ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/220805290

Dynamic light amplification for immersive environment rendering

Conference Paper · December 2009

DOI: 10.1145/1670252.1670303 · Source: DBLP

CITATIONS
0

READS
154

2 authors:

Andrei Sherstyuk
University of Hawai'i System
52 PUBLICATIONS **594** CITATIONS
SEE PROFILE

Anton Treskunov
30 PUBLICATIONS **220** CITATIONS
SEE PROFILE

All content following this page was uploaded by Anton Treskunov on 22 May 2014.

The user has requested enhancement of the downloaded file.

Dynamic Light Amplification for Immersive Environment Rendering

Andrei Sherstyuk*
Avatar Reality

Anton Treskunov †
Samsung

Abstract

Two common limitations of modern Head Mounted Displays (HMD): the narrow field of view and limited dynamic range, call for rendering techniques that can circumvent or even take advantage of these limiting factors. In order to improve visual response from HMDs, we propose a new method of creating various lighting effects, by using view-dependent control over lighting. Two implemented examples are provided: simulation of a blinding effect in dark environments, and contrast enhancement. The paper is intended for the audience interested in developing HMD-based Virtual Reality applications with improved scene illumination.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques;

Keywords: view-dependent lighting, improved HMD response.

1 Introduction

Viewing with a tracked HMD in immersive Virtual Reality (VR) is fundamentally different from desktop and projector based systems. Unlike stationary displays, a tracked HMD allows viewers to continuously "sample" the scene in all directions by using natural head movements, creating an integrated image of the environment. This process is similar to what happens in reality, where eyes are actively scanning the scene, focusing on important features.

Human eye has a very high dynamic contrast ratio, approximately one to million. Spatial resolution, across the visible field, on the contrary, is relatively low. We are able to see objects in sharp detail only if they fall on a very small area of the retina, called fovea centralis, where most light photoreceptors are concentrated. Projected outwards into the viewable scene, this area is about the size of the full moon, or 15 degrees. Everything outside of this high-acuity area is basically a blur [Williamson and Cummins 1983].

Building on [2009], we advocate an idea that rendering for wearable displays must account for and, whenever possible, take advantage of these features of human vision. This approach will help to circumvent the limitations of the current displays. In particular, low contrast and brightness of modern HMDs do not allow to reproduce the whole range of light intensities that the human eye can perceive. As a result, scenes with wide range of luminance often look unconvincing in VR, which significantly affects the sense of presence. We propose a framework for rendering immersive environments, based on the principle that objects' appearance is view-dependent. In this work, we focus on improving lighting part of the rendering process, and provide a palliative solution to the problem of limited brightness and contrast of available HMDs.

*e-mail: andrei@avatar-reality.com

†e-mail:anton.t@sisa.samsung.com; work performed while at USC.

2 Related work

At the first glance, the problem described above falls into the scope of High Dynamic Range (HDR) rendering. HDR imaging techniques were initially developed for photography and then were applied to computer graphics [Larson and Shakespeare 1998]. HDR rendering aims to compensate for the limits (insufficient contrast and brightness) set by traditional graphics pipeline designed for display devices and show as much detail as possible in areas with extreme illumination. Being computationally expensive, HDR techniques were first used predominantly in off-line rendering. With the advent of powerful and inexpensive video cards, HDR methods moved into domain of real-time graphics and were implemented in graphics engines, both commercial (CryENGINE [1], Unreal [2], Unigine [3]) and open source (Ogre [4]). The typical set of real-time HDR and related techniques include

- *tone-mapping*, which allows to transform a wide range of scene radiance values onto displayable intensities interval allowed by display hardware [Devlin et al. 2002];
- *glow effects*, or color bleeding of bright areas over the edge of occluding objects, which makes these areas look brighter than they are displayed [Spencer et al. 1995];
- *lens flare*, a visual effect which appears when camera is directed towards a very bright light source, such as the sun [King 2000];
- *eye accommodation*, an effect that mimics how human eyes adjust to changes in lighting conditions [Ferwerda et al. 1996].

Real-time HDR rendering became very popular in video games, because images it produces are visually appealing. However, not all of these methods seem to be suitable for immersive VR applications, for a number of reasons.

