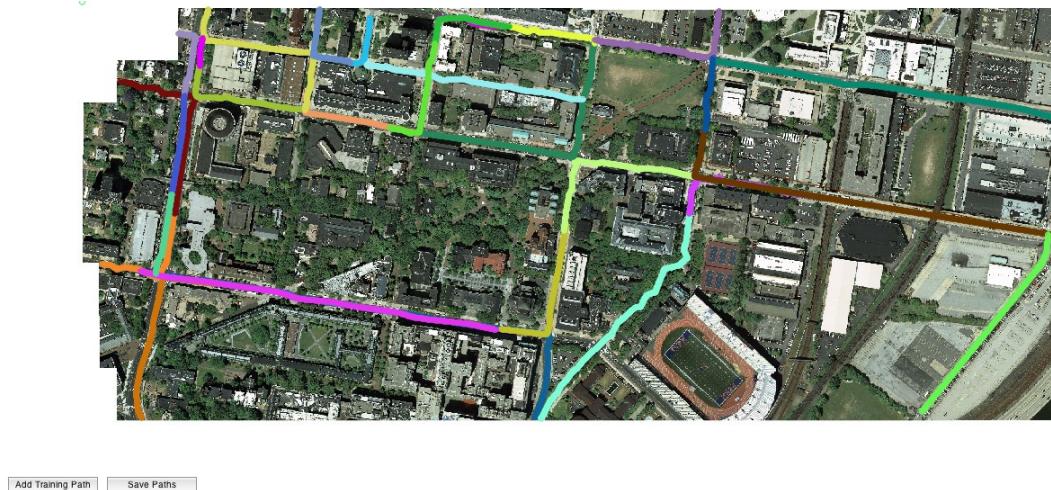


# ESE650 Project 5: Path Planning

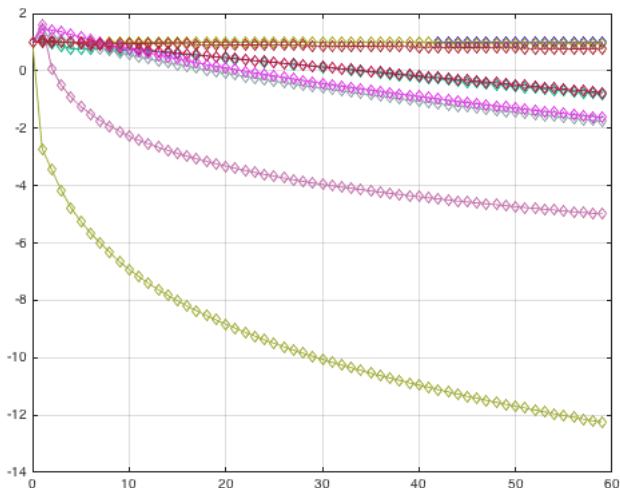
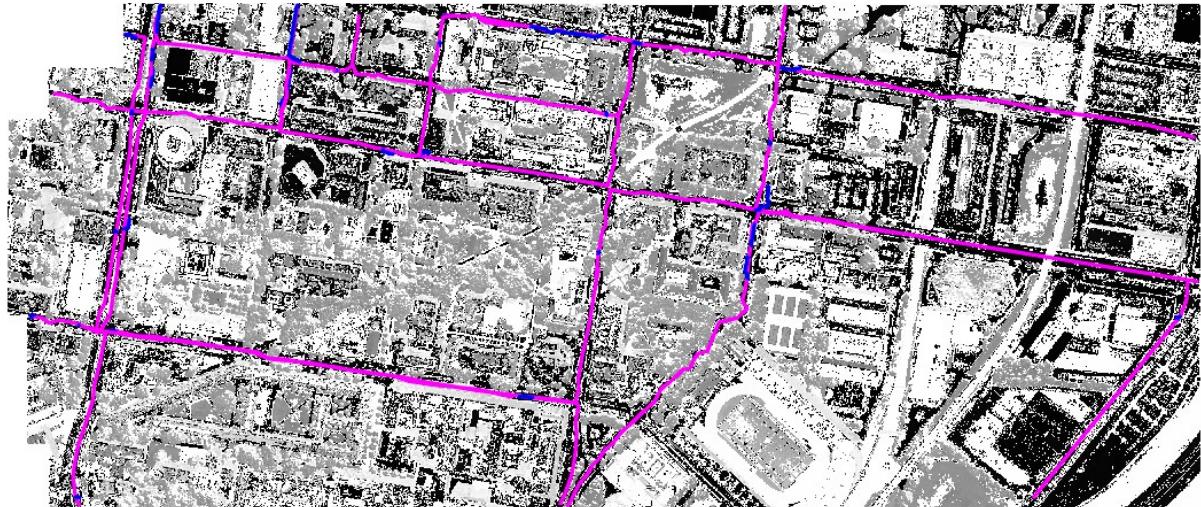
## Approach:

1. Learned weighting of feature layers using gradient Descent with regularization to train two different exponential cost maps. One for walking and one for driving.
2. The feature layers were also different for the training of the walking and driving cost maps. For the walking map I used 24 k-means clusters using euclidian distance on 2 image layers converted from RGB to CIE 1976 L\*a\*b\* and finally to gray scale feature layers. I then used the program 'colorThresholder' to create roughly 12 hand picked color layers for the driving map.
3. The map, feature layers and expert paths are all scalable in the code and I used a scale of  $\frac{1}{4}$  the original image size for both cost maps.
4. In order to speed up training, I trained each expert path on a patch of the image rather than the whole map. This decreased training time significantly.
5. Training on the driving paths converged after about 60 iterations and about 300 for the walking paths. I used a fairly large value for  $\eta$  of 1e-3 with a  $\lambda$  value of 1e-10. I set my gradient descent's convergence determination to be the current cost minus the previous cost and if it was less than 1e-3, I considered the cost map to be converged.
6. I also created a simple GUI to generate expert training paths.



