Non-photorealistic Radiosity Rendering

John Brosz and Dustin Lindsay

April 30, 2004

1 Background

Cel-rendering (a.k.a toon shading, comic rendering, or cartoon rendering) is a method of non-photorealistic rendering where scenes are drawn using solid colors separated by silhouette and contour lines. This method was first introduced (to the best of my knowledge) by [Decaudin].

Two key methods are used to perform cartoon shading. In the simpler approach, the polygons of the model are simply filled with a solid color. In the other, more interesting approach, shadowed areas of the model are represented with darker areas of the solid color. This additional complication gives the viewer cues as to the shape and lighting of characters and scenes [Graf]. To achieve boundaries between the lit and unlit areas of the models and Phong lighting equation is modified to:

$$C = A_l * A_m + max(L \cdot N, 0) * D_l * D_m$$

where A=ambient, D=diffuse, l=light, m=material, L=unit vector from light source, N=unit normal of material at the point of intersection, and C is our resulting color.

This lighting equation is then used to create an intensity map for the model or scene. After the intensity has been calculated a set of colors and thresholds is then determined. This is a mapping of the range [0-1] achieved by the dot product of N and L to a set of colors (usually between two and four different colors). The intensity map for the model is then colored based on the chosen thresholds.

This shading techniques works very well for cartoons and when used in conjunction with stylized strokes [Lake et al.] can be used to achieve sketch-style drawings

2 Problem

This shading method works well for simple sketches with a point light source, but when we desire conditions where more than one light source and/or diffuse lighting conditions are present it is clear that such simple shading alone is notsufficient. For such complex environments a radiosity solution is often used to create photorealistic renderings of such environments, so it makes sense that we would use some information from radiosity to create non-photorealistic renderings of these environments.

Before I go any further, it is important to address the question: why would we want NPR renderings of such complex environments? There are several applications that

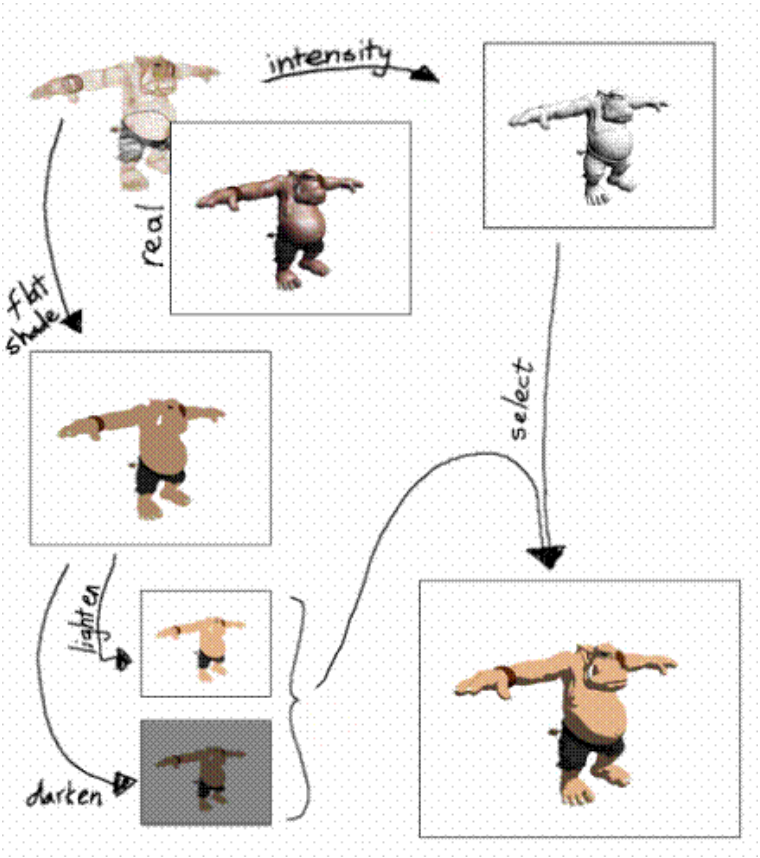

Figure 1: The above figure is from [Graf] and illustrates this process very well.

readily come to mind. The first and foremost are architecture sketches and paintings. These works are used to portray the general lighting conditions of a building while it is being planned. A photorealistic rendering is not as desirable for such scenarios because exact room details are sometimes not known or desired or because they produce output that is too complicated when a broad approximation is desired. Another application is in reproducing paintings and other art work where light plays are large role.

