Scene Edge Classification from Image Sequences

Kelleher Guerin[1], Pyry Matikainen
Computational Photography 15-463
Final Paper, 12/16/08
[1] kguerin.optics@gmail.com

Abstract - There is often interest in identifying geometry in a scene. Edge detection is an established method of looking at scene geometry, but naive edge detection in a complex scene discerns little about the physical geometry; light and shadowing effects especially in outdoor scenes can cause edge detectors to mistakenly classify texture and shadow edges as geometry edges. However, with a sample of several images of a scene where the illumination is changing over time, one can correctly classify different scene edges by removing lighting effects and textures. We present a method of using intrinsic images and time varying intensity gradient to identify geometry edges, texture edges and shadow edges. We first remove shadows, then look at per pixel gradient intensity variance to segment normals in the scene, and use the boundaries of those segments to find geometry edges. The texture edges then are the edges which have not been classified as geometry or shadow.

1. INTRODUCTION

The classification of geometry in a scene is often desired for many vision applications. Robots can use geometry to detect obstacles and potential driving paths. Augmented reality applications need robust notions of scene geometry for the proper overlay of augmented sprites and 3D models. Tracking algorithms often rely on robust classifications of geometry to determine movement. For a simple stationary scene, an edge detector is a reasonably good way to look at geometry, especially in a scene with discrete, none overlapping objects on simple backgrounds. Often programs using edge detection can robustly determine separate objects and use this for labeling. However, the scene characteristics mentioned are very ideal and are not often found in most locations, especially in the outdoors. Because of this, complex scenes do not benefit from naïve edge classification, that is, edge classifiers that arbitrarily look at every edge response in the image. Identifying the type of edge detected could provide more information about geometry and other scene aspects. There are three types of edges which are of interest in the scene. Edges that form the physical boundaries of scene geometry referred to as geometry edges, edges which are intensity boundaries of texture, herein called texture edges, and the boundaries of lighting and shadow, herein called shadow edges. By finding a way of classifying these three edge types, you can determine scene geometry by isolating geometry edges. Further, you can reconstruct the image from these geometry edges into an image which shows only geometry. In order to achieve this, you must have a scene which has time varying light intensity.

2. PREVIOUS WORK

There has been much work in processing scenes to extract geometric information about them. Weiss et al.[1] has worked on creating intrinsic images of a scene. Essentially, given a time and illumination varying sequence of images, the X and Y spatial derivatives of each image are taken, and then a temporal median filter is applied to each X and Y derivative image stack, show in Figure 2. These median gradient images are then used to reconstruct a reflectance only image and an illumination only image using a pseudo inverse, shown in Figure 1.

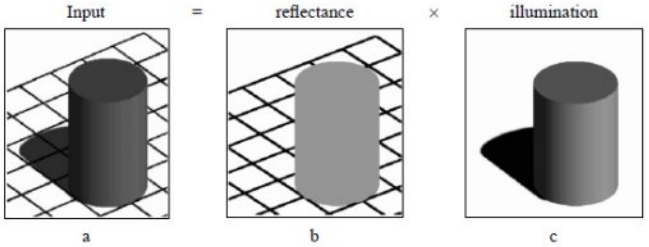

Fig.1 Weiss et al, intrinsic images

Other work in geometry classification in a scene has been done by Koppal et al.[2] where sequences of images were used, again with time varying lighting. The method used per pixel time varying intensity. These intensity functions

[1] Weiss, Y., "Deriving intrinsic images from image sequences," *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on* , vol.2, no., pp.68-75 vol.2, 2001

[2] S. Koppal, S. Narasimhan. Clustering Appearance for Scene Analysis. In Proc. CVPR, 2006

were then clustered vie their maximum values. The intention was that similar surface normal points would have similar maxima, and therefore be clustered together according to canonical trajectories, showin in Figure 3. The edges were then found by looking at the boundaries between these clustered normals.

