Interactive Relighting in Macro Photography

Sean Meador, Alexis Yee Lyn Chan, Andrew Adams and Marc Levoy

Computer Science Department
Stanford University

Abstract

We present an interactive relighting of macro photography using image-based rendering. An object is sequentially lit with a light emitting diode at 25 different vertices of a once-subdivided icosahedral hemisphere and photographed. The virtual lighting condition of the scene is represented by a light hemisphere. We design a user interface that allows the user to create an arbitrary number of light sources at any point on the light hemisphere. The per pixel luminance is calculated using a distance-weighted average of the luminance of similarly located pixels on the image stack. Finally, we present an interface for macro photography illumination with user-specified colors.

I. Introduction

Image-based rendering is the technique of generating novel views of a scene from a set of pre-acquired images of the scene. Using image-based rendering for interactive relighting allows for a simple reconstruction of the scene without requiring complex geometry or bump map modeling. This is suitable for a user-specified relighting interface because the processing time is independent of the scene complexity. The tradeoff of using image-based rendering instead of geometric modeling is that the viewpoint of the scene and the background scene are fixed. We view this as a reasonable tradeoff for the development of a user interface for the interactive relighting of macro photography. Further development of the relighting interface would benefit from user-specified viewpoints and backgrounds.

In this paper we look at related applications of image-based rendering and previous implementations of time-multiplexed illumination using spherical and hemispherical structures. We acquire sets of images lighted sequentially in a hemispherical dome. The dome is fitted with 25 light emitting diodes, one at each vertex of the once-subdivided icosahedron structure that forms the hemisphere, except the top-most vertex which was replaced by the camera (Figure 1). The nodes of the light hemisphere in the graphical user interface are calibrated to match the vertices of the dome. The visual representation of the viewing hemisphere as a light hemisphere gives the user an intuitive approach towards relighting the scene. We chose bilinear interpolation to reconstruct the scene under novel lighting conditions because it has a reasonably good visual result, with faithful recreation of shadows and highlights [Masselus04] and could be developed rapidly within the time constraints that we had for this project.

Figure 1: Left:A once-subdivided icosahedral hemisphere with the top vertex and its adjacent edges replaced with the Canon A640. Right: The dome with matte black paper to reduce reflection from the lighting structure.

II. Related Work

A. Image-Based Rendering Applications

[Masselus04] presented a thorough analysis on reconstruction techniques in image-based relighting rendering. They introduced the multilevel B-Spline technique which produced the most coherent relighting during fast variation of incident illumination. Visual artifacts appear for zero-order hold, biquadratic polynomial fitting and spherical harmonics reconstruction techniques but these methods require very few coefficients which is important for real-time relighting. Linear interpolation, B-Spline and wavelet transform reconstruction produce reasonably good visual result. Other works used image-based rendering to construct opacity hulls for three dimensional photography [Matusik02] and to create non-photorealistic images using depth discontinuity detection [Raskar04].

B. Interactive Relighting

[Wong97] developed an image-based rendering method based on the light field and Lumigraph method that allows interactive control of illumination. By increasing the range of images to different viewpoints and different illuminations, they constructed an apparent bidirectional reflectance distribution function of each surface element and reconstruct the correct user perspective view for varying lighting conditions. [Wong01] introduced a representation of panoramic images that would allow the interactive relighting in addition to allowing the zooming, panning, and tilting of the panorama.

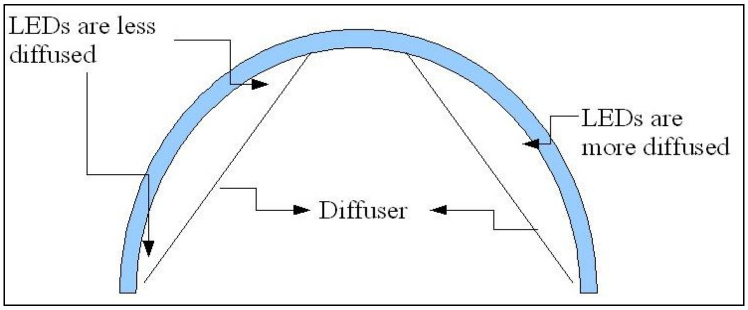

Figure 2: With a single layer of parchment paper as diffuser, the light from LEDs at top and bottom vertices is less diffused compared to the light from LEDs at the middle vertices

