

CS5310_assignment1

Chenxi Qu

May 2024

1 Do something interesting to the image

```
1 ../bin/lab1 geraniums.ppm lab1.ppm
```

First, I changed the order of green and red to try to output only the leaves, but there was some noise. To fix this issue, two ideas came to mind:

1. Blur the original image first, then perform the color separation.
2. Use a method similar to shadow mapping's Percentage-Closer Filtering (PCF), which reduces noise by sampling a circle of surrounding kernels to compute weights when each point is sampled.

I can do this change in the last section, so leave this method right here.

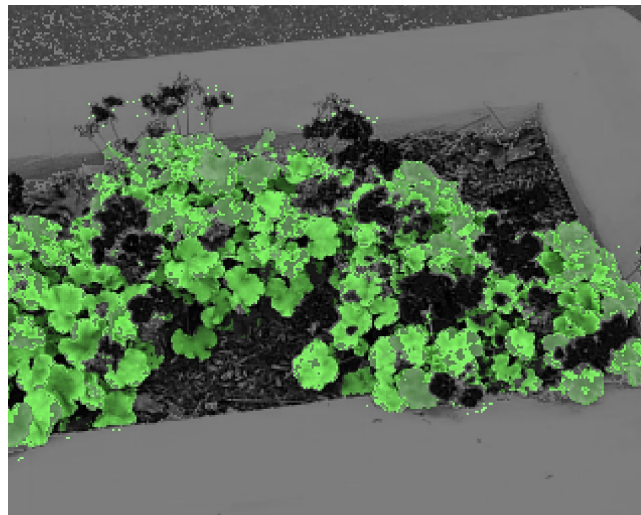


Figure 1: Brief output of green leaves

2 Implement green/blue screen compositing

Try to mask the blue and green channel, and try to combine it with the origin background, here is what I got.

```
1 ../bin/mask powerpuff.ppm mask.ppm
```

Trying to use offset dx and dy now.

Two problems:

1. The front image is so big, trying to add a scale funtion before combine.
2. The eyes are missing, because the current blue and green channel blocks much more than except.

2.1 add scale

A new scale function has been added, which now scale the size of the image, the command is below.

```
1 ../bin/combine powerpuff.ppm geraniums.ppm mask.ppm 20 -20 0.5 output.ppm
```



Figure 2: first temp output of combine the graphics

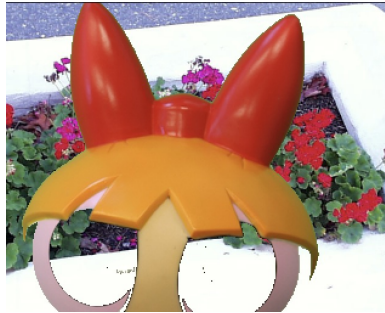


Figure 3: add offset to the combined graph

2.2 updated mask algorithm

After studying the image carefully, the eyes are composed of black and white, so I changed the algorithm so that if the color of the rgb is black, or pure white, it always counts as white and doesn't count as part of the background. Now the result seems better.

```
1 ../bin/mask powerpuff.ppm mask.ppm
```

2.3 add blur background

It feels like if you can blur the background, the result should be cool. My algorithm just adds a kernel just like the pcf in shadowmap and samples around the circle to average the blur. And below is my final result

3 Conclusion

I added a couple of extra algorithms as below:

1. modifying the scale algorithm to distinguish not just green, but also black and white.
2. scale the size of the image instead of just the offset
3. blur the background.

4 In the future

Furthermore, maybe in the future I would do:

1. Jaggedness removal between characters and background
2. The bottom right corner of the eye is still partially missing, and I'd like to fix this by sampling different eye colors separately and finally combine together instead of one.



Figure 4: add scale to the combined graph



Figure 5: change the mask function and recalculate the result



Figure 6: final result