First, rendering in games assumes the use of a stationary display device: a computer screen or a television set. As a result, the model layout and lighting are optimized for viewing the scene as a predominantly static image. All the visual information is delivered to users at once, post-processed for the entire frame. In other words, the scene is rendered without considerations where exactly is the user's focus of attention.

Secondly, certain HDR effects, such as lens-flare and cross-flare, reproduce artifacts that can only happen when conventional optics is used, for example, a physical camera. In VR, the visual content is delivered by two virtual cameras, which are free from flare. Therefore, use of lens-flare effect is not justified in VR, unless viewers are offered virtual binoculars or some other viewing-aid devices.

Finally, effects that mimic adaptation of human eyes to changes in light brightness may cause discomfort. These simulated effects will be competing with the actual physiological adaptation. It is not clear, if concurrent real and simulated accommodation will enhance or degrade user experience in VR.

[1] http://www.crytek.com/
[2] http://www.unrealtechnology.com/
[3] http://unigine.com
[4] http://www.ogre3d.org

In summary, picture-perfect rendering provided by real-time HDR techniques, is perfect for pictures, or stationary displays, when the whole scene fits into a single view frame. For dynamic viewing in immersive VR applications, when a perpetual motion of user's focus of attention can be estimated, other methods are required.

3 View-Dependent Lighting

Our approach is based on an observation that for non-panoramic HMDs, orientation of the user head approximates his or her gaze direction. Field of view (FOV) of most popular HMD models is relatively low, ranging between 40 and 60 degrees diagonally [Bungert 2006]. To compare, human visual field extends to 180 degrees horizontally and 75 degrees vertically. Limited field of view leads to a "tunnel vision" effect which is generally regarded as one of the most objectionable drawbacks of HMDs.

However, this disadvantage becomes a very helpful feature, for tasks that require estimation of the user gaze direction. The tunnel vision effect prompts viewers to turn their heads more actively, in order to see the scene outside of the visible HMD frame. As a result, fast saccadic eye movements that happen in real life, are replaced by wide and relatively slow head rotations in VR. Head rotation can be reliably tracked by a variety of available devices, and used as an approximation of the view direction.

3.1 Algorithm

On each cycle of the main graphics loop, each light source is checked against the current user position and orientation. If the light source appears to be in the high acuity area of the user's gaze (approximated by the head direction, as described above), the light intensity is magnified to match the higher eye sensitivity in this area.

For different types of light sources, the magnification factors are computed differently. For positional lights, this factor is inversely proportional to the distance from the position of the light source to the center of the viewing plane, in the camera space. For directional lights and spotlights, the direction of the light source must also be taken into account: if the camera and the light source are oriented towards each other, the light becomes brighter. The view-dependent light control algorithm for different types of light sources is summarized in Figure 1.

for all active light source L **do**

$$factor \leftarrow \begin{cases} -L.direction \bullet cam.direction, & \text{for directional light;} \\ 1/\text{distance\_to\_screen\_center}(L.pos), & \text{for positional light;} \\ 1/\text{distance\_to\_screen\_center}(L.pos) - L.direction \bullet cam.direction, & \text{for spot light.} \end{cases}$$

$L.intensity \leftarrow L.intensity * factor$

end for

Figure 1: *Summary of the view-dependent lighting control. The values returned by dot product $\bullet$ are clamped to $[0; 1]$ range. Evaluating distance to screen center requires transformation of light position from world to camera space. An alternative way of computing distance factor is shown in Figure 2.*

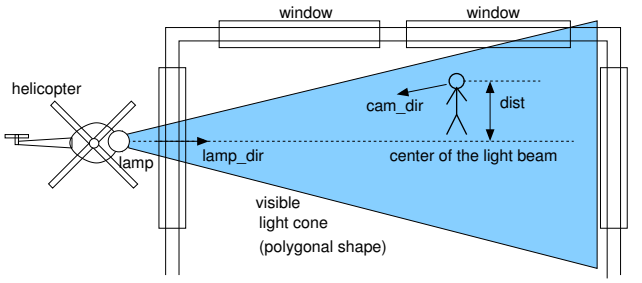
3.2 VR System

The algorithm was implemented in open-source 3D visualization system Flatland [2002]. Flatland runs under Linux with OpenGL graphics API. It provides stereoscopic rendering for both active and passive stereo configurations. In our system, both types of displays were used: Virtual Research V8 HMD (640 x 480 pixels, 60° FOV, passive) and 5DT800 HMD (800 x 600 pixels, 40° FOV, active). For motion tracking, Flock of Birds by Ascension was used, running in 9 feet extended tracking range.