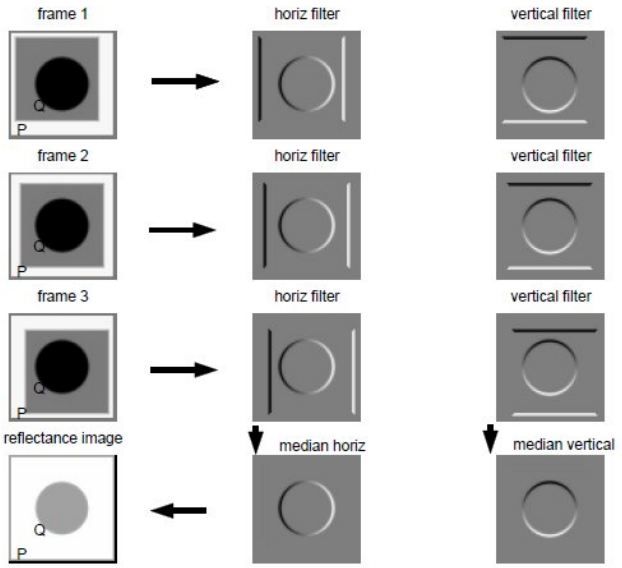

Fig. 2 Gradient Images and reconstruction

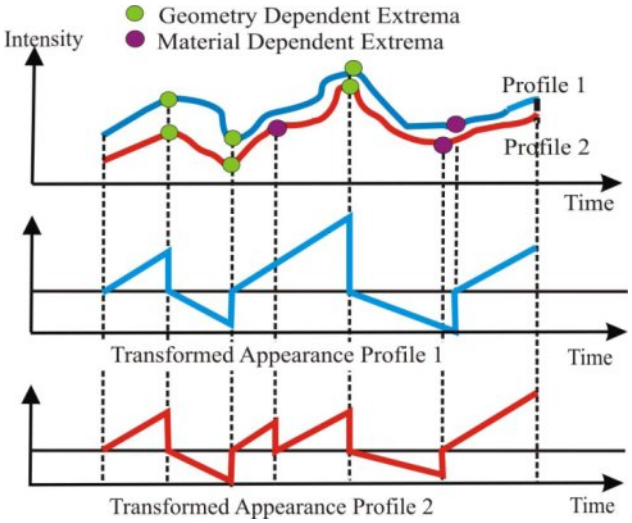

Fig. 3 Pixel appearance profiles

3. GEOMETRY EDGES

For our implementation, we used a sequence of images taken from a University of Arizona webcam (fig. 4), and from an idealized scene constructed in the lab (fig, 5). For the Arizona dataset, images were taken ~20 min intervals for 50 images spread over the entire cloudless day. The idealized scene in the lab used 14 images. Both sequences vary in time with lighting moving in an arch over the scene.

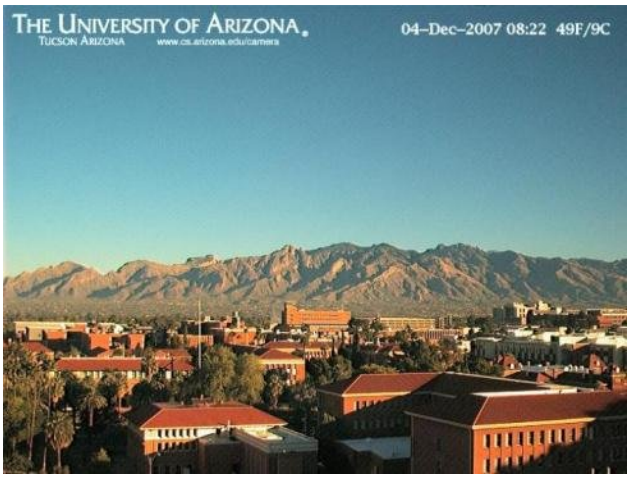

Fig. 4 Arizona webcam sample frame

Fig. 5 Lab scene frame

Our edge segmentation starts out by using Weiss' method of taking X and Y derivative gradients of a sequence of images. We first take each image and convert to LAB colorspace, using the L channel for all of our implementation. We then create a matrix of all the images in the sequences:

$$\mathbf{M = [\ H \times W \times N\ ]}$$

where H is the image height, W is the image width and N is the number of images in the sequence (Fig. 6).

