ESE650 Project 5:

Cost Learning and Path Planning

Nischal K N
nischal@seas.upenn.edu

Abstract
Imitation learning is an advanced behavior to replicate the previously supervised action. This is accomplished by learning the parameters of the model through the various supervised examples. In this project we learn the cost of various features of a satellite map through hand labeled routes to build a cost map for driving and walking. This is accomplished using gradient decent to choose the weights for each feature set.

Contents

1. Introduction

In this project, the aim is to generate routes between any two points in a satellite image of penn campus. It has to support 2 modes, walking and driving. To accomplish this, we first hand label viable routes for both the modes. Based on the hand labeled paths weights we learn different cost maps for each mode. The cost map is a linear combination of weights multiplied to the feature map which are derived from the satellite image. The objective is to learn the weights of each feature map that minimizes the difference in the Dijkstra path and hand labeled path. This is achieved through gradient decent.

2. Algorithm

2.1 Hand Labeling

A number of paths are hand labeled using the ginput matlab function. GetMapCellsFromRay function is used to obtain the path between the clicked points from the ginput. Fig. 1a and Fig. 1b shows the hand labeled paths for driving and walking modes respectively.

2.2 Features

A total of 24 features were used as the feature set. It included 16 Kmeans cluster in RGB color space with each cluster as a feature set and a unique feature map for unique objects such as white roof tops, black roof tops, tree cover, foot paths, etc either in HSV, YCbCr or Lab color space depending on which can capture the features better. They were manually picked using the color thresholder app in Matlab.

Additionally to each of these feature maps, morphological operations were performed to reduce the effect of stray pixels. The series of morphological operations were

1. **spur** using bwmorph
2. **clean** using bw morph
3. **erode** using imerode with a disk structural element of size 2.
4. **dilate** using imdilate with a disk structural element of size 2.

In few cases such as gray and black roof tops, quite a few roads were also present in the feature map. To remove these, the property of eccentricity was used. First the connected components are found out. Then for each of these connected components the area and eccentricity was thresholded. Anything above the threshold was consider as a road and hence removed as roads are more elongated and hence have higher eccentricity. Hence this preserved rooftops only.

To all the binary images, a Gaussian blur was applied. The list of features used and the operations performed on them are shown in Table 2.

2.3 Gradient Descent

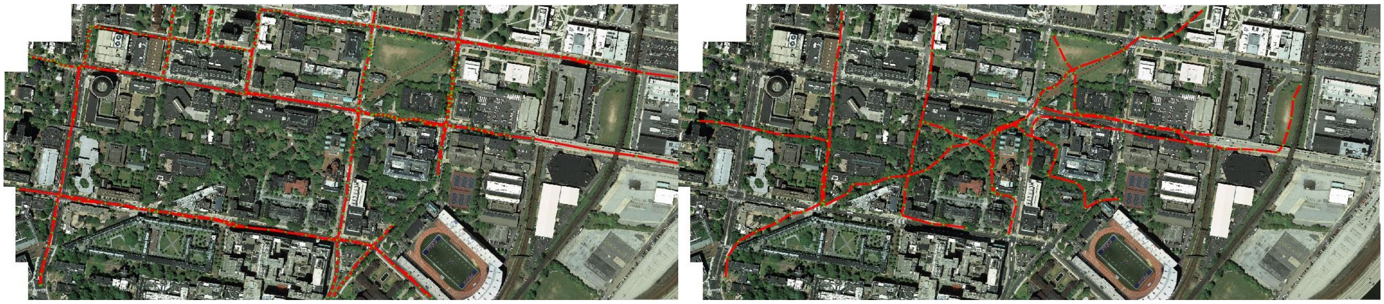
The following steps were performed to determine the weights of the feature maps

1. Weights w were initialized uniformly and the cost map $C(x,y)$ was calculated using

$$w_i = 1, \quad \forall i$$
$$C(x,y) = e^{\sum_i w_i F_i(x,y)}$$

   Where $F_i(x,y)$ is the feature map and i is the feature.

2. For each pair of training start and end points, Dijkstra path was created using the previously computed cost map $C(x,y)$. The path generated be (x',y').



(a) Hand Labeled Drive paths **(b)** Hand Labeled Walk paths

Figure 1. Hand labeled paths

3. The gradient or the difference between hand labeled and Dijkstra map is calculated for each feature and the sum over each sample path is taken. Gradient is calculated as

$$\frac{\delta J}{\delta w_i} = \sum_{(x,y)} F_i(x,y)e^{\sum_i w_i F_i(x,y)} - \sum_{(x',y')} F_i(x',y')e^{\sum_i w_i F_i(x',y')}$$

$$\frac{\delta J}{\delta w_i} = \sum_{(x,y)} F_i(x,y)C(x,y) - \sum_{(x',y')} F_i(x',y')C(x',y')$$

   Where (x,y) are hand labeled paths and (x',y') are Dijkstra generated paths.

4. The weights are updated using this gradient

$$w_i = w_i - \eta \frac{\delta J}{\delta w_i}$$

   Where η is the learning rate.