## Training Results

Driving Cost Map: after convergence

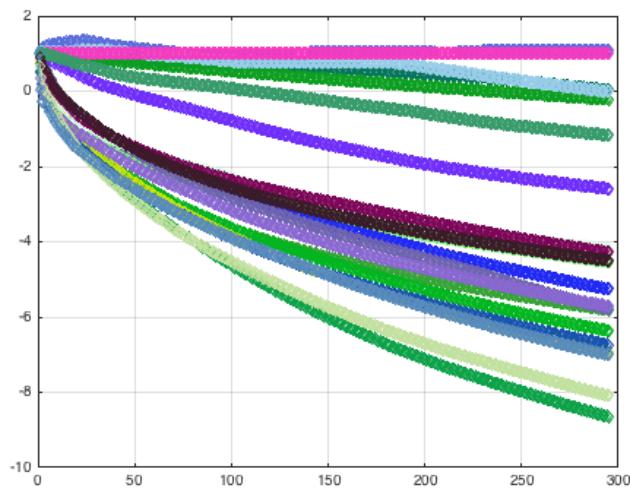


Graph of driving training weights

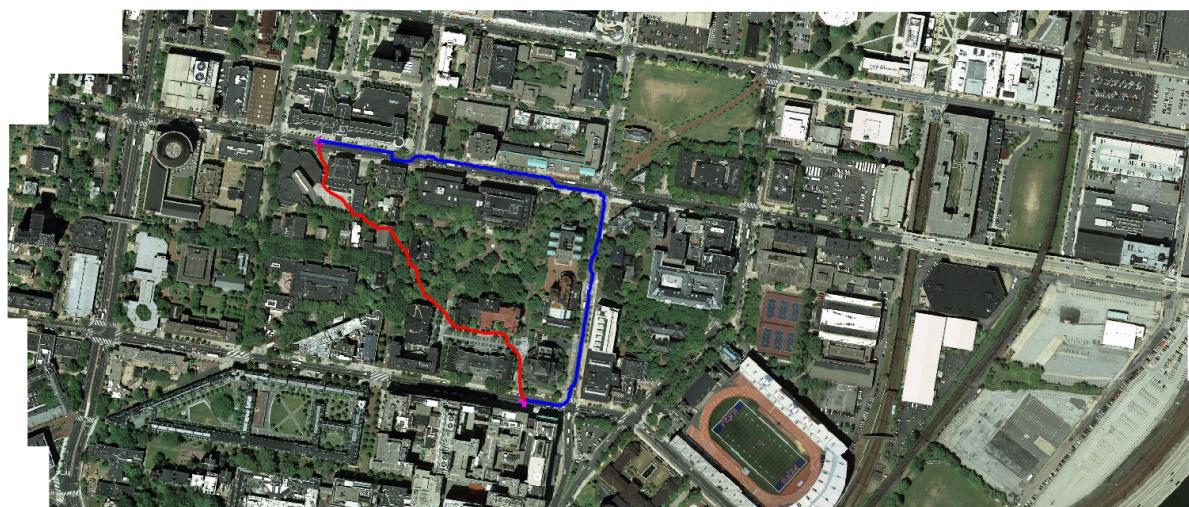
Walking Cost Map: after convergence

