Part2:

I used the method in A. R. Smith and J. F. Blinn, “Blue Screen Matting,” Proc. ACM SIGGRAPH, pp. 259–268, 1996.
As the theorem 4, alpha can be rewritten directly as

$$1 - \frac{\Sigma_{col}^{[RGB]}(col_{c1} \; - col_{c2})o(col_{b1} - col_{b2})}{\Sigma_{col}^{[RGB]}(col_{b1} - col_{b2})o(col_{b1} - col_{b2})}$$

when o is Hadamard product
Then it would be much quicker because it uses only entry wise product and the total time consumption would be O(mn) only.

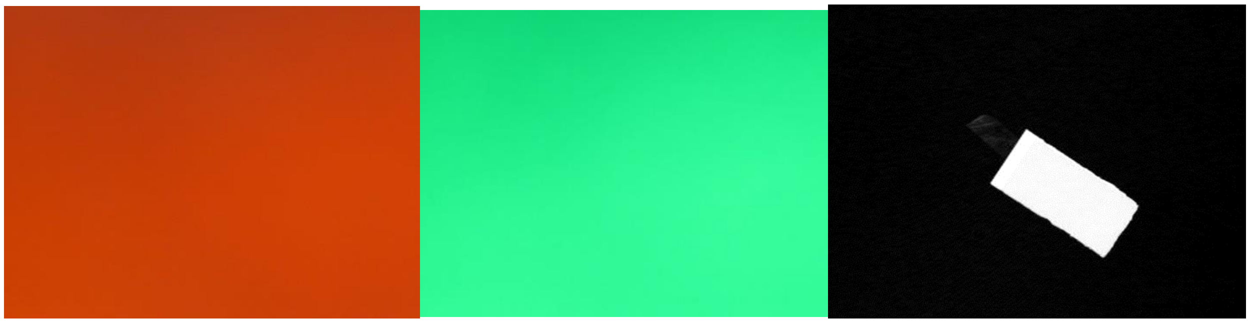
Part3:

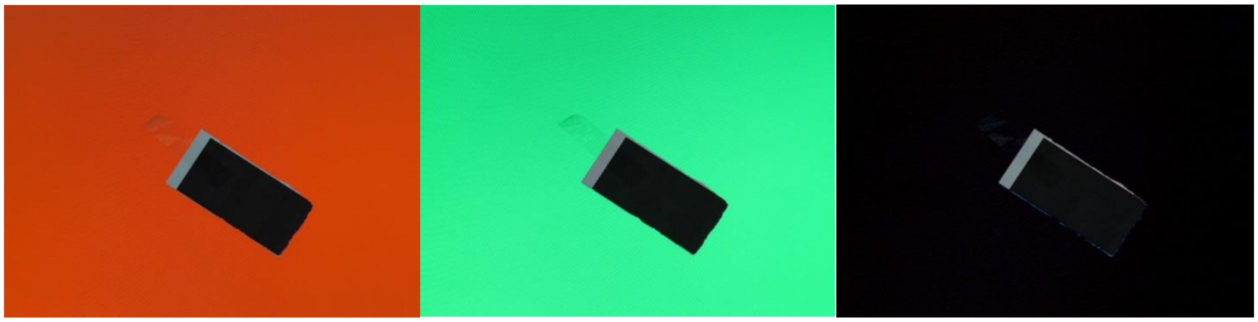
The triangulation matting method is a good way to solve the generalized problem on color matting. However, it also has some limits.

