999), 1–12. 2

[Tat05] Tatarchuk N.: Irradiance volumes for games, 2005. 1

[UGLY08] Unger J., Gustavson S., Larsson P., Ynnerman A.: Free form incident light fields. *Computer Graphics Forum 27*, 4 (2008), 1293–1301. 1, 2

[UKL*13] Unger J., Kronander J., Larsson P., Gustavson S., Löw J., Ynnerman A.: Spatially varying image based lighting. *To appear in Computers and Graphics* (2013). 2

[UWH*03] Unger J., Wenger A., Hawkins T., Gardner A., Debevec P.: Capturing and rendering with incident light fields. In *EGRW '03: Proceedings of the 14th Eurographics workshop on Rendering* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 141–149. 2

[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 (July 2005), 1098–1107. 3

[WLF*11] Wang O., Lang M., Frei M., Hornung A., Smolic A., Gross M.: Stereobrush: interactive 2d to 3d conversion using discontinuous warps. In *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling* (New York, NY, USA, 2011), SBIM '11, ACM, pp. 47–54. 2, 4

[WMG*09] Wald I., Mark W. R., Günther J., Boulos S., Ize T., Hunt W., Parker S. G., Shirley P.: State of the Art in Ray Tracing Animated Scenes. *Computer Graphics Forum 28*, 6 (2009), 1691–1722. 6

[XLJ*09] Xu K., Li Y., Ju T., Hu S.-M., Liu T.-Q.: Efficient affinity-based edit propagation using k-d tree. *ACM Trans. Graph. 28*, 5 (Dec. 2009), 118:1–118:6. 4