C. Once-subdivided Icosahedral Structures

[Malzbender01] used a once-subdivided icosahedral template to acquire photographs of a surface under varying lighting conditions to construct polynomial texture maps. They stored coefficients of a biquadratic polynomial per texel for the reconstruction of surface color under varying lighting conditions. The reconstruction from polynomial texture maps muted sharp specularities and softened hard shadows. However, they presented reflectance transformation operations to compensate for the surface smoothing. Polynomial texture maps can be used with changes in specularity and specular exponent for specular enhancement, addition of a gain factor to exaggerate the diffuse reflectance properties for diffuse gain, and light direction extrapolation.

[Debevec02] used a once-subdivided icosahedral sphere to implement realistic compositing of an actor with a background environment during a live performance. In addition to visible light, they utilized infra-red light to perform a matting of the actor for a realistically lit moving composite of the actor over a novel background. They developed "The Lighting Control Program" which uses a light sphere to visually represent the viewing sphere of a scene. We emulated this approach in our development of the user interface.

[Debevec05] used a similar geodesic hemisphere for the postproduction relighting and reflectance transformation of an actor. We implemented the single light basis and triangle basis investigated in their work and found that for the scale of our dome, the single light basis gives us a higher range of lighting conditions (Figure 5).

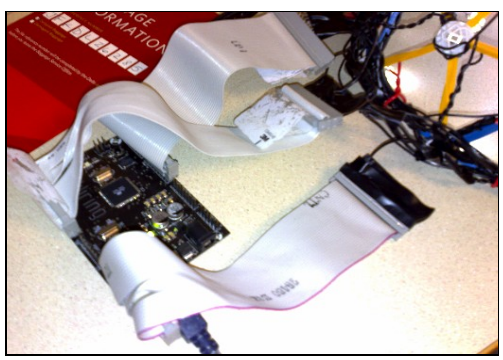

Figure 3: The two negative terminals of each LED are connected in parallel to a 110 Ohm resistor and grounded. The two positive terminals of each LED are connected in parallel to an output pin on the board. The board can be powered using Universal Serial Bus cable connected to a computer or using a regulated power supply.

III. Apparatus

A. The Structure

The lighting structure has 70 edges, 25 vertices and 35 faces. The top vertex of the once-subdivided icosahedron is replaced with the camera. On each of the 25 vertices, an LED is placed facing inwards. The radius of the base of the dome is 13cm and the height of the dome is 12 cm. The structure is constructed using Zome Tools Adventurer Kit, with the blue edges measuring 6cm and the yellow edges measuring 5cm (Figure 1). To reduce reflection from the edges of the structure, the dome is covered with black paper with a matte surface.

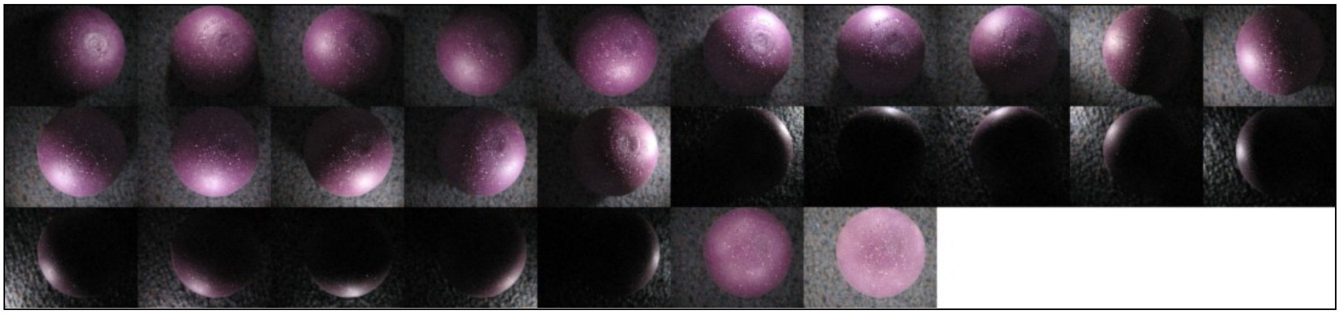

Figure 4: From top left, in order of left to right, top to bottom: The first 5 images are scenes with LEDs at the top vertices lit sequentially. The next 10 images are scenes with LEDs at the middle vertices lit sequentially. The following 10 images are scenes with LEDs at the bottom vertices lit sequentially. The next image is a scene with all top vertices forming a ring around the camera lens lit up. The last image is a scene with all LEDs lit simultaneously.

