Proceedings of the International Conference on Intelligent Computing and Communication for Smart World (I2C2SW-2018)
14-15 December 2018

Offline Navigation:GPS based assisting system in Sathuragiri forests using Machine Learning

N.Prabhu Ram
Department of ECE
Kongu Engineering College
Erode, India
prabhuramn4186@gmail.com

K.Sandhiya
Department of ECE
Kongu Engineering College
Erode, India
sandhiyasaras15@gmail.com

Vibin Mammen Vinod
Department of ECE
Kongu Engineering College
Erode,India
vibin.ece@kongu.edu

V.Mekala
Department of ECE
Kongu Engineering College
Erode,India
mekalav.ece@kongu.edu

***Abstract*—Location aware services are the need of the hour in places where communication facilities are limited. The lack of locationbased services in dense forested areas with poor network connectivity has resulted in severe losses both human life and monetary in many a places around the world. The case of floods in Sathuragiri forest in Virudhunagar district of Tamil Nadu in May 2015 is a classic case of such deficiencies. This paper presents an efficient and robust offline based navigation system suitable for the Sathuragiri forest which can be customized for varying geographical regions. The dataset for Sathuragiri forest was downloaded from Google Maps and applied K-means Clustering to identify the possible centroids. The calculations were carried out for two cases namely plane surfaces and sloppy terrains wherein the altitude was also taken into consideration. A standalone GPS module with other accessories was designed to indicate the current position. Using Haversine Formula, the distance to all the nearby centroids are calculated and the system displays the path with minimum weight for the traversal. Various methods of clustering have been performed and desired result was achieved from K-means clustering**

Keywords—Offline Navigation, K-Mean, Haversine,Centroid Positioning

I. INTRODUCTION

Sathuragiri forest in Viruthunagar district,TamilNadu which is famous for SundharaMahalingeswar temple where large number of people visit every year. Flood and natural calamities are unpredictable. This region is prone to sudden floods. There were many incidents of missing people. After eight pilgrims were washed away by flash floods in May 2015, the district administration of Madurai as well as Virudhunagar tightened security arrangements on the hill, such as over 1500 officials are involved in the work on the foothills and also a voluntary organization has also established VHF communication facility. However, it is difficult for the individual who lost their path due to natural calamities or wild life attacks and to reach safer zone.

With the development of mobile Internet, people get convenient service by electronic gadget. The application like google maps, imaps provides location based service of the current location by GPS positioning or network positioning. With the combination of GPS and using Google Map API, the navigation system provides functions such as current location, get the navigation route, address query and view historical location records. However the facility of getting path information through online navigation system is insufficient in these regions as it is remote areas. It is hard to load google maps to locate in the Sathuragiri forest due to absence of network services. Thus, in this region the offline based navigation system will be suitable.

Offline navigation system is designed by measuring great circle distance using Haversine formula inmachine learning technique. The Machine learning technique in which it deals with programming, where the system learns automatically and improves with its experience without being explicitly programmed. Machine learning is commonly classified into supervised and unsupervised learning. Supervised learning is a learning algorithm in which datasets being trained with respect to labeled values (called the training data set). Unsupervised learning is a type of machine learning algorithm used to draw inferences from datasets consisting of input data without labeled responses. The exploratory data analysis in finding patterns from the unlabelled data sets can be modeled by unsupervised learning method of cluster analysis. The unlabelled data sets are clustered into a groups by the measure of similarity metrics such as finding Euclidean distance or probabilistic distances.

II. LITERATURE SURVEY

Prathilothamai et.al.,[1] proposed "Offline Navigation : GPS based Location Assisting System", in which the application is used for assisting Visually impaired people to reach their destination in offline using magnetic sensor to guide them. In this entire route information database is downloaded by the administrator using Google direction APIs.

Asmita Singh et al.,[2] proposed "Offline Location Search using Reverse K-Means Clustering & GSM Communication", in which the input the centroid from ground reference position through mouse clicks on the geographical map, if desired location is found then corresponding image of the location will be loaded in the system to display.

Pharpatara et.al.,[3]proposed "Shortest path for aerial vehicles in heterogeneous environment using RRT", in which a special graph called tree is used. The starting and ending point is chosen and a tree is to be rooted. It Samples points at random and connects to the graph. This point searches for the nearest point and moves in straight line to it and also searches for collision free trajectory to reach the end point. From this methodology, an idea of estimating the shortest distance without any obstacle through RRT* algorithm has been inferred.

