# ESE 650 Project 5 Report

Anushree Singh

April 8, 2014

## INTRODUCTION:

In this Project our aim is to do path planning for a robot in a given map environment using imitation learning. This is done by extracting the features of the colored map (aerial view of the Penn campus) provided by Google maps. Depending on these features and the mode of travel (pedestrian or car), certain weights was associated with these features which were corrected using the gradient descent method thus generating a cost map. This cost map was fed to the Dijkstra's algorithm that generated a minimum cost path. Hand labeling of the routes was done which was then used to optimize the path generated via Dijkstra's Algorithm.

## FEATURE SELECTION:

The most important part of this project was feature selection, as it greatly affected the cost maps that were generated. I have given my algorithm a total of ten features. These features are mostly dependent on two color spaces of the transformed map, i.e. L*a*b and HSV. L*a*b helped bring out the pedestrian paths pretty well whereas HSV helped extract the roads. I used K-means in order to cluster the color spaces in 4 clusters for L*a*b and 5 clusters for HSV. And my $10^{th}$ feature is edges.

Feature 1: This feature is taken from the L*a*b color space and it describes the road, which is the asphalt color.

Feature 2: This feature is taken from the L*a*b color space and it describes the vegetation (especially trees and dense green grass, etc.)

Feature 3: This feature is taken from the L*a*b color space and it describes the lighter vegetation and beaten paths.

Feature 4: This feature is taken from the L*a*b color space and it describes the buildings (lighter than roads).

Feature 5: This feature describes the edges in the map (good for differentiating the buildings from roads).

Feature 6: This feature is taken from the HSV color space and it describes the vegetation.

Feature 7: This feature is taken from the HSV color space and it describes the lighter vegetation.

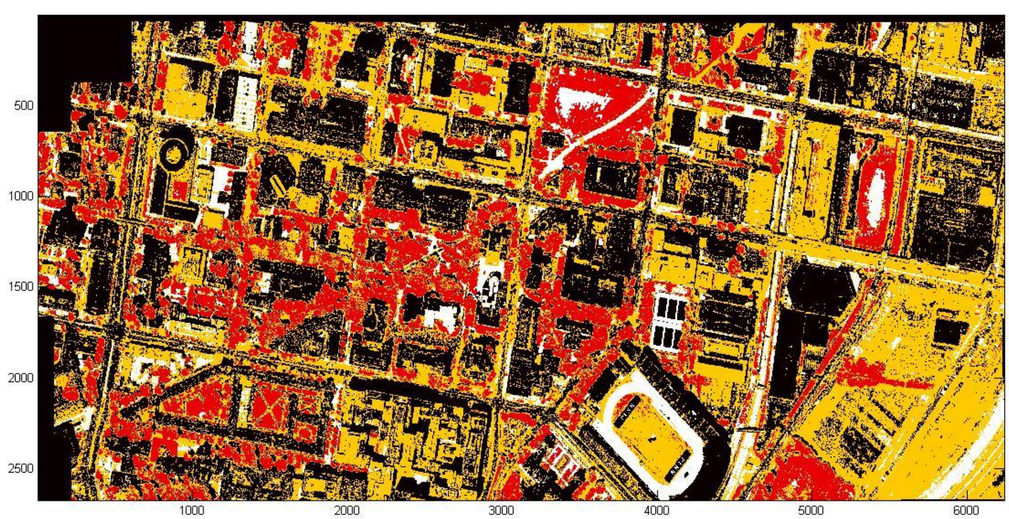
Feature 8:  This feature is taken from the HSV color space and it gives the buildings.

Feature 9:  This feature is taken from the HSV color space and is not much informative.

Feature 10:  This feature is taken from the HSV color space and it describes the road

Fig: Clustered colors in L*a*b space (4 clusters) :


Fig: Clustered colors in HSV space (5 clusters) :