3.3 Example 1: Programming a Flying Projector

In Flatland, each light source is a node in the scene graph, with its own $update()$ function. These functions are executed in the main graphics loop, after updating all scene transformations and before lighting is applied. During $update()$, each light source inquires the system for the current position and orientation of the camera and adjusts its intensity accordingly.

If a light source is attached to an object with visible geometry, calculations of light amplification factors must also account for additional light that comes from that object. For example, a spotlight may be placed at the apex of a semi-transparent polygonal cone, which will make the light beam visible, shown as shaded area in a top diagram in Figure 2. In this scene, a very bright spotlight is attached to a helicopter object that flies around a building, looking into windows. When the user moves closer to the axis of the beam, the visible light cone must look brighter. The standard OpenGL spotlight will not provide this effect, because it does not account for the viewer's position inside the cone. Figure 2 shows how light magnification is calculated for the helicopter lamp. Rendered results are shown in Figures 3 and 6 Video clip is available [5].



1: **procedure** HelicopterLightUpdate(Camera cam, Light $lamp$)
2: $dist \leftarrow$ distance from cam position to the $lamp$ light beam line
3: $D \leftarrow \begin{cases} 1 & \text{if } dist < dist_{min}; \\ 0 & \text{if } dist > dist_{max}; \\ \text{falls off linearly} & \text{otherwise.} \end{cases}$
4: $A \leftarrow -lamp.direction \bullet cam.direction$
5: $lamp.intensity \leftarrow c_0 + c_1 * D + c_2 * A$
6: **end procedure**

Figure 2: *A scene layout and $update()$ function of the helicopter lamp. The light intensity is modified according to the relative position and orientation of the light with respect to the viewer. Rendered results are shown in Figure 3. Notes. Line 3, calculating distance factor D: constants $dist_{min}$ and $dist_{max}$ were set at 1 and 10 meters, respectively. Line 4: Angle factor A falls off as a cosine. Line 5: c_0, c_1, c_2 are user-defined parameters, useful values are 5, 5, 6.*

[5] http://www.tri.jabsom.hawaii.edu/tri/video/projects-triage/full/helicopter2.avi

