

Grabbing Real Light - Toward Virtual Presence

Pascal Gautron*

Jean-Eudes Marvie†
Technicolor Research & Innovation

Cyprien Buron‡

Abstract

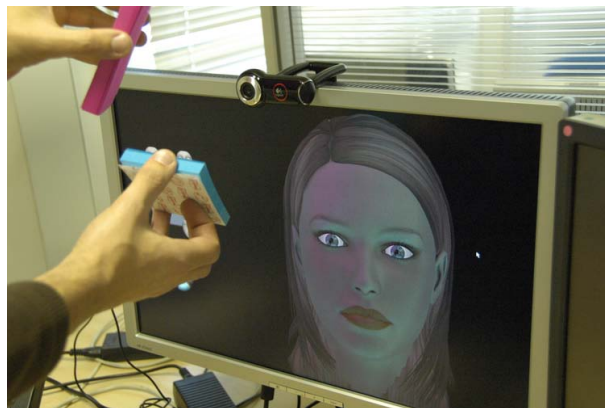
Realistic lighting of virtual scenes has been a widely researched domain for decades. Many recent methods for realistic rendering make extensive use of *image-based lighting*, in which a real lighting environment is captured and reused to light virtual objects. However, the capture of such lighting environments usually requires still scenes and specific capture hardware such as high end digital cameras and mirror balls. In this paper we introduce a simple solution for the capture of plausible environment maps using one of the most widely used capture device: a simple webcam. In order to be applied efficiently, the captured environment data is then filtered using graphics hardware according to the reflectance functions of the virtual surfaces. As the limited aperture of the webcam does not allow the capture of the entire environment, we propose a simple method for extrapolating the incoming lighting outside the webcam range, yielding a full estimate of the incoming lighting for any direction in space. Our solution can be applied to direct lighting of virtual objects in all virtual reality applications such as virtual worlds, games etc. Also, using either environment map combination or semi-automatic light source extraction, our method can be used as a lighting design tool integrated for post-production.

CR Categories: K.6.1 [Management of Computing and Information Systems]: Project and People Management—Life Cycle; K.7.m [The Computing Profession]: Miscellaneous—Ethics

1 Introduction

Realistic lighting of virtual scenes has been a research challenge since the emergence of computer graphics. Many multimedia applications benefit from this research, ranging from video games to virtual training or movie post-production. Also, the capture of real environments under the form of environment maps has been widely investigated during the past decade. However, the capture of such lighting environments requires specific hardware, such as high end digital cameras, mirror balls, or HDR video capture devices.

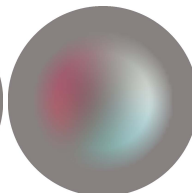
We propose a simple solution for the capture of plausible environment maps using one of the most widely used capture device: a simple webcam. In order to be applied efficiently, the captured environment data is first filtered using graphics hardware according to the reflectance functions of the virtual surfaces. As the limited aperture of the webcam does not allow the capture of the entire environment, we introduce a simple method for extrapolating the incoming lighting outside the webcam range, yielding a full estimate of the incoming lighting for any direction in space. Our solution can be applied to direct lighting of virtual objects in all virtual reality applications such as virtual worlds, games etc. Also, using



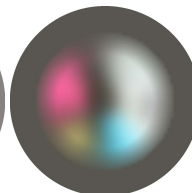
(a) Real lighting capture using webcam



(b) Virtual mirror ball



(c) Diffuse filtering



(d) Glossy filtering



(e) Real-time rendering using environment maps (b,c,d)

Figure 1: Starting with a capture of the surrounding environment using a simple webcam (a), our method maps the captured image onto a virtual mirror ball (b). The obtained spherical information is then filtered in real-time using graphics hardware to obtain the desired filtered environment maps, such as diffuse (c) or glossy (d). The environment maps are then used for real-time shading of virtual objects (e).

simple light source extraction solutions, our method can be used as a lighting design tool integrated in post-production workflows.

This paper is organized as follows: after a brief description of related previous work on environment capture and lighting, Section 3 describes our capture setup. In Sections 4 and 5, we show how the captured information is filtered and extrapolated to obtain a full description of the lighting environment. Section 6 presents the results obtained using our method, in the contexts of virtual reality and lighting edition.

*e-mail: pascal.gautron@technicolor.com

†e-mail: jean-eudes.marvie@technicolor.com

‡e-mail: cyprien.buron@technicolor.com

2 Previous Work

Our approach is based on the classical environment mapping technique, which consists in considering that virtual objects are lit only by distant light sources. Consequently, for a given incoming direction, the incoming lighting is considered the same regardless of the location in space. The acquisition, filtering and rendering of environment maps have been studied intensively in many computer graphics works, such as [Heidrich and Seidel 1999; Ramamoorthi and Hanrahan 2002; Krivánek and Colbert 2008].

