Shadow Segmentation
from Multiple Light Source Images

Ran Halprin, Efi Arazi School of Computer Science, Interdisciplinary Center Herzliya
Submitted to Prof. Yacov Hel-Or

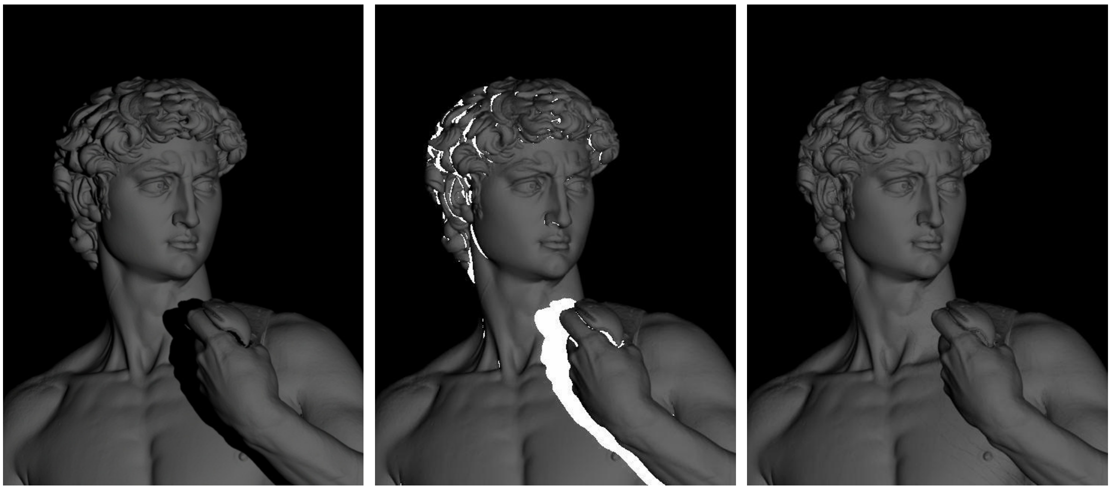

Figure 1: Original image, original image with shadows highlighted, image after reconstruction

Background and problem definition

The topic of Polynomial Texture Maps (PTM) was introduced at Siggraph 2001 Proceedings [Malzbender 01]. In this article, a method is proposed to create relatively realistic texture mapping. This is done by sampling the same 3D object from the same view point, but with several discrete light directions. Using these samples, a continuous two dimensional formula can be calculated to interpolate each pixel in the image with any light direction desired. Combined, an image can be created with any light source direction.

This work has brought about some problems, specifically with regard to cast shadows in some of the images. When we create the general formula for each pixel (see [Malzbender 01] for a full explanation), we assume that each pixel illumination conforms to the simplified illumination formula:

$$I = \overline{N} \cdot \overline{L}$$

Meaning the Illumination level (scalar value) of the pixel is determined by multiplying its Normal vector with the Light vector (vector which points toward the light source). The model is simplified by disregarding various constants not necessary for good interpolation.

The problem with this formula, is that is depends on convex plains. When the plains are concave, it causes some illumination samples not to conform to the formula:

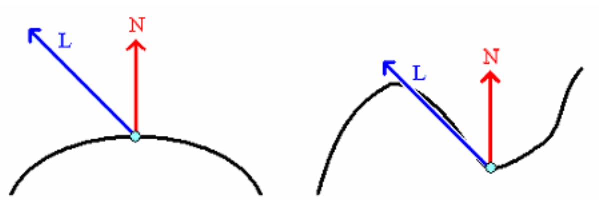

Figure 2: Two points with the same normal and light vectors have different illumination. The right one's light vector is blocked by the "hill" on the left of the plain causing it to be black.

See figure 2. Under the assumption that the light source is a single point, the in the left image the pixel would have a fairly light illumination, but in the right image, the pixel would be pitch black, as the light will be blocked by the "hill".

As the general continuous formula is created, such samples cause erratic results in the interpolation, as sample points do no all conform to the formula that we seek. These off-samples cause various errors in interpolation as seen in figures 3 and 4 (Single dimensional examples estimated for Illustration purposes only). Using higher degree curves causes "ringing" effects and are therefore also not useful.