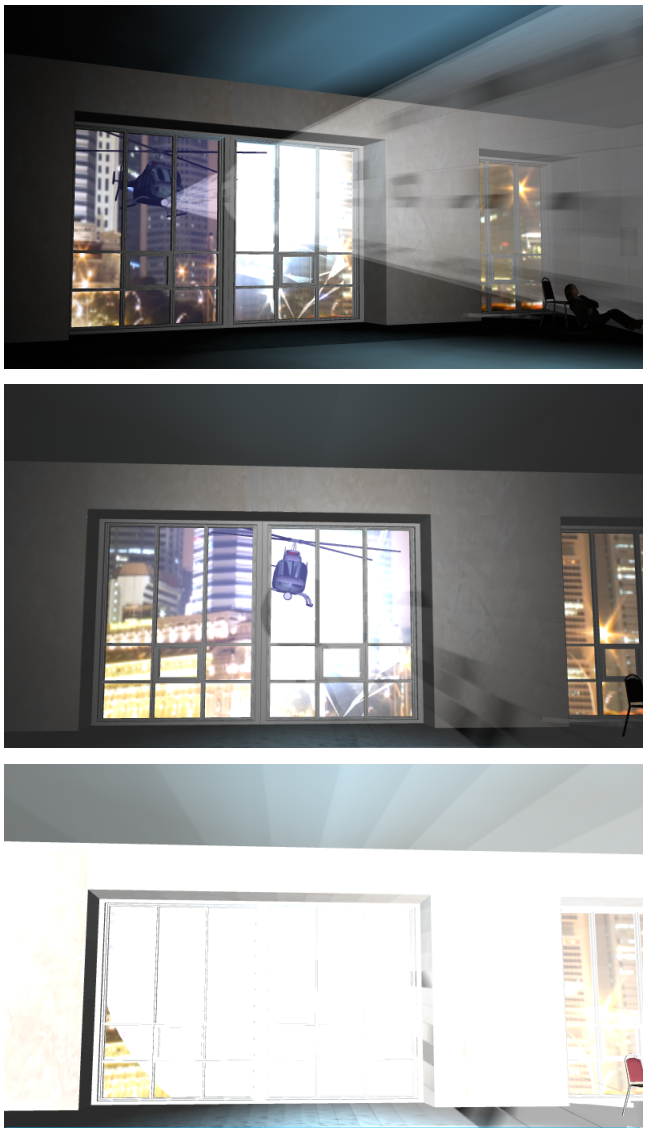


Figure 3: *Rendered examples from the helicopter scene. When the light points away from camera, lighting with standard OpenGL spotlight produces acceptable result (top). When the light points directly into the camera, standard lighting looks unnaturally flat (middle). Dynamic light amplification creates the desired blinding effect (bottom).*

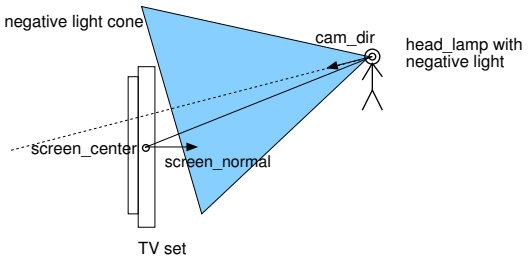
This scene was designed for training first responders in hostile environments [Vincent et al. 2008]. The helicopter object with a blinding light was modeled and programmed to add a sense of drama to training scenarios.

3.4 Example 2: Luminous Objects and Negative Lights

In addition to conventional light sources, the scene illumination may be affected by presence of various luminous objects. Examples include: a window with bright daylight outdoors, a computer or television screen. We implemented this extension by adding the $update()$ method to their data structure and registering each such object twice in the scene graph: as a drawable entity and as a light source. This double role of luminous objects allows them to participate in lighting calculations. Implementations of the $update()$ method are object-specific. The object must evaluate its own importance for the current viewing conditions and make appropriate adjustments to scene illumination.

An interior scene with a television set, illustrated in Figures 4 and 5, provide an example how luminous objects are processed. The TV screen is textured with a very bright image of a desert, with the sun in direct view. The texture is applied in decal mode, making the TV screen exempt from lighting calculations. In order to make the TV screen look brighter, a negative light is applied, that darkens all other objects in view. The negative light was implemented as an OpenGL spotlight, parented to the user head object and oriented in the same direction. Its intensity is controlled by TV screen's $update()$ function, listed in Figure 4. Note that the round window in Figure 5, top, is also a luminous object and can be processed in the same manner as the TV screen. For that purpose, the negative light is shared by all luminous object, collecting their contributions, Figure 4, line 5.

Other types of luminous objects may require more accurate evaluation of their contribution to scene lighting. Objects that are very large or are extremely bright must calculate their exact pixel footprint on the viewing frame. This can be done by a number of available techniques, such as [Sekulic 2004] and [Bittner et al. 2004].



0: $head\_lamp.intensity \leftarrow 0$
...
1: **procedure** TV_Update(Camera cam, Light $head\_lamp$)
2: **if** TV object is visible **then**
3: $A \leftarrow -screen.normal \bullet cam.direction$
4: $D \leftarrow (screen.center - cam.pos) \bullet cam.direction$
5: $head\_lamp.intensity += c_1 * A * D$
6: **end if**
7: **end procedure**

Figure 4: *Enhancing perceived brightness of a TV screen object, using a negative spotlight, attached to camera. The spotlight has maximal intensity c_1 = 5, when the TV screen is centered in user view ($D = 1$) and oriented towards the camera ($A = 1$). Rendered images are displayed in Figure 5.*

3.5 Notes on View Dependent Level of Detail

In this work, we explored only the lighting part of the rendering process. Shape representation is another very important task that can be optimized by adding view-dependent controls. Objects that fall within the current line of sight of the viewer, should be displayed at the maximal geometrical resolution, or level of detail (LOD). Objects in the peripheral area may be represented with lower LOD models. Dynamic control over LOD is widely used in real-time graphics applications, first and foremost in video games, because it allows to increase visual complexity of the scene without compromising rendering speed. The most commonly used methods of

