Specularity Replacement in a Contour in a Single image

Stephen Akers, Nadia El Mouldi, Thomas Weihe
MSCS
St. Olaf College
Northfield, MN 55057
akers@stolaf.edu, elmoul1@stolaf.edu, weihe@stolaf.edu

Abstract

We present our approach of detecting specularity based on the luminance of pixels, synthesizing texture from a diffuse sample and replacing the specularity with the synthesized texture.

After applying a luminance threshold to differentiate between specular and diffuse reflection in pixels, we use an implementation of the Efros-Leung texture synthesis algorithm to select a diffuse sample from the original image to generate a continuation of the texture to replace pixels in a contour in an image affected by specular reflection.

The specular detection method left either some specularity in the image, or it replaced diffuse pixels with the synthesized texture as well, resulting in an inaccurate depiction of the diffuse component of the image. The version of the Efros-Leung texture synthesis method we implemented greatly reduces the natural diffuse color variation of the texture.

1 Introduction

When specular reflection is present on an object, the color information is lost to varying degrees. For this reason, detecting and replacing specular reflection with the diffuse texture can restore valuable information about a given image. Although research exists on extracting a diffuse and specular map from an image, restoring diffuse pixels from specular pixels is still a challenging problem in computer vision.

First, specularity can sometimes be mistaken for a light source as it can have high luminance. Second, predicting a natural continuation of a texture can be difficult when only a small sample is used.

In our paper, we talk about our approach of detecting the specular and the diffuse components in a contour in a single image using Laskowski's luminance formula [3]. We use one of two separate methods of texture synthesis, one using texture multiplication, and one based on the Efros-Leung algorithm [1], to replace the detected specular pixels. Although the algorithm has been studied and has achieved results to some extent[2], such efforts have been limited, and further research into similar algorithms is necessary.

2 Background

2.1 Luminance

The luminance of a pixel can be used as a tool to predict whether a pixel belongs to a diffuse or a specular area. The luminance values also allow us to differentiate between specular areas and light sources. Laskowski's proposed method [3] detects light sources based on the luminance of the pixels. The first step of the algorithm is to calculate the luminance of the pixels using the formula:
where R,G and B are the values of the red, blue and green channels respectively.

$$L(R, G, B) = 0.2126 \times R + 0.7152 \times G + 0.0722 \times B$$

Then, using the Monte Carlo method, we select a random set of points from the image, of which we select the point of largest luminance, eliminating the others in the set. After repeating this process a number of times, the result is an image with potential detected light sources.

The second step of the algorithm is to reduce the number of the detected light sources. By choosing the most luminant pixels and eliminating the points in their neighborhood, we reduce the number of points per possible light source.

The last step is to reduce the number of points to a single point differentiating each light source. To do this, we determine a line between all the points in a single area source of light using the Bresenham's algorithm. If the difference between the luminance of the points is small, we calculate the mean of these points to end up with one single point referring to each light source.

2.2 Texture Synthesis

The Efros and Leung [1] algorithm is the most commonly used way of synthesizing a texture from a finite sample. Aguerrebere et al. [2] implemented this algorithm to create a new image from a seed component and a sample image. We start by finding the next pixel to be synthesized, then looking at the patch of pixels surrounding it, and after that comparing them to pixel patches in the given sample. The following formula expresses the distance of similarity, denoted as d, between pixel patches, denoted as N, of pixels p and p':

$$d(\mathcal{N}(p), \mathcal{N}(p')) = \frac{1}{\sum_{i \in \mathcal{N}_0} G_\sigma(i)} \sum_{i \in \mathcal{N}_0} (A(p'+i) - B(p+i))^2 G_\sigma(i),$$

where A is the sample image, B is the image to be synthesized, G_σ is the Gaussian with standard deviation σ, $N_0$is an n by n window centered on origin, p' is a pixel in image A, and p is a pixel in image B. After the distances are calculated, the minimum distance d_m between N(p) and all patches in A is calculated:

$$d_m(p) = \min_{p' \in A} d(\mathcal{N}(p), \mathcal{N}(p')).$$

Next, the set of pixels most similar to p is determined using the expression:

$$S_\varepsilon(p) = \{p' \in A : d(\mathcal{N}(p), \mathcal{N}(p')) \leq (1+\varepsilon) d_m(p)\}$$

where ε is the tolerance parameter for similarity between the patches. Once the set S_ε has been determined, a patch from the set is randomly chosen and the center pixel of the patch is used to fill the next pixel.

3 Method

Texture Synthesis, while in the original context is used to generate entirely new images, also provides a solution to the issue of what data should be used to replace pixels that are affected by specularity. In our research, we use a texture synthesis method based off of the implementation of the Efros-Leung algorithm by Aguerrebere et al. [2] to facilitate the removal of specularity from contours.

Our version of this process varies from the original in several ways: Firstly, the texture is not being synthesized on the edge of the sample onto an unfilled pixel; it is being used to replace pixels that are identified as specular, using the luminance threshold described above. Secondly, the pixel to be filled only uses the pixels above it, or the pixel patch, which prevents interference with the comparison of other nearby pixels affected by specularity. Finally, the closest-matching pixel patch is always chosen, rather than a randomly chosen patch from the group of the closest-matching patches.