3 Approach

Firstly, using a technique such as the dot product between the surface normal and the direction of the nearest light source (or even several light sources) will not sufficiently portray complicated lighting scenarios, especially in cases where diffuse reflection is a contributing factor.

A great deal of work has been done creating complicated methods of realistically

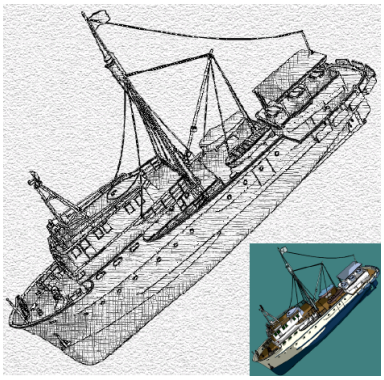

Figure 2: An example of pencil sketches style rendering with cartoon rendering inset from [Lake et Al].

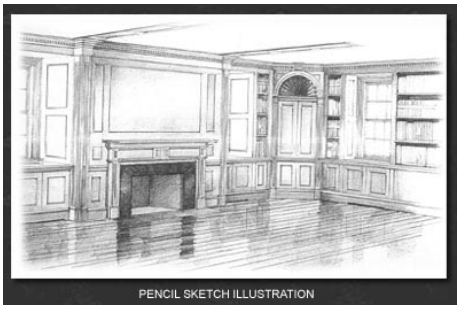

Figure 3: An example architecture sketch provided by Dr Costa Sousa (real source unknown).

simulating light in numerous environments. Radiosity, photon mapping, and ray tracing, are but a few methods that exist to simulate light. Rather than recreate such simulations the goal of this work is to use and reinterpret the results of these simulations to achieve our goal.

There are many applications that combine many rendering techniques. Two noteworthy examples that I considered for this work are POV-Ray (a ray tracer that incorporates radiosity, photon mapping, sub-surface scatting and many other effects), and radiance (a monte-carlo based ray tracer that incorporates radiance calculations giving very accurate lighting solutions). In this work I have exclusively used POV-Ray as it worked after a day of attempting to compile it (as opposed to Radiance where I spent two days attempting to compile and have not yet succeeded). However, the technique I have proposed should work very well with any simulation where radiosity values are calculated.

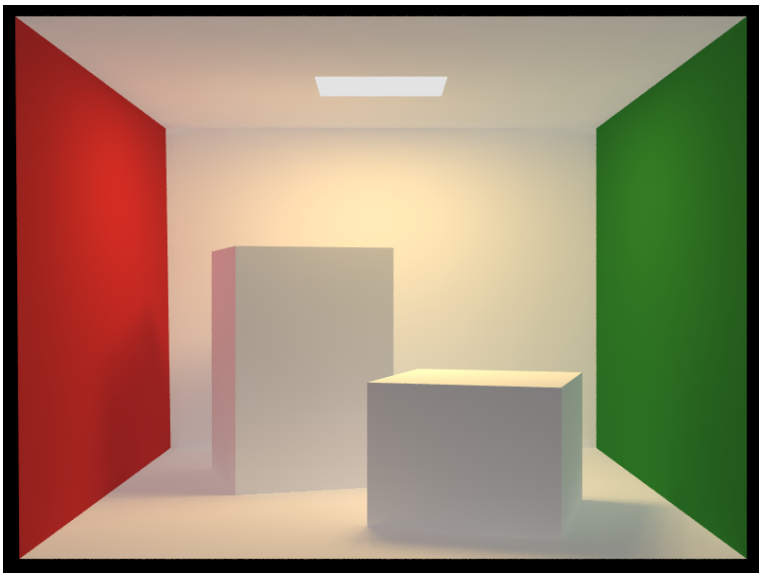

Figure 4: The Cornell box as rendered by POV-Ray.