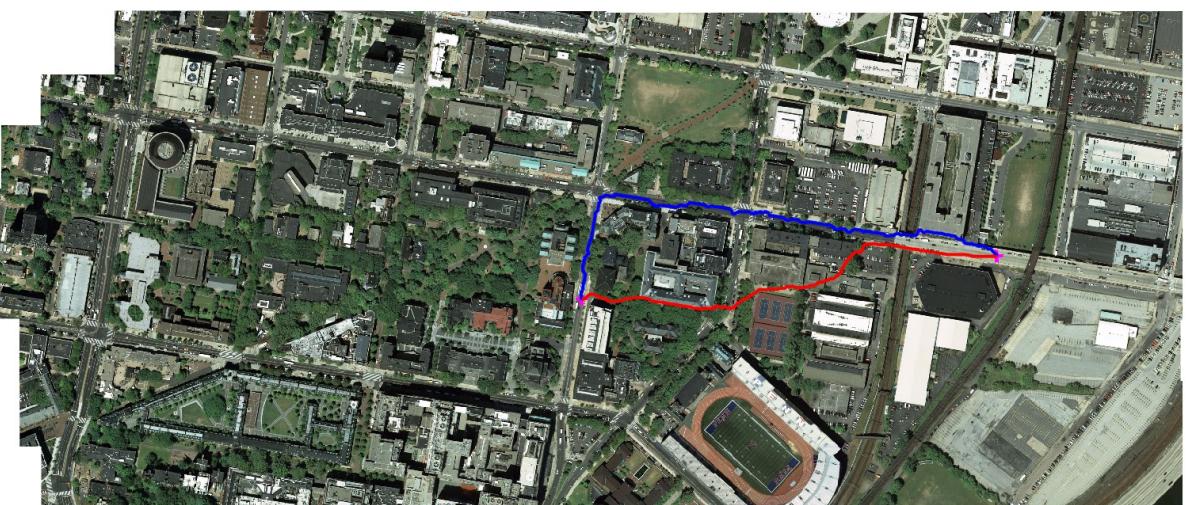
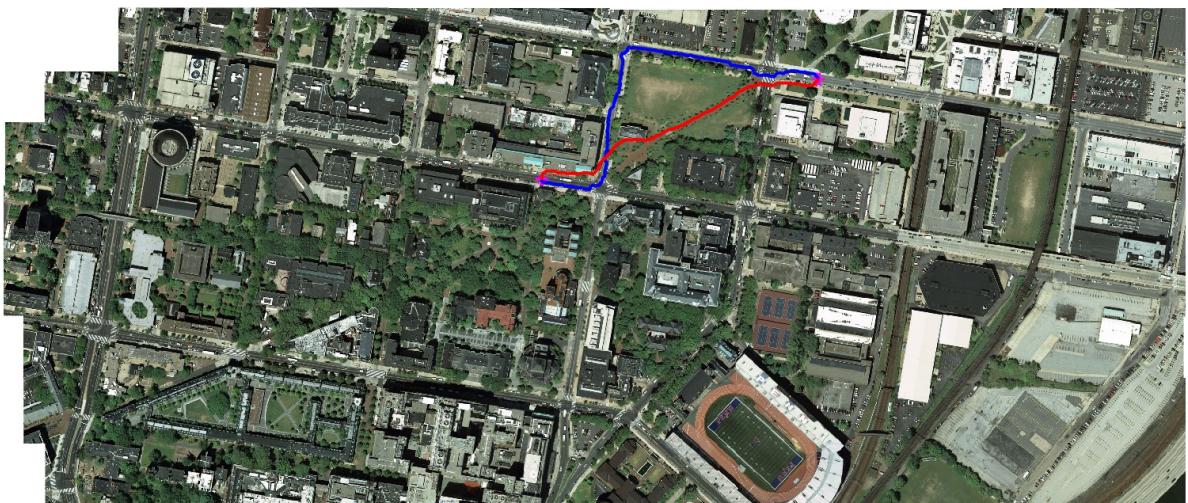
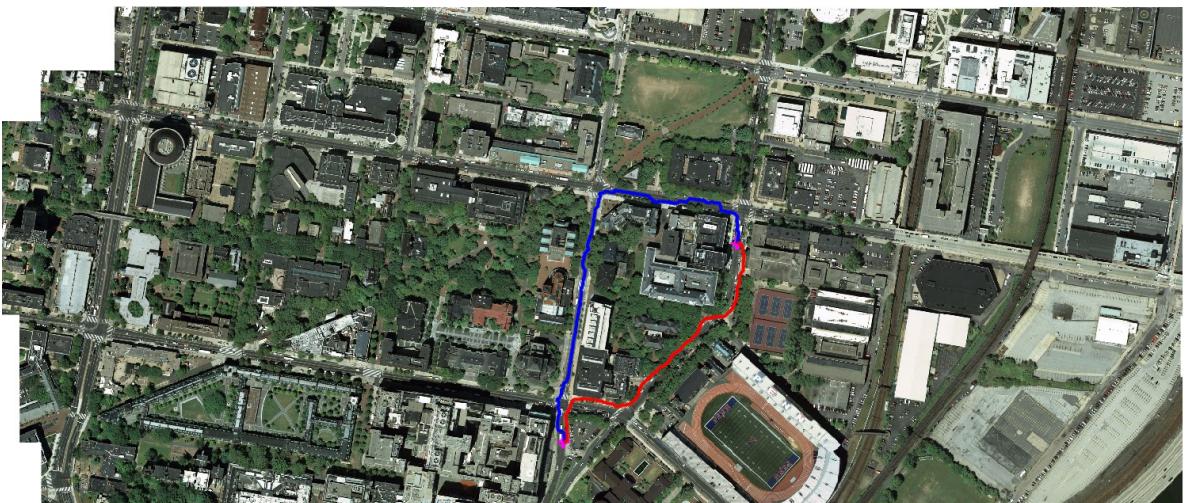


Weights Graph from 1 to 300 iterations



Testing Results (Blue is the driving path and Red is the walking path)



