

Computer vision course

Lab 5 - Segmentation

Pietrobon Andrea

27 maggio 2023

1 | Task 1

This first task was not complicated in the selection of the type of cluster to be used or in its implementation but it was very long in the selection of the image preprocessing methods and in the selection of the correct values. In the end I chose to use a mix of both before and after segmentation to remove the noise that plagued the second and third images. I used Gaussian blur, Erosion and Dilation also increasing the brightness of the image to remove the darkest areas. As the main method instead I used Otsu's thresholding. After numerous unsuccessful attempts to try the different values I implemented sliders as in the first task of lab 4 to find the correct parameters and then insert them in the code. The Figure 1.1 is the one that gave the best result while the other 2 (Figure 1.3 and Figure 1.5 still kept some noise).



Figura 1.1 First image before segmentation.



Figura 1.2 First image after segmentation.



Figura 1.3 Second image before segmentation.



Figura 1.4 Second image after segmentation.



Figura 1.5 Third image before segmentation.

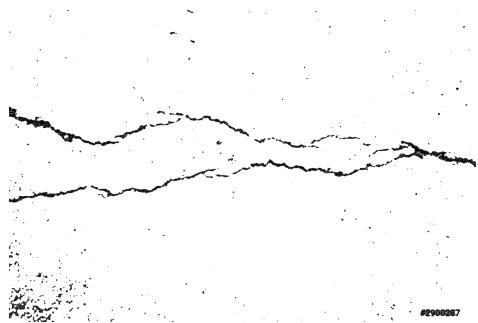


Figura 1.6 Third image after segmentation.

2 | Task 2

I spent the majority of the time to improve the kmeans performance as much as possible so as to effectively classify the image. Finally I also used cv::erode to "delete" useless elements in the image. Unfortunately I couldn't overdo it otherwise I would have missed too many items. In this case a small portion of a box was misclassified and I couldn't fix it without increasing the erosion too much. (However, a possible solution could be to use cv::Houghlines to delimit the line of the roads so as to fix the missclassified elements before this line with a for loop.) As can be seen in Figure 2.2, the image without erosion it has many badly classified elements. While in Figure 2.3 it is possible to see the elements classified correctly with the exception of the box previously described.



Figura 2.1 Original image.



Figura 2.2 Image without cv::erode.

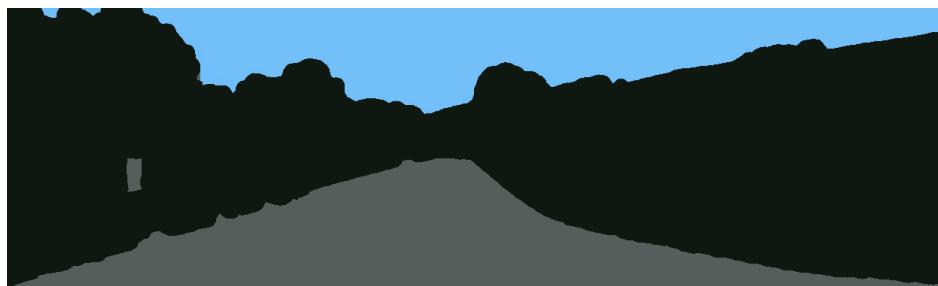


Figura 2.3 Final image with cv::erode.

3 | Task 3

I tried Otsu's thresholding and k-means in different variations but unfortunately none of them worked. The shirts were always clustered incorrectly. In the end I opted for a simple method, that is through the color range. I have selected the color range of the shirts. Then I looked for the circles in the image so as to find the ball which, having a similar color to the shirts, ended up in the main mask. Found the ball with HoughCircles I built a mask around it to remove it from the elements of my interest. In the end I then used cv::bitwise and to remove everything that is not of interest to me from the image and return the final image with only the shirts. In the images below (Figure 3.1 and Figure 3.2) you can see the starting image and the final result. Note: I tried to use cv::bitwise but when I compiled with cmake the image came out differently so I used a double for loop.



Figura 3.1 Original image.

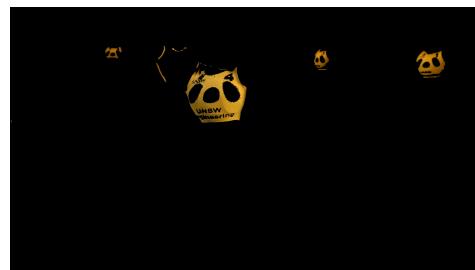


Figura 3.2 Final Image.