Implementation

In order to capture any of the following shot the following steps were taken.

1. I set up the camera (my phone) by setting it against a stack of post it notes on a table.
2. I place a removable backing down and take a photo of it
3. I place the object and take a photo of the object and the backing
4. I remove the backing and take a photo with the object and without the back
5. I remove the object and take a photo of the backing

Some exceptions to these step occurred occasionally when I had to improvised to get the matting to work correctly

The camera I used was a sony Z5 smartphone camera with the following settings

- Resolution 23MP with an aspect ratio of 4:3
- ISO is fixed to auto
- No changes to the white balancing settings

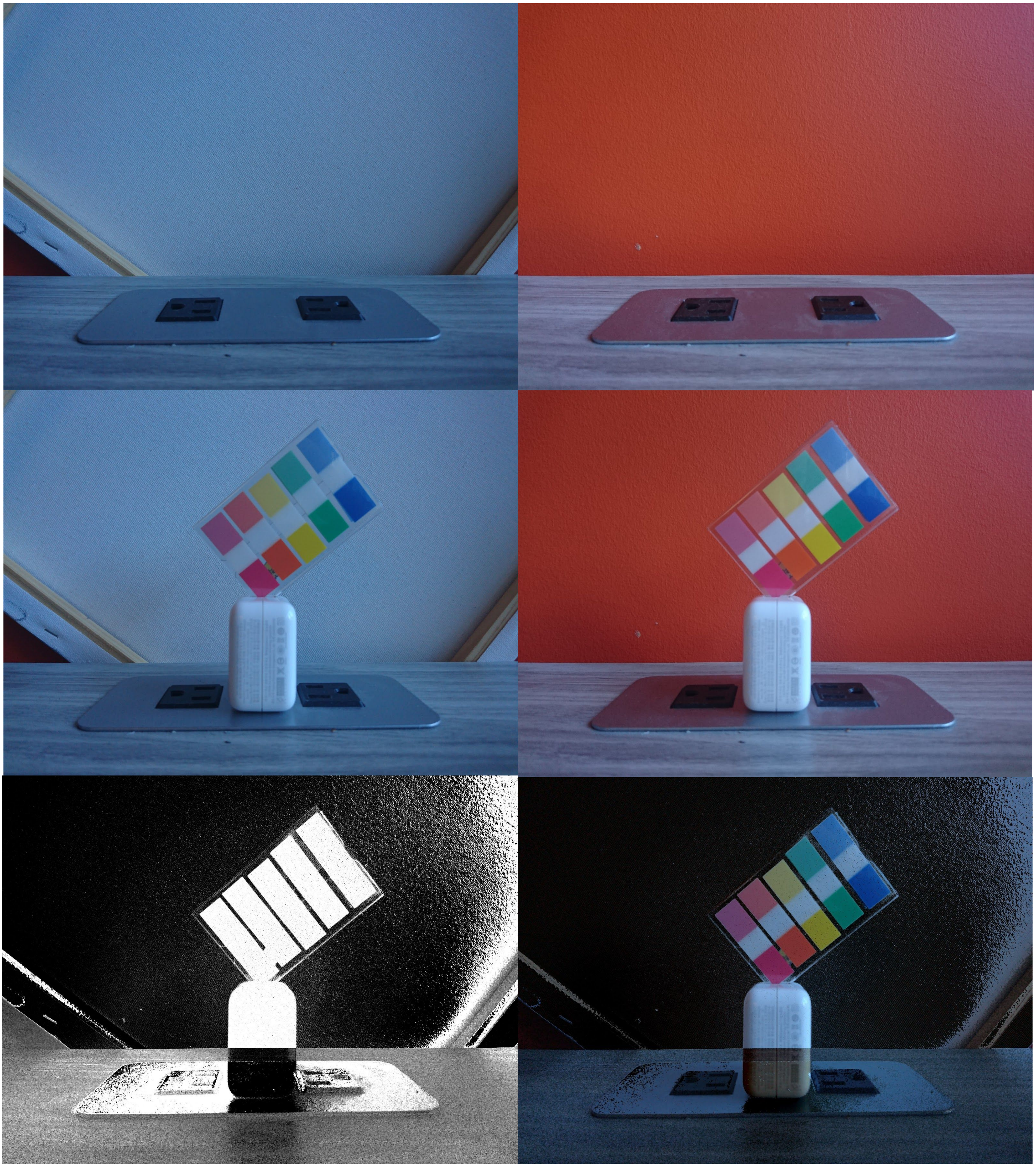
Below are conclusiones that I was able to draw from experimentation.

