CMSC426 Project 1

Pavan Ravindra, Sabrina Zhou, Gaurav Guglani

1 Single Gaussian

Detecting the ball with the single gaussian method was fairly easy to write up. The results were also actually surprisingly good, considering that only a single gaussian is used to model the probability for the pixel colors in the training set.

Really, the only non-straightforward part of building the single gaussian model was the selection of the prior for the orange ball. Initially, we were planning on adding another cluster for "not ball," but then we realized that there isn't really a single gaussian that can be constructed to model all of the possible colors in the background. Modelling the colors of the ground, wall, chairs, etc. with a single gaussian isn't feasible, but perhaps a GMM (more on this below) could be used to model the background colors such that at least one different gaussian models each of the different sources of variation in background color.

Now, back to selection of a prior. Since a TA mentioned on Piazza that we could use a threshold for single gaussian and we decided to use a single color cluster, calculating and comparing the posterior probability to our threshold became very easy. Below, we have a derivation for all of the variables that we were able to combine into our "threshold".

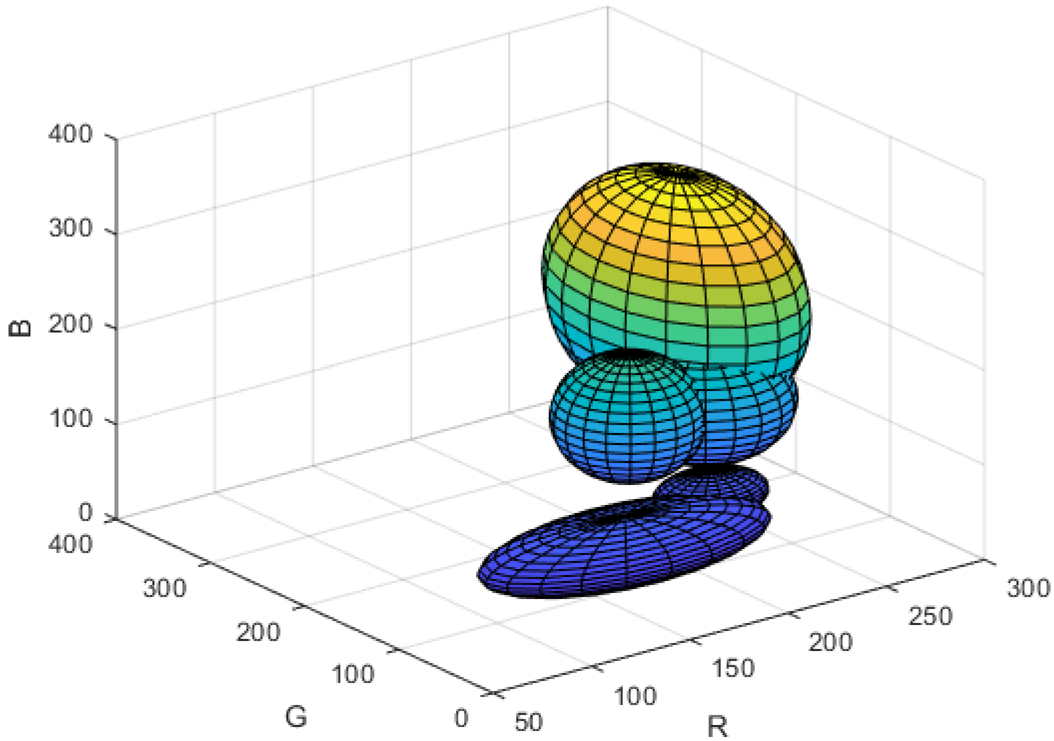
Based on what we show above, we could just choose a single threshold, tau, to use as our threshold.

2 GMM

We chose to use RGB-space, ran randomized initialization as specified in the module, and we used k = 5 for our number of Gaussians. The GMM algorithm is stronger because it allows for multiple gaussians to describe the colors present in the training set of images for the orange ball. There are natural variances in the RGB color values of the ball, even within a single image. In the single gaussian model, these variances have to be accounted for by a single gaussian alone. This means that the single covariance matrix for this gaussian has to find a balance between mapping too high of a probability to colors that are not in the original training set and mapping too low of a probability to certain colors that are in the original training set. In the GMM model however, we can have multiple gaussians (as the name implies), and each of these gaussians has its own mean and covariance matrix. Even for values that are not very common in the training set (consider the part of the ball that is reflecting light from the lightbulb in the room, which is a very small part of the set, but it should still be captured by the model), these rare colors can be weighted down by the pi value for that particular gaussian.