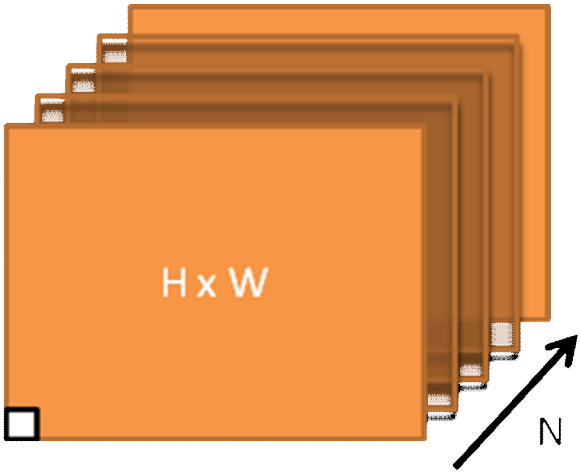

Fig. 6 Image Matrix

This image matrix then allows us to create a 1 x N time varying intensity vector for each pixel location in the image.

$$M(i,j,:) = \{p(i,j)_1\ p(i,j)_2\ \ldots\ p(i,j)_N\}$$

We then take the X and Y gradients (derivatives) of the each image in the spatial domain, and combine these gradients using a sum of squared method. This gives us two indications of the pixel identity, one in the intensity domain, and one in the 2D gradient domain. Shown in Figure 7 is the intensity and in Figure 8 the gradient of three pixels selected from the image (Figure 9). Point A is in shadow for part of the sequence, point B is a texture edge and point C is a geometry edge. Note the spike in the gradient behavior of A from the shadow movement across the scene.

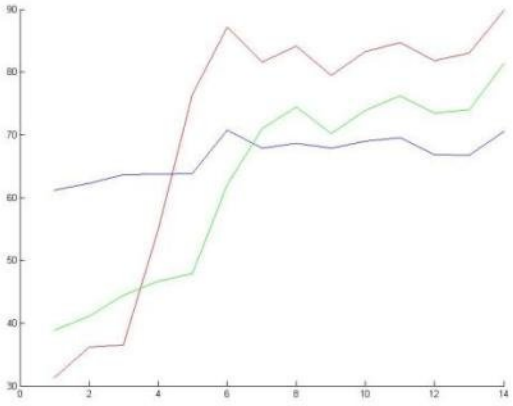

Fig. 7 Pixel intensity values

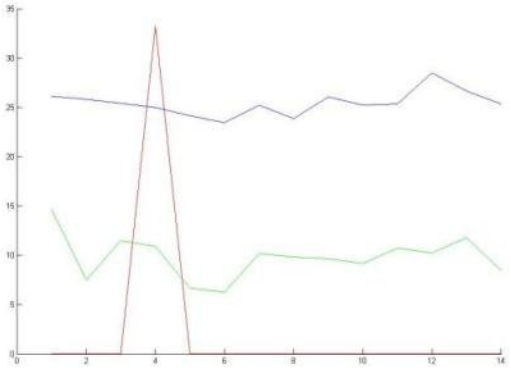

Fig. 8 Pixel gradient intensity values

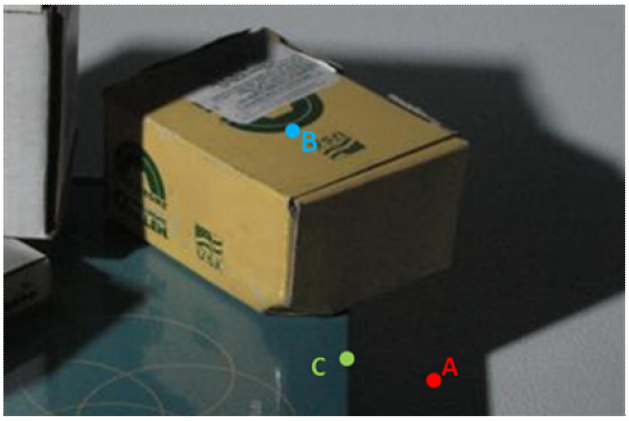

Fig. 9 Three pixels from the image

We can use the variance of these gradient intensity vectors to classify the edge type. By taking a variance measurement of each pixel vector and throwing away the top and bottom 25% of the data, we create an outlier rejecting variance. By taking a threshold of this variance, we can only select the pixels that correspond to edges which are ideally geometry edges (Figure 10).