5. The new cost map is calculated using the equation in step 1.

6. The termination condition is determined based on the difference between the costs of hand labeled path and Dijkstra path.

$$J = \sum_{(x,y)} C(x,y) - \sum_{(x',y')} C(x',y')$$

   If the difference is less than the threshold, the process is terminated, else its repeated from step 2.

The final weights after convergence is shown in Table 1 for both driving and walking.

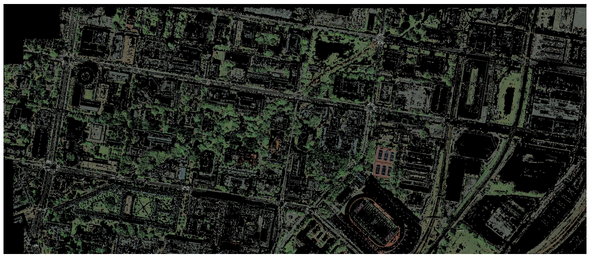
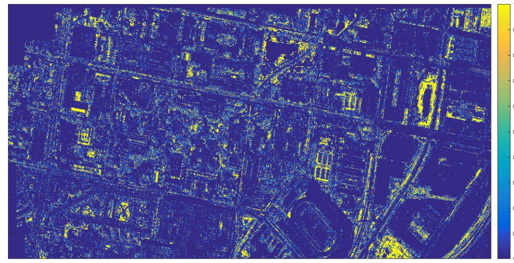
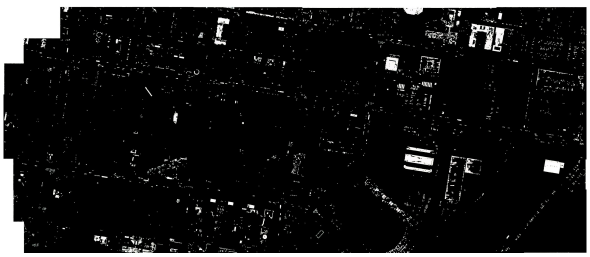
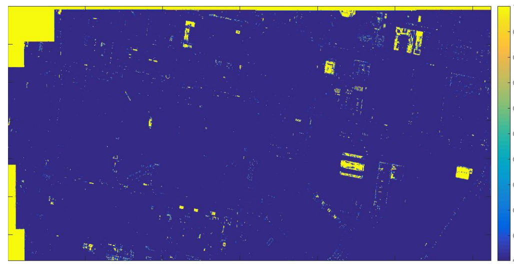
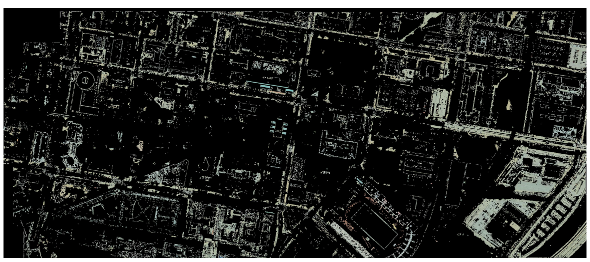
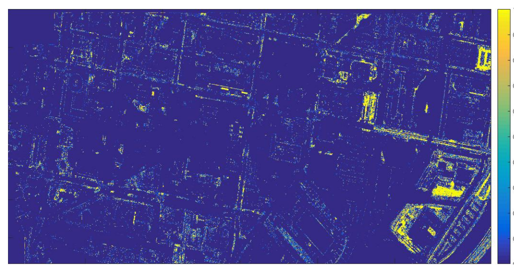
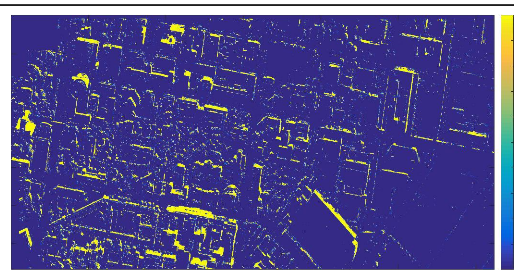
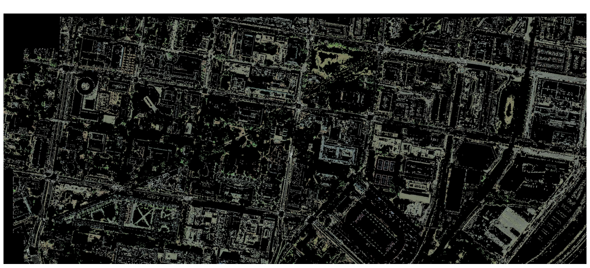
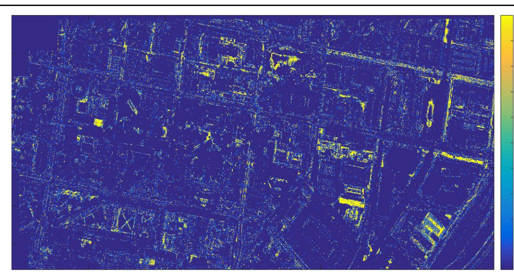
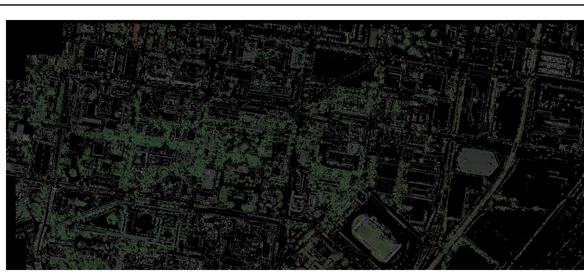
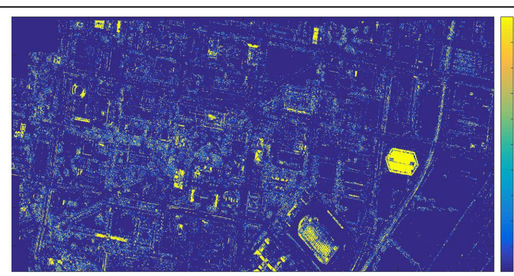
3. Experiments and Results