For this experiment a borrowed a nikon camera from a friend and set the ISO to 120 and the had no lights on.

In an attempt to improve speed the first thing that I tried to do was remove the noise from the image. (I outline my technique in the efficient implementation section) from these results I assumed that dark photos with large amounts of noise would have difficulties matting. Below is 4 images that were grabbed in quick succession in a dark setting. Although the background in textured I would like to note that the char which is not textured has a similar amount of noise confirming my suspicions.

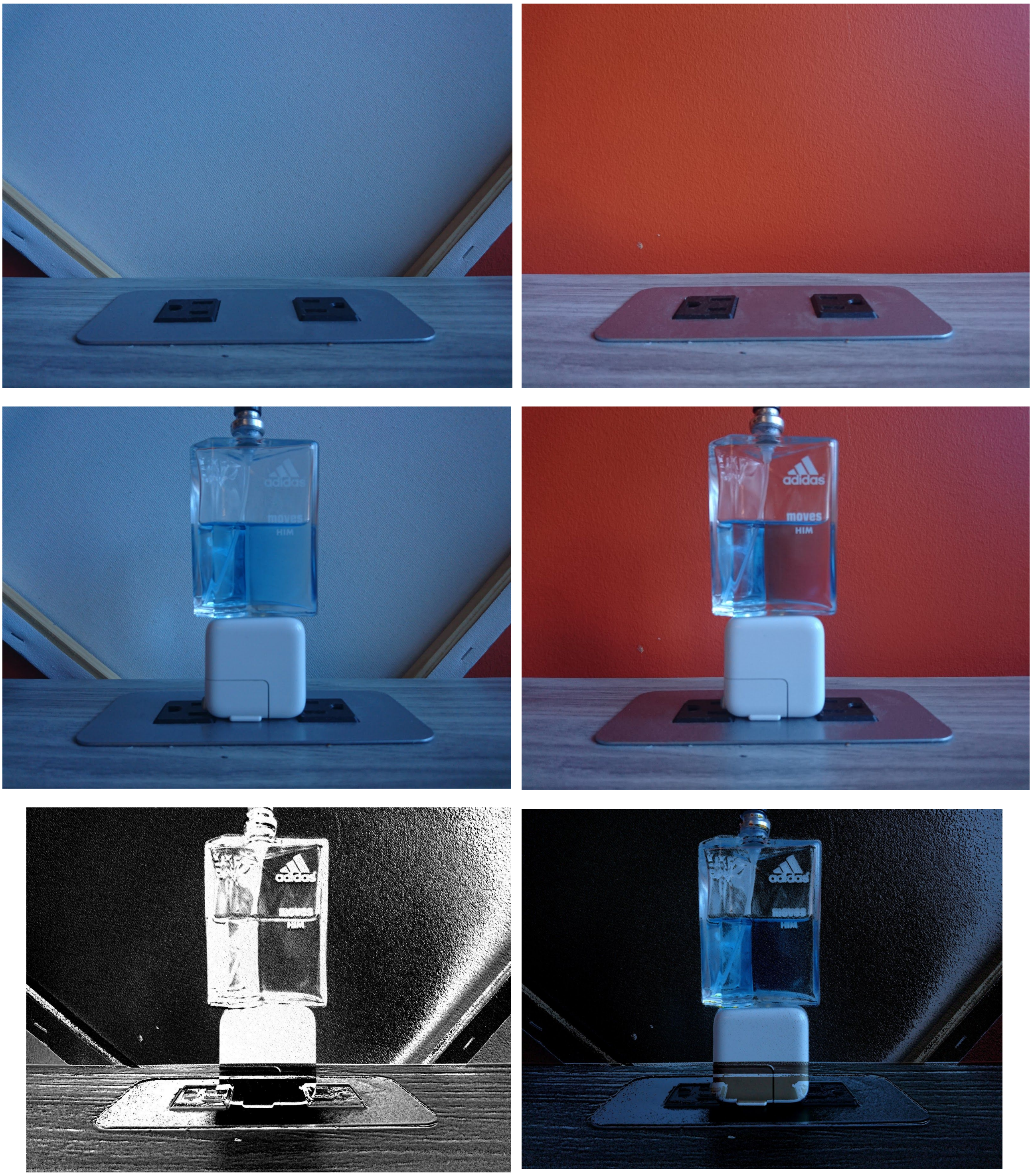


After experimenting with noise I believe that it is important to show with adequate lighting that the matting technique could be done with backgrounds that are flat and relatively uniform.