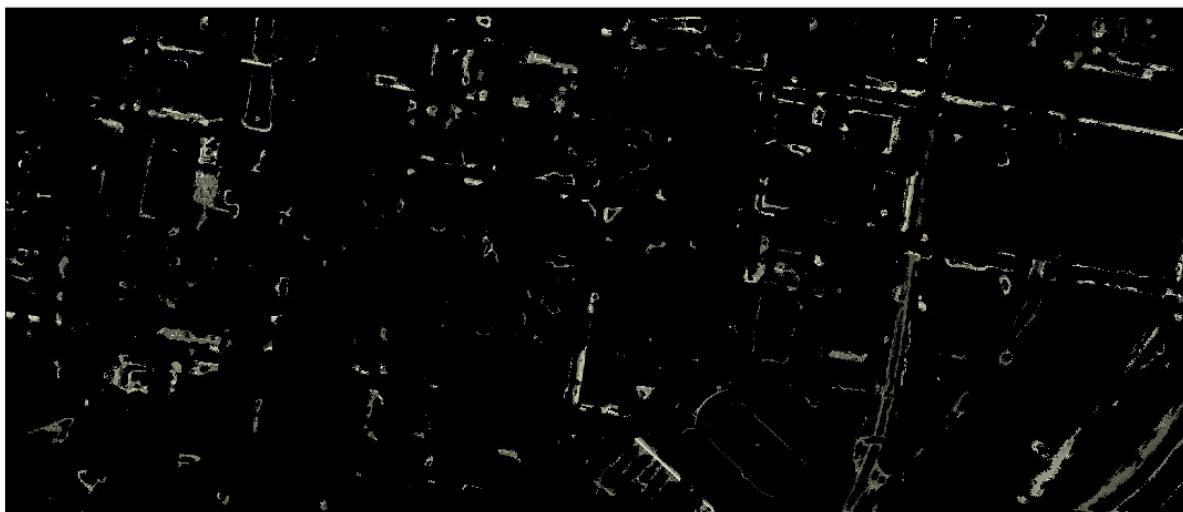


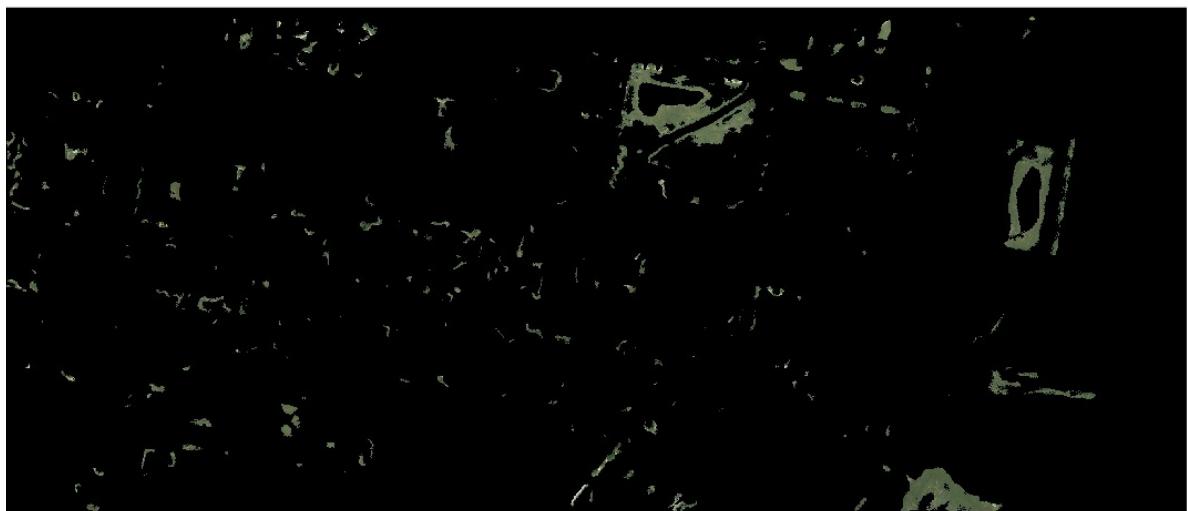
## Conclusion

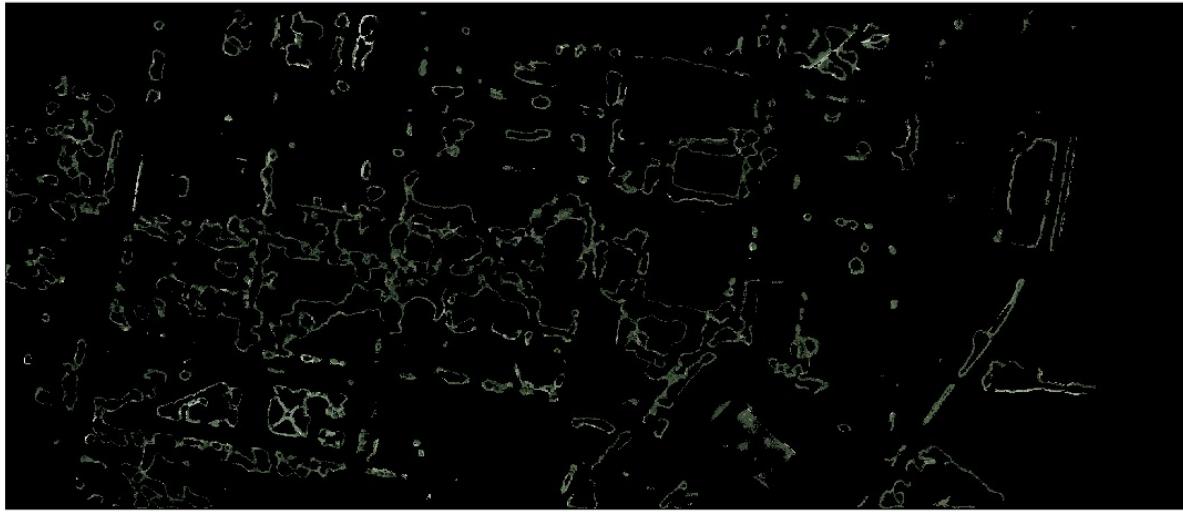
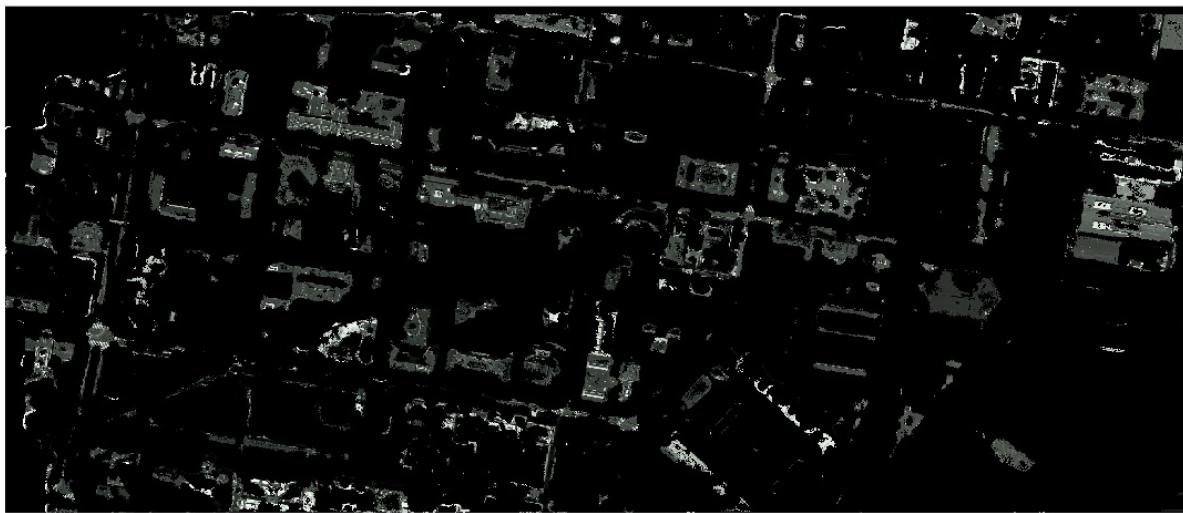
As you can see my results turned out fairly well. Only a few buildings were crossed over by the walking model, but the driving paths were almost perfect except for the above image where the car goes off the lighter colored bridge.