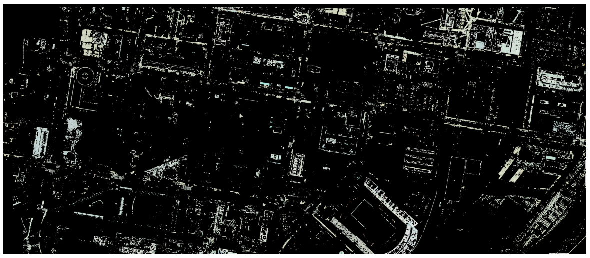
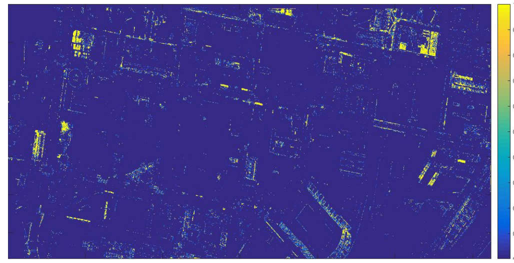
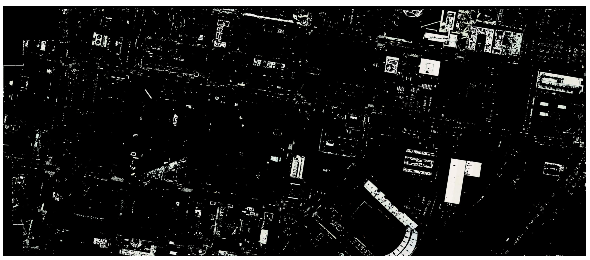
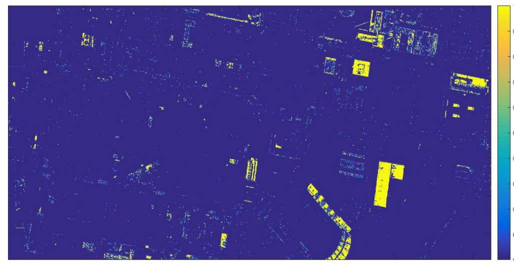
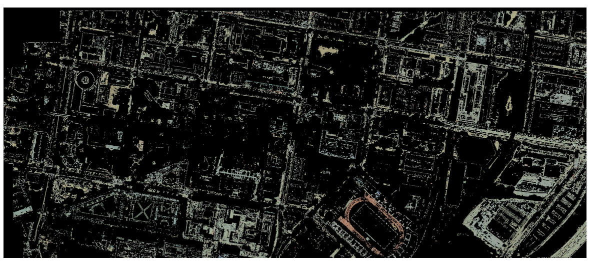
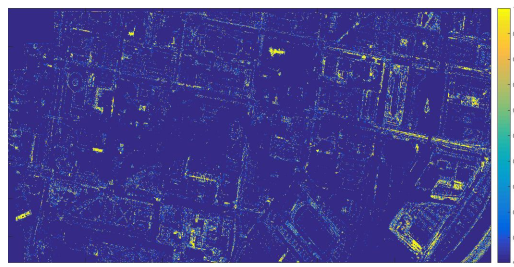
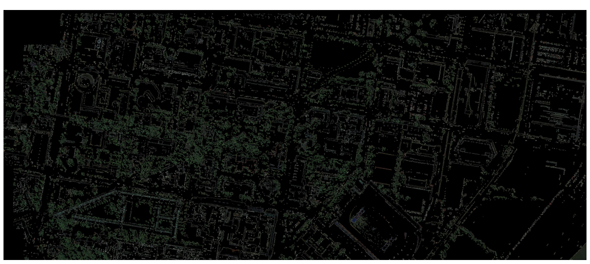
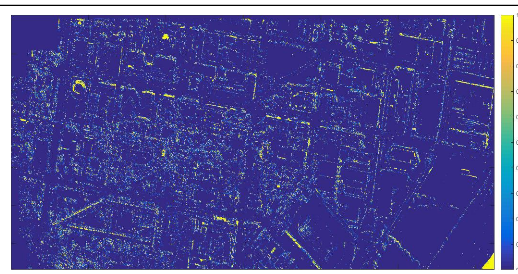
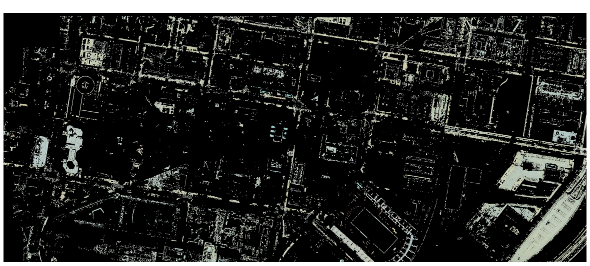
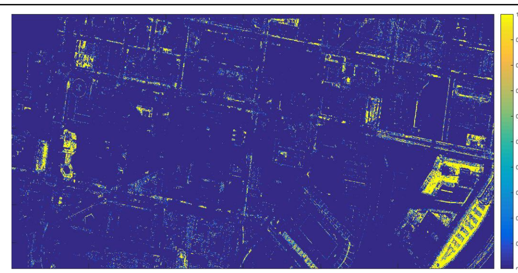
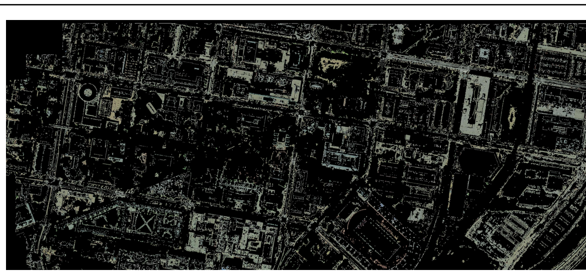
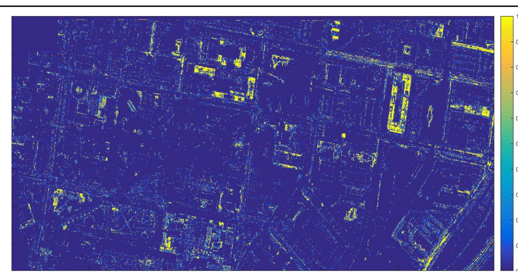
The cost map learned from the above process are shown in Fig. 2 for driving and Fig. 3 for walking. The performance on test points is shown in Fig. 4 for driving and Fig. 5 for walking.

