Creative relighting in compositing based on z depth

Johannes Saam[*]
National center of computer animation Bournemouth University

Abstract

The faster turnaround times in modern vfx productions force companies and artist to come up with ideas to speed up every part of the pipeline and to be able to change the final image fast. One of the most time consuming steps is the rendering step. Once the parameters are set rendering takes place for a long time. Should lighting and surface properties change re-rendering is veryexpensive and time comsuming. Interactive relighting as a part in compositing is very helpful to archive faster turnarounds. Problems occur because additional data is required to archive visually pleasing results. Antialiasing poses also a problem because image data based on physical properties of the scene and camera cannot be anti-aliased due to resulting in wrong results. The technique we present in this paper is completely based on the depth information of the scene. Depth layers are never anti-aliased and always available because this information is needed for several things like depth of field simulation. We can extract surface normal and point position from a z-Depth and relight the scene with that. More complex effects like refractions can be simulated to a certain extent. Another use for this technique is that depth layers can be extracted with the help of stereoscopic imagining and or optical flow from original footage. Our technique helps to relight even shot live footage.

Keywords: relighting, compositing techniques, image proccessing

1 Introduction

As compositing is the last step in a vfx production it is the closest step to the final image. Retouching can be needed, and final decisions are made in the last moment. These changes could affect the whole pipeline and be very costly. Beeing able to change the final image here without any revisiting of the graphics pipeline makes production cost effective and faster. A single artist that can change or add lighting information at the end of a pipeline is very powerful to enhance the production pipeline. As common relighting engines rely on rendering of different passes, being able to relight is not always the case. A system that builds on the already existing image information has therefore eligibility. The Z depth provides enough information to get the job done and should be available on almost every 3d rendering.

2 Related work

Relighting solutions till now rely on multiple data inputs. Different kinds of approaches exist to pre-compute relevant data. The first attempt to relight 3D scenes uses a deep framebuffer to write several passes with scene data. This data can be formed from user set attributes or gathered information from rendering. [Gershbein 2000] presents a fast way to design lighting for cinematic purposes. Pixar takes that a step further with presenting Lpics. [Pellacini 2005] presents a tool where lighting TD‘s can interactively pre-light scenes with instant feedback. These techniques pre-light the scene and help to design the shot. We focus on the change of lighting in vfx productions after the design and rendering phase. An interesting way to collect data from film footage is presented in [Nishio 2004]. Cheslack-Postava presents a method to extract environment illumination data from the human eye. This technique could be combined with ours to match 3d geometry to actors in the scene. [Kautz 2005] introduces the first time an approach with spherical harmonics for low-frequency relighting. [Cheslack-Postava 2007] and [Ng 2004] improved that technique with high-resolution light transport matrices and triple wavelets. Global illumination solutions for real time use these techniques efficiently to pre-compute radiance transfer. [Haan 2006] ported this idea to cinematic prelighting. All these ways of gathering data are very complex and can not be applied on the fly in a production and need the pre computation of scenes to be able to work. Our system is designed to be quick and always aviable to the artist. After the rendering process took place.

3 Calculating the positon data

To be able build up a shading model from the z depth information we need several different essential values. Lighting is based on the position of the shading sample, its surface normal and several light properties. In a first step we compute the position of the currently shaded pixel in camera space and code the resulting vector in RGB channels, where r = x, g = y and b = z. The z position is already known. The z depth equals the z position in camera space. To compute the x and y position accurately we need the cameras field of view as follows:

$$P_x = sin(\frac{fov}{2}) - \frac{x \times sin(\frac{fov}{2})}{width} \tag{1}$$

[*]e-mail: jo(at)johannessaam.com

$$P_x = sin(\frac{fov}{2}) \times ar - \frac{y \times sin(\frac{fov}{2}) \times ar}{height} \quad (2)$$

Where P is the position, fov is the field of view, ar is the images aspect ratio, width is the image width and height is the image height. P is referred form now on as the position of the shaded point.