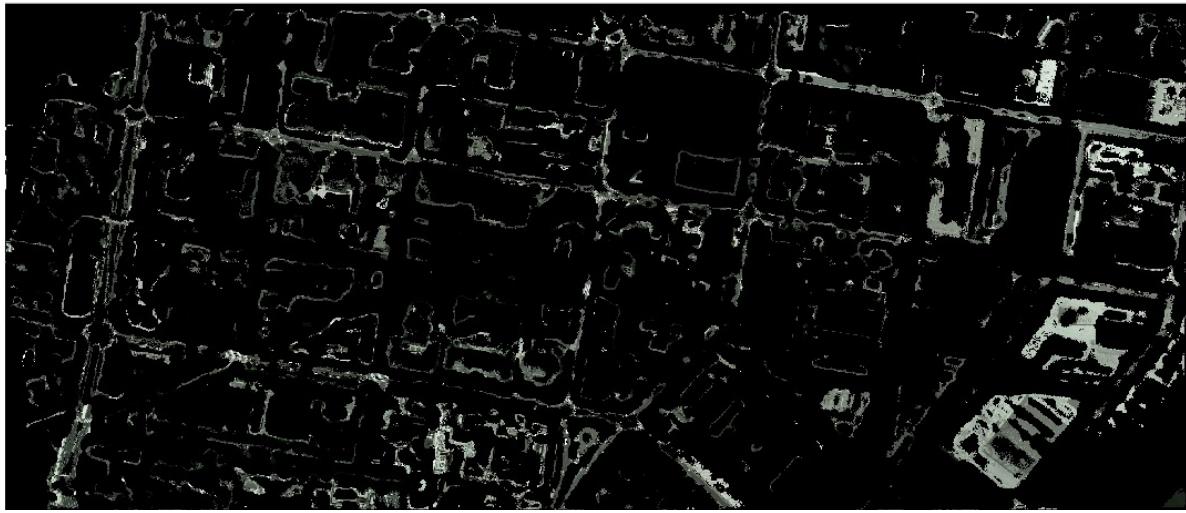
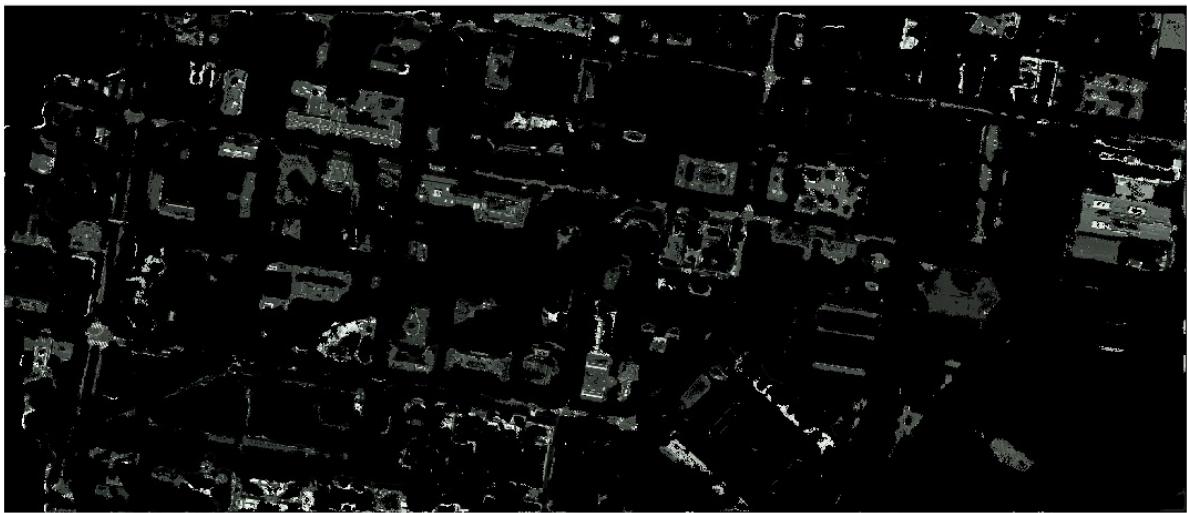
## Feature Maps