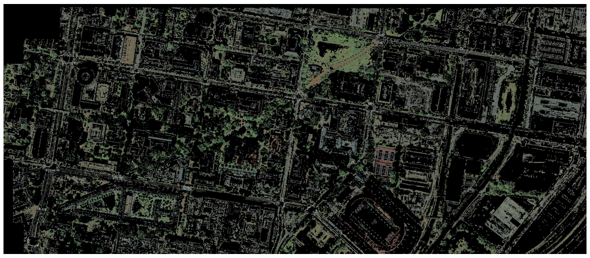
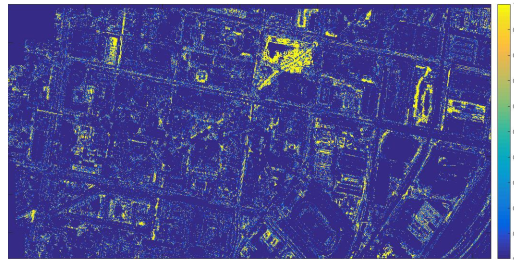
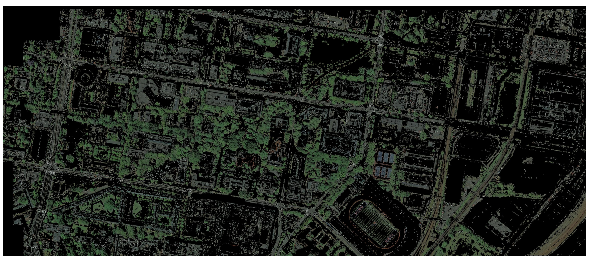
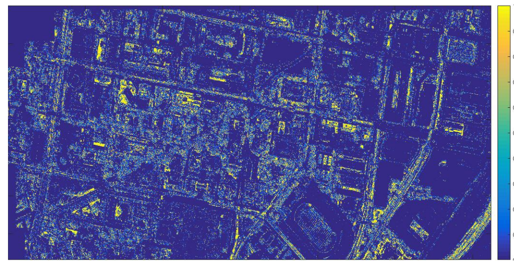
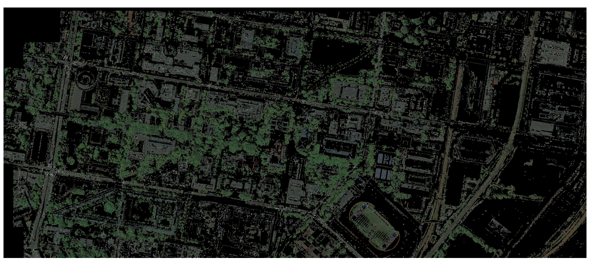
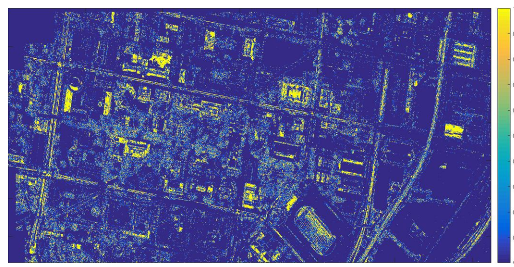
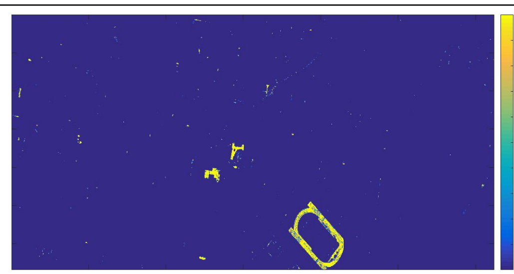
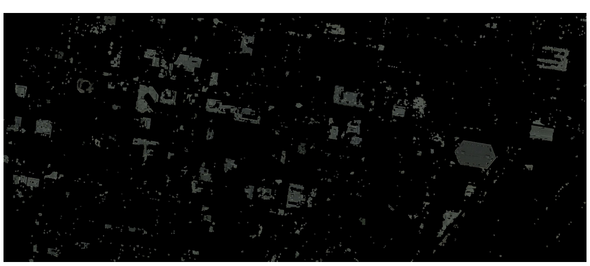
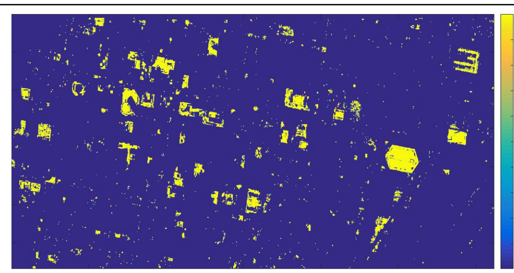
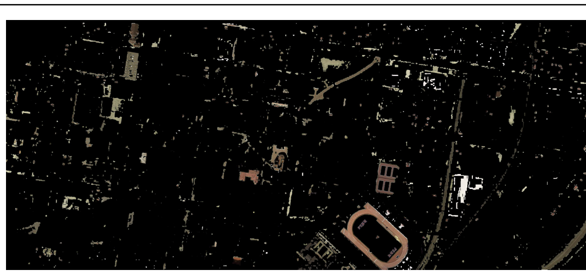
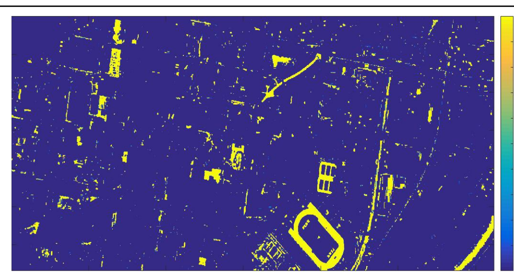
Table 1. Weights for feature map