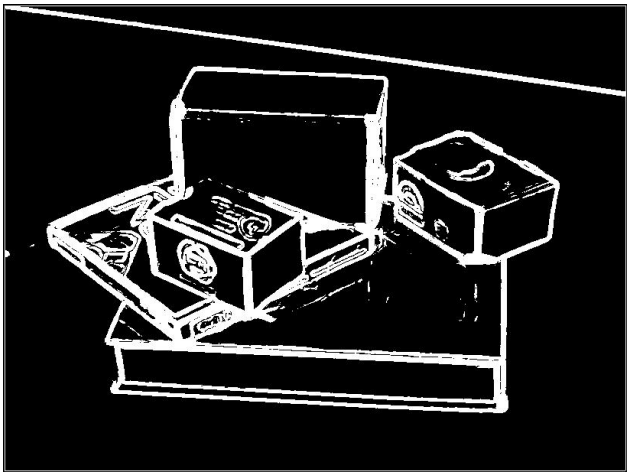

Fig. 10 Geometry edge classification

Note in figure 10 how some texture edges, such as the decals on the boxes have been misclassified as geometry. This is because the thresholding method does not account for all pixel variances; since many pixels in texture and geometry behave very closely. However, this is quite robust to shadow and lighting edges, and does a reasonably good job of selecting only geometry.

4. SHADOW EDGES

Returning to our original X and Y gradients of the image sequences, we can now take each of those sequences and median filter them in the time domain. Then using Weiss' method, we can use the pseudo inverse to reconstruct the intrinsic image. This gives us an image that is totally devoid of illumination and is only composed of the reflectance of the scene (Figure 11). From there it is straightforward to subtract an original L image from the intrinsic image to get an illumination only image (Figure 12).

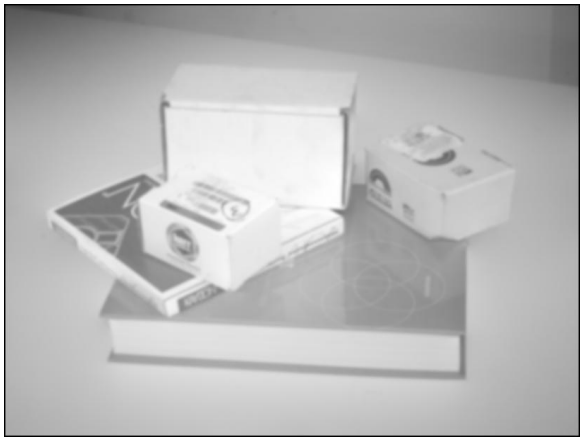

Fig. 11 Median reconstructed intrinsic image

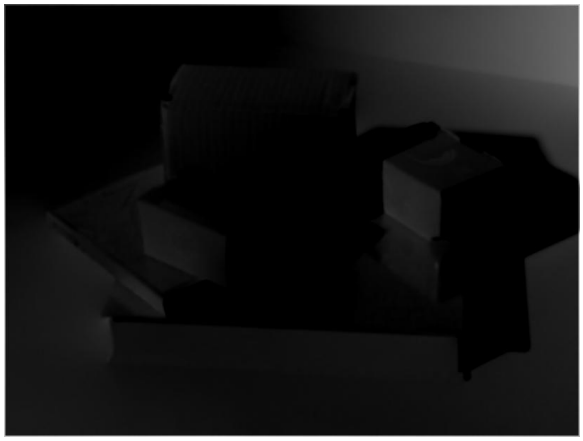

Fig. 12 Illumination only image

Using a method of thresholding and edge detecting this image (using a Canny edge detector) we are able to extract the illumination or shadow edges of the scene (Figure 13).