A simple protocol for the acquisition of environment maps was thoroughly investigated, starting with Debevec and Malik [Debevec and Malik 1997] and numerous enhancements as those described in [Stumpfel et al. 2006]. The technique is based on a high quality digital camera taking a set of pictures of a mirror ball using different exposures. The pictures are then combined to create a high dynamic range image representing the incoming lighting from any direction. Note that the need for several pictures with variable exposures restrain the use of this method to still images. High dynamic range capture of dynamic scenes has also been investigated, but require specific hardware [Havran et al. 2005; Mannami et al. 2007].

Our method described hereafter focuses on the use of inexpensive, off-the-shelf hardware for plausible lighting acquisition.

3 Capturing Real Lighting

In this paper our aim is plausible, dynamic lighting of virtual objects. Therefore, the capture device has to be located as close as possible to the objects displayed in the virtual world. For real-time virtual reality applications, the location of choice for the webcam is then directly above or under the display to ensure the spatial coherence of the lighting (Figure 1).

As described in the previous Section, classical environment map capture considers taking a picture of a mirror ball to acquire lighting over the entire sphere around the viewer. In our case, the webcam tend to have a very limited aperture, which is far from even a wide-angle lens. As our aim is the reconstruction of the entire environment lighting, we first map the acquired data onto a virtual sphere as shown in Figure 2, yielding an angular map of the environment. As described in Debevec *et al.* [Debevec and Malik 1997], a direction \mathbf{d} corresponds to a pixel $(x, y) = (\mathbf{d}_x, \mathbf{d}_y)$ in the target angular map, where:

$$r = (1/\pi) \arccos(\mathbf{d}_z / \sqrt{\mathbf{d}_x^2 + \mathbf{d}_y^2}) \quad (1)$$

Likewise, we can associate a pixel (x, y) of the captured video to a direction \mathbf{d}_{cap} :

$$\mathbf{d}_{\text{cap}} = \left(\tan \frac{\gamma_x}{2} \left(\frac{2x}{\text{Res}_x} - 1 \right), \tan \frac{\gamma_y}{2} \left(\frac{2y}{\text{Res}_y} - 1 \right), 1 \right) \quad (2)$$

γ_x and γ_y are the respective horizontal and vertical apertures or the capture device, and Res_x and Res_y are the horizontal and vertical resolutions of the captured image. Using this correspondence, the captured image is then mapped onto an angular map, yielding a partial representation of the environment map. The two next sections consider the filtering and extrapolation of the partial environment map to obtain lighting information over the entire sphere.

4 Environment Filtering

A classical method for lighting diffuse and glossy surfaces using an environment map relies on a preprocess in which the environment is

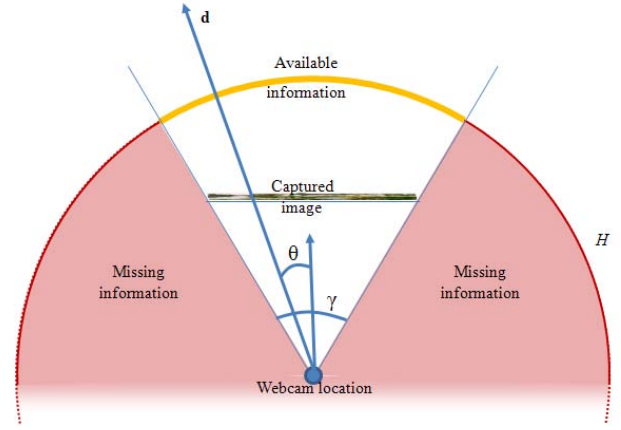


Figure 2: Virtual mirror ball for acquisition of incoming lighting information: the webcam captures the lighting incoming from a small subset of the directions of the sphere. A coherent extrapolation of the unknown parts must be performed to obtain a plausible environment map.

convolved with the target surface reflectance function. More explicitly, for each direction \mathbf{d} of the environment, the process integrates the incoming lighting on the hemisphere H around \mathbf{d} :

$$L_{\text{diffuse}}(\mathbf{d}) = \int_H \cos \theta L(\omega) d\omega \quad (3)$$