Feature	Driving Weights	Walking Weights
K-means, K = 1	-0.4807	0.7781
K-means, K = 2	0.0900	0.9951
K-means, K = 3	-0.1071	0.8336
K-means, K = 4	-1.0135	0.8306
K-means, K = 5	-0.8823	0.8794
K-means, K = 6	-0.0458	0.8255
K-means, K = 7	0.3423	0.9021
K-means, K = 8	0.2456	0.9357
K-means, K = 9	-0.8598	0.8122
K-means, K = 10	0.6176	0.7909
K-means, K = 11	0.5919	0.8376
K-means, K = 12	-0.9072	0.8991
K-means, K = 13	-0.9023	0.7589
K-means, K = 14	-0.9404	0.6724
K-means, K = 15	-0.9351	0.7738
K-means, K = 16	0.8765	0.9710
Black roof	-0.8460	0.8327
Brown roof	-0.1785	0.5296
Grey roof	-0.3161	0.7979
Green trees	1.0849	0.0357
Dark Grey	-0.0882	0.7150
White roof	0.6745	0.9744
Cream roof	0.9684	0.5953
roads	-1.1784	-0.6059

Table 2. Features

Feature	Masked RGB Image	Feature	Post Processing
K-means, K = 1			Gaussian Blur
K-means, K = 2			Gaussian Blur
K-means, K = 3			Gaussian Blur
K-means, K = 4			Gaussian Blur
K-means, K = 5			Gaussian Blur
K-means, K = 6			Gaussian Blur

Feature	Masked RGB Image	Feature	Post Processing
K-means, K = 7			Gaussian Blur
K-means, K = 8			Gaussian Blur
K-means, K = 9			Gaussian Blur
K-means, K = 10			Gaussian Blur
K-means, K = 11			Gaussian Blur
K-means, K = 12			Gaussian Blur