In this experimental evaluation I used my Canon SX50Hs camera. Commonly I would use 1/50s exposure time, 14mm focus length. The photo would be 640*480 in 180 dpi
special camera settings for one picture will be shown. Most of the background would be the picture on cdf computer screens. I supported the computer screen, the object and the camera with a fixed position and angle. When I tried to take the pictures, 1) take the back1 2) place the object and take the comp1 3) change the background on computer to take comp2 4) remove the object to take back2. In this case the object will not be moved between comp1 and comp2, which maximizes the accuracy of the result

This would be an example:

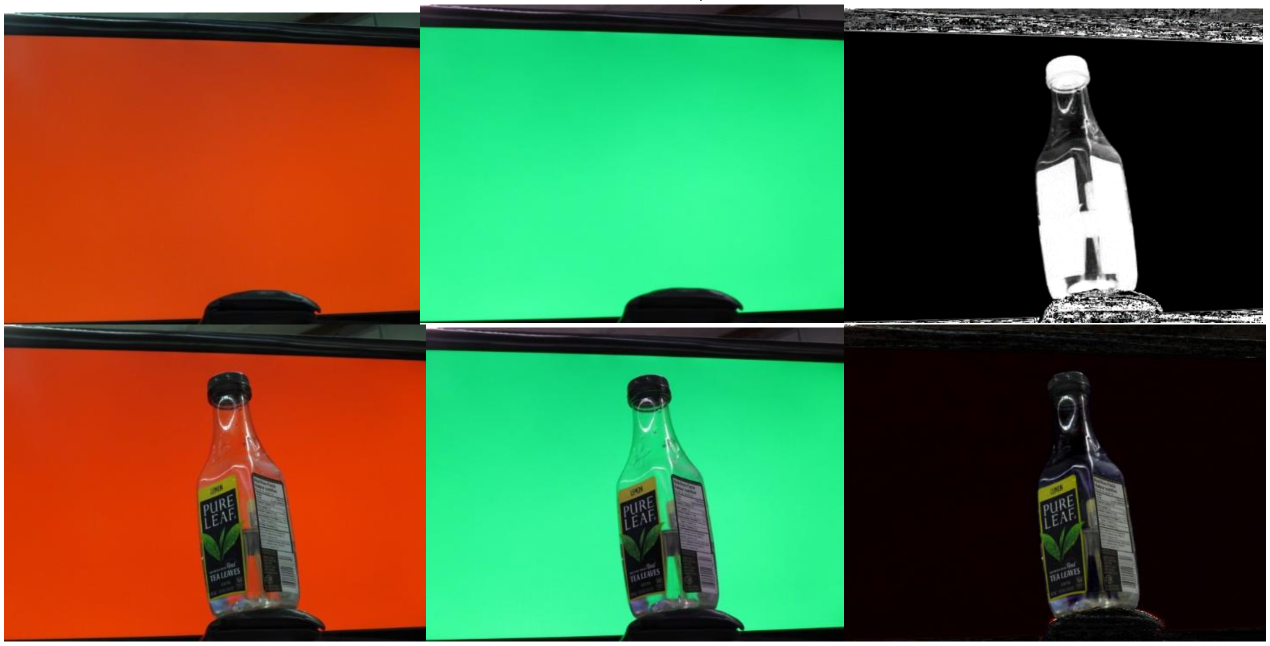
It's clear to see it's the black paper stuck on the green or red screen. The sticker is identified clearly in both alpha and col pictures. However, there is a small detail: the transparent tape is very fuzzy on the color file and it's not activated in alpha file. It's easy to see that the triangulation might not be very sensitive to the transparent objects.



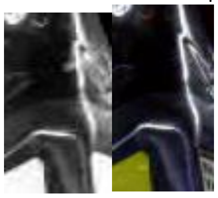


Problem1: transparency

Another object is a water bottle. In the example below, the trademark "PURELEAF" is clear in both alpha and col pictures. However, the bottleneck is transparent, or semi-transparent. As a result, the alpha values on bottleneck is not activated well. In some circumstance, it would be identified.