This step is usually performed on the CPU by explicitly sampling all the possible pixels of the mirror ball image and hence requires a huge amount of computations, especially for high-definition images. As our approach is the real-time convolution of such environments using graphics hardware, we use a simplified filtering method inspired from [Krivánek and Colbert 2008]. For each pixel p of the virtual mirror ball image, we first determine the corresponding direction \mathbf{d} . Then, for diffuse surfaces, we take a set of random directions of the sphere according to a cosine distribution around \mathbf{d} . Using Monte Carlo integration, Equation 3 becomes:

$$L_{\text{diffuse}}(\mathbf{d}) \approx \pi \sum_{i=1}^N \frac{L(\omega_i)}{\cos \theta_i} \quad (4)$$

In the context of real-time computation, this integration must be carried out in a very small amount of time. On a high resolution image, we could afford only a very small number of samples (typically 5) per environment direction even on the latest graphics hardware. Using such a small number of samples yields very noisy results both in space and time, and hence light speckles and flickering in the final image. As the diffuse or moderately glossy lighting tends to vary smoothly over directions we can reduce the resolution of the generated filtered environment map to very small values, such as 128×128 . The sampling quality of each pixel of the filtered environment map can then be higher (typically 50 samples/pixel) while still meeting the real-time constraints.

As described in Section 3, the available information on the incoming lighting is partial due to the limited aperture of the webcam, and needs to be extrapolated.

5 Environment Extrapolation

The aperture of the webcam being limited, a straightforward way of extrapolating the incoming lighting is the extension of border pixels of the captured image over the rest of the sphere (Figure 3(a)).

In this case, each border pixel spans a very large solid angle in the environment map, yielding a possibly inaccurate and unstable estimate of the surrounding lighting. Another simple extrapolation method consists in using a user-defined value for the missing parts of the environment. However, an inappropriate value yields incoherent lighting in the final image (Figure 3(b)). Instead, we choose to use an average of the captured incoming lighting on the unknown parts of the hemisphere. The environment map generation is then divided into two parts: first, we generate the partial environment maps as detailed in the previous Section. At the same time, we keep track of the percentage of sample directions corresponding to unknown lighting values. Such samples yield a zero contribution to the lighting incoming in the target direction. The output of the partial environment map generation is then an RGBA texture, where the RGB channels contain the convolved lighting values and the alpha channel stores the amount of directions corresponding to unknown values.

A good notion of the average lighting is provided by the center pixel of the partial diffuse environment map (Figure 3(c)): the directions used during the convolution are mostly located within the range of the webcam. Therefore, the pixels of the final environment map are obtained through a weighted sum of the partial environment map L_{part} and this average value L_{avg} . The weight value is the alpha channel of the partial map, the amount of directions corresponding to unknown lighting values. The extrapolated lighting is then defined by:

$$L_{final}(\mathbf{d}) = \alpha L_{avg} + (1 - \alpha) L_{part} \quad (5)$$

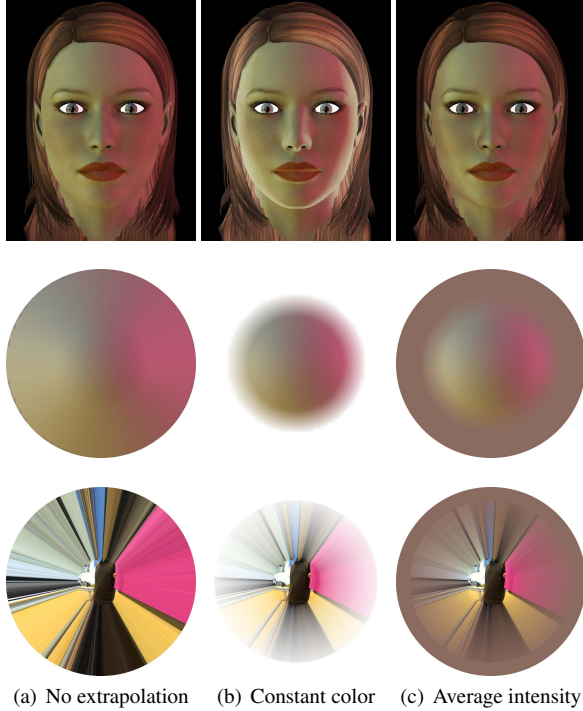


Figure 3: Lighting extrapolation is crucial to reconstruct a plausible environment map: without extrapolation (a) the pixels on the edge of the acquired image are extended over a large part of the hemisphere, possibly leading to important over- or under-estimations of the surrounding lighting. With a fixed, user-defined default value (b), the lighting coming from grazing angles looks incoherent with the front lighting. Using the middle of the convoluted environment map yields an approximate yet coherent lighting information, hence increasing the rendering realism.