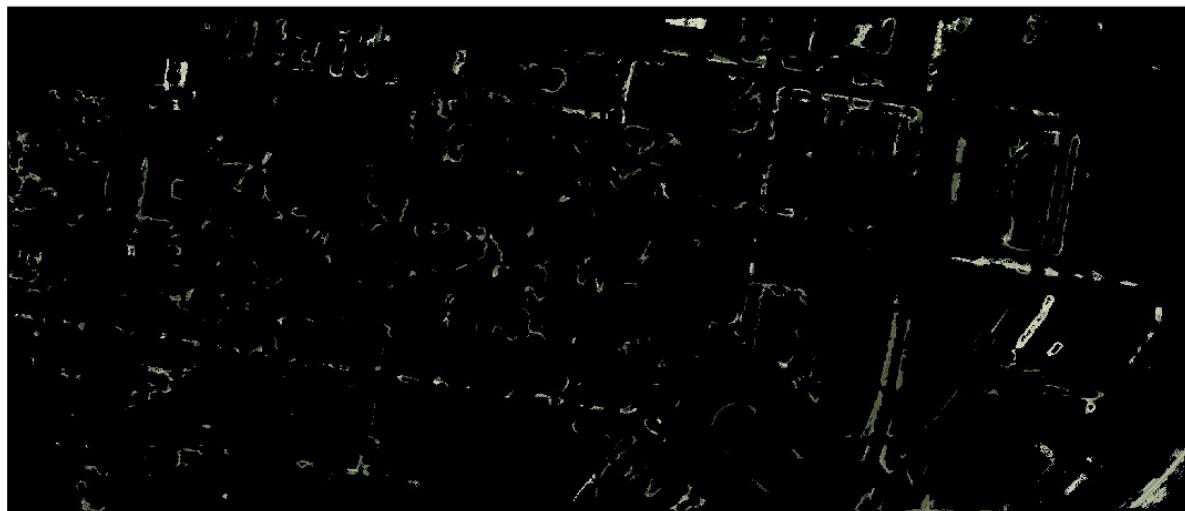
### K-means Layers

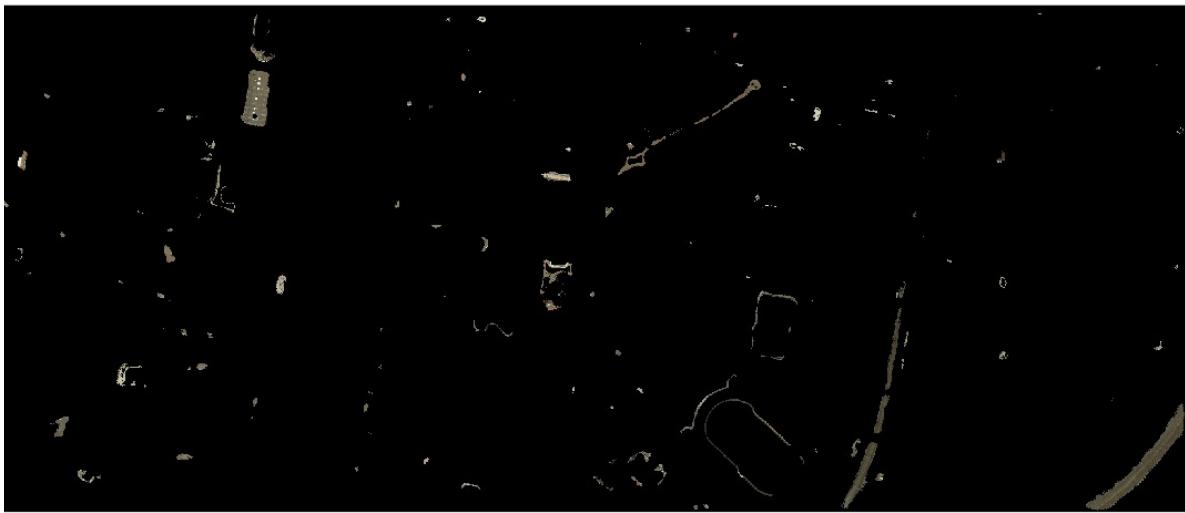
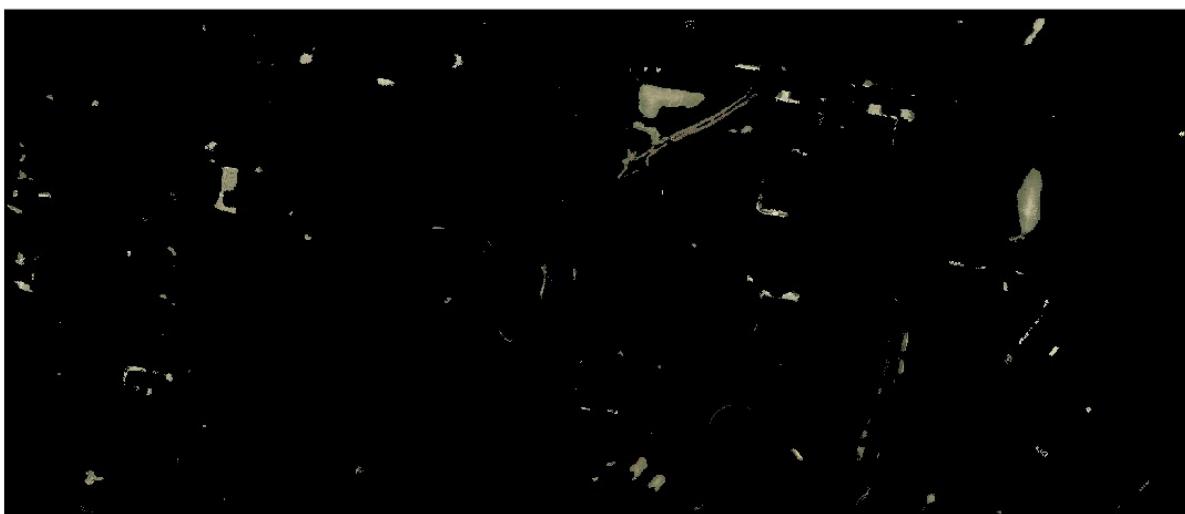