Based on the above reasoning, the GMM algorithm is stronger than the single gaussian algorithm, since it better accounts for variation in the original data set. Even if the original dataset is truly sampled from a single gaussian, the optimization of the pi weights of each gaussian can allow the GMM to essentially perform like a single gaussian model.

Below is a plot of the Guassian ellipsoids from our GMM model (found in GMM.mat).



3 Calculating Distance

The training data followed an inverse square model, which we were able to generate a model for using the Matlab fit command. The input value was the area of the ball in the image (in pixels), and the output value was the distance of the ball from the camera.

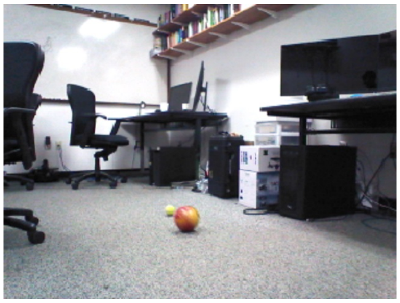
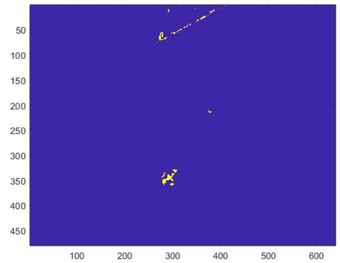
4 Results

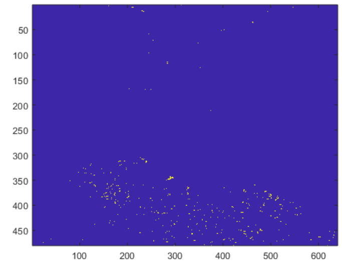
In this section, for each test image, we provide the provided test image, the posterior probability map for single gaussian, and the posterior probability map for the GMM algorithm (in that order).

An important note to be made about the probability maps: we incorporated the threshold into our map, so the points that light up as yellow are the points that were selected to be above the cutoff threshold that we decided for both the single gaussian and GMM. This allows us to make more informed comments in the discussion of our results about how the posterior probability maps actually affect the results. After the posterior probability maps were calculated, regionprops was used to select the largest present centroid in the image, and the area of that image was used to calculate the distance, as mentioned above.

The predicted distance for single gaussian and GMM are below the images (again, in that order).

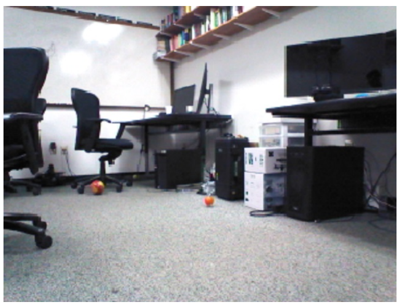
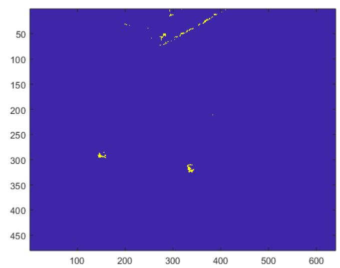
4.1 Image 1

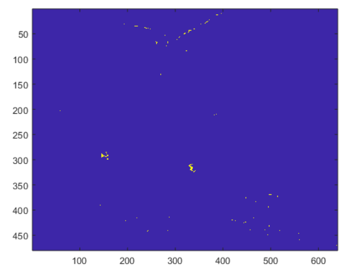





186 (single gaussian), 227 (GMM)

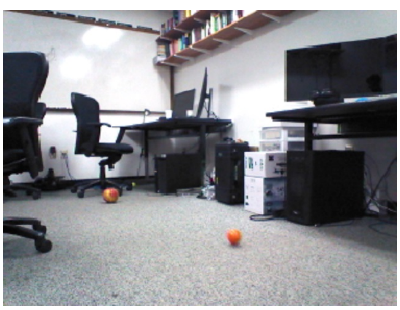
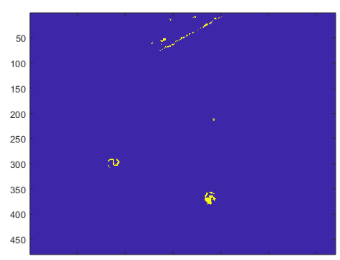
4.2 Image 2