B. The Lights

The LEDs fitted on the dome are 7.62 mm square, 4-terminal, clear, non-diffused, white LEDs with manufacturer code VLWW9600. Each LED has a 30 degree of half-intensity and total flux of 2.2 lm. A 110 Ohm resistor is connected to both cathode terminals of each LED. To increase the visual quality of linear interpolation of specular objects, we added diffusers by covering the LEDs with parchment paper. A single layer diffuser would diffuse the LEDs at the middle vertices more than the LEDs at the top and bottom vertices because of the hemispherical shape of the dome (Figure 2). More layers of parchment paper are added to the LEDs at the top and bottom vertices to overcome this.

The LEDs are lit one-by-one from the top to bottom vertices. Then the 5 LEDs at the top vertices that form a ring around the camera lens are lit up. After that, all the LEDs are lit up simultaneously (Figure 3). Each lighting condition lasts for 10 seconds. The time-multiplexed illumination is programmed using a microcontroller programmed with the [Wiring] programming environment (Figure 3).

C. The Camera System

A Canon A640 is programmed using [CHDK], a firmware add-on for Canon's compact digital cameras, based on DIGIC I and DIGIC II processors to take capture images at 5 second intervals, with 0 second intervals between shots and a 5 second delay to the first shot. The Basic script is adapted from Keoeeit's [Ultra Intervalometer] to capture images with the “macro” setting turned on.

IV. Basis Selection

Our microcontroller program allows us to turn on and off any set of light source for each frame to produce arbitrary binary patterns. Given a set of linearly independent lighting patterns, one can recover individual lighting directions by inverting a linear system. [Debevec02] compared a single, triangle, and Hadamard lighting basis. Their results favored a triangle basis for a large scale stage because it provides more illumination and hence reduces image noise. Our LEDs have a typical luminous intensity (I_v) of 1760 candelas and we photograph objects that are only a few inches away from the light source. Thus, using a triangle basis caused observable image saturation, loss of detail, and focusing issues (Figure 5). A Hadamard lighting basis requires an even larger number of lights to be simultaneously illuminated. Therefore, for macro photography with our structure we found that using a single lighting basis gives the best results.

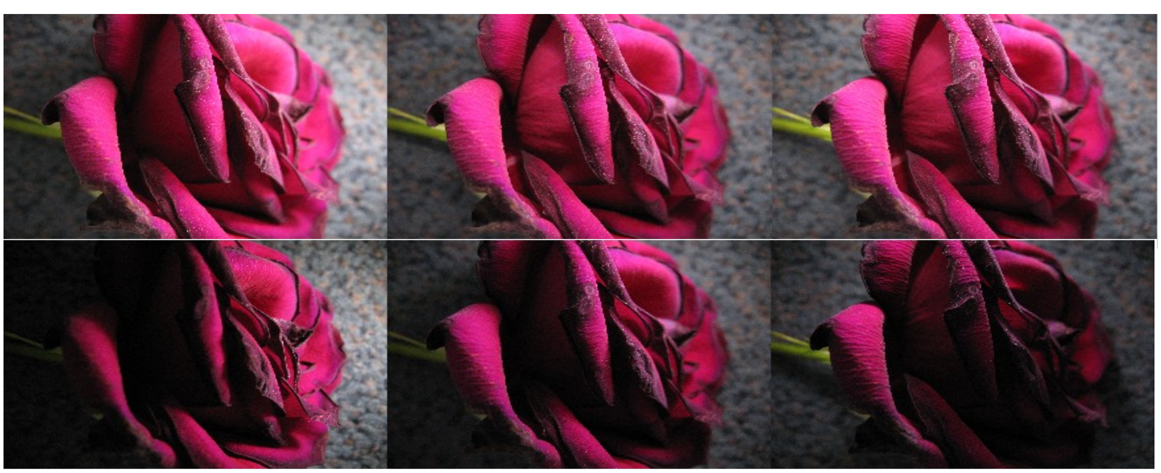

Figure 5: Top Left, Center and Right: Images taken using triangle basis functions. Bottom Left, Center and Right: Images taken using single light basis functions. Using single light basis functions allowed a higher range of image details with less light saturation.