6 Applications and Results

The results presented in this paper have been generated using a Logitech QuickCam Pro 9000 plugged on a laptop running an Intel Core 2 Duo 2.4GHz CPU with a QuadroFX 1600M GPU. The original image is captured at 800×600 , and converted to a 600×600 angular map. The angular map is then filtered to generate a diffuse and a glossy environment maps with resolution 64×64 . The entire capture and filtering process runs at 55fps on our laptop. With the rendering of the virtual face (40K polygons) using the generated environment maps and high quality diffuse and normal maps, the entire process runs at 25fps.

6.1 Reconstructed Environment Maps

Our method aims at a plausible yet imprecise reconstruction of the environment lighting through extrapolation. However, we have compared our extrapolated environment maps with actual, acquired environment maps. The capture was performed as in [Debevec and Malik 1997] by taking a picture of a mirror ball using a high quality digital camera. The picture was then converted to an angular map and filtered to obtain a diffuse environment map (Figure 4(a, b)). A picture of the same environment has been captured using a webcam located at the same location as the mirror ball (Figure 4(d, e)). Figures 4(c, f) show the rendering of a virtual face using the actual and extrapolated lightings with similar white balances. Our tests show that even though the webcam-based environment extrapolation misses parts of the environment, our method provides a coherent lighting reconstruction compared to the actual capture.

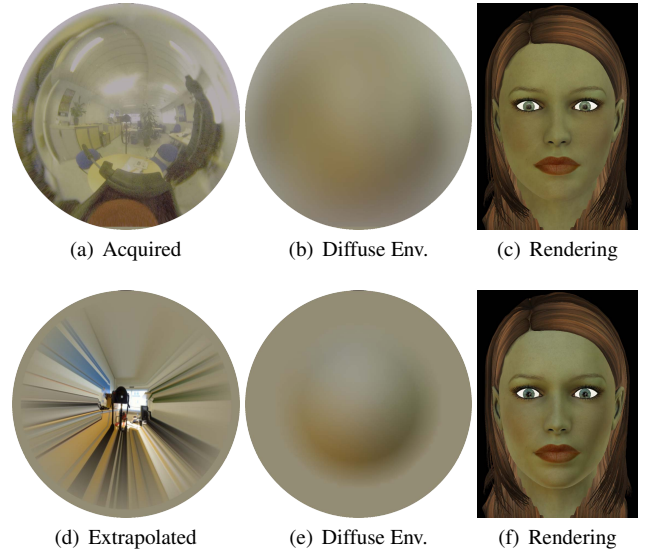


Figure 4: Comparison between actual and extrapolated environments map. Upper row: The actual environment (a) is captured using a mirror ball and a Nikon D70 digital camera. This acquired environment undergoes an offline diffuse convolution (b). Both environment maps are used to render a virtual face (c). Lower row: a picture of the environment is taken using a webcam and extrapolated using our method (d). A diffuse environment map is generated on-the-fly as in Section 4 (e), and a virtual face is rendered using those maps (f).

6.2 Virtual Reality

The sensation of immersion in virtual worlds is greatly enhanced by the coherence between the real and virtual lightings: this coherence tends to seamlessly integrate the displayed virtual components into

the real world. As shown in Figure 1, one of the first applications of our method is virtual conferencing. Each user is represented by an avatar in the virtual world. As our method provides plausible, dynamic lighting reconstruction, the gap between the real user and the virtual avatar tends to be reduced, thus enhancing the presence of the avatar. A wide range of other applications can benefit from our method, such as virtual training or collaborative virtual worlds.

6.3 Lighting Edition Tool

As our method allows the user to directly interact with the lighting of the virtual scene, our work can be used as a basis for an interactive light editing tool. Starting with a 3D model, the user can use real light sources to build a prototype of the virtual lighting environment. The reconstructed environment map can then be used as is to light the virtual scenes in production. Another possibility relies on the combination of several environment maps to build the target lighting: as shown in Figure 5, the lighting is captured several times, then combined to obtain the desired effect. Also, the user could use one of the above methods to generate an environment map and then perform point light source extraction [Cohen and Debevec 2001] corresponding to the major lighting elements of the map.