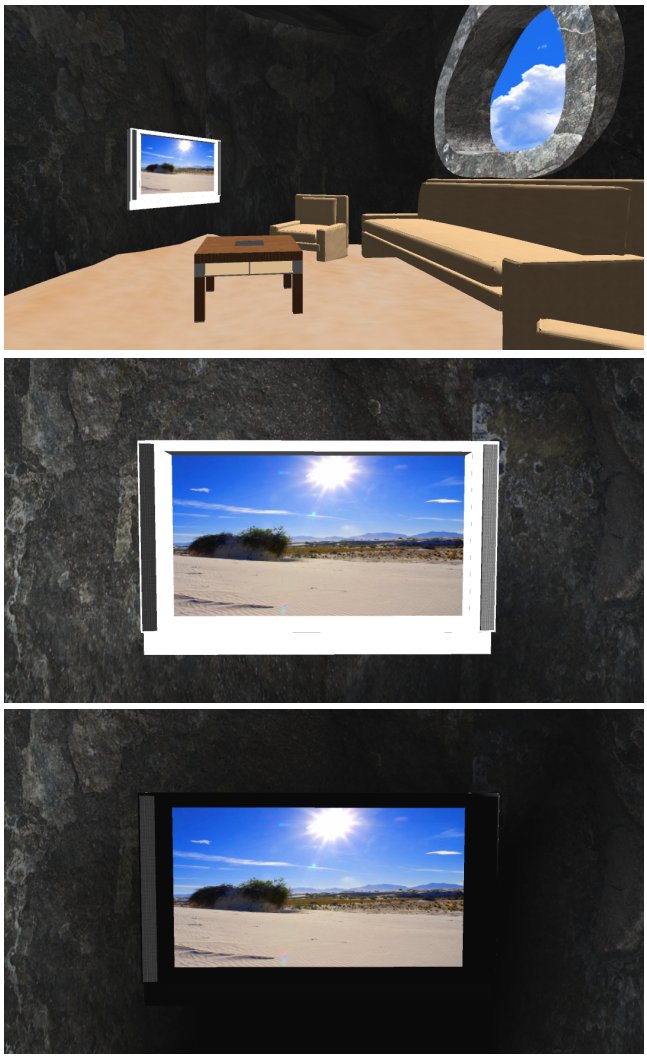


Figure 5: *A scene with a wall-mounted television set, with its screen textured in decal mode. Top: general room layout. Middle: close-up view of the TV set under normal illumination. Bottom: a negative light is applied, which darkens the wall and the TV frame and speakers. The TV screen, unaffected by this new light, appears now brighter.*

selecting LOD models are based on distance from the camera or the pixel footprint of the object on the viewing plane [Akenine-Möller et al. 2008]. At the time of this writing, view-based methods for selecting LODs have not yet been used in immersive VR systems, although the potential of this approach was clearly demonstrated by Ohshima et al. [1996].

Interestingly, the idea of variable spatial resolution for immersive VR rendering was recently implemented in hardware. The near-panoramic Head Mount Display Wide5 (150° FOV), has two actively updated viewports for each eye, built as a display-in-display. The wider viewport that covers the whole scene, is implemented on a larger display, at 800 by 600 pixels. The smaller central area of the same viewport is rendered on an enclosed display, which has the same number of pixels 800 x 600. This layered design ensures that the objects along the current line of sight are displayed in more detail. The results of pilots studies are presented in [Bolas et al. 2007].

4 Conclusions

Our contribution can be summarized as follows: we extended the basic OpenGL-style lighting process by taking the user position and gaze direction into account. This extension allowed to create various lighting effects and improve the overall visual response from Head Mounted displays, in immersive VR settings.

The view-dependent lighting control, presented in this paper, is not intended to simulate nor account for processes that happen in the human eye during adaptation to light. These mechanisms are rather complex. They include fast optical adjustments (pupil dilations and contractions, which happen in seconds) and slow chemical processes that happen in the retina and take minutes. There is also a large hysteresis with respect to the direction of changes in light intensity.

Instead, we used a more pragmatical approach, based on the similarity between the narrow field of view, typical for common HMDs, and the small size of high-acuity vision area in a human eye. From that perspective, many problems of improving lighting for HMD rendering have simple geometric solutions, that can be easily programmed for most VR systems and applications.