V. Graphical User Interface

We present a novel graphical user interface (GUI) that allows the user to intuitively and easily relight the scene. Our user interface is implemented in C/C++ leveraging off of the Fast Light Toolkit (FLTK). We chose [FLTK] for its speed, image processing convenience, and OpenGL support. The graphical user interfaces has three main components:

1. a “light hemisphere” for changing light direction
2. a properties box for changing light characteristics (color, etc.)
3. an image stack showing the source images and a composite resultant image

A. Light Hemisphere

A natural graphical representation of scene lighting is with a hemisphere. The light hemisphere allows the user to manipulate the direction of illumination in the scene. First, we render a sphere using OpenGL. Lights are positioned a unit distance away from the center of this sphere and oriented in the normal direction to its surface. By clicking and dragging the knobs, the user may reorient this normal direction around the sphere. This is accomplished by projecting the cursor’s window coordinates into object coordinates using gluUnProject. From these object X and Y coordinates, we calculate the Z coordinate by constraining the location on the unit sphere:

$$Z_{obj} = \sqrt{1 - X_{obj}^2 - Y_{obj}^2}$$

We have no lighting from behind the object, which further constrains that Z be positive. Using this stereographic projection the user can easily manipulate the light source. A natural consequence of this is that the disk has a larger radius than the sphere projected onto it. This contributes to a somewhat awkward experience if the user desires side lighting as it requires cursor movement beyond the sphere window. The user may also add up to eight light sources, which matches the typical maximum allowed by OpenGL.

The next step is calibrating between the light hemisphere and the discrete lighting directions provided by our physical dome. To do this, we meticulously measured the position of each vertex in our dome. For convenience we assume a unit diameter for the dome. Our measurements were computed with precision of up to four decimal places. We introduce error in the physical measurement of the pieces and the ratio of large to small parts. This error is hard to quantify, but is on the order of $\pm$ 0.01 length units. We represent these vertices as yellow points in the light hemisphere window (Figure 6). We number each internal node with the same numbering system used for the image vertices.

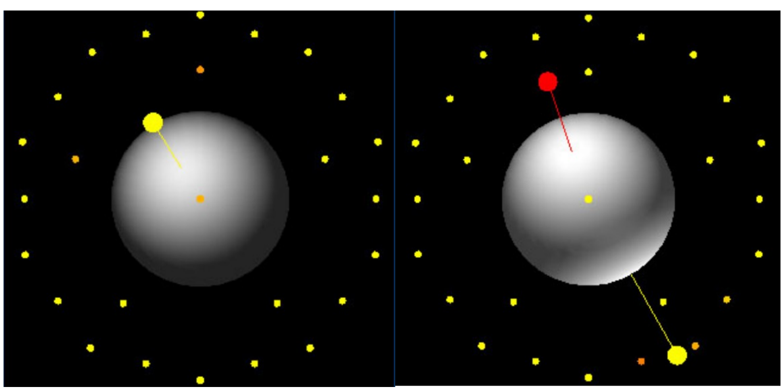

Figure 6: Left: Light hemisphere with 25 nodes and 1 light source. Right: Light hemisphere with 2 light sources.

B. Image Stack

We take 26 basis images with a 10 megapixel camera. Loading all of these images into memory would require upwards of 260 MB of memory. Also, real time relighting would be impossible as this would require iteration through over 10 million pixels per change in illumination. On average machines this is prohibitively slow. Therefore, images are downsampled before being loaded into memory using FLTK machinery. However, once the desired lighting conditions are acquired, one can apply this state to a high resolution in a single pass.

The 26 small images are stored in a reflectance map along with each associated light direction vector. These images are also calibrated to each node using the numbering scheme mentioned earlier. We display the images to the user in a 3x3 grid. For each light created by the user, we display the interpolated single-light-source images on the perimeter of this grid. The destination image is located in the center and is a composite of these single-light-source images. Color or other property manipulation is applied only to the currently selected light source (Figure 7).