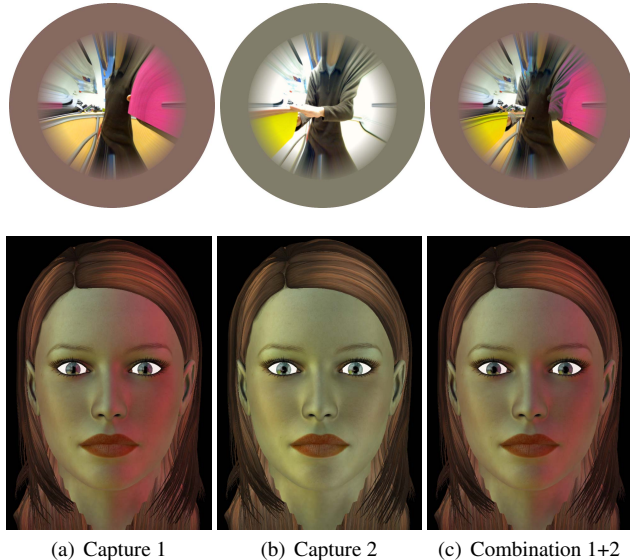


Figure 5: Lighting is captured several times (a,b), and then combined to create a new environment map (c).

6.4 Discussion: Artificially Widening the Aperture

As the aperture of a webcam is typically small, most of the spherical information is obtained through extrapolation as shown in Figure 2. A simple method for reducing this problem without introducing prohibitive artifacts consists in artificially widening the aperture of the webcam. Thus the image spans a larger part of the sphere, reducing the amount of extrapolated data. While physically incorrect, this solution emphasizes the movement of real objects around the camera, hence enhancing the sensation of immersion of the virtual object into the real world. For lighting design, this provides a simple and intuitive way of creating light sources for any direction of the hemisphere.

6.5 Discussion: High Dynamic Range

Usually, detailed environment maps are provided as high dynamic range (HDR) images generated from a set of photographs taken

with different exposures. In our case, simple digital cameras such as webcams do not offer real-time capture of HDR images. In the spirit of [Kang et al. 2003; Bennett and McMillan 2005], future work will consider temporal bracketing to increase the dynamic range of the captured lighting by dynamically modifying the exposure of the webcam sensor.

7 Conclusion

This paper introduces a simple and inexpensive method for extracting lighting information out of the images provided by a simple webcam or other capture devices. We first remap the captured images onto an angular map and extrapolate the lighting information in unknown parts of the surrounding environment. The extrapolated environment map is then filtered in real-time using graphics hardware for high quality rendering. While approximate, our method provides plausible environment maps usable for virtual reality and multimedia applications, and runs in real-time even on low-end graphics hardware. We also consider the use of our solution in the context of lighting design using environment combination. While providing visually pleasing results, our method could be particularly enhanced by approximate HDR capture using dynamic exposure changes and image merging in the temporal domain.

References

- BENNETT, E. P., AND MCMILLAN, L. 2005. Video enhancement using per-pixel virtual exposures. In *Proceedings of SIGGRAPH*, 845–852.
- COHEN, J., AND DEBEVEC, P. 2001. LightGen, HDRShop plugin. <http://gl.ict.usc.edu/HDRShop/lightgen>.
- DEBEVEC, P., AND MALIK, J. 1997. Recovering high dynamic range radiance maps from photographs. In *Proceedings of SIGGRAPH*, 369–378.
- HAVRAN, V., SMYK, M., KRAWCZYK, G., MYSZKOWSKI, K., AND SEIDEL, H.-P. 2005. Interactive system for dynamic scene lighting using captured video environment maps. In *Proceedings of Eurographics Symposium on Rendering*, 31–42.
- HEIDRICH, W., AND SEIDEL, H.-P. 1999. Realistic, hardware-accelerated shading and lighting. In *Proceedings of SIGGRAPH*, 171–178.
- KANG, S. B., UYTENDAELE, M., WINDER, S., AND SZELISKI, R. 2003. High dynamic range video. In *Proceedings of SIGGRAPH*, 319–325.
- KŘIVÁNEK, J., AND COLBERT, M. 2008. Real-time shading with filtered importance sampling. In *Computer Graphics Forum*, vol. 27, 1147–1154.
- MANNAMI, H., SAGAWA, R., MUKAIGAWA, Y., ECHIGO, T., AND YAGI, Y. 2007. Adaptive dynamic range camera with reflective liquid crystal. *J. Vis. Comun. Image Represent.* 18, 5, 359–365.
- RAMAMOORTHY, R., AND HANRAHAN, P. 2002. Frequency space environment map rendering. In *Proceedings of SIGGRAPH*, 517–526.
- SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of SIGGRAPH*, 527–536.
- STUMPFEL, J., JONES, A., WENGER, A., TCHOU, C., HAWKINS, T., AND DEBEVEC, P. 2006. Direct HDR capture of the sun and sky. In *SIGGRAPH 2006 Courses*.