# GRADIENT DESCENT:

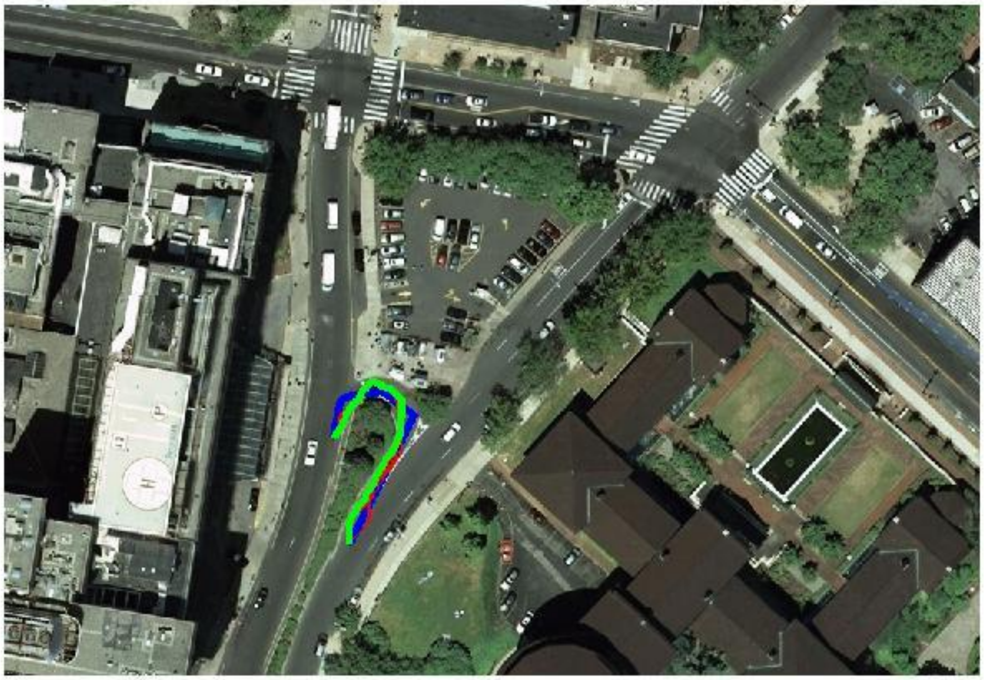
A number of paths were hand labeled to set the desired path routes (as seen in the figures below as blue). This was done using matlab's ginput and the Bresenham's algorithm for line drawing.

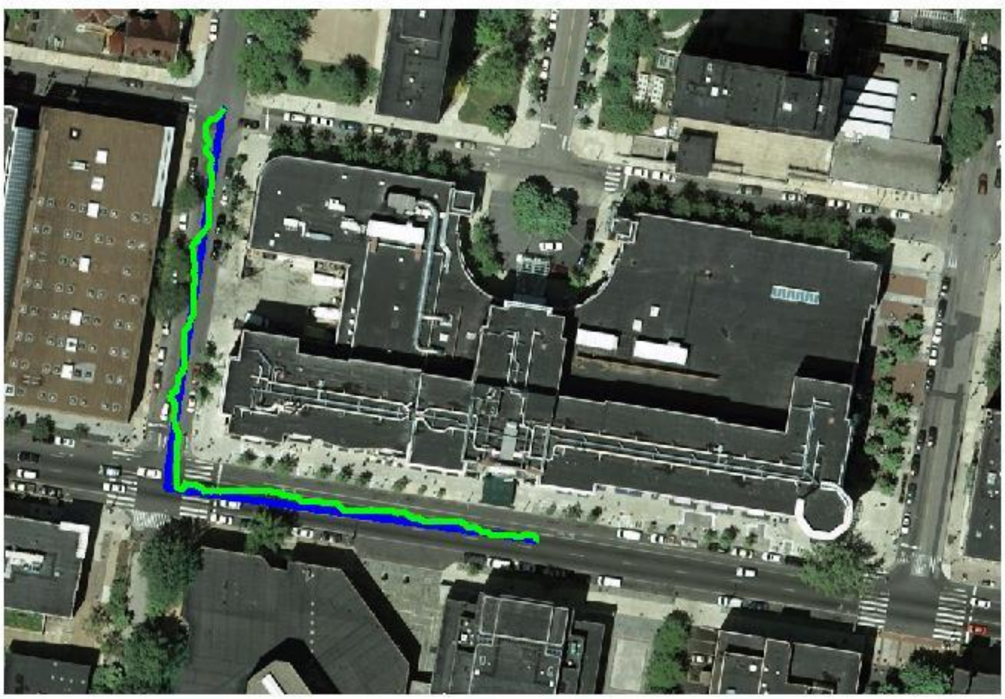
Based on the features described above and the mode of travel a certain set of weights were set to initialize the gradient descent. Using this weight vector and the set features a cost map was generated. Using the Dijkstra's Algorithm an optimum path was created. The difference, of the cost of the desired path multiplied by the feature vector of all the pixels in the path and the cost of the optimum path multiplied by the feature vector of all the pixels in the path, was calculated. This difference was scaled by a factor eota. The scaled difference was subtracted from the weight vector to give the new weight vector. This was done iteratively for 20 iterations also keeping a check on the scaled difference divergence.