The phenomenon of the horizontal cut on the white image block is a result of the “backing” in this case the desk being blocked by the object. This is a result of the Cdelta in the matting equation being constant in that area so we don’t have a solution under theorem 3 of Smith and Blinn-SIGGRAPH 1996.pdf. See optimization section for more details.

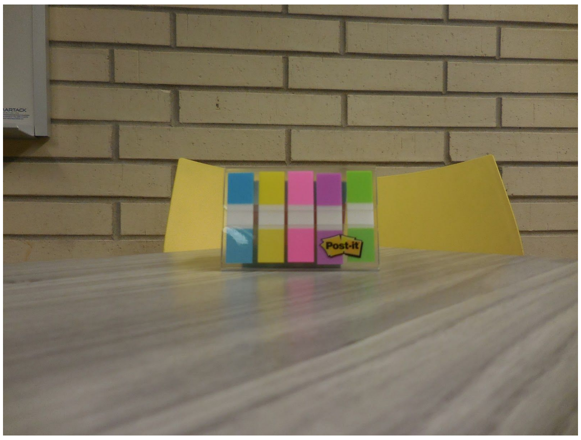
Below we tool a photo of a cologne bottle that has different levels of opacities from differing thickness of the glass bottle

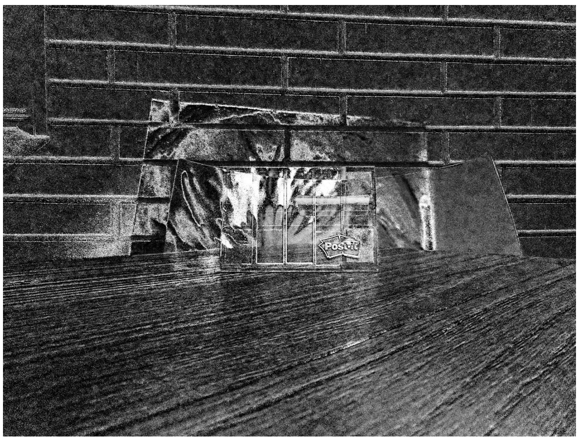


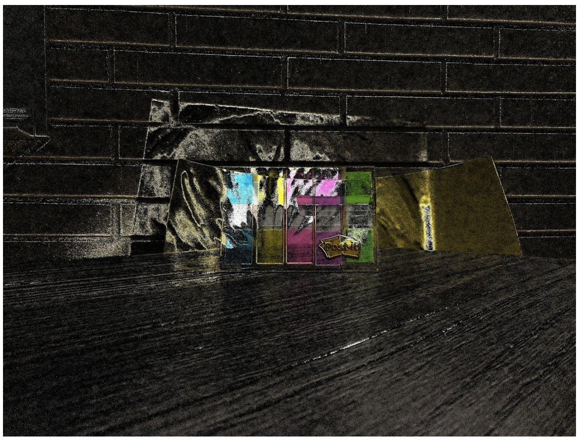
In the alpha image you can clearly see that it thinks that the left side of the image deserves a larger alpha value but for the composite images you can see that it is only refracting the light from behind the bottle. This is because the refraction changes the light shining through the bottle so that the assumption that the transparency of the image pixel wise is false.