4 Calculating the normal data

The normal data is only based on the position of the rendered pixel in camera space. First the image gradients in x and y are computed for every channel separately. The resulting vectors represent the surface slope in x and y. The cross product of these vectors is the resulting surface normal. The image gradients are computed efficiently by subtracting the image value at x+1 from the current x value. Y is done accordingly.

$$V_x = I_{x+1} - I_x \quad (3)$$

$$V_y = I_{y+1} - I_y \quad (4)$$

$$N = V_x \times V_y \quad (5)$$

V_x and V_y are the slope vectors. N is referred form now on as the surface Normal of the shaded point. Now all the relevant data for relighting is extracted from a single depth map.

5 Relighting

To relight the image the collected data is used to fake 3D lighting. For every pixel the positional and normal data is fed into a lighting system that calculates a light vector L. Three different light types are chosen, Point, directional and spot lights. The L vectors are calculated as follows:

Point Light:

$$L_{xyz} = P_{light} - P \quad (6)$$

Directional Light:

$$L_x = sin(\theta) \times cos(\phi) \quad (7)$$

$$L_y = sin(\phi) \quad (8)$$

$$L_z = cos(\phi) \times cos(\theta) \quad (9)$$

Where ϕ and θ are the light angles in spherical coordinates.

Spot Light:

$$L_{xyz} = P_{light} - P \quad (10)$$

With a cone angle and a cone delta angle to define the typical spot light cone and a smooth falloff. Lambertian diffuse shading is then calculated by dotting L with N. Specular highlights can be added according to Phongs shading model.

$$spec = (I.R)^{roughness} \quad (11)$$

Where R is the reflection vector from I pass N and roughness is the specular roughness according to Phongs lighting model. To achieve realistic lighting all the lights can be attenuated with the length of L which equals the lights distance to P, and tinted with any color.

6 Advanced Shading

To maximize the visual impact of the result more shading phenomena can be simulated. By converting any vector to uv coordinates on a infinite environment sphere a simple texture lookup can be done to simulate reflections and refractions.

$$S = \frac{acos(y)}{\pi} \quad (12)$$

$$T = \frac{\pi + atan2(z,x)}{2\pi} \quad (13)$$

Where s and t are the uv coordinates on the environment sphere. These coordinates can be transferred to the environment map image space by:

$$I(x) = s \times width \quad (14)$$

$$I(y) = t \times height \quad (15)$$

A lookup in the environment based on the reflection vector from the incident vector and the normal returns reflections. The same in the direction of the refracted vector of these components provides refractions. Reflection and refraction combined with Fresnel‘s law results in realistic glass shading. A lookup based on the surface normal gathers environment lighting. If a diffuse convoluted image is chosen realistic environment lighting is archived. Bent normals can be brought into the system. Bent normals are the surface normal bent to the average of the unconcluded samples on a hemisphere around N. Environment lookups in the direction of the bent normals results in an environment lighting with fake shadows. Non photorealistic shading can be simulated my reflecting custom painted comic like environment spheres in the direction of N.

7 Future work

A know issue is the camera dependency of the system. A new feature allowing the camera data to be fed into the system would provide camera independency. This is needed for moving cameras. This is rather easy to add to the system. More work has also to be done in the area of shadowing. Calculating accurate shadow is possible by transferring the information of P into a point cloud. Based on this point cloud ray tracing will be possible. Realistic shadows ambient occlusion and sub surface scattering can be done on these point clouds. Right now shadowing can be done by exporting the light data back to a 3D package and rendering it there. Another interesting area which is usually only explored with huge rigs, is the relighting of real live footage. In [Wegner 2005] a method to extract different information from shot footage is described. With our method a two camera setup to extract depth data from footage would be enough to recover all data neccesary ot relight a real person. Reflections and refractions would be the possible, too. Also all the mathematics are manly just vector math which can be ported easy to the grapics card. Doing that a huge speedup would increase the usability of the system.

