CS3150

Professor Feng Jiang

Xiaosong Wang

Project#1

This is a very challenging project. I had a hard time and tried a lot of method on how to select the useful pixels in cut1 and merge it into cut2. Then I found there is video about that. The Team record video really helps me in successfully implement the goal.

First import the packages I need for this project.

Text

Description automatically generated

Then used the code provided in the instruction to cutoff the head and neck region which will be used in this program.

A person posing for the camera

Description automatically generatedA person posing for the camera

Description automatically generated

Since the cut1 and cut2 have different size, we need to resize them so that I can find the useful pixels in cut1 on the mask position in cut2. If I didn’t resize the images, the error message occurs.

Text

Description automatically generated

Error message if didn’t do the resize.



Then apply skin\_rgb\_threshold method to both images. Since two image are both in pretty good conditions and qualities, directly applying the methods can detect the skin well.

Text

Description automatically generated

A person with collar shirt

Description automatically generatedA person with collar shirt

Description automatically generated

Our purpose is to find the pixels in cut1 which is located in the mask position in cut2, so I did the subtraction between to image. After subtracting out the common part. The left part should be mask part.

Text

Description automatically generated

A picture containing chart

Description automatically generatedA picture containing chart

Description automatically generated

I’ve tried several different thresholds to find the best mask image such as 35 and 45, and found out 40 is the best one.

A picture containing chart

Description automatically generatedA picture containing chart

Description automatically generated

However, after my testing, the thresholds didn’t influence a lot. Because later opening and closing can also fix the problem.

Next, applying opening and closing to the mask for removing the outer noise (the hair differences between two images) and the holes in the mask area.

Text

Description automatically generatedA picture containing chart

Description automatically generated

I’ve also tried several kernels such as 6 by 6 and 8 by 8, and found out 7 by 7 is the best one.

A picture containing chart

Description automatically generatedChart

Description automatically generated

After I found the area which we need the pixels in that area in cut1, I found that there is an obvious edge between the merged image which is what I don’t want.

Text

Description automatically generatedA person posing for the camera

Description automatically generated

If I do dilation to the mask and makes the mask part a little bigger, I could pick more pixels in cut1, so that it will fill in the edge area. After my testing, applying twice dilation will generate the best result (still can see the edge if applying only once dilation).

A person smiling for the camera

Description automatically generated

I’ve also test erode the mask also just for curious, and obviously the result is worse.

A person posing for the camera

Description automatically generated

Lastly, after getting the pretty good merged image, I applied a Gaussian filter trying to smooth the image a little bit.

Text

Description automatically generatedA picture containing application

Description automatically generated

This is all I got for the project. It’s a challenging project definitely, especially in how to pick the useful pixels and merge two image together. The video helps a lot.

Also, I spent a lot time in finding the threshold which can remove the outer boundary in the difference image of cut1 and cut2, such as trying to apply different filters to it. However, sadly I didn’t successfully implement that. Thus, I used the easier way which is defining the best threshold after examining different ones.

Overall, it’s a very interesting project and I learnt a lot from this.