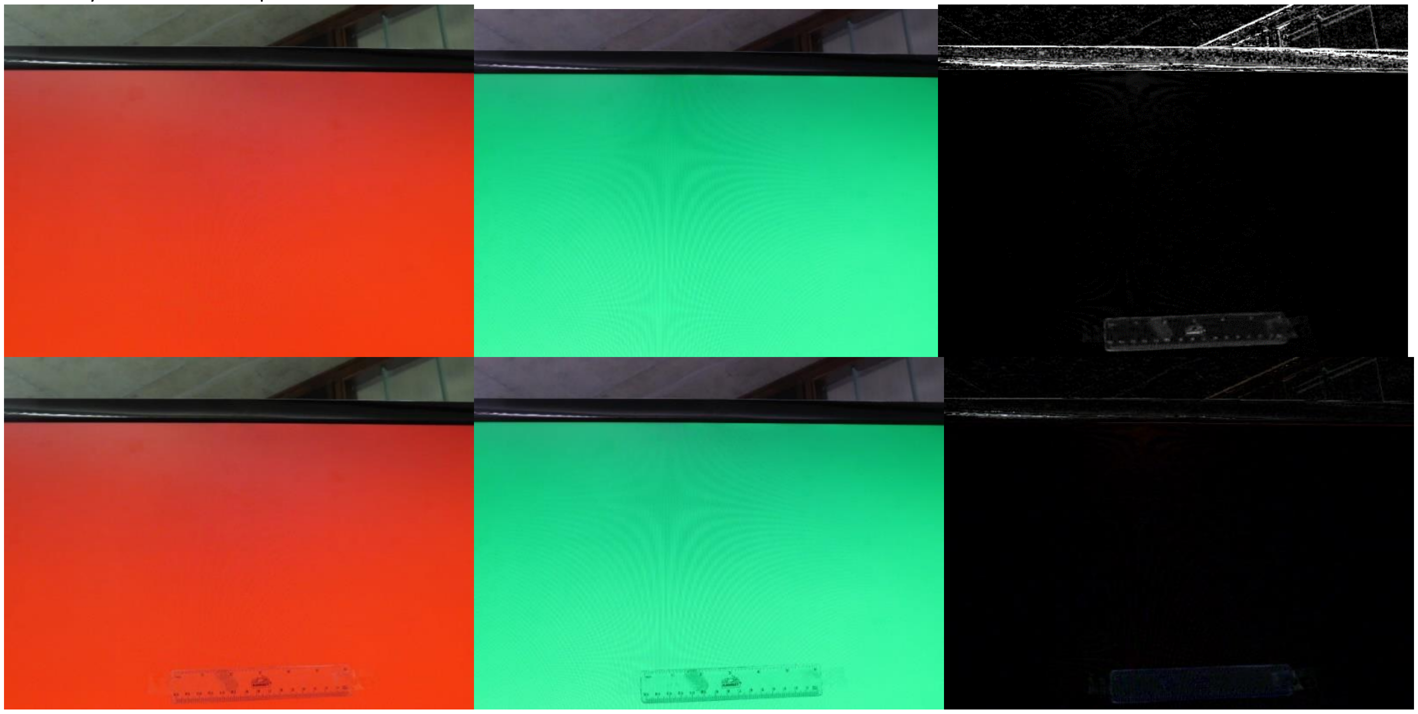


Especially for this part on the bottleneck. Is this really identified as part of the neck? Or some part of it would be treated as the background? It's clear to see as a human, as we will consider object continuously in pixels as a bottle. The computer doesn't.



Alpha col

Another example is a transparent ruler with sticker. From the pictures on comp1 and comp2, we can see the ruler at the bottom of the screen. In alpha picture it's even weaker than the top background and it's not able to identify the ruler in picture col.