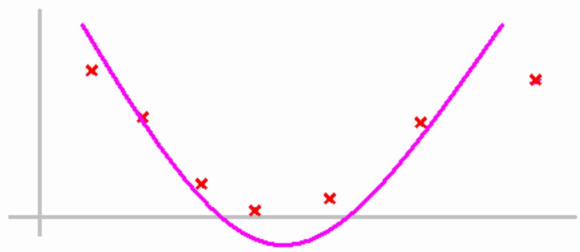

Figure 3: Good samples create good interpolation (2^{nd} deg. curve)

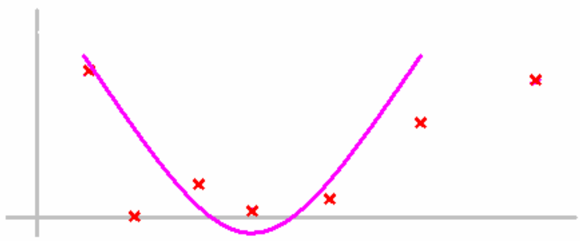

Figure 4: A single shaded sample causes large errors in interpolation

Proposed solution

The general idea – before interpolation, we will try to remove all samples which are shaded. First we remove all images in which this pixel is black (very near 0%) or bright (very near 100%). We then detect problematic images by finding the estimated normal vector for each pixel and removing the images in which the pixel is much different from the estimation produced by using the normal and light vectors.

More specifically - Since we have multiple images, each with a different light source and different pixel illumination, we can widen the formula:

$$\begin{aligned} I_1 &= \bar{N} \cdot \bar{L}_1 \\ &\dots \\ I_n &= \bar{N} \cdot \bar{L}_n \end{aligned}$$

Or in a matrix form:

$$\begin{pmatrix} I_1 \\ \dots \\ I_n \end{pmatrix} = \bar{N} \cdot \begin{pmatrix} \bar{L}_1 \\ \dots \\ \bar{L}_n \end{pmatrix}$$

Thus, we can reverse engineer the vector N by using minimal squares to isolate the vector N. We define:

$$M = \begin{pmatrix} \bar{L}_1 \\ \dots \\ \bar{L}_n \end{pmatrix}$$

And then:

$$\bar{N} \approx \left[(M^T M)^{-1} M^T \begin{pmatrix} I_1 \\ \dots \\ I_n \end{pmatrix} \right]$$

Now that we have a good estimation of the Normal vector for the specific pixel, we can create the estimated pixel illumination for each pixel in every image. Then we can easily find pixels which differ by a certain pre-selected degree from the margin, and exclude them from the interpolation.

Results and conclusions

As seen in Figure 1, this method provides a generally good estimation of cast shadows and allows their easy removal to create a clean image.

Beyond the application of this technique in PTM images, it could be used to create images of objects which naturally contain cast shadows when lighted upon (see Figure 1 for an example), when a single camera position and a single light source at a time are available.

Open Problems

In the theoretical world cast shadows from a single point light source would be easily segmented, as any shaded pixel would be black or very near it. In real images the light source is not a single point, and the image is made of discrete samples of the theoretically continuous original image.

These causes the shadow edges to be less defined, and usually range over more then a single pixel (i.e.: blurred). Usage of lossy compression enhances this symptom.

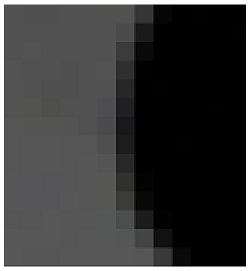

Figure 5: Extreme zoom of a shadow edge in an original image

When attempting to estimate the normal vector for pixels which lay on an edge, they are often not neglected, due to their non-extreme values. Some sort of edge detection and enhancement could solve these problems.

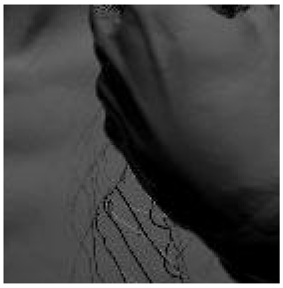

Figure 6: Shadow edges traces in a reconstructed image appear

Another common problem is with pixels that are shaded in most images. When there are very little samples to work with, the interpolation is generally completely off, causing reconstructed images to have various unsuitable values.

Figure 7: Example of a "stray" bright pixel above the eye

A general direction for a solution for this problem would be to use various filters, such as median filters, on the target images. Another possibility is to work on the Normal vector level, as nearing pixels should have similar Normal vectors, and if a vector does not conform, it could be replaced by a mean of its neighbors.

References

[Malzbender 01] – Tom Malzbender, Dan Gelb, Hans Wolters, "Polynomial Texture Maps", available online at: http://www.hpl.hp.com/research/ptm/