In order to give our texture synthesis a natural variation, the pixel chosen from the most similar patch is randomly offset from the center by zero, one, or two pixels. Similar to Aguerrebere et al.'s method, our method scans through the image for pixels to replace and look for pixel patches similar to that of the pixel to be filled, in the sample image. Once the most similar patch is located in the sample, a pixel from the patch in the sample is chosen to replace the pixel affected by specularity.

If a given contour contains multiple textures, multiple samples can be used to replace specularity in the image. The dataset of images we worked with frequently contained two distinct textures. We differentiated between these two textures by looking at the difference between RGB values.

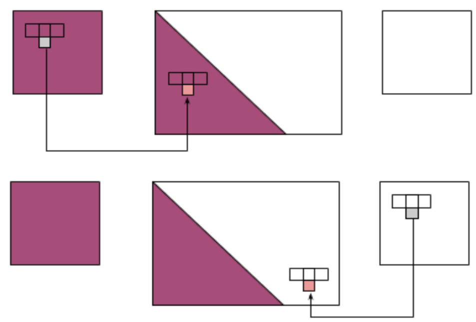

Figure 1: Different samples (far left and far right) are used to replace specularity in different parts of the image. (middle)

4 Results

The Efros-Leung texture synthesis algorithm we used to replace specular pixels was moderately successful with images of certain textures, although with loss of detail from the original images. This technique needs refining as it is imperfect at replicating the exact texture of an object with the parameters we chose, and it leaves unrealistic over-frequent repetitions of patterns in the image. It also greatly reduces the natural diffuse color variation of the texture. Also, the luminance threshold provides only superficial specularity detection which may include a nontrivial number of diffuse pixels with the specular pixels in what it classifies as specular. It also handles multiple textures within an image only with a rather narrow manual specification of the differing RGB values. Runtime is also an issue with this method as a single 1188 x 792 PPM image file can take 40 to 60 minutes to be processed with this technique, although runtime can be decreased with a substantial tradeoff in accuracy by decreasing the number of chunks the program compares to the blank pixel chunk.

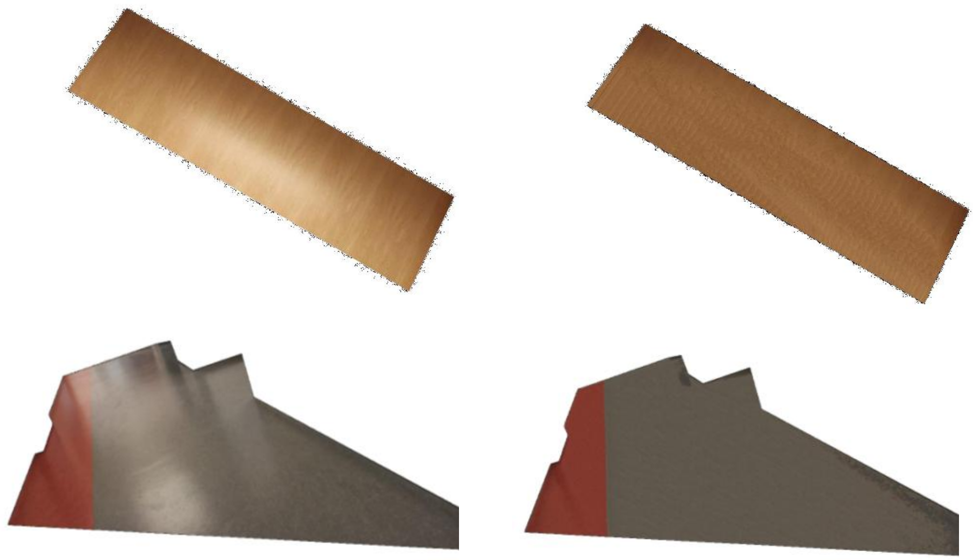

Figure 5: Right: Images modified using our texture synthesis specular pixel replacement technique. The wood contour utilized one sample, whereas the floor contour utilized two. (Note: original background of images was black)

5 Future Work

We based our work on using one single image to detect and remove specularity but future development can consider using multiple images from the same scene taken from different angles. By aligning and comparing them, we can extract a more accurate estimation of the diffuse component of the contour.

A more focused specular detection technique is still necessary to retain the highest amount of original data from an image. Too much pixel-level color variation is lost using the current texture synthesis configuration for it to be a convincing replacement of the original specular pixels in most situations. Also, a more automated color and texture segmentation method would be required in order to make the application of the multiple-texture techniques viable on a large scale, due to the sheer variety and complexity of RGB color value differences between textures.

References

[1] A.A. Efros, T.K. Leung. Texture Synthesis by Nonparametric Sampling. University of California, Berkeley, Computer Science Division. Berkeley, CA.

[2] Aguerrebere et al. Exemplar-Based Texture Synthesis: the Efros-Leung Algorithm. IIE, UdelaR, Uruguay and Télécom ParisTech, LTCI CNRS, France.

[3] Laskowski, M. Detection of light sources in digital photographs. In Proceedings of the 11th *Central European Seminar on Computer Graphics. Central European Seminar* on Computer Graphics, 2007.

[4] R.T. Tan, K. Ikeuchi. Separating Reflection Components of Textured Surface using a Single Image. Proc. *IEEE Intl Conf. Computer Vision*, 2003.