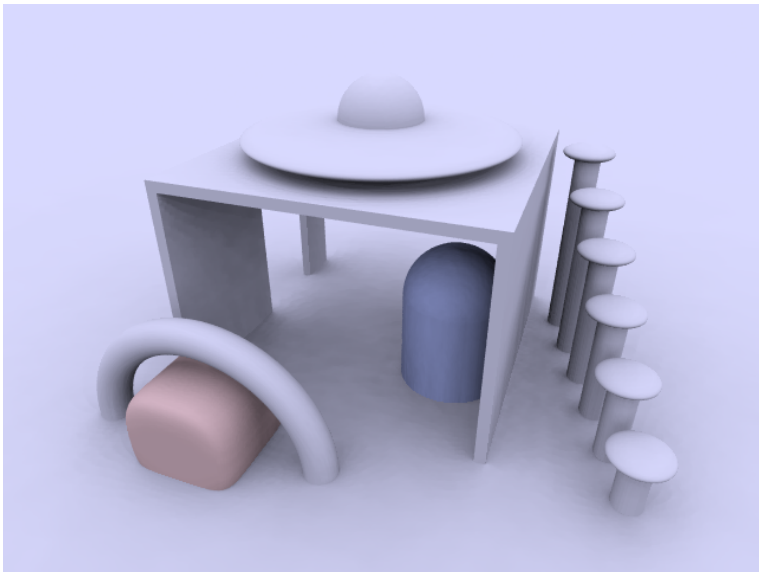

Figure 5: A radiosity test scene as rendered by POV-Ray.

4 Trivial Solution

The first obvious and trivial solution is to simply take the image output by our radiosity rendering, extract some kind of intensity value from each pixel and then use these

intensity values to re-color the image using the original material colors. The extraction I have used for the intensity values is a simple conversion to grey scale where

$$I = 0.299 * red + 0.587 * green + 0.114 * blue.$$

This trivial extraction yields the following results:

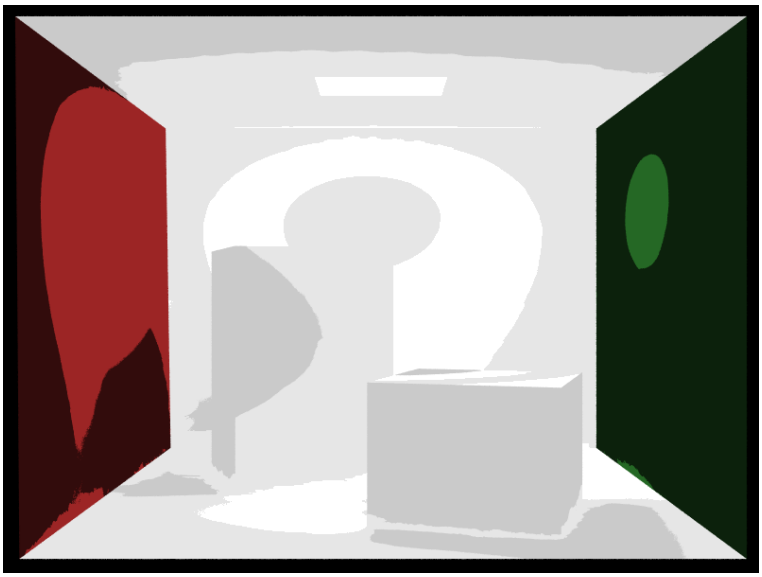

Figure 6: Cornell box NPR rendering produced with trivial solution. Four threshold colors have been used.

The algorithm used to achieve these results is:

- Radiosity is calculated for the scene.
- Each pixel's color for the final image is calculated.
- The pixel's color is cast into a grey scale intensity value.
- The intensity value determines the category of color to use, the color is taken from the material properties.