Feature	Masked RGB Image	Feature	Post Processing
K-means, K = 13			Gaussian Blur
K-means, K = 14			Gaussian Blur
K-means, K = 15			Gaussian Blur
K-means, K = 16			Gaussian Blur
Black roof			YCbCr, Gaussian Blur, morphological operations, eccentricity
Brown roof			YCbCr, Gaussian Blur, morphological operations

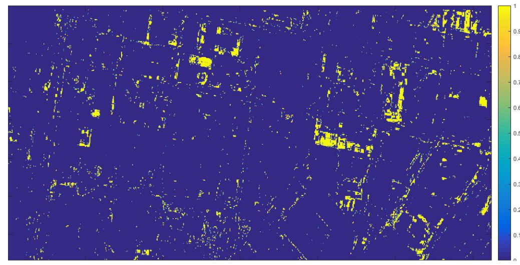
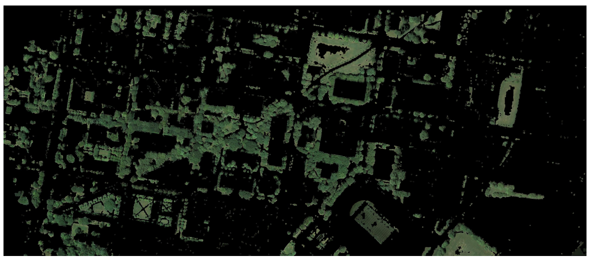
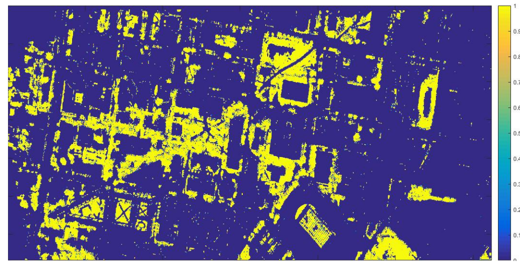
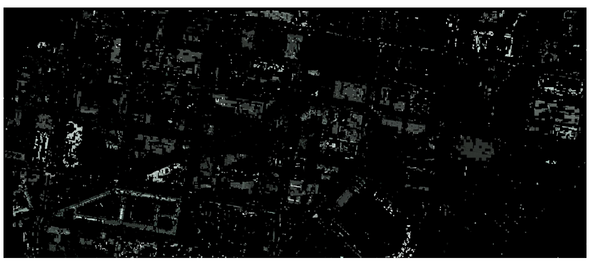
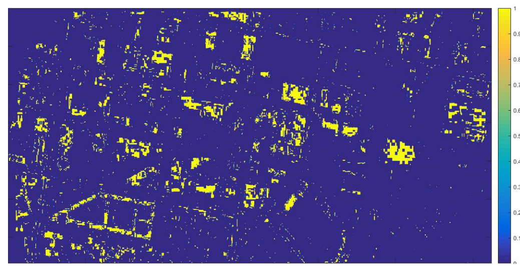
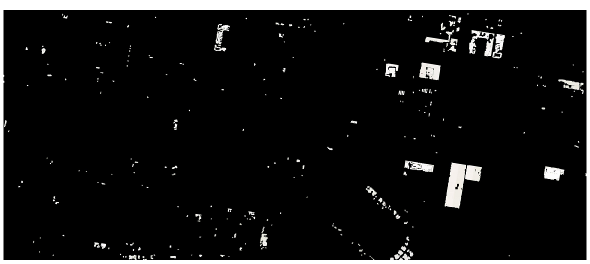
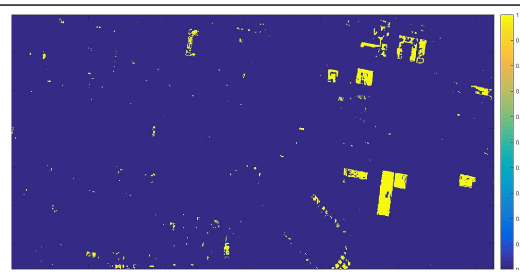
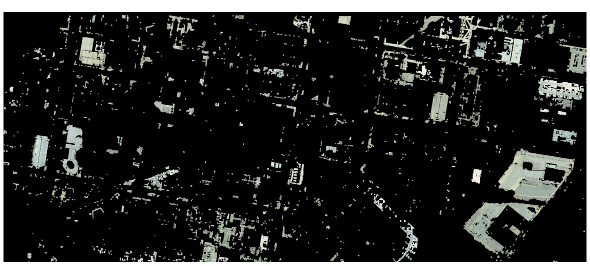
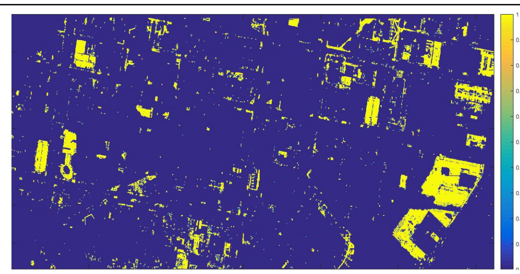
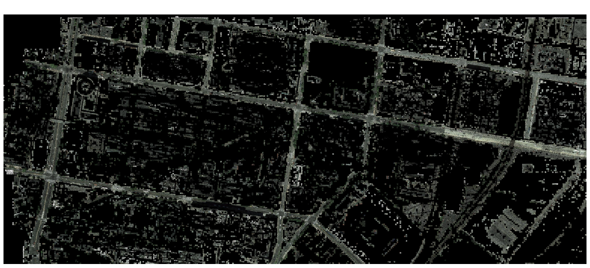
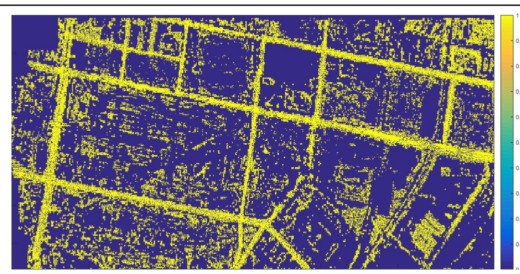
Feature	Masked RGB Image	Feature	Post Processing
Grey roof			YCbCr, Gaussian Blur, morphological operations, eccentricity
Green trees			HSV, Gaussian Blur, morphological operations
Dark Gray roof			YCbCr, Gaussian Blur, morphological operations, eccentricity
White roof			HSV, Gaussian Blur, morphological operations
Cream roof			YCbCr, Gaussian Blur, morphological operations
Roads			Lab, Gaussian Blur, morphological operations, eccentricity