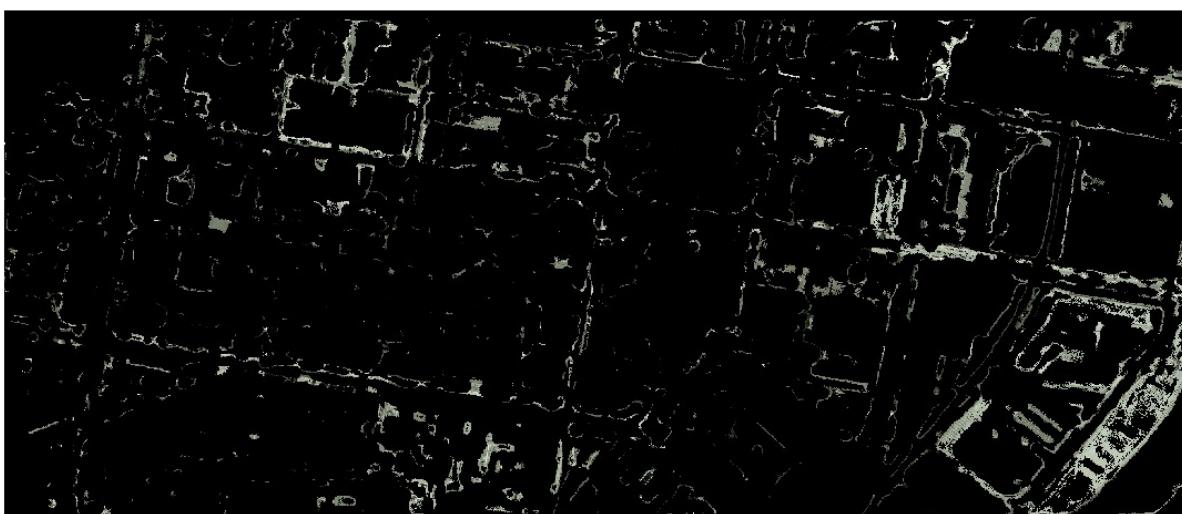
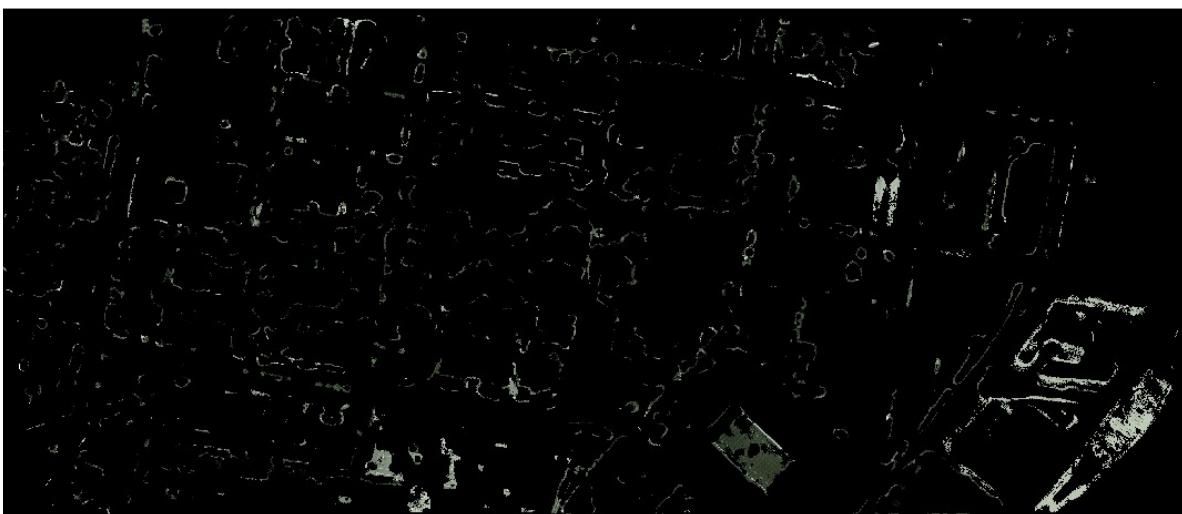


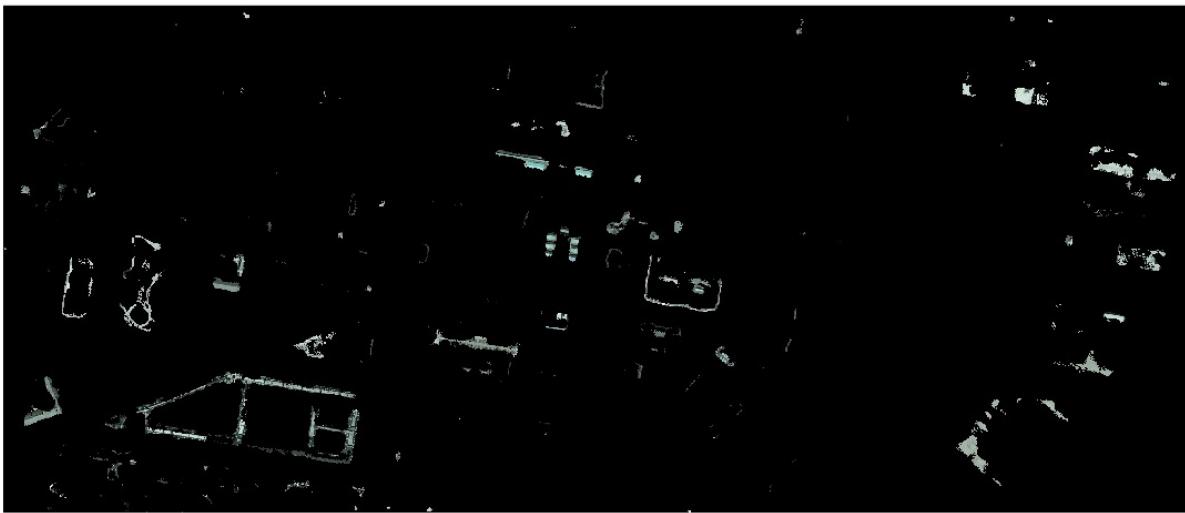
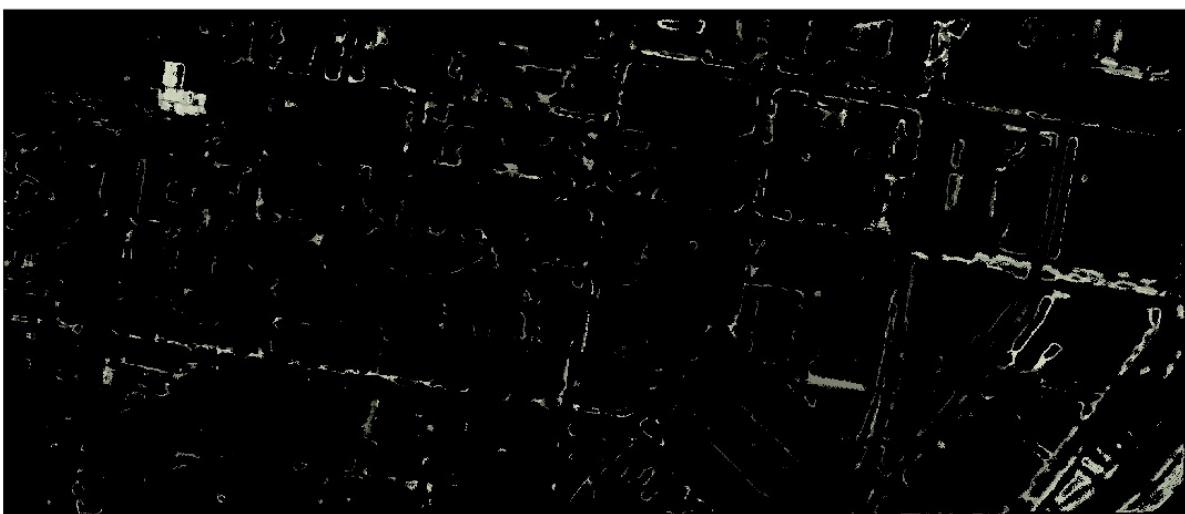