VI. Interpolation

A. Single Light Source

During light source movement and manipulation we track the distances to the three nearest nodes from the current light source. These distances give us a weighting metric to do linear interpolation between the nodes. Given a current light position$(x_l, y_l, z_l)^T$, we associate each of the three source photographs s_i with a weight w_i where

$$w_i = \frac{1.0}{\sqrt{(x_l - x_i)^2 + (y_l - y_i)^2 + (z_l - z_i)^2}}$$

To ensure that the composite image has the same brightness as the source images, we mandate that the weights at each pixel sum to one. This is accomplished by normalizing the weights. We also clamp the value of w_i to slightly less than one to avoid dividing by zero in the above equation. The three weighted source images are then added together to produce the final composite. For each pixel $p_j(x, y)$ in the composite single-light-source image we compute:

$$p_j(x, y) = \sum_{1}^{3} w_i * s_i(x, y)$$

B. Multiple Light Sources

Creating a final composite image from multiple light sources is a natural extension from the interpolation technique for single light sources. In order to have correct and realistic illumination from multiple light sources, we must have additive interference amongst the lights. Simply adding the pixels from single-light-source images p_j will over-saturate shadows. In fact, we must know something about the surface normals of the object in the image to do this correctly. In practice, the user frequently wishes to light an image with illuminants that have little interference, such as a side light and a key light. With this in mind, we use a naive approach which takes the maximum of the pixel values over all of the sources. For each pixel $c(x, y)$ in our final composite image we compute:

$$c(x, y) = \max_{i=(1 \ldots n)}\left(s_i(x, y)\right)$$

This approach has the advantage of conserving shadows and provides blending along seams between different light sources.