Figure 2. Cost map for Driving

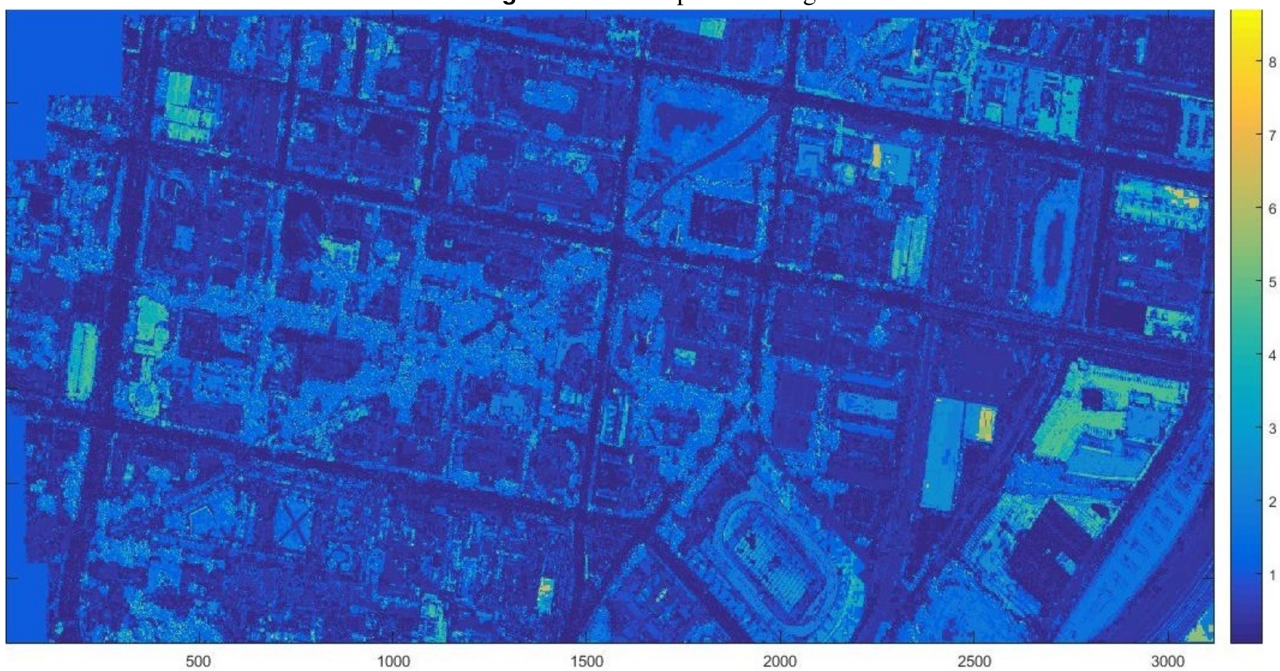


Figure 3. Cost map for Walking

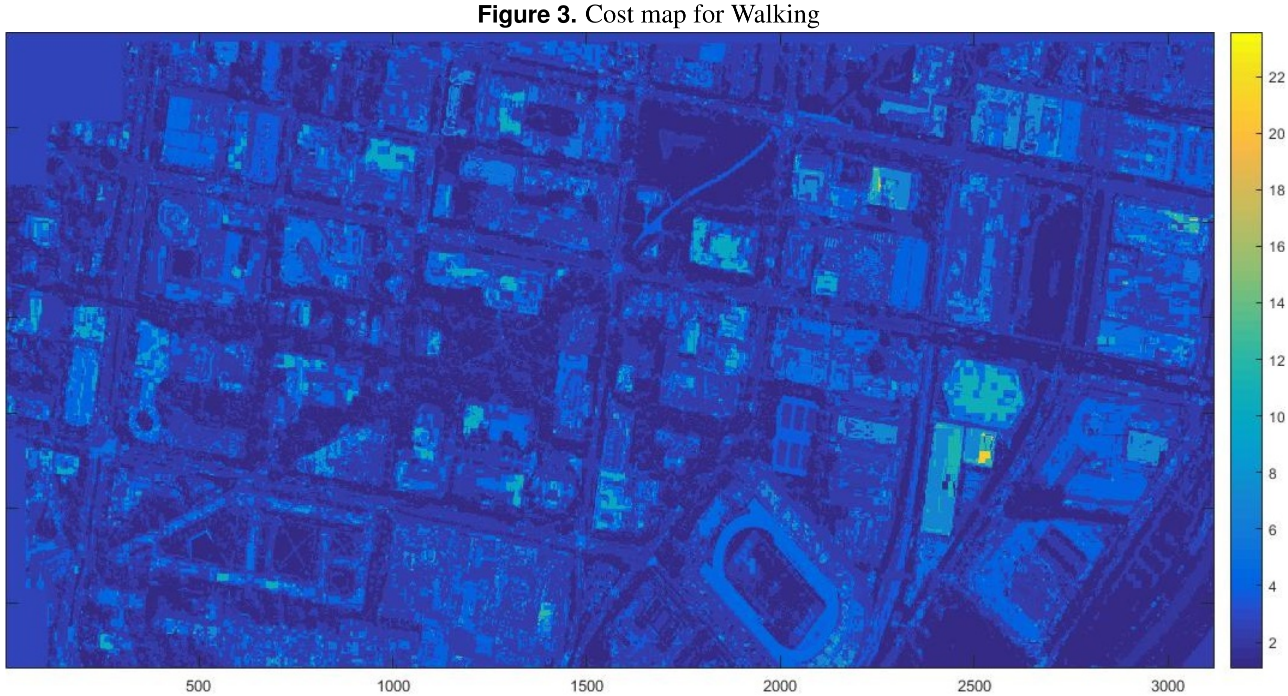