8 Thanks

The autors like to thank Joao Montenegro, Peter Lewis, Gerard Keating and Malcolm Childs from the NCCA master courses for ideas and testing. Special thanks to Dacydd Morris for modeling the wolf morph used in the results.

9 Results

Here we present some of the images rendered with our system implemented in the foundary's NUKE. Every step is packaged in a node for efficient node based arrangement of the compositiong.

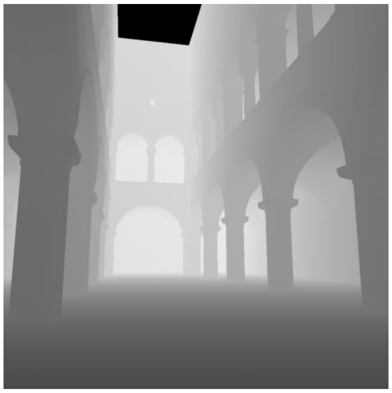

Figure 1: Example of a depth pass on the sponza atrium

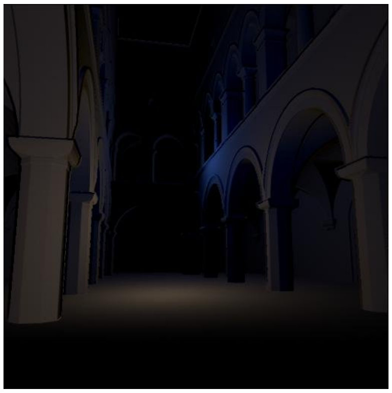

Figure 2: The sponza atrium with two point lights

References

CHESLACK-POSTAVA, E. 2007. 4d compression and relighting with high-resolution light transport matrices.

GERSHBEIN, R. 2000. A fast relighting engine for interactive cinematic lighting design.

HAAN, M. 2006. Direct-to-indirect transfer for cinematic relighting.

KAUTZ, J. 2005. Fast, arbitrary brdf shading for low-frequency lighting using spherical harmonics.

NG, R. 2004. Triple product wavelet integrals for all-frequency relighting.

NISHIO, K. 2004. Eyes for relighting.

PELLACINI, F. 2005. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography.

WEGNER, A. 2005. Performance relighting and reflectance transformation with time-multiplexed illumination.