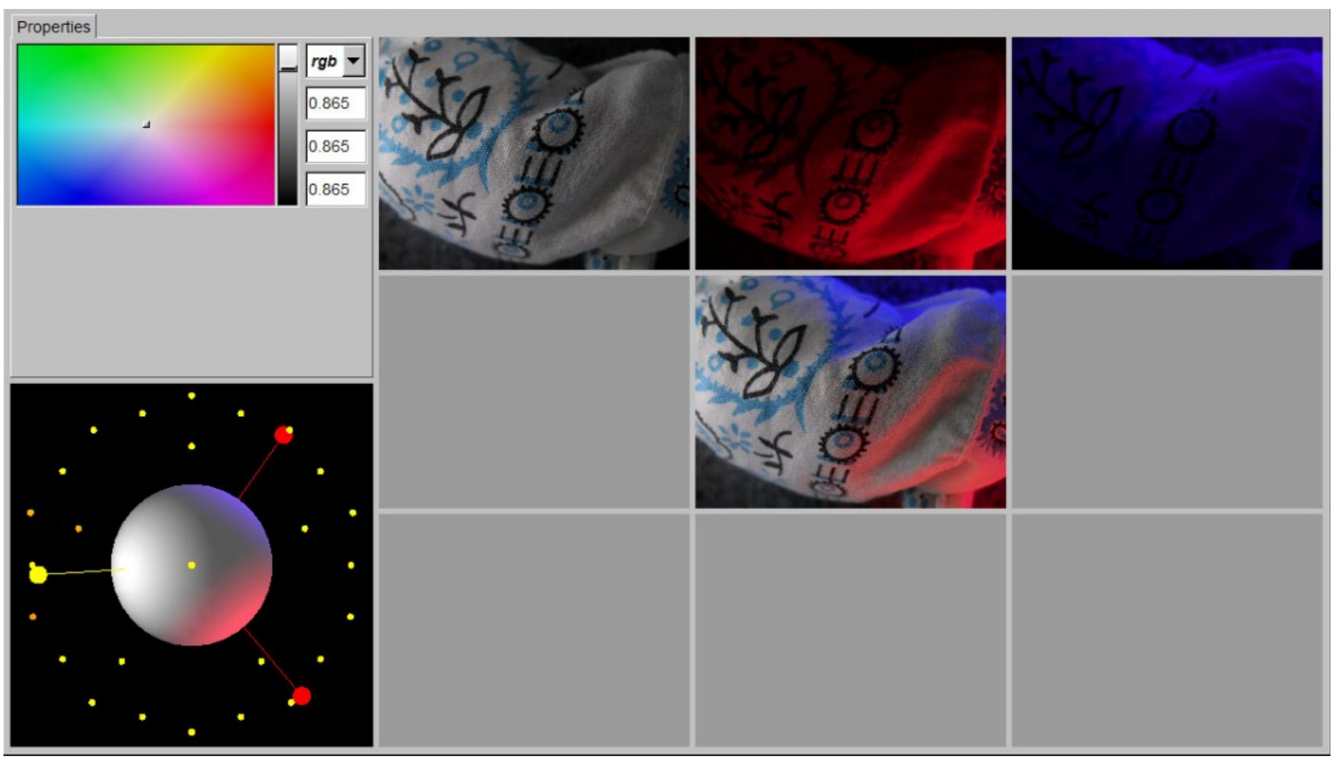

Figure 7: The graphical user interface. At bottom left is the light hemisphere. At top right is the 3x3 grid with the interpolated single-light-source images on the perimeter of this grid and the composite of these single-light-source images in the center. At top left is the color grid. Color or other property manipulation is applied only to the currently selected light source

C. Light Recoloring

Performing light recoloring is also a simple extension onto the interpolation technique described above. The user specifies the desired light color as an RGB value. This value is then scaled between 0 and 1 and is multiplied with the current RGB color value of each pixel in the source image. Given a three tuple $(R_L G_L B_L)$ representing the light color, we compute the new RGB pixel values as follows:

$$R' = \frac{R_L * R}{255}, G' = \frac{G_L * G}{255}, B' = \frac{B_L * B}{255}$$

Note that no further processing must be done to form a composite image from multiple different-colored light directions.

VII. Results

Figure 5 shows the difference between triangle basis lighting conditions and single light basis lighting conditions. Figure 4 shows the image stack of an object with a specular surface. Figure 8 shows the image stack of an object with specular surface and subsurface scattering. Figure 9 shows the image stack of an object with diffuse surface and subsurface scattering.

Camera_view.AVI shows the viewpoint of the camera during the sequential changes of lighting conditions. The video has very low image resolution because of the low lighting. This is not a problem in image acquisition because the camera is configured to auto-focus with optical zoom for a macro setting.

Figure 7 shows the complete interactive relighting interface for macro photography.

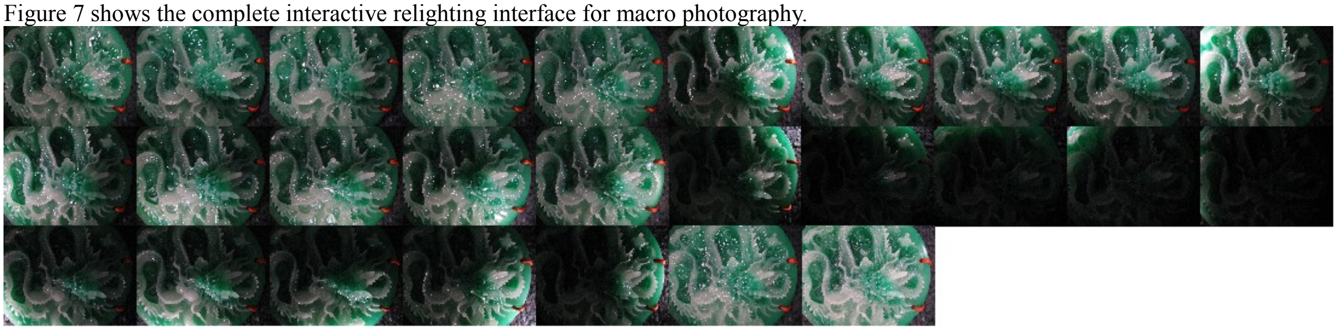

Figure 8: Image stack of an object with specular surface and subsurface scattering.

Figure 9: Image stack of an object with diffuse surface and subsurface scattering.