Figure 4. Test set results for Driving

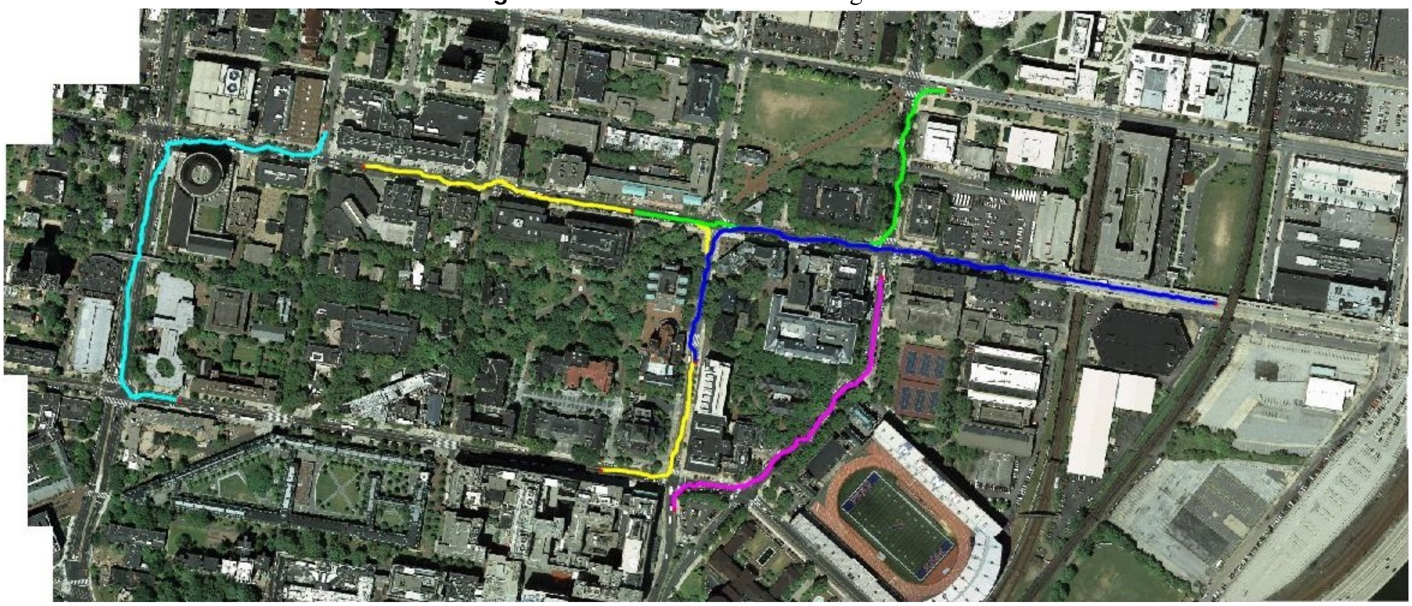


Figure 5. Test set results for Walking