4.1 Analysis

When looking at the resulting images, although some of the lighting information is conveyed by the resulting grey scale intensity, it is obvious that the original material colors distort the result. For example, a well lit dark colored object will likely still be thresholded to a dark light value because it does not become light enough in color to be considered lit. The opposite occurs with a light object that is dimly lit. Another drawback to this technique is that depending on the final intensity levels of the scene

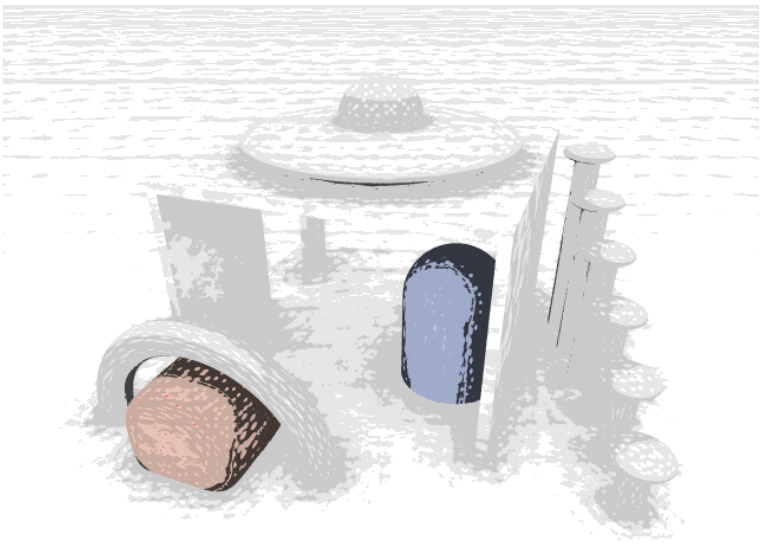

Figure 7: NPR rendering of test scene produced with trivial solution. Four threshold colors have been used.

it is often necessary to spend a lot of time adjusting the threshold levels to achieve decent looking results. An obvious benefit of this method is that such thresholding can be performed on any sort of image (even photographs) without any knowledge of underlying light calculations.

It is my conclusion that this method does not do a good job of portraying the light present in complex scenes. It may be possible to achieve better results by using more threshold levels and spending long periods adjusting the threshold values, but the results of this do not solve the underlying problem that the actual light in the scene is not accounted for.

5 Radiosity Thresholding

A better idea for thresholding the light intensity in the scene is to threshold based on the calculated radiosity values. This quickly yields two alternatives for coloring. In first coloring method we use the material properties of the object being rendered (the same as our trivial solution). The alternative coloring method is to use fractions of the diffuse color calculated by the radiosity solution.

The results achieved using the first method of coloring are shown in figure 8. The results achieved using the second method of coloring are shown in figures 9 and 10.

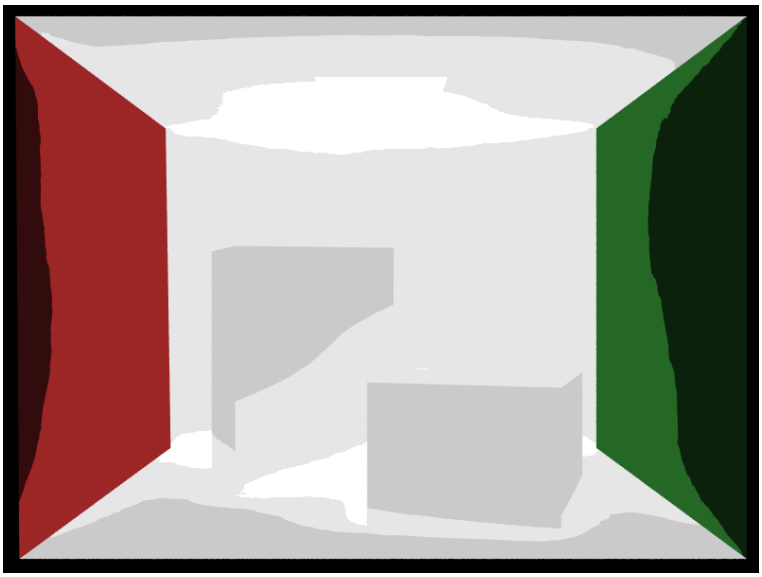

Figure 8: Cornell box NPR rendering produced with the first coloring method (and radiosity thresholds). Four threshold colors have been used.