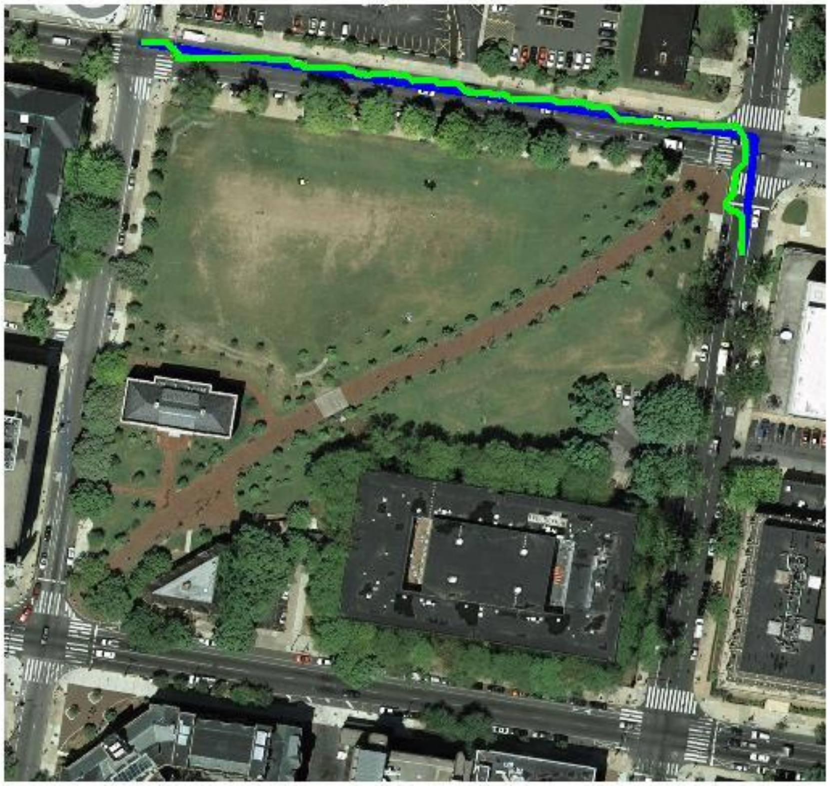
The final weight of this gradient descent is used to calculate the final cost map which essentially is passed to the Dijkstra's Algorithm to calculate the final trajectory of the robot.

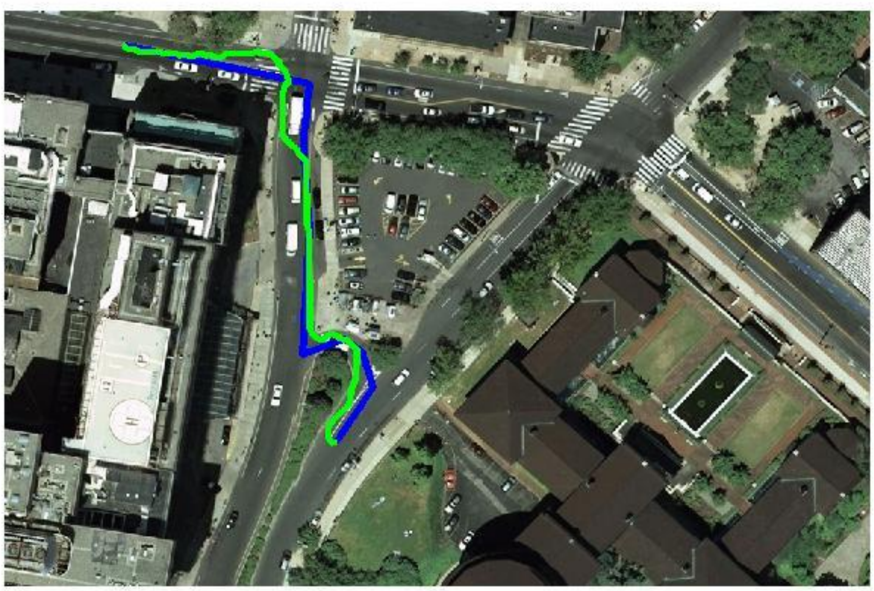
Examples of hand labeled path and matching path:

The green line indicates the path chosen by the gradient descent algorithm, whereas the blue line indicates the path given by hand labeling.