The described system was tested and proved useful for two HMDs, V8 (60° FOV) and 5DT-800 (40° FOV). We believe that it will work for most consumer-grade HMDs, with FOV up to 60 degrees. For panoramic and near-panoramic displays, the assumption of equality between the gaze direction and head direction will fail, and true eye tracking will be required. However, it seems unlikely that panoramic HMDs will dominate the market of wearable displays, at least, not in the nearest future. On the contrary, low and medium grade displays remain in high demand. The long-standing V8 model by Virtual Research was recently re-engineered from NTSC to full 1280 x 1024 resolution. The popular low-cost Z800 HMD by eMagine is now distributed by several companies, in ruggedized form, with better shape factor and improved peripherals. The optics remained the same, though, in both cases: 60° and 40° diagonal field of view, respectively. That gives reasons to believe that the view-dependent light-control mechanism, described in this work, will remain valid and useful for some considerable time.

5 Acknowledgment

Many thanks to Kin Lik Wang for helping with the preparation of video materials for Example 1, and Pavel Senin for offering his photographic talents in Example 2.

References

AKENINE-MÖLLER, T., HAINES, E., AND HOFFMAN, N. 2008. *Real-Time Rendering 3rd Edition*. A. K. Peters, Ltd., Natick, MA, USA.

BITTNER, J., WIMMER, M., PIRINGER, H., AND PURGATHOFER, W. 2004. Coherent hierarchical culling: Hardware occlusion queries made useful. *Computer Graphics Forum 23*, 3 (Sept.), 615–624. Proceedings EUROGRAPHICS 2004.

BOLAS, M., PAIR, J., HAYNES, K., AND MCDOWALL, I. 2007. Display research at the University of Southern California. In *IEEE Emerging Displays Workshop*.

BUNGERT, C. 2006. *HMD/headset/VR-helmet Comparison Chart*. http://www.stereo3d.com/hmd.htm, last updated on Dec. 27, 2006.

DEVLIN, K., CHALMERS, A., WILKIE, A., AND PURGATHOFER, W. 2002. Star: Tone reproduction and physically based spectral rendering. In *State of the Art Reports, Eurographics 2002*, The Eurographics Association, D. Fellner and R. Scopignio, Eds., 101–123.

FERWERDA, J. A., PATTANAIK, S. N., SHIRLEY, P., AND GREENBERG, D. P. 1996. A model of visual adaptation for realistic image synthesis. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, 249–258.

FLATLAND. 2002. *The Homunculus Project at the Albuquerque High Performance Computing Center (AHPCC)*. http://www.hpc.unm.edu/homunculus.

KING, Y. 2000. 2d lens flare. In *Game Programming Gems*, Charles River Media Inc, 515–518.

LARSON, G. W., AND SHAKESPEARE, R. 1998. *Rendering with radiance: the art and science of lighting visualization*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

OHSHIMA, T., YAMAMOTO, H., AND TAMURA, H. 1996. Gaze-directed adaptive rendering for interacting with virtual space. In *VRAIS '96: Proceedings of the 1996 Virtual Reality Annual International Symposium (VRAIS 96)*, IEEE Computer Society, Washington, DC, USA, 103.

SEKULIC, D. 2004. Efficient occlusion culling. In *GPU Gems*, Addison-Wesley, 487–503.

SHERSTYUK, A., AND TRESKUNOV, A. 2009. Dynamic light amplification for head mounted displays. In *VRST '09: Proceedings of the 16th ACM Symposium on Virtual Reality Software and Technology (VRST 2009)*, ACM, Kyoto, Japan, 103.

SPENCER, G., SHIRLEY, P., ZIMMERMAN, K., AND GREENBERG, D. P. 1995. Physically-based glare effects for digital images. *Computer Graphics 29*, Annual Conference Series, 325–334.

VINCENT, D., SHERSTYUK, A., BURGESS, L., AND CONNOLLY, K. 2008. Teaching mass casualty triage skills using immersive three-dimensional Virtual Reality. *Academic Emergency Medicine 15*, 11, 1160–5.

WILLIAMSON, S. J., AND CUMMINS, H. Z. 1983. *Light and color in Nature and Art*. John Wiley and Sons, New York, NY.

Figure 6: *"Blinded by the virtual light". A helicopter with a bright projector lamp is orbiting an office building, "looking" into windows. When the beam light hits the viewer's eye, the light intensity is magnified, producing a blinding effect (bottom image). For details, see Section 3.2 in the text.*

View publication stats