VIII. Conclusion

For this project, we used image-based rendering with time-multiplexed illumination and bilinear interpolation reconstruction to create a user interface for interactive relighting in macro photography. Lighting the scene with a hemispherical geodesic dome allows for a simple and fast implementation of scene rerendering with novel illumination. However, this limits the viewpoint and background of the scene. More complex reconstruction techniques could be implemented to either increase fidelity of the rendered scene or to create interactive relighting of macro photography with contrast enhancements.

IX. Future Work

The lighting structure could be built for a Digital Single-Lens Reflex camera with a macro lens for better focusing. An extra support structure has to be constructed to ensure minimal movement when the DSLR is attached to the dome. Fitting the DSLR on a tripod would allow for a wider range of possible background environments. Our current implementation would only accommodate flat surface backgrounds.

An interactive relighting of macro scenes with user-specifiable viewpoints could be implemented using light field rendering [Levoy96] with multiple viewpoints [Wong97]. This allows for a larger degree of freedom in interactive relighting and would be useful for rapid relighting prototyping.

Foreground matting and the compositing of background with foreground while preserving realistic lighting conditions would be useful for the implementation of interactive relighting of macro scenes with user-specifiable backgrounds. [Debevec02]'s approach of using infra-red light for the matting of human actors has limited potential in macro photography because it would only be useful for subjects or objects that emit heat.

For higher fidelity to realistic lighting in scenes with dynamic incident illumination, future work could implement the multilevel B-Spline reconstruction technique introduced in [Masselus04].

Finally, future work could feature an interactive relighting of macro photography with contrast enhancements by constructing a polynomial texture map from the acquired image stack and modify the reflectance properties to enhance specularity or increase diffuse gain [Malzbender01].

References

[CHDK] http://chdk.wikia.com/wiki

[Debevec02] Debevec, P., Wenger, A., Tchou, C., Gardner, A., Waese, J., Hawkins, T., "A Lighting Reproduction Approach to Live-Action Compositing", *ACM Transactions on Graphics*,Vol. 21, Issue 3, July 2002, pp. 547-556, ISBN 1-58113-521-1.

[Debevec05] Wenger, A., Gardner, A., Tchou, C., Unger, J., Hawkins, T., Debevec, P., "Performance Relighting and Reflectance Transformation with Time-Multiplexed Illumination", *ACM Transactions on Graphics (SIGGRAPH 2005)*, Vol. 24, Issue 3, July 2005 pp. 756-764.

[FLTK] http://www.fltk.org

[Levoy96] Levoy, M., Hanrahan, P., "Light Field Rendering", In *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 1996)*, 1996, pp. 31-42, ISBN 0-89791-746-4.

[Malzbender01] Malzbender, T., Gelb, D., and Wolters, H, "Polynomial Texture Maps", *Proceedings of SIGGRAPH 2001* (August 2001), pp. 519–528, ISBN 1-58113-292-1.

[Masselus04] Masselus, V., Peers, P., Dutré, P., Willems, Y., "Smooth Reconstruction and Compact Representation of Reflectance Functions for Image-based Relighting", *Eurographics Symposium on Rendering 2004*, June 2004.

[Matusik02] Matusik, W., Pfister, H., Ngan, A., Beardsley, P., Ziegler, R., McMillan, L., "Image-based 3D Photography using Opacity Hulls", *ACM Transactions on Graphics,* Vol. 21, Issue 3, July, 2002, pp.427-437.

[Ultra Intervalometer] http://chdk.wikia.com/wiki/UBASIC/Scripts:_Ultra_Intervalometer

[Raskar04] Raskar, R., Yu, J., Ilie, A., "A Non-Photorealistic Camera: Detecting Silhouettes With Multi-Flash", *ACM SIGGRAPH 2003 Technical Sketch*, 2003.

[Wiring] http://www.wiring.org.co

[Wong 97] Wong, T., Heng, P., Or, S., Ng, W., "Image-based Rendering with Controllable Illumination", Rendering Techniques 97: *Proceedings of the 8th Eurographics Workshop on Rendering*, June, 1997, pp. 13-22.

[Wong01] Wong, T., Heng, P., Fu, C., "Interactive Relighting of Panoramas", *IEEE Computer Graphics and Applications*, Vol.21, No.2,Mar/Apr, 2001, pp. 32-41.