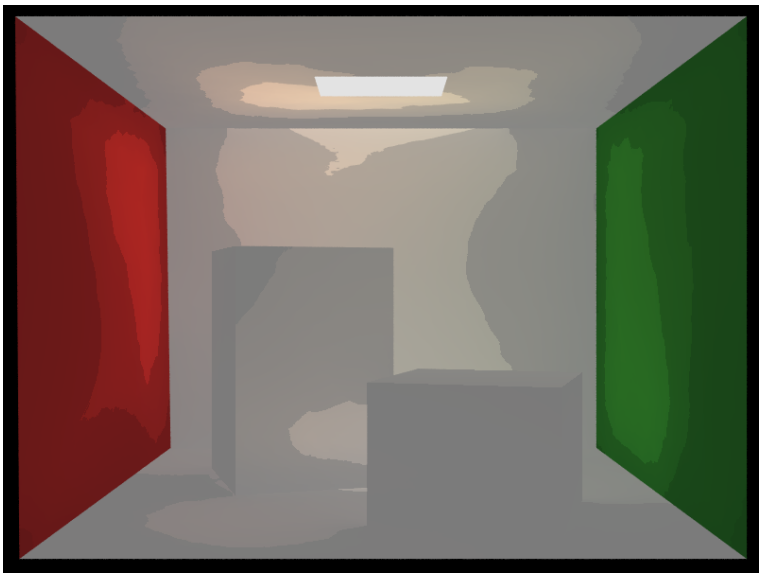

Figure 9: Cornell box NPR rendering produced with the second coloring method (and radiosity thresholds). Four radiosity thresholds have been used.

5.1 Analysis

For the first method of coloring, we can immediately get a better impression of the major areas of lighting in the scene. Unfortunately the problem of finding good thresh-

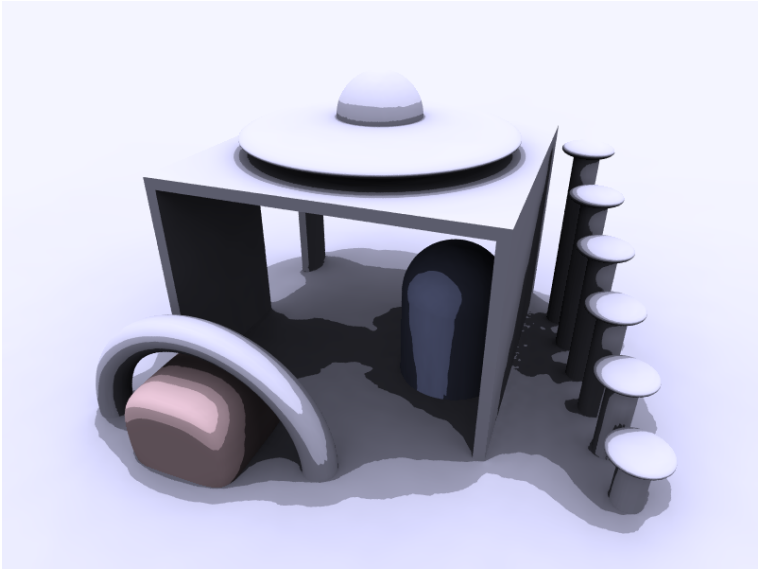

Figure 10: NPR rendering of test scene produced with the second coloring method (and radiosity thresholds). Four radiosity thresholds have been used.

old levels still exists. I spent a large amount of time for each image finding better thresholds than my initial predictions.