Fig. 13 Thresholded and Canny shadow results

6. TEXTURE EDGES

Texture edges were found by taking a canny edge sequence of an L image and subtracting the identified geometry and shadow edges. The resulting edges must be from texture. These results are show in sequence in figure 14.

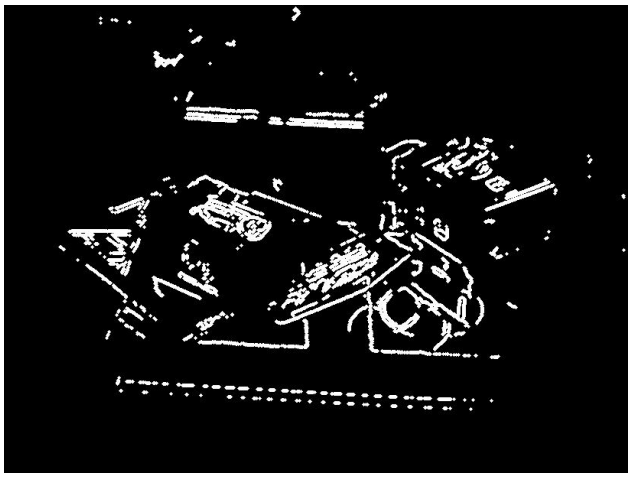

Fig. 14 Results of subtraction, texture edges

6. RESULTS

From these three classifications of geometry edges, texture edges and shadow or illumination images, we were able to create RGB composite images, where red shows the shadow, green shows geometry and blue shows texture edges. The results for the ideal scene and for Arizona are shown in figures 15, 16. There is relatively robust classification of geometry edges and shadow edges, though texture edges are somewhat misclassified with geometry. Also of note, the lettering in the Arizona image has been classified as geometry, but since there is no illumination change there, the lettering is considered texture. This suggests there are slight problems in the intensity variance thresholding method.

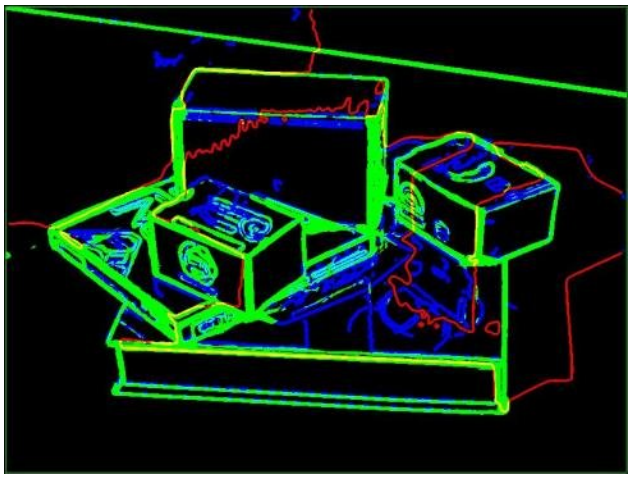

Fig. 15 Lab scene with classified edges

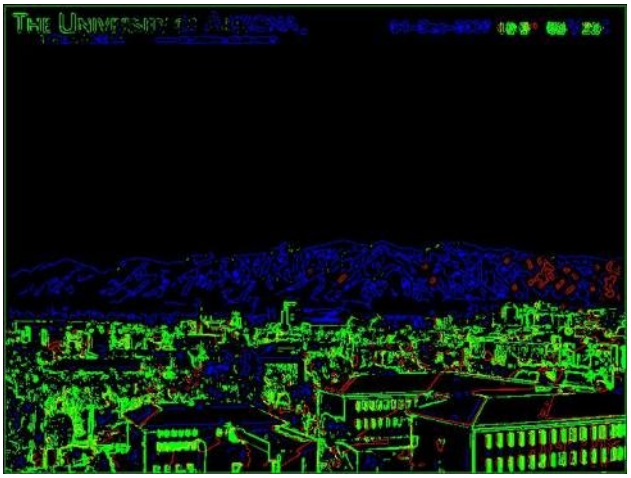

Fig. 16 Arizona classified edges

Another interesting result is the reconstruction images from edge only gradient space. To create these, we take our gradient median images and zero out everything except the desired classified edges, the reconstruct using Weiss' method pseudo inverse. The results for geometry (figure 17) texture only (figure 18) and illumination only (figure 19) are shown. Ideally we would want the geometry edge only reconstruction to be basic shapes without any texture, but due to the misclassification of some texture edges, we still see some texture in the reconstruction.