Shi Na et.al.,[4]proposed "Research on k-means Clustering Algorithm-An Improved k-means Clustering

Algorithm", in which K-means clustering algorithm is used to calculate the distance between each data object and all cluster centers in each iteration. The efficiency is improved by framing a plain data structure to buffer some information in every epochs, which is to be used on next epoch. This methodology can avoid computing the Euclidean distance of each data sets to the cluster centers repeatedly, thereby reducing the training time.From this methodology, an idea on theperformance of k-means clustering can be improved by constructing a simple data structure in storing information of centroids at every iterations in Euclidean space was derived.

Youguo Li et.al.,[5] proposed "A Clustering Method Based on K-Means Algorithm", in which the traditional K-means algorithm will randomly chooses the initial focal points and gather the remaining sample dots in accordance with the distance and classification will be iteratively repeated until reasonable classification is done. This method focuses on energy decrease in the direction of search, so if the initial cluster point is not proper the whole algorithm will sink into local point. The largest minimum distance algorithm is used to determine the K initial cluster focal points and then combined with traditional algorithm to accomplish better classification. From this methodology, an idea of determining K value in the K-means algorithm which will increase the stability and cluster precision.

III. Existing Methodology

In the existing method [1],an mobile application developed especially for visually impaired people. It uses Google's direction API for deriving the route data. The route map is downloaded based on the points selected by the administrator and these points are saved in the local database. To load another available paths, the user should the refresh the application to reload the dataset in local database. The data are downloaded as in JSON format and this requires less data usage as the whole map is not loaded.

The limitation in this methodology is refresh to reload data set into the local database system. This is due to accommodate an entire valuable data in the local database.

IV. Proposed Methodology

The above limitation in the existing methodology is overcome by using entire dataset into the local system as database. It uses Google's direction API for deriving the route datasets .The limitation of storing entire dataset can be managed by reducing the dimension from higher level to optimum level by using K-means clustering algorithm. The distance between unknown position to known position can be measured by using Haversine formula.

A. K-Means Clustering

K-means clustering is a type of unsupervised learning, which is used when there is unlabeled data (i.e., data without defined categories or groups). This algorithm is used to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. Data points are clustered based on feature similarity.

Suppose the target point is x and xi denotes the mean of the cluster Ci, the sum of square error function is defined as follows

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} |x - x_i|^2 \quad (1)$$

E is the sum of square error function of all objects in the datasets. The distance of the square error function is Euclidean distance, which is used for determining the nearest distance between each datasets and cluster center. The Euclidean distance between one vector x = (x1,x2,x3,... xn) and another vector y = (y1,y2,y3,...yn), the Euclidean distance d(xi,yi) is given by

$$d(x_i, y_i) = \left[\sum_{i=1}^{n} (x_i - y_i)^2 \right]^{\frac{1}{2}} \quad (2)$$

The algorithm works as follows,

- First initialize k points, called means, randomly.
- Categorize each item to its closest mean and update the mean's coordinates, which are the averages of the items categorized in that mean so far.
- The process is repeated for a given number of iterations and at the end, clusters are formed.

B. Haversine Formula

The Haversine formula determines the great-circle distance between two points (Latitude, Longitude) pairs on the sphere of earth.The formulae for calculations on the basis of a spherical earth (ignoring ellipsoidal effects) – which is accurate enough for most purposes [In fact, the earth is very slightly ellipsoidal; using a spherical model gives errors typically up to 0.3% [6]. It is a special case of a more general formula in spherical trigonometry, the law of haversines, that relates the sides and angles of spherical triangles

The Haversine formula is given by

$$hav(\Theta) = hav(\varphi_2 - \varphi_1) + \cos(\varphi_1) \times \cos(\varphi_2) \times hav(\lambda_2 - \lambda_1) \quad (3)$$

where

- φ_1, φ_2: latitude of point 1 and latitude of point 2,
- λ_1, λ_2: longitude of point 1 and longitude of point 2.

Let the central angleΘ between any two points on a sphere be

$$\Theta = \frac{d}{r} \quad (4)$$

where:

- d is the distance between the two points (along a great circle of the sphere)
- r is the radius of the sphere of earth.