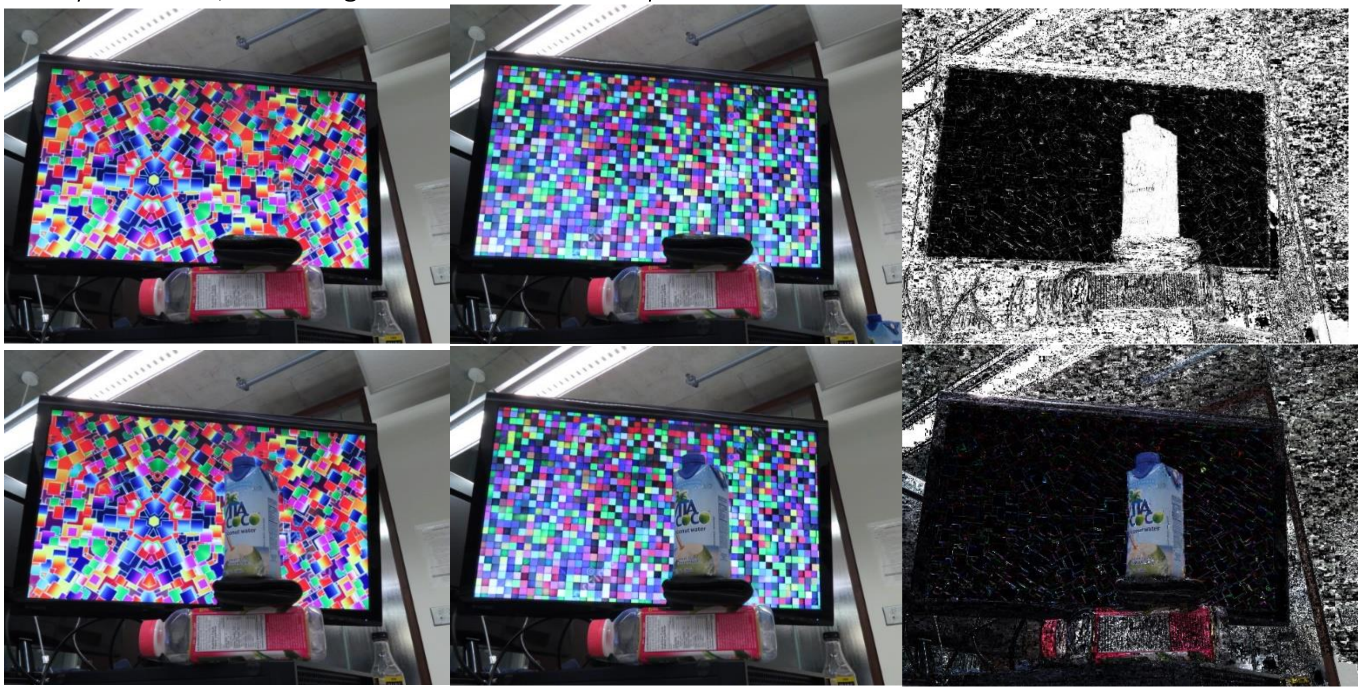


In fact, the transparent object has part of the color information in the background. Actual color = coefficient * material color + (1-coefficient) * background color.
The ruler has much smaller coefficient, then the actual color would be much closer to the background color. Thus, it would be harder to identify.