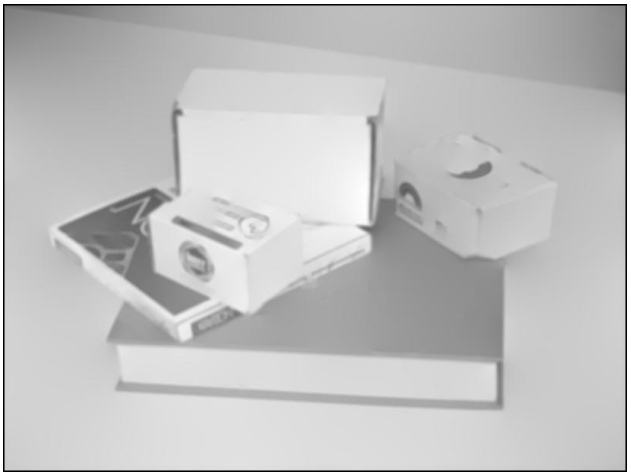

Fig. 17 Geometry edge only reconstruction

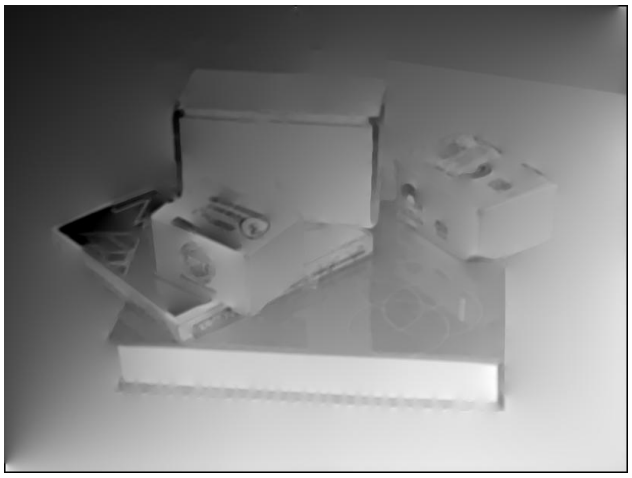

Fig. 18 Texture edge only reconstruction

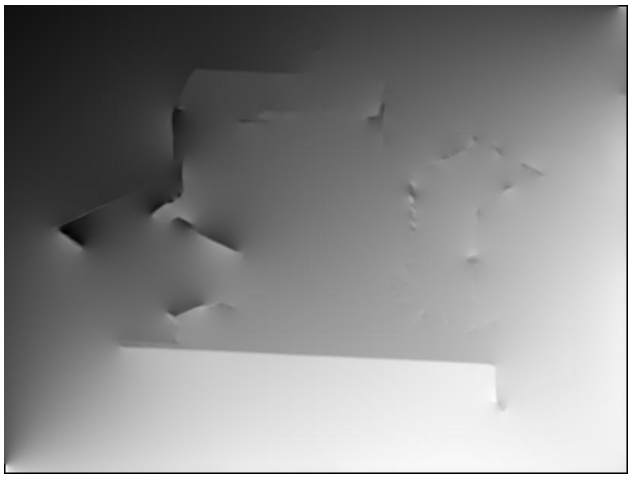

Fig. 19 Shadow edge only reconstruction

7. CONCLUSIONS

We present a relatively robust method of classifying edges in a scene as geometry, texture and shadow edges. This method has some trouble differentiating between geometry and texture, but with more refinement, this misclassification could be mitigated. This method does require a fairly specific scene, one with changing illumination and periodic imaging, but the uses for this classification could still benefit vision applications.