PHOTOREALISTIC PLACEMENT OF A MEDIATED REALITY OBJECT USING LIGHTSPACE COMPARAMETRICS

Mohit Kansal, Tae-Hyung Kwon, Billal Belmellat, Corey Manders, Steve Mann

University of Toronto
Department of Electrical and Computer Engineering
10 King's College Rd., M5S 3F5, Toronto ON, CANADA

ABSTRACT

This paper explains a new method used to realistically place a 3D object into an everyday scene. Using powerful tools such as OpenVIDIA [1] and Videorbits the reality mediating engine described robustly places a character into one's video image, tracks the image (using only sequential video frames) and transforms the object accordingly. A new lighting concept utilizing lightspace comparametrics yields information that realistically globally illuminates the object placed in the scene.

Past papers attempting to mimic lighting conditions in a room are not nearly robust enough to be utilized in any sort of real world applications. The lighting algorithm discussed is a new and very robust technique that can easily be implemented on devices such as the EyeTap [2].

1. INTRODUCTION

Creating a true mediated reality based object has been a goal of many researchers for quite some time. The advent of the Augmented Reality Toolkit [3] was a big step towards achieving this goal. However, the AR toolkit focused on image transformation and alignment on marked planes, leaving out photorealism and natural occuring planes. For there to be any possibility of creating a real-world mediated reality application work must be done to increase the photorealism of objects and to use natural occurring planes in our everyday environment.

There have been several schools of thought relating to photorealism both in the detection and implementation of lighting characteristics of a scene. However, time and time again the complexity and calibration problems with past algorithms hinder any sort of realistic real-time applications.

Using radiosity to compute global and direct illumination has been explored by people such as [4], though results are positive, proper implemention requires many characteristics of the scene must be known prior to rendering. Characteristics such as reflectance are not practical to obtain in a real-time dynamic scene.

The work of Sato et. al [5] computed the radiance of a scene by using an omni-directional stereo algorithm based on images with varying shutter speed. Though effective, the complicated nature of the algorithm makes it impractical for real-time everyday use.

Besides complexity issues, an intrinsic problem in previous research is the lack of consideration of the camera response function. As will be discussed later, many are unaware of the hidden response function inherent in all cameras and hence without its consideration true photorealism cannot be achieved.

We will now describe the process of how the object is properly placed in the scene. The first step is the actual recognition of the plane in question. The second involves the detection of the amount of ambient light present in the current situation. And the third, the placement of the object with the appropriate texture modification and position transformation.

However, before we can describe the actual process we must briefly explain the key concepts used, namely OpenVIDIA (the framework in which the work is presented), Videorbits (the tracking algorithm used) and Lightspace Comparametrics (the fundamental basis in which lighting is detected and implemented on the object).

1.1. OpenVIDIA

The name OpenVIDIA derived from its open source and implementation on nVIDIA graphics hardware utilizes both OpenGL and Cg Fragment Shaders. This is done to take full advantage of very advanced modern day graphical processing units or GPUs. Discussed in detail by the creator James Fung [1], GPUs bode an incredible number of transistors when compared to normal CPUs while lacking the need for cache, hence, making GPUs very effective at performing matrix operations. For example, the NV35 GPU onboard the GeForce FX5900 contains a full 30 million more transistors than an Xeon CPU. The idea behind the OpenVIDIA project is to exploit this computing power to speed up computer vision applications while freeing up the main CPU for other use. In general, OpenVIDIA provides a framework for low-cost, fast computer vision processing, which is a large step towards making developed applications more easily acceptable to a larger computing audience. The application discussed is fully implemented in this framework and special care is taken to minimize computing power in general, a more detailed discussion will take place in areas where this is a concern.

1.2. Videorbits

Videorbits registers sequential video frames from a non-dynamic scene and calculates the related co-ordinate transformation. The camera that captures the video is free to pan, tilt, rotate about its optical axis and zoom. The algorithm is very useful for many mediated reality applications and is taken advantage of in the current application under discussion.

In our case, the passing of sequential frames from our camera movement yields us the appropriate co-ordinate transformation of the four corner points of the plane of interest.

1.3. Lightspace Comparametrics

1.3.1. Image Range Compression

Most cameras do not provide an output that varies linearly with light input. Instead, most cameras contain a unique non-linear dynamic range compressor, as illustrated in Fig. 1 which varies widely in its response function according to the particular camera system.

1.3.2. Range Compression Today

When range compressors were built into video cameras for the purpose of capturing data to be reproduced remotely, the display devices were all televisions with largely the same response to a video signal. Today, one may capture images with no notion of what the image may be displayed on. The archived images may be displayed on analog or digital televisions, video projectors, EyeTap devices, print, volumetric displays, etc., just to name a few of the current possibilities. In the future, the number of options will certainly grow. Most importantly, the range expander in each of these devices will most likely vary. This means that to accurately display the image, careful calibration is needed. In many cases, the calibration may be useless. This is simply because the range compression used by particular camera companies is proprietary, prohibiting accurate representation by arbitrary display devices. In essence, range compression results in a large probability of the compressor and expander not matching, resulting in improper representation of the subject matter.