The second method of coloring achieves much better results than either of the two previous attempts. With the four threshold levels used we get a very good impression of the lighting in the scene. Furthermore, these threshold values are easily obtained simply by storing the largest radiosity value and then thresholding to equally spaced fractions of this value (0.66, 0.33, 0.01) we achieve good results for almost all scenes. When viewing the resulting image we can easily determine the areas that are directly receiving light from the areas that are shaded. By using the calculated colors we do get a small amount of shading instead of the flat tones we were originally looking for. However, the amount of shading is quite small and thus not a large concern.

6 Unobtained Goals

To achieve better renderings that are more comprehensible silhouette lines are needed. This is something I wanted to do but have not had time to accomplish. This should be quite straightforward. My plan is to use an image-space silhouette extraction approach [Isenberg] where edge detection is used on the depth, normal, and object buffer to find the silhouette lines. Unfortunately, POV-Ray does not include these buffers and I have not taken the time to write the edge-detection code.

The next thing I would like to achieve is replacing the thresholded colors with the pen and ink textures shown in figure 11. This should be easily obtained by constructing

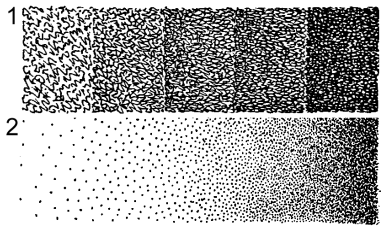

Figure 11: Real ink textures.

a buffer that stores threshold category for each pixel of the image. Then it is a simple operation to use the texture corresponding to each threshold category to color the pixel. To achieve results where the underlying objects are visible, it is first necessary to draw the silhouette lines (otherwise it is very difficult to make out the contents of the scene).

7 More Results & Conclusion

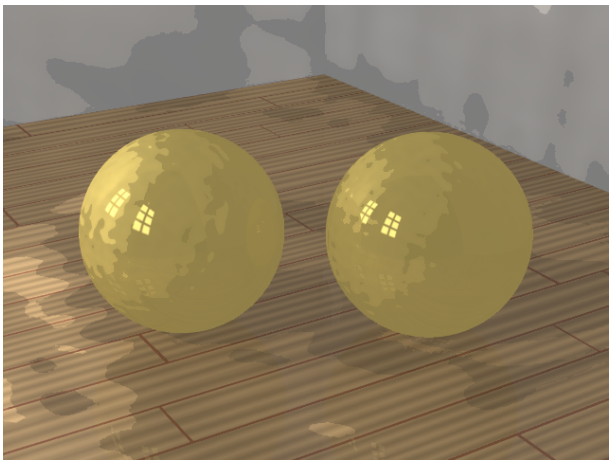

Figure 12: Scene of metallic balls lit by a pair of French windows.

Two conclusions can be drawn from this work. The first is that the trivial approach does not do a good job portraying the light present in the scene. The second is that by thresholding based on the results of the radiosity calculation we can achieve results that accurately convey the approximate lighting of the scene.

8 Future Work

I would like to implement this rendering method within Radiance. It would be interesting to examine these results as I suspect that Radiance's radiosity calculation is of

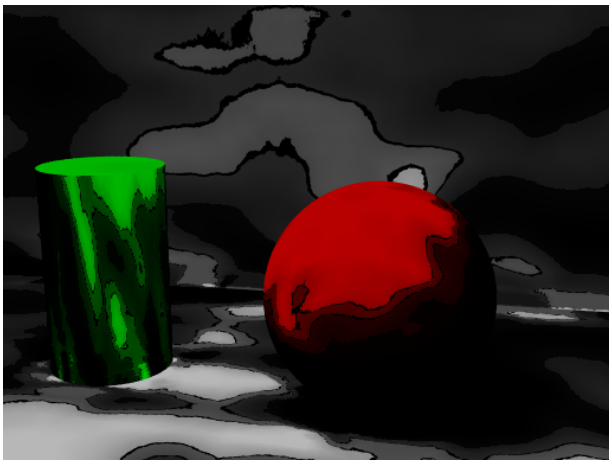

Figure 13: A cylinder and sphere.