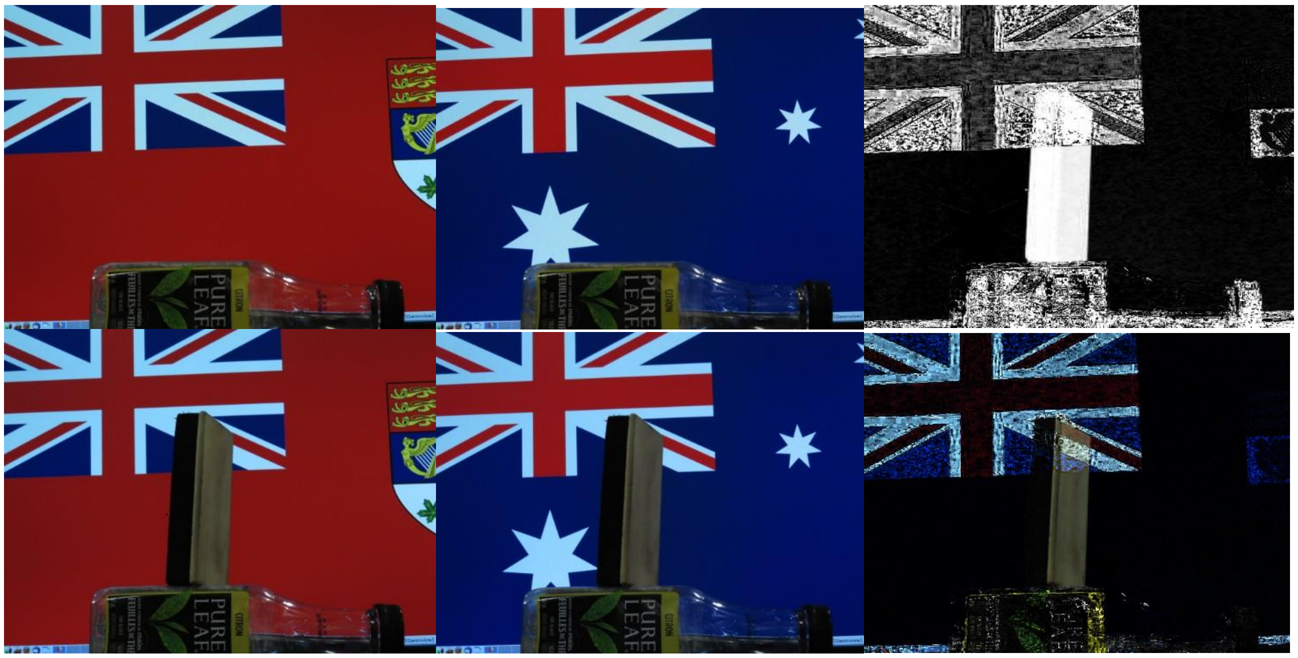
Problem 2: background similarity

When the two backgrounds are similar, the similar part would be easier to have higher alpha, thus it would be failed to identify the foreground. Sometimes the alpha value for the background would be even higher than the foreground!

For example, this set of pictures. The object is a coconut juice bottle and the background should be different mosaic. However, the camera is zoomed out so part of the computer edges, the ceiling, the wall, the support object are also included in the image. From the col and alpha, we