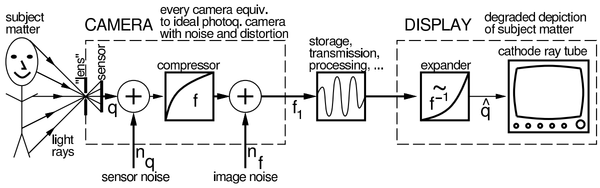

Fig. 1. **Typical camera and display :** Light from subject matter passes through lens (typically approximated by simple algebraic projective geometry, e.g. an idealized "pinhole") and is quantified in units "q" by a sensor array where noise n_q is also added, to produce an output which is compressed in dynamic range by a typically unknown function f. Further noise n_f is introduced by the camera electronics, including quantization noise if the camera is a digital camera and compression noise if the camera produces a compressed output such as a jpeg image, giving rise to an output image $f_1(x, y)$. The apparatus that converts light rays into $f_1(x, y)$ is labelled CAMERA. The image f_1 is transmitted or recorded and played back into a DISPLAY system where the dynamic range is expanded again. Most cathode ray tubes exhibit a nonlinear response to voltage, and this nonlinear response is the expander. The block labelled "expander" is therefore not usually a separate device. Typical print media also exhibit a nonlinear response that embodies an implicit "expander".

1.3.3. Comparametrics

Lightspace is a mathematical framework that describes a model of the way in which light is represented by a camera. In this instance we are using lightspace to model the way light interacts with a scene or object [2].

Understanding that there is range compression present when an image is represented by pixels (say in ppm format or jpeg format), the values of the image will not yield an accurate representation of light unless the pixels are converted to lightspace values. To do this, there are various methods which are beyond the scope of this paper. However, options which exist to solve for this compression function are comparametric equations, covered in [6], solving for splines or piecewise linear parts, covered in [7][8], or superposimetric methods, covered in [9][10].

Using this theoretical framework there exists an intuitive method to naturally darken or lighten the image. For instance, given a pair of images taken with a camera with a known response function, the relative gain between images is estimated and either of the pair is lightened or darkened to bring it into the same exposure as the other image. Similarly, any computer-generated information in a mediated or augmented scene is brought into the appropriate exposure of the scene. A more precise overview will be discussed in the following section.

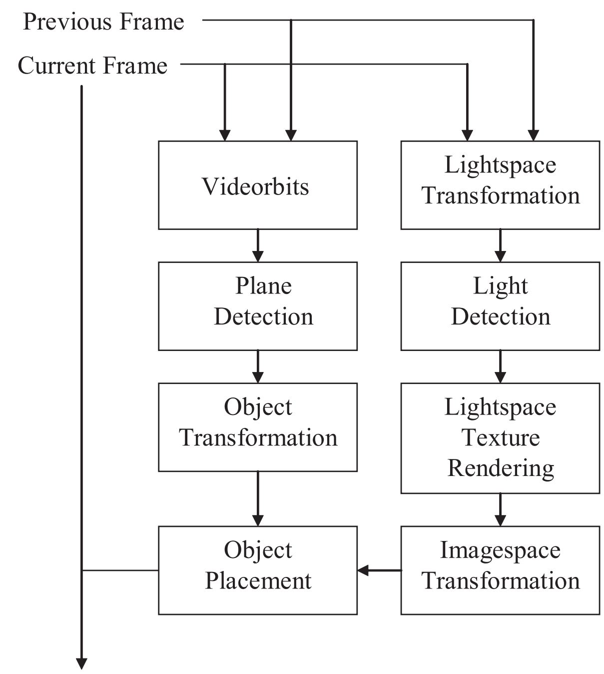

Fig. 2. **Block Diagram of Process :** The ongoing process shown in the diagram depends solely on the current and previous video frames. If we follow the first column we can see how the frames are passed through orbits to renew plane co-ordinates after of which applying the plane detection technique will yield us the new object transformation. The second column highlights the lighting factors in the scene. After both frames are transformed into lightspace, light changes are found and then applied to the objects texture. Finally, the object texture is transformed back into imagespace and placed on the screen along with the current frame.

2. OVERVIEW OF ALGORITHM

Shown in Figure 2 is a block diagram summarizing the overall process used. A full discussion of the key steps will now take place.

2.1. Plane Recongnition

The initial plane recognition is determined from the user's input. By clicking on the four corners of the plane of interest and ensuring that our camera's distortion has been corrected we can determine the plane's normal vector. From the components of the normal the transformation of the object can be accomplished with ease.

2.2. Light Detection

The detection of light is done by comparing sequential frames. After the previous and current frames have been transformed into lightspace, a pixel by pixel comparison takes place. Comparing each R, G and B value between frames, namely the amount of light falling on the sensor in question, as is shown in Figure 3, the gain of light on the sensors is found. By taking a cumulative sum over all sensors an average ambient light gain is found. This difference in lighting is applied to the texture of the object being placed.