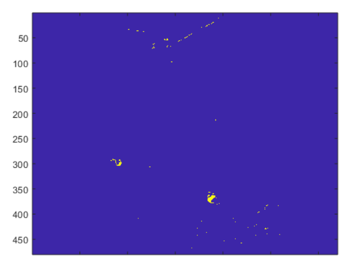





260 (single gaussian), 261 (GMM)

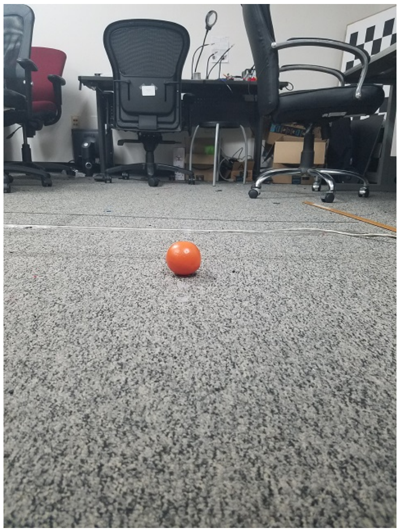
4.3 Image 3

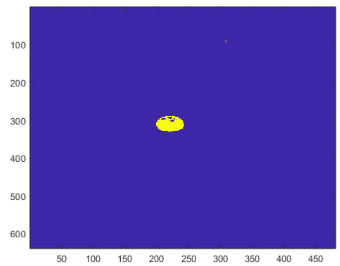



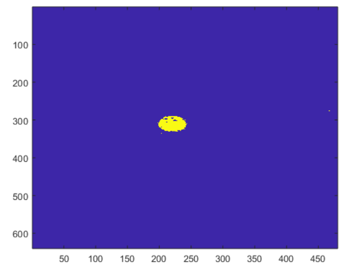


176 (single gaussian), 225 (GMM)

4.4 Image 4

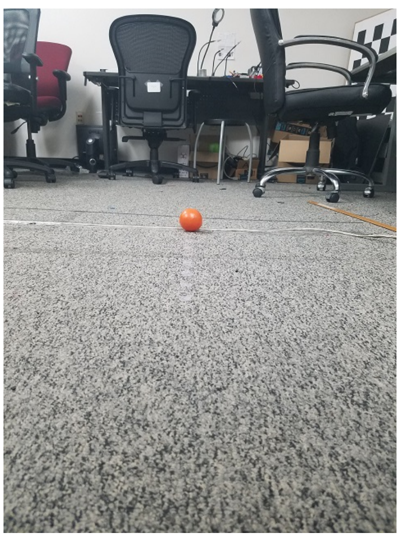


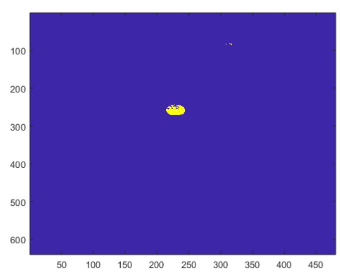


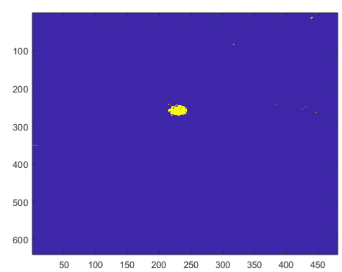


87 (single gaussian), 88 (GMM)

4.5 Image 5

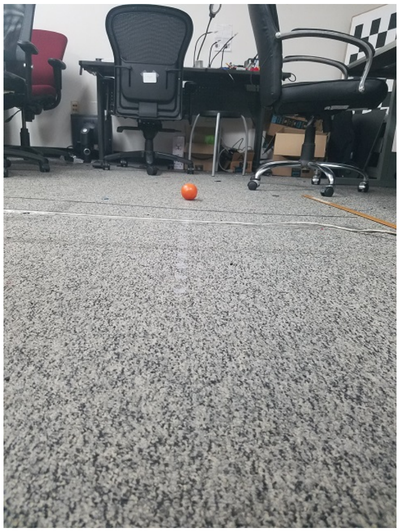


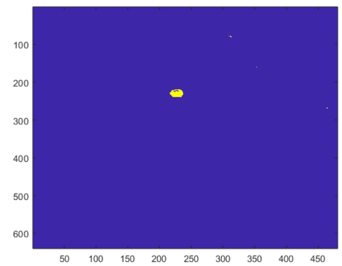


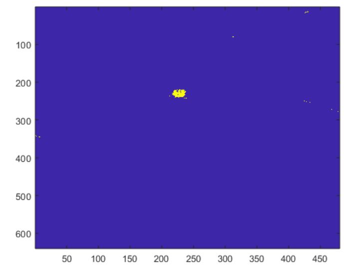


128 (single gaussian), 134 (GMM)