All of the above have been using the same backing, the reverse of a canvas and an orange wall. But my first attempt was with a brick wall and painting as my backing the textures show through because of small changes in the focusing from my camera in the following image.

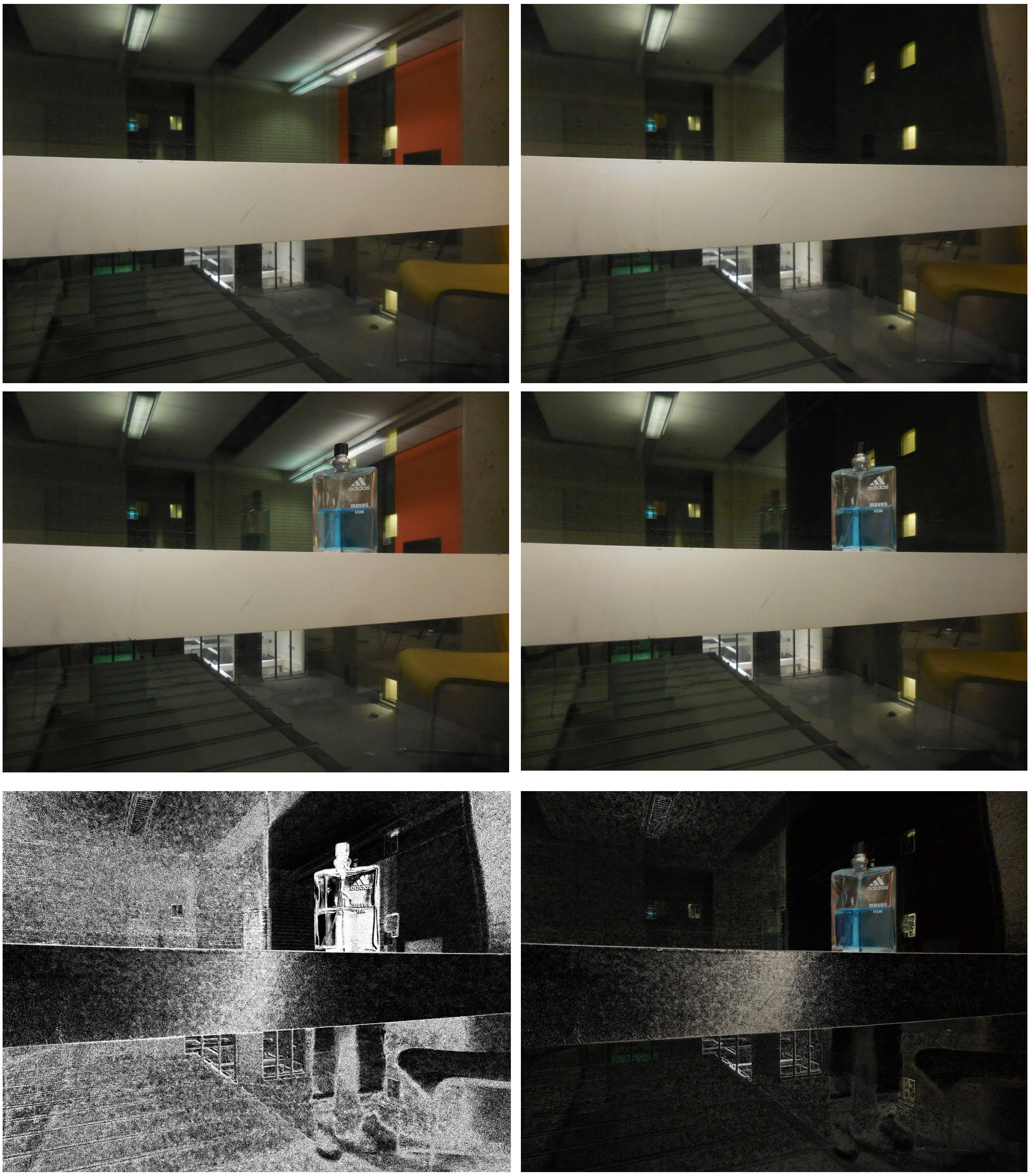








The final interesting result was unconventional. Although I followed all of the steps above the backing was a reflection through a double walled window. Since the window was uniform although the backing was a very textured scarf the backings texture didn't show in the alpha image