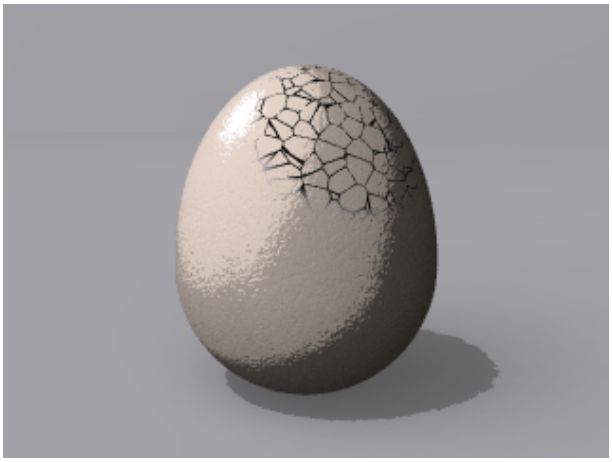

Figure 14: A bump-mapped egg.

higher quality than POV-Ray's radiosity calculation.

9 References

P. Decaudin, "Cartoon-Looking Rendering of 3D-Scenes", INRIA Research Report, Number 2919, June 1996.

C. Graf, "A tutorial on how to create cartoon shaded figures from 3d models". Retrieved from http://www.cs.uni-magdeburg.de/ cgraf/NZ/COMPSCI715/Project/Tutorial.html.

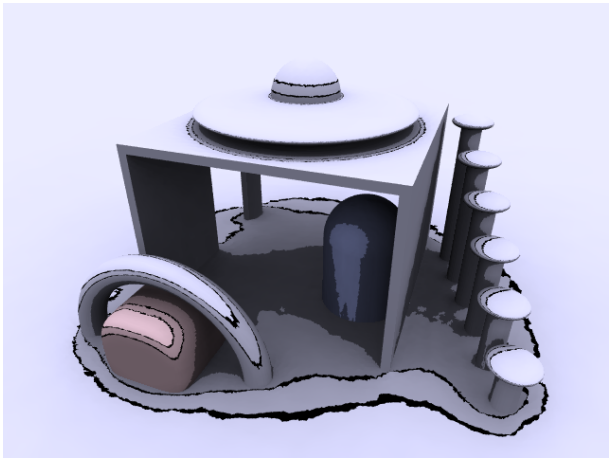

Figure 15: The radiosity test scene where intensity values bordering between the existing thresholds are set to black. Produces an interesting outline effect.

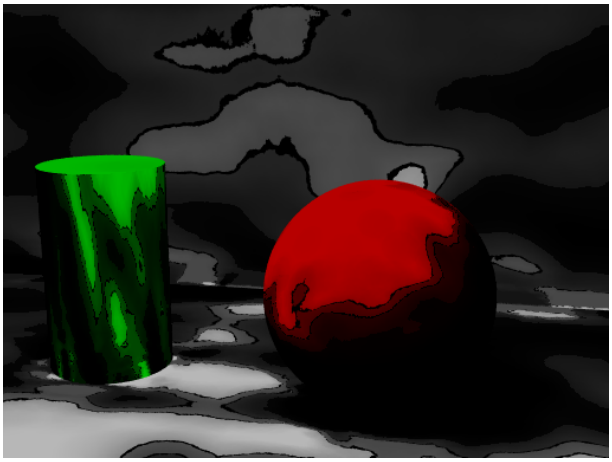

Figure 16: A cylinder and sphere with threshold borders rendered.

T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte, ”A developer’s guide to silhouette algorithms for polygonal models”, IEEE Computer Graphics and Applications, July/August 2003, pp 28-37.

A. Lake, C. Marshall, M. Harris, M. Blackstsein, ”Stylized Rendering Techniques for Scalable Real-time Animation”, ACM Non-photorealistic Animation and Rendering (NPAR)Symposium, (2000).