Comp Photography (Spring 2015) Final Project

Jonathan Hudgins
jhudgins8@gatech.edu

Automated Background Replacement

Replace an unwanted background in a picture with a more professional background.

Replacing Background

I took some family photos at a formal event but only realized afterwards how awful the background was. While photoshop provides manual tools to fix individual photos, I would like to automatically replace the background (especially the clutter) in a large number of photographs automatically.

Input

Output

Pipeline

input

materials

mask

background

output

Materials

I attempted various methods (see “What Did Not Work” below), but the most interesting progress was from trying to identify materials using k-means clustering. In computer graphics, materials are usually a collection of textures and material properties (like specularity) that usually feed into vertex and pixel shader programs. Reversing the process I attempt to recover distinct materials.

Attributes

To form material clusters each pixel needs a set of attributes. Here is a list of attributes I used:

- Each channel of original image
- Each channel of a blurred image
- Each channel of blurred image minus original normalized (“Delta”)
- Each channel of inverted gaussian filtered image (“Inverted”)
- X, Y location

Identified Materials

1. All Attributes

2. Color Only

3. Blurred Only

4. Color and Blurred

5. Color, Blurred and Position

6. Blurred, Delta and Inverted

Material Image Analysis

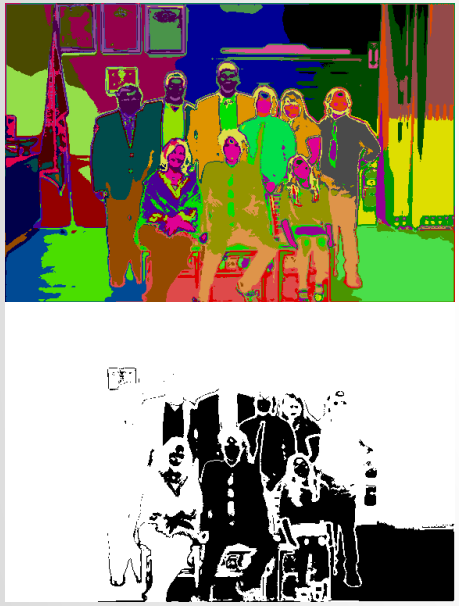
Images 1, 5 and 6 seem to do a pretty good job at determining materials. While images 2, 3 and 4 have a lot of static.

This seems like a promising start in identifying objects in the scene.

Which Materials are Background

I created a simple block around the photograph of known background. Then, for all pixels in that block, mark the associated materials as background materials.

Resulting Masks



1. All Attributes

5. Color, Blurred and Position

6. Blurred, Delta and Inverted

Combining results

Perhaps because of the randomness of clustering and the bleeding of materials, these masks are disappointing. However, if we overlay the masks (a pixel is foreground if it is foreground in ***any*** of the three masks) we get decent results. Then using laplacian blending here is the final image.

Next Steps

- Experiment with other filters to generate different attributes (sharpen, emboss, blurrier, non-symmetric, etc.)
- Logic to exclude materials from background (average proximity to center)
- Identify human (or non-human) shapes
- Use neural network over patch to determine if center pixel is background

… Next Steps

- Determine background materials from colored pixels with highest mode
- Use image pyramids to better segregate materials
- Use distribution-based clustering (Expectation Maximization) to create background probabilities
- Exclude anti-aliased pixels from background mask (pixels on the edge)
- Combine results from various approaches

What did not work? Why?

- Edge detection was not particularly helpful -- because we want to determine entire region not just the boundary.
- Black is such a common color that it happens in background and foreground.
- Clustering is a fairly expensive operation (even when using an image reduced to 1/8th each dimension -- 1/64th total number of pixels)

Conclusion

While the resulting image was less successful than hoped, the intermediate steps show a good deal of potential. With more investigation, this could produce professional level background removal.

References / Pointers

Third party python imports

- OpenCV
- scipy.signal
- sklearn.cluster

While I did some internet investigations, I did not find anything directly useful. Most hits for “static background removal” related to removing the static background from a set of images with a moving object.

Credits/Thanks

- flickr user “rubyblossom” for the background