still can see the coconut juice bottle very clearly. However, the background in col is also very clear!



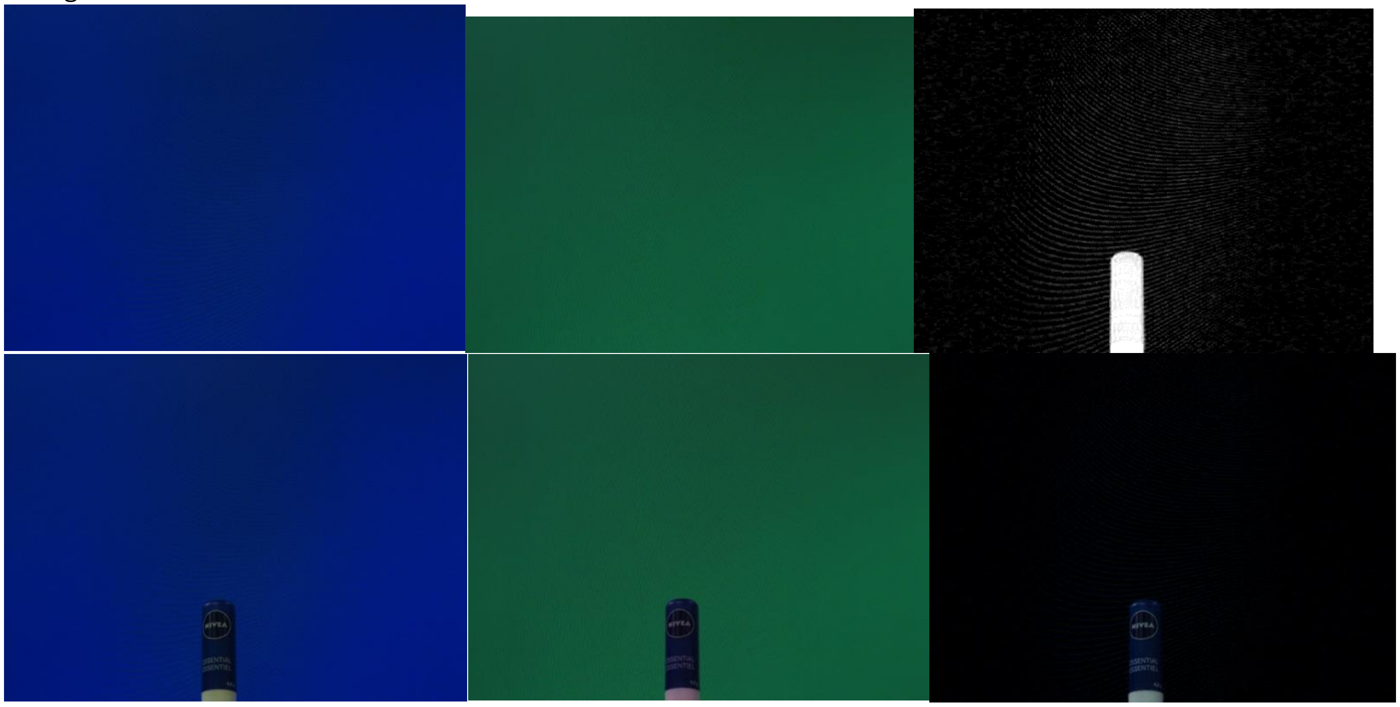
Another set of pictures. The backgrounds are Australia national flag and Canadian Red Ensign flag and the object is a whiteboard eraser.
There is a British flag at the top left corner at both backgrounds and a PURELEAF bottle at the bottom. In alpha and col we can see the eraser is identified clearly. However, the British flag and the bottle are also identified. Although they are not as clear as the eraser, it's still able to identify them!