The image is able to be well isolated and the backing being so far away does not have much of an effect on the refraction in the bottle using the reflection seems to be a valid way to solve for images where light will refract the back.

Written question

In my third set of images above I showed the the refraction of the light through glass caused the alpha values to be artificially high, this was because the light refracting through the image invalidates the assumption that each pixel is independent of other pixels in the image. The operation that we perform on the vase and flowers is done pixel wise so without this assumption we are blind to the image as a hole.

Efficient Implementation

The code the runs in my submission uses the results of theorem 3 from Smith and Blinn-SIGGRAPH 1996.pdf. In it they outline that the alpha from the matting equation is the sum of the differences of the foreground all divided by the sum of the differences of the background. This is implemented componentwise so the resulting solution runs exceptionally fast for processing of a large image the code takes ~ 0.13 seconds.

This was not the first time reduction that i implemented and i feel that my previous attempt is worth discussing. In my commented out code I used a set constant nosie_level and if the difference of the foreground minus the background for both sets of images was less then the set noise level we would set the alpha for that pixel to zero.

This intuitively has a nice appeal to it. Below i butcher the mathematical intuition behind the concept.
If we are trying to determine

$$1 - \alpha_0$$

Then we have

$$C_\Delta = C_0 - \alpha_0 C_k$$

If $C_\Delta \simeq 0$ then

$$0 \simeq C_0 - \alpha_0 C_k$$

But we are trying to isolate the new image from the background so

$$C_0 = \alpha_0 C_k$$

Is not a good solution to the equation so we have

$$0 \simeq \alpha_0$$

$$0 \simeq C_0$$

Is the desired solution. The only thing that is left fuzzy is what do we mean by $C_\Delta \simeq 0$.
In the given images I found that the most visually appealing solution would be to noise_level = 15. Below on the left is through this noise reduction process the right is the normal process.

Video live Mating
Using polarization movie motivation
So I have come across 2 solutions to this. The first solution that I personally had thought of was to use the blurring of the background image in order to capture the same image at 2 separate focal lengths

Proposed method. You would need a camera that is able to save focus settings and reuse the same settings at a different time.

You take a photo focussing the tip of a person's nose and a second image focusing on the person's ears the 2 photos will have the person's face in focus but the background will be blurred. When the person's face has moved then you would take a second set of photos at the same focal length of the first to get the backing at a later time in the video.

This method has some harsh restrictions.

1. You have to have a camera that is able to focus very accurately to recreate the blur at a later time
2. You need to save a pair of focal lengths for every frame in your video
3. The background can not change drastically from frame to frame.

Hypothesis

If you were to implement this the best backing would be far away and have a texture that at has variation in colour so that different colors emerge from different blurring effects.

In reality I was unable to find any papers on this exact idea. It would probably be to difficult to test in practice but is still an interesting idea.

Idea number 2

Motivation

When you put on older 3D glasses that are polarized you're not able to look at your phone. When I had first noticed this in a 3D movie theater had thought that it was a particularly good application since you shouldn't look at your phone in a movie theater anyways. But the image distortion is different for each eye.

Proposed solution

For each frame of your movie you shoot the frame with 2 cameras and have the backing be a computer monitor. Each camera is given one of the lenses from a pair of 3D glasses. Because your background is uniform you can just use a second set of images that are taken with the cameras before the movie starts with a blank background.

After searching I was able to find papers supporting this idea. I have cited one below

McGuire, Morgan, and Wojciech Matusik. "Real-time triangulation matting using passive polarization." *ACM SIGGRAPH 2006 Sketches*. ACM, 2006.