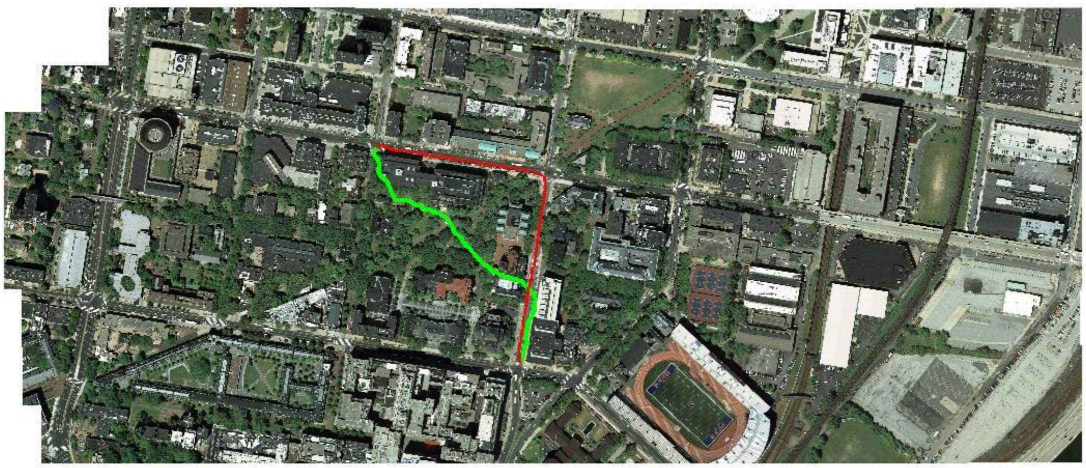


# RESULTS:

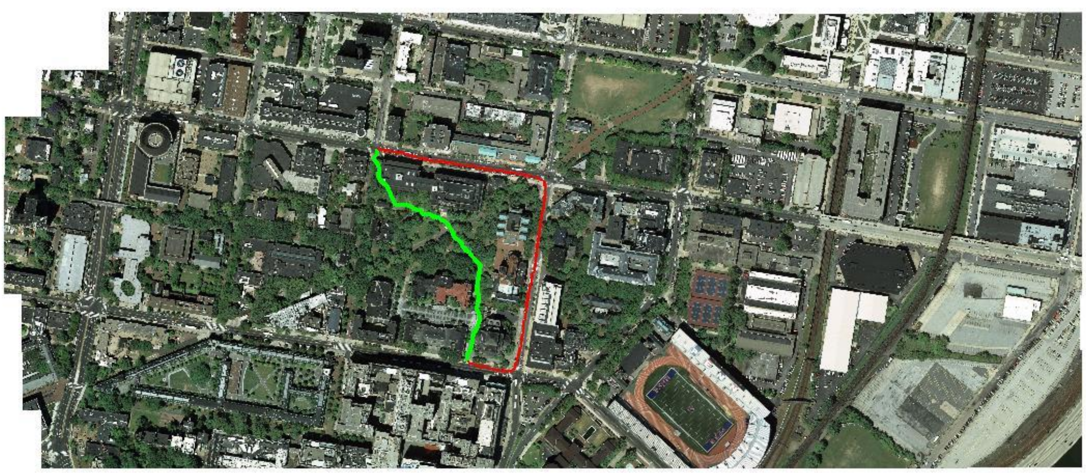
Results of some of the sample paths can be seen below:

The green line represents the path planned for a robot that imitates a pedestrian and the red line indicated the path planned for the robot that imitates a car.

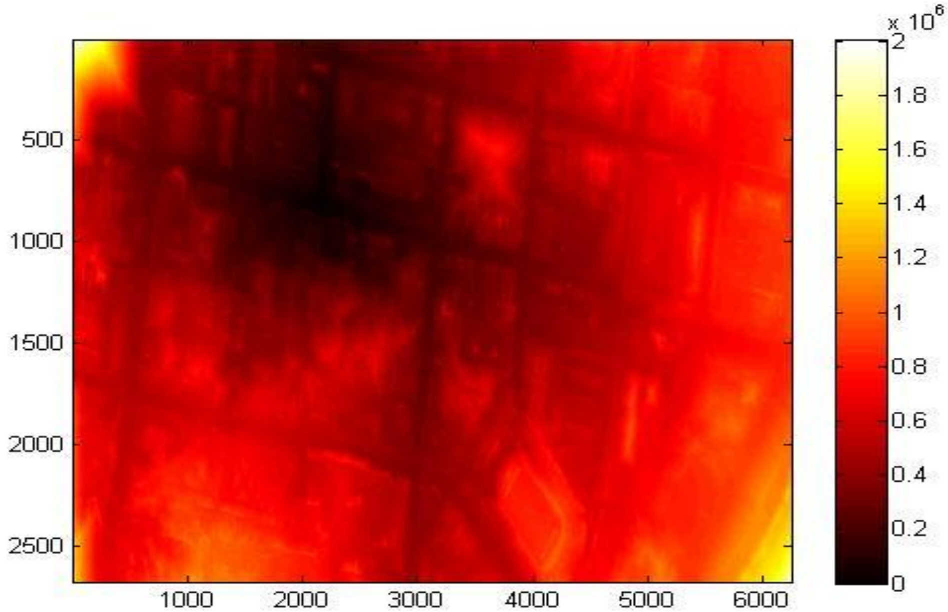
Path1 :