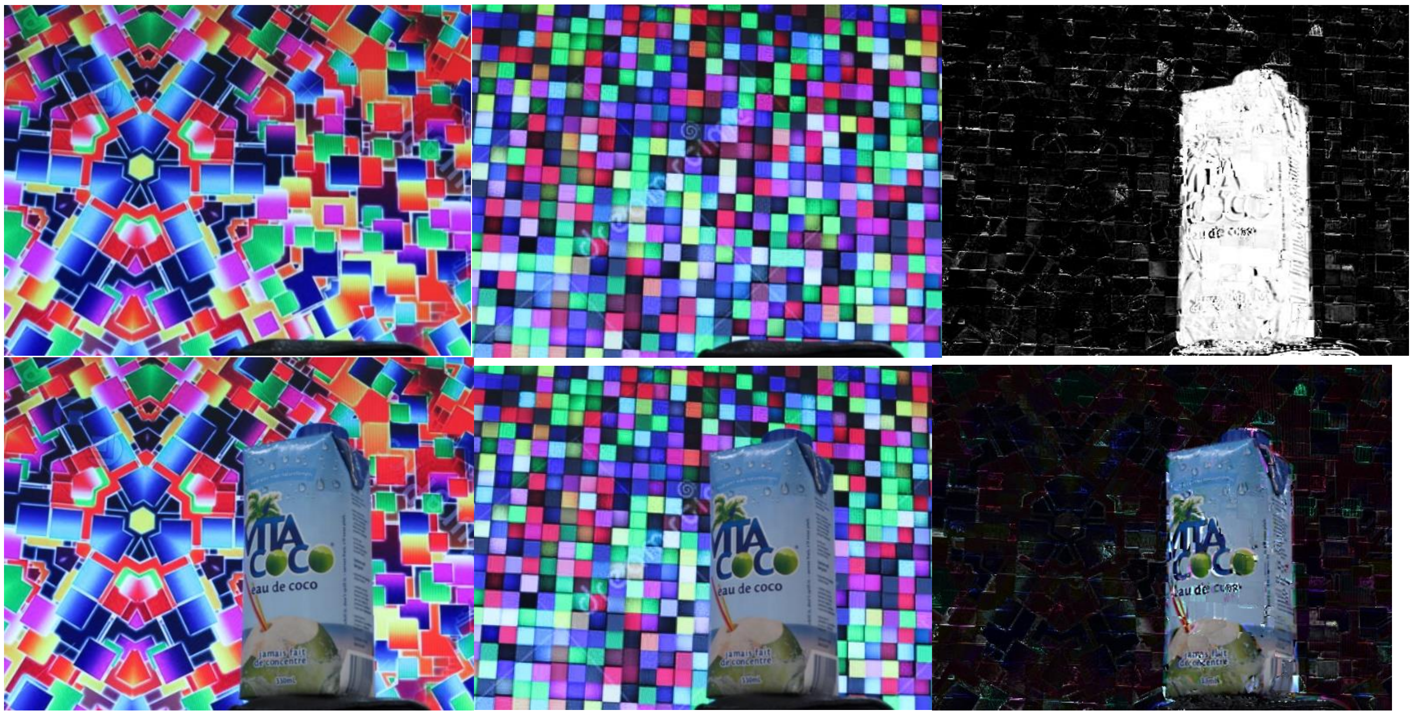
In the formula of alpha in “Blue Screen Matting,” (A. R. Smith and J. F. Blinn, 1996), both differences of composite images and background images are tending to zero. Sometimes it may happen that the alpha is closer to zero and sometimes it is closer to one. It's not like the object, whose all of the alpha values are closer to one. However, even part of the values do like this, then it's able to show a semi-transparent image on the col picture.