To solve for the distance d, apply the inverse haversine hav^{-1} to the central angle Θ or use the arcsine (inverse sine) function, then from equation (4)

$$d = r \times hav^{-1}(h) = 2 \times r \arcsin(\sqrt{h}) \quad (5)$$

where $h = hav(\Theta)$

The haversine formula works good for numerical computation even at small distance, unlike calculations is based on the spherical law of cosines. The Earth radius "r" varies from 6356.752 km at the poles to 6378.137 km at the equator.

The datasets consists of Latitude, Longitude in signed decimal degree format instead of DMS(Degree Minute Second) format and altitude in meter.

C. Data Extraction and Preprocessing

The datasets are represented as in the form of Latitude, Longitude and Altitude as an array of set. The dataset are formatted into signed decimal degree as shown in Fig. 1.

```
Forest_Latitude_Longitude Roadmap with Dimensions and Value = (2451, 2)
 [[  9.76644    77.58524  ]
 [  9.76641    77.5853   ]
 [  9.76635    77.58552  ]
 ...,
 [  9.76582    77.73756  ]
 [  9.7038563  77.7161977]
 [  9.7658231  77.7375615]]
Forest_Latitude_Longitude_Altitude with Dimensions and Value = (9885, 3)
 [[   9.76885   77.6219   911.36   ]
 [   9.74163   77.62968  911.18   ]
 [   9.77331   77.59734  932.83   ]
 ...,
 [   9.75709   77.65314  625.96   ]
 [   9.76018   77.60395  625.89   ]
 [   9.77381   77.64456  368.46   ]]
```

Fig. 1. Dataset after preprocessed

The latitude and longitude values of roadways around Sathuragiri forest have been extracted from Googlemaps[7]. The dataset_1 of road maps can be exported in .kml or .kmz format. The latitude, longitude with altitude as dataset_2 are entered manually from enetplanet[8].Totally 2451 unique points of dataset_1(Latitude & Longitude) of roadways around Sathuragiri forest are extracted from Google maps exported file. The roadmap path is selected by user in the Google maps to be exported. Totally 9885 unique points of dataset_2 (Latitude, Longitude, Altitude) are collected from enetplanet at random clicks within the range of 100m between each points.

The dataset_1 can be extracted by parsing using minidom method from xml.dom package in python script. The dataset_1 have been preprocessed by removing splitters like "\n", " "etc., and reforming into numpy array using numpy package. The dataset in numpy format will be easy for numerical computation.

The altitude of mountain which are selected at various range. The distribution of altitudes are shown in Fig.2.

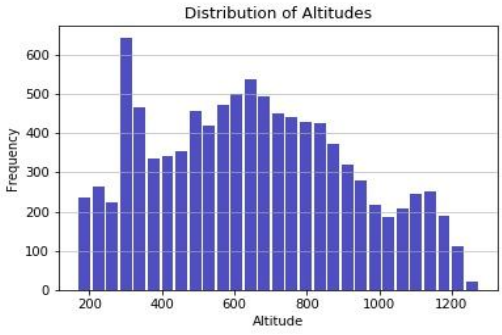

Fig. 2. Histogram of Altitude

In the Fig.2, x- axis represents the altitudes at various positions(Latitude, Longitude) in the forest and y-axis represents the frequent occurrence count of altitude of the forest.

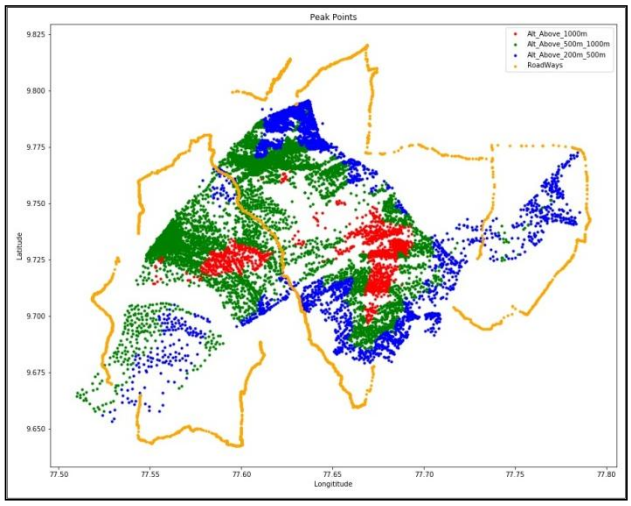

Fig. 3. Forest Region with Peak points

In the Fig.3, dataset_2 of 9885 unique points of forest region with different range of altitude is represented. The region in red color is the representation of LAT_LON with the peak points are above 1000m, the region in green color with the peak points are between 500m and 1000m, the region in blue color with the peak points are between 200m and 500m and the region in orange color is roadways around forest.