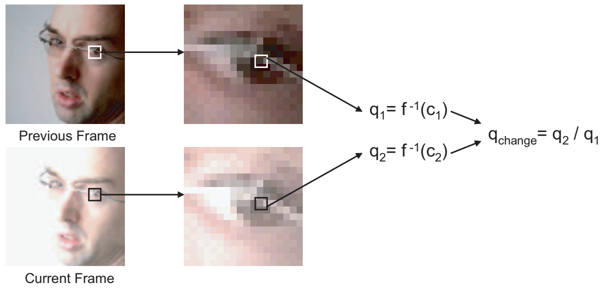

Fig. 3. **Light Detection Process :** This simple diagram depicts the light detection process. Following a lightspace transformation, a comparison between corresponding pixels shows us the light gain between the frames at that particular pixel.

2.3. Object Transformation

After the appropriate light and plane attributes have been determined, the data is implemented on the OpenGL object [11]. The plane attributes obtained through Videorbits computations are easily fed into a position transformation function implicit in the OpenGL framework, however in our case the lighting functions within OpenGL are not used. More accurately, a newly created lightspace filter was used to ensure that the lighting on the character was done properly. By transforming the textures of the object into lightspace and then multiplying by the appropriate light gain factor, the character should update its texture appropriately with any ambient light changes.

3. RESULTS AND FUTURE WORK

Example screenshots are shown in Figure 4. After the plane of interest is determined by the user, the placement of the object is aligned with the normal of this plane very well. Lighting changes also seem to follow corresponding changes in the light background. However, due to the continuous reliance on sequential frames it was found that one unusual frame, perhaps caused by a minor obstruction, would throw

Fig. 4. **Resulting Screenshots :** This example shows two screenshots of an animated skeleton atop a building on the University of Toronto campus. By simply aiming the camera out of a window and allowing the user to indicate what plane they are interested in, the skeleton is put into place and reactive to any camera movement or ambient light change. The first of the images is under a high quanitity of light input and the second under little light input. For our testing purposes, the lighting is controlled by modifying gain or exposure on the camera independent of the algorithm applied.

off lighting on the character indefinetly, in which case the lighting characteristics had to be reset.

The next step in the improvement process is to filter out unusual frames so that the lighting gain factor stays consistent with overall ambient lighting. Also, work towards more complicated light detection mainly directional light needs to be investigated in conjunction with ambient light to improve realism of the placed object. Current work is focused on improving the issues discussed and implementing the algorithm on current EyeTap prototypes.

4. REFERENCES

[1] James Fung and Steve Mann, "Computer vision signal processing on graphical processing units," *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, May*, pp. V93–V96, 2004.

[2] Steve Mann, *Intelligent Image Processing*, John Wiley and Sons, November 2001, ISBN: 0-471-40637-6.

[3] H.Kato, M. Billinghurst, I. Poupyrev, K. Imamoto, K. Tachibana, "Virtual object manipulation on a table-top ar environment," 2000.

[4] George Drettakis, Luc Robert, and Sylvain Bougnoux, "Interactive common illumination for computer augmented reality," in *Proceedings of the Eurographics Workshop on Rendering Techniques 97*. 1997, pp. 45–56, Springer-Verlag.

[5] Imari Sato, Yoichi Sato, and Katsushi Ikeuchi, "Acquiring a radiance distribution to superimpose virtual objects onto a real scene," pp. 137–160, 2001.

[6] S. Mann, "Comparametric equations with practical applications in quantigraphic image processing," *IEEE Trans. Image Proc.*, vol. 9, no. 8, pp. 1389–1406, August 2000, ISSN 1057-7149.

[7] F. M. Candocia, "A least squares approach for the joint domain and range registration of images," *IEEE ICASSP*, vol. IV, pp. 3237–3240, May 13-17 2002, avail. at http://iul.eng.fiu.edu/candocia/Publications/Publications.htm.

[8] A. Barros and F. M. Candocia, "Image registration in range using a constrained piecewise linear model," *IEEE ICASSP*, vol. IV, pp. 3345–3348, May 13-17 2002, avail. at http://iul.eng.fiu.edu/candocia/Publications/Publications.htm.

[9] C. Manders, C. Aimone, and S. Mann, "Camera response recovery from different illuminations of identical subject matter," in *Proceedings of the IEEE International Conference on Image Processing*, Singapore, Oct. 24-27 2004, pp. 2965–2968.

[10] C. Manders and S. Mann, "Determining camera response functions from comparagrams of images with their raw datafile counterparts," in *Proceedings of the IEEE first International Symposium on Intelligent Multimedia*, Hong Kong, Oct. 20-22 2004, pp. 418–421.

[11] "Cal3d character animation library, http://cal3d.sourceforge.net/," .