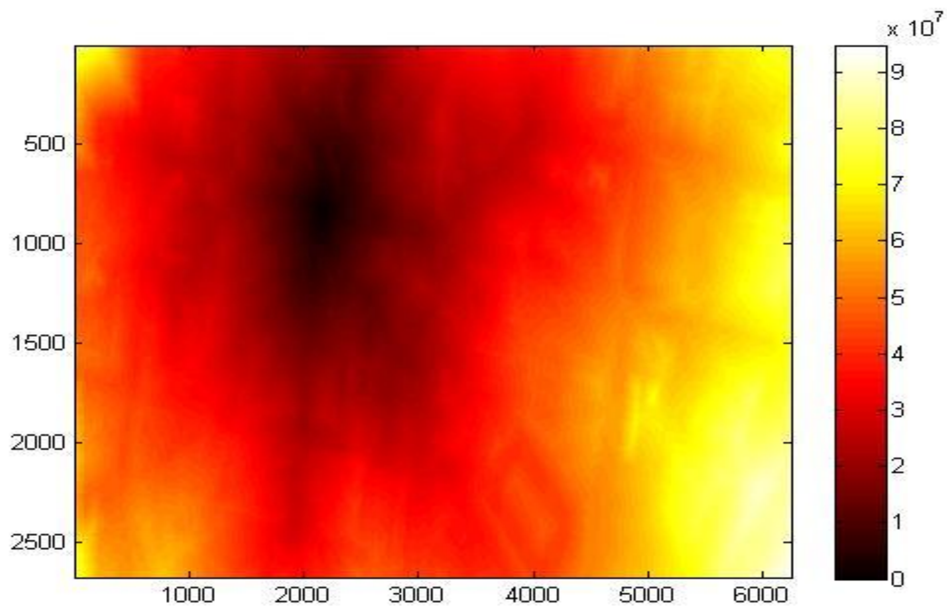
Path 2:



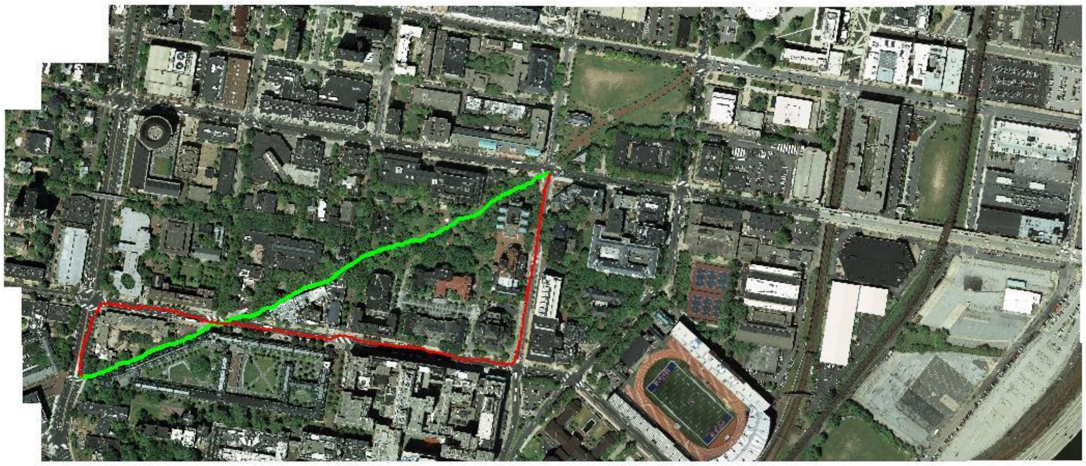
Cost to go for car ,path2 :




Cost to go for pedestrian ,path2 :



Path 3:



# FUTURE WORK:

Future work would be to implement the max margin learning method as described in the RatCliff et. al. which is one of the famous techniques in re-enforcement learning.