4.6 Image 6

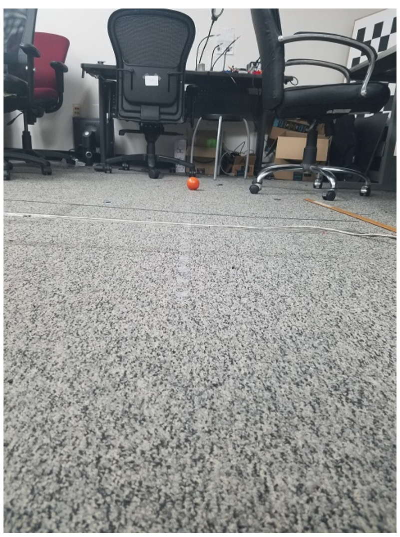


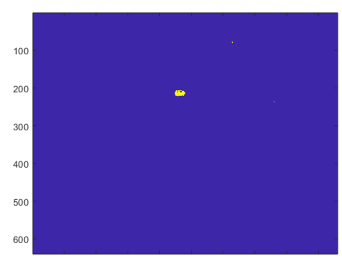


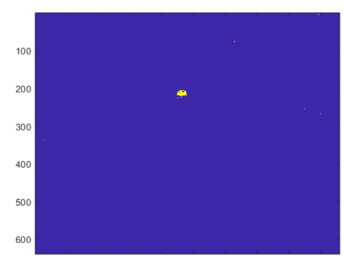


179 (single gaussian), 179 (GMM)

4.7 Image 7

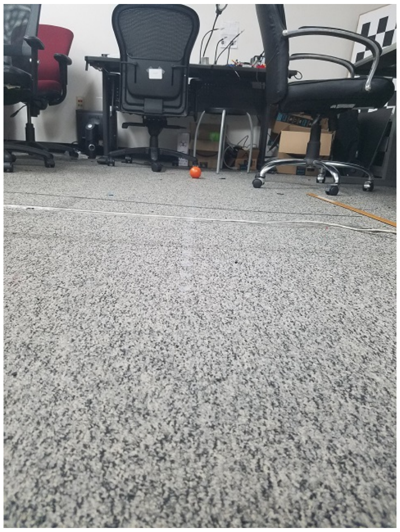


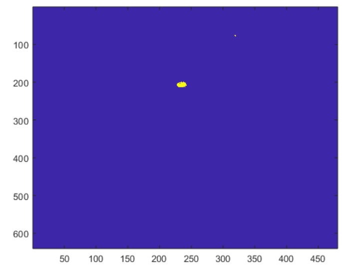


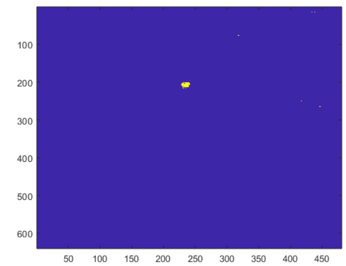


243 (single gaussian), 275 (GMM)

4.8 Image 8







261 (single gaussian), 258 (GMM)

5 Discussion of Results

5.1 Image 1

Image 1 contains two "distraction" balls, neither of which are the original orange ball that the models were trained on. This is perhaps the most glaring demonstration of the weaknesses of the single gaussian model, since the single gaussian model identified the red ball (and parts of the bookshelf) as the original orange ball, whereas the GMM was much better at seeing that most of the input image was just noise.

5.2 Images 2 and 3

These images are similar to Image 1, but the orange ball is actually present in this image. However, the selected pixels for both models were part of the real orange ball. In Image 2, the results are pretty similar, but in Image 3, the GMM chooses a smaller region for the ball, so the resulting distance was larger.

5.3 Images 4-8

These images aren't as interesting, because they don't have any "distractions" in the image, and the single gaussian approach and GMM both yield similar probability maps.

6 Overall Discussion

In the above discussion, it's clear that in the presence of distractions, the GMM performs significantly better than the single gaussian model. However, the GMM appears to be a little more conservative in picking orange ball pixels, which is particularly prominent in Image 3.

Our GMM algorithm seems to be really good at phasing out distraction objects, but it seems like it may be a little too conservative with selecting orange pixels that are truly part of the ball.