The size of dataset is very huge, when computing an entire datasets for measuring distance between every two points using (3) & (5) will consume more time and energy. The method to reduce the dimension of the dataset is by removing some of the points at random fashion, but it will not be a good solution as some needful points in the dataset will be vanished. Hence, an alternate approach is by clustering at appropriate size using k-means clustercan reduce the dimensions of dataset. Thus reducing the computational time in measurement and also power consumption on implementation on small handheld computer like raspberry pi board.

V. RESULTS & DISCUSSION

A. Reducing Dimension using K-means clustering

In the k-means clustering, every data objects will be assigned to certain cluster center[4]. An idea behind the dimension reduction is by considering cluster center points as dataset points. In this cluster center points is to be considered if it lies in the actual dataset, else other cluster center points will be discarded by which the dimensionality of an entire datasets will be reduced at optimum.

The k-means clustering is executed by using sklearn package in python. The sklearn package is from scikit-learn, a simpleand efficient tools for data analysis, an open source with build in numpy, scipy and matplotlib packages (supporting package for numerical computation). In this sklearn package, the k-means problem is executed by using Lloyd's algorithm or Elkan's algorithm[9], an improved version of k-means clustering.

The syntax of the k-means class in sklearn library is as follows

- Model = KMeans(<no. of clusters>,<method of initiation>,<random number generation for centroid initialization>).fit(<dataset>)
  - <no. of cluster> points will be initiated by the ratio of 1,2,6,24,120.
  - <method of initialization> will be kmeans++
  - <random number generation for centroid> will be 2 as an integer number will make the randomness deterministic.
  - <dataset> will be either forest region always or will use only for initial condition and for iteration the results of centroid centers will be used as dataset.
- Model.cluster_centers_ will returns the centroid points of each clusters.
- The sample code of kmeans cluster is shown on Fig.4.

```
kmeans_alt = KMeans(n_clusters=int(Forest_Lat_Lon_Alt.shape[0]/i),init = 'k-means++', random_state=1).fit(center_alt)
center_alt=kmeans_alt.cluster_centers_
labels_alt = kmeans_alt.labels_
```

Fig. 4. Sample Code

The Kernel size of the k-means cluster for the dataset_2 is iterated as trial and error method. It has been iterated from 2 to 4 and further the cluster of center points are very minimum to be as similar to that of actual datasets. It is finalized as the number of cluster centersto be equal to total size of an entire dataset_2 divided by 3.(i.e., Kernel size = 3295) and fitting with training dataset as clustered centers as tabulated in Table I. Similarly the kernel size of the k-means cluster for the dataset_1 is iterated and finalized as the number of cluster centers to be equal to total size of an entire dataset_1 divided by 3.(i.e., Kernel size = 817) and fitting with training dataset as clustered centersas tabulated in Table II.

The total dataset_2 is 9885 sets of latitude, longitude and altitude. At every iteration starting from iter_value 2 to 4, the kernel size are tabulated. Of this dataset_2 entire points of 9885, each data point is iteratively assigned to one of the K(Cluster center size) groups based on the features of similarity.

TABLE I. FITTING DATASET_2 WITH DIFFERENT KERNEL SIZE

Kernal Size	Actual Dataset =9885		Number of similar Cluster center value as same as actual datasets	
	iter_ Value	*KernalSize = Cluster center size*	*Fitting with entire Dataset (E)*	*Fitting with cluster centers (C)*
$\frac{Total\ Dataset\ Size}{iter\_Value}$ (E)	2	4942	3010	3004
	3	3295	1340	1432
	4	2471	751	754
$\frac{Cluster\ center\ Size}{iter\_Value}$ (C)	2	4942	3010	3004
	3	1647	312	249
	4	411	36	7

In the Fig. 5, it is clearly shows that the optimum measure is obtained on fitting cluster centers with an entire dataset_2. In that iter_value at 3 is chosen, since the number of similar cluster centers with respect to acutal dataset_2 is about 14.49%, which will be optimum in measuring elevation distance in the enitre forest with easy computation.

The total dataset_1 is 2451 sets of latitude and longitude. At every iteration starting from iter_value 2 to 4, the kernel size are tabulated. Of this dataset_1 entire points of 2451, each data point is iteratively assigned to one of the K(Cluster center size) groups based on the features of similarity.