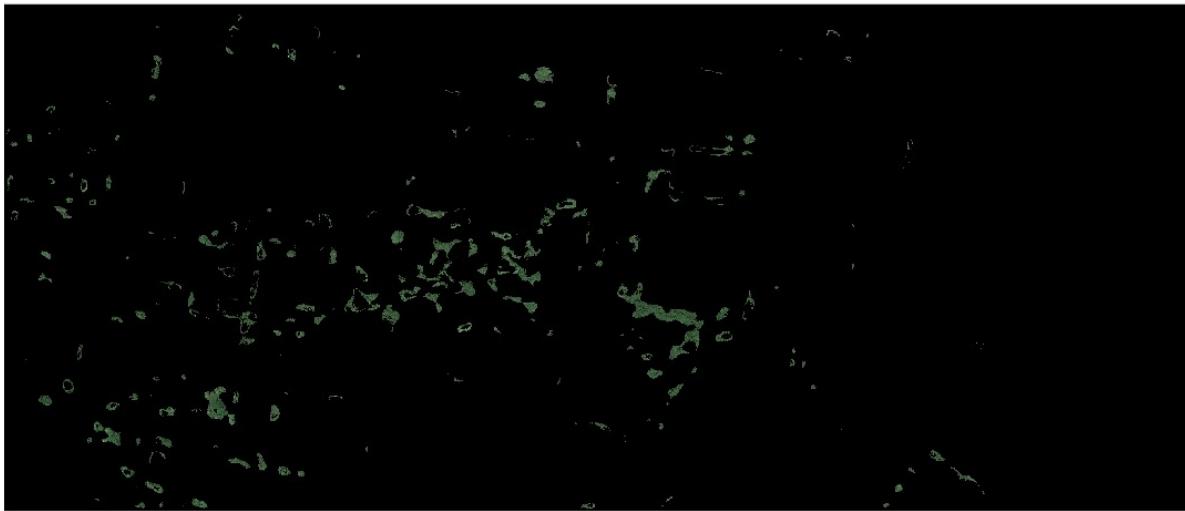










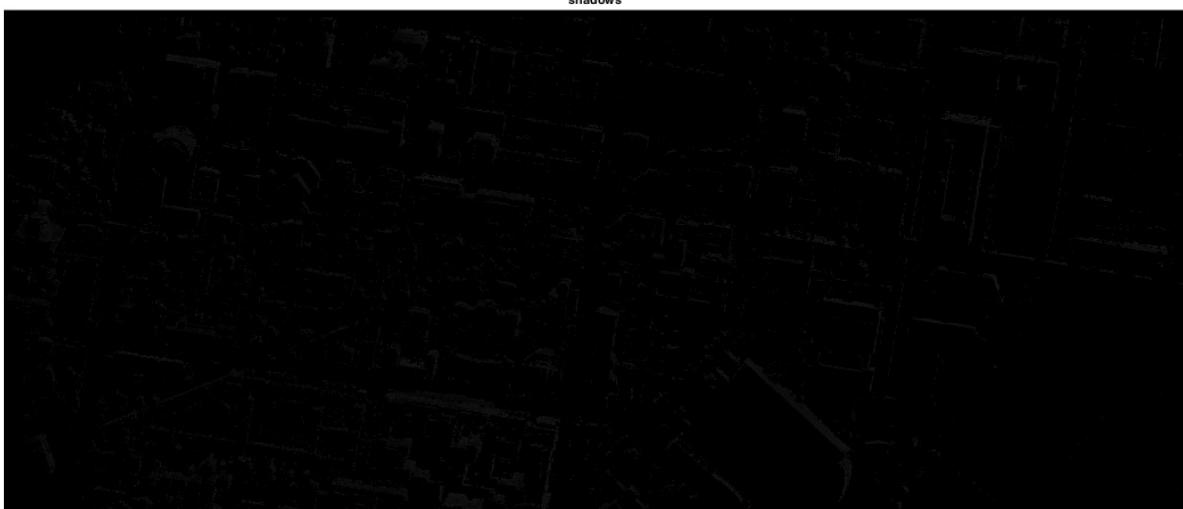


**ColorThresholder features follow:**





shadows



trees