3 color between object and background

Surprisingly, even the colors between the object and the background are close, it's still easy to identify the object.
The background of the following set of images are blue and green, when the object is my lip balm.
We can see that the alpha values at the lip balm are high within expectation. However, no response for the background.



This set of photos is still about the coconut juice bottle and the background are still mosaic. The only difference is that the camera is zoomed in such that the background photo fill the whole photo. It seems that the algorithm is confused a little bit by the mosaic background, however, the bottle is still identified. The messy alpha at the background is more related to the color on mosaic. It's possible that the two mosaic pictures have similar color at some specific positions, for example, the white border between boxes. Hence some of the values would be activated.



Above all, the similarity between object and the background could be ignored, or much smaller than expectation. The reason is that there are two different backgrounds. If both two backgrounds have the similar color, then it may raise the alpha to one, but the reason would be the similarity between backgrounds but not with the object anymore!

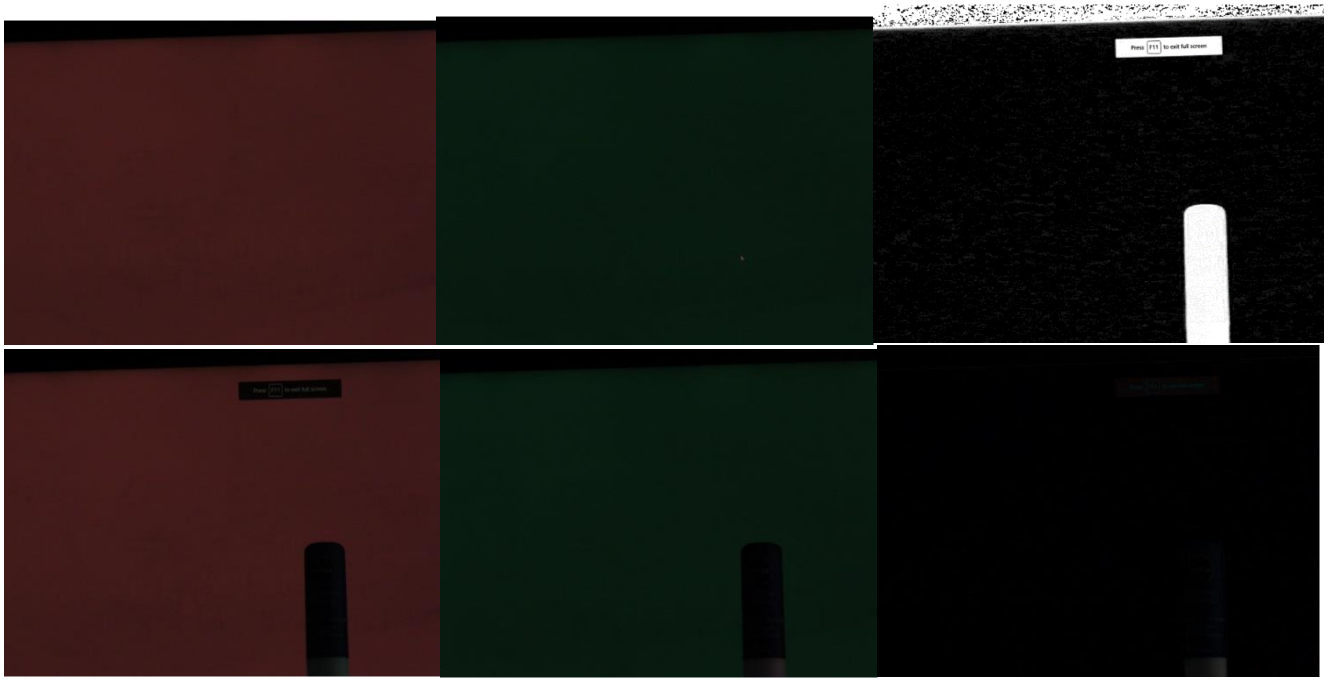
4 lightness

What happened if it is very dark?

The set of photos have the backgrounds in red and green. The object is the lip balm used above. However, the whole circumstance is very dark. (Don't care about the words in red_comp and alpha. That's a typo.)

Interestingly, the object is still very clear. Although it's almost impossible to find it on color image as it's too dark to see, but it is still found on alpha. This is also a limit: you can still find the object, but the color information is lost. Nobody can find the lip balm is blue in this circumstance.

It's also important to prevent unexpected errors. In this example, the full screen message was popped up in red comp image and it is saved directly into the alpha frame. As the formula of alpha, c_1 at the message bar is much smaller (as black is smaller number) than expected, then the alpha is much bigger. That's why the message bar is activated on alpha.



Above all, the lightness will not confuse the algorithm. The object is still able to identify. However, it's hard to keep the origin object color as the lack of light. Theoretically, it's a camera problem, but not a matting one.

Above all,

I have talked about some limits related to triangulation matting.

Transparency would make it harder to detect or split from the background.
The background similarity will add huge noise to the alpha to make you hard identify.
The influence between object and background can be almost ignored.
The lightness will not affect the matting to identify the object, except it's too dark to contain any color information. However, the object doesn't have most of its color information in real life due to the lack of the light.

Part4:

Although the second cloth is black, it's not the same black everywhere. The white background is white almost everywhere but the second one doesn't. If you open the two pictures flowers-backB.jpg and flowers-compB.jpg, the second picture is bluer than the first, while the background A keeps the same.

Let's analysis the pixel value at these two points: 2 means [810, 40]; first letter means background/comp; second letter means background A/B.

```
>>> ba = np.array([0.56078431,0.5254902,0.51372549])
>>> bb = np.array([0.08235294,0.09411765,0.08627451])
>>> ca = np.array([0.58039216,0.54117647,0.51372549])
>>> cb = np.array([0.11372549,0.09803922,0.07843137])
>>> ba2 = np.array([0.54117647,0.50588235,0.49019608])
>>> bb2 = np.array([0.03921569,0.07843137,0.17254902])
>>> ca2 = np.array([0.49803922,0.46666667,0.46666667])
>>> cb2 = np.array([0.0627451,0.09019608,0.16862745])
```

These two points at the left side doesn't like black points but closer to blue points. More seriously, ca2 is a little bit darker than ca(the reason might be the shadow on the background), and cb2 is lighter than cb. Thus, ca2-cb2 = [0.43529412, 0.37647059, 0.29803922] which is much smaller than expected, compared with ba2-bb2.
According to the formula thm4 in Blue Screen Matting(A. R. Smith and J. F. Blinn), the alpha is much bigger at [810,40]