

Report

October 27, 2019

1 CMSC426, HW03, Shiyuan Duan, Report

1.1 Introduction

In this project, we are implementing a color segmentation method by clustering color points in a RGB color space. The idea is that we fit a gaussian model or a GMM to the desired color, orange in this case, that is represented as a 3D point on a RGB color space. Then we use the trained GMM on the test-image to find the pixels that fits the GMM the most. Finally we will be looking for the distance from the object to the camera. There are several ways of doing it, in our case we will be using the object's area.

1.2 Obtaining training data

We are obtaining our training dataset by cropping the pixels that represent the orange ball and collect them into a csv file(ball.csv). This can be done by executing crop.py

1.3 Single Gaussian Model

When it comes to problems like clustering data points, it is usually done by gaussian models. The simplest model is a single gaussian model on a 3D color space.

However, in this case, the data points are not clustered in a way that the entire dataset can be well described by a single gaussian model. Even though most of the data points are captured but there is still a significant amount of data points that are not captured by the single Gaussian model. In order to strengthen our algorithm, we have to use a GMM to describe our data set

1.4 Gaussian Mixture Model

We use the code from HW02 to construct a gaussian mixture model to describe our dataset. The initialization methods are the same and discussed in the code and the only difference is that instead of using 3 gaussian models we will be using 7 gaussian models. With 7 gaussian models we can successfully describe the entire dataset with our GMM. However, there are limitations to this methods. One of the most notable limitation is that, since we are representing each pixel on a RGB color space and cluster them, our algorithm is only sensitive to colors.

1.5 Estimating distance

Since we have a GMM that describes the color of the orange ball, what we can do is to set a cut-off and suppress every pixel that has a low p-value according to our model. The idea is that we want to make every pixel that's not on the ball black and make the ball white. Then we can calculate the

area of our ball. Given that there's a quadratic relationship between the area and the distance, we can fit our training set onto a linear regression model and estimate distance in the test set.

1.6 Conclusion and limitations

We can see with a gaussian mixture model we can successfully cluster colors. This algorithm is really strong however it also has some limitations. For example one of the limitations is that it is really sensitive to color. The object must have distinctively different color from the background color. Also my test result shows that, if the image is really blurry, this algorithm fails to capture its true area because only the part in the middle is captured and the boundary is blurry so it is not included in the area. This will cause our algorithm fail to capture the object even though it is in the image. This is what happened to test image 2. Even I could capture the orange ball in test image 3, I falsely identified its distance because only the center part of the ball is captured so the algorithm will think that the ball is small so it is far away from the camera.