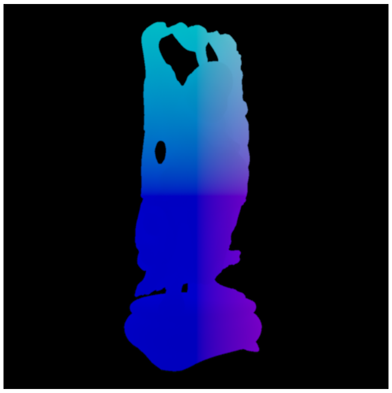

Figure 3: P derived from a z depth on the happy buddha statue

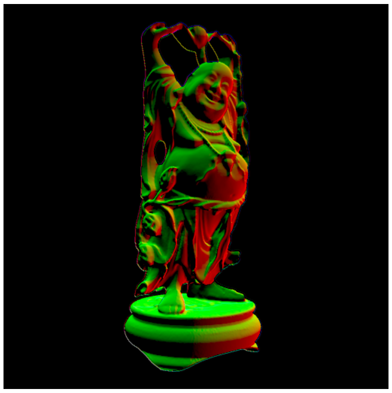

Figure 4: N derived from P

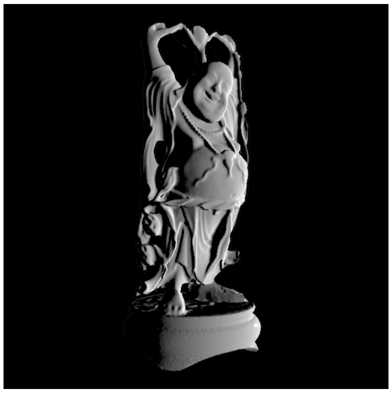

Figure 5: Happy buddha with direct lighting

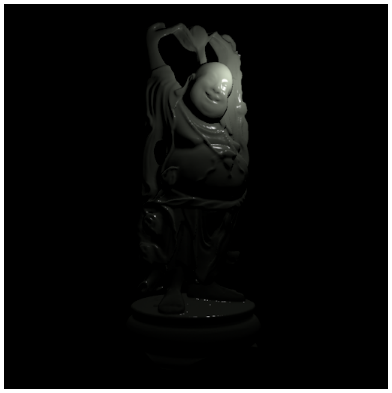

Figure 6: Happy buddha with one point light

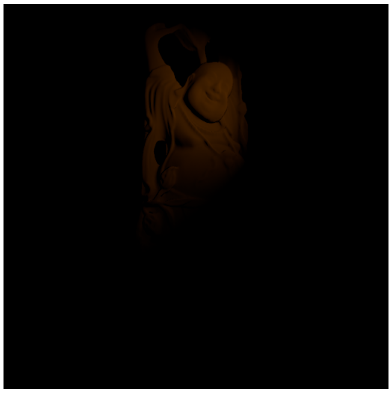

Figure 7: Happy buddha with one spot light

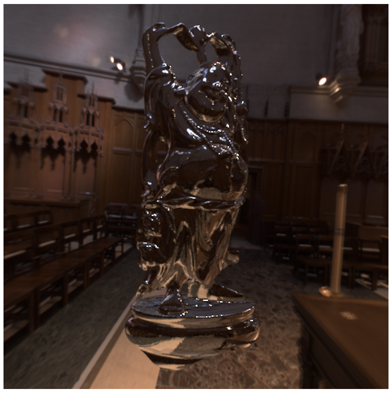

Figure 8: Happy buddha with a glass material in grace cathedral

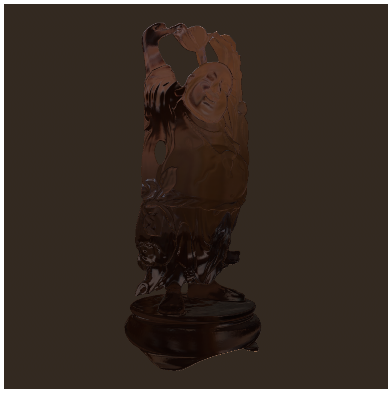

Figure 9: The refraction component of the glas material

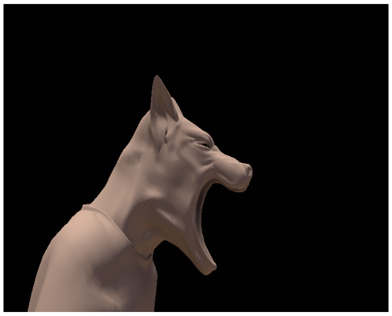

Figure 10: Werewolf model with diffuse convolution lookup

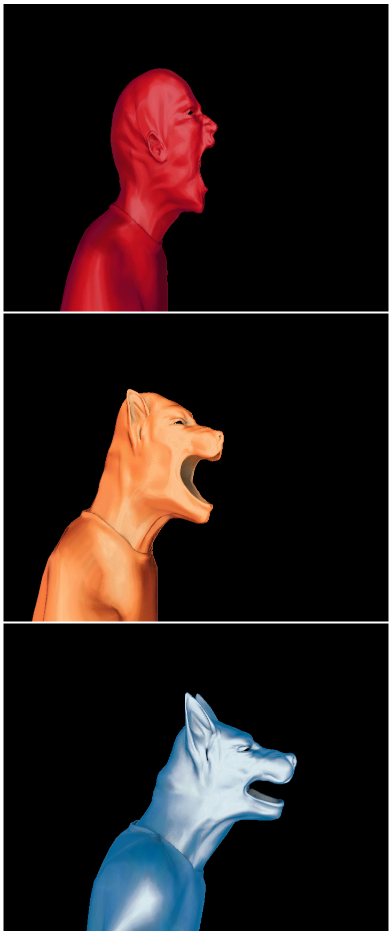

Figure 11: Non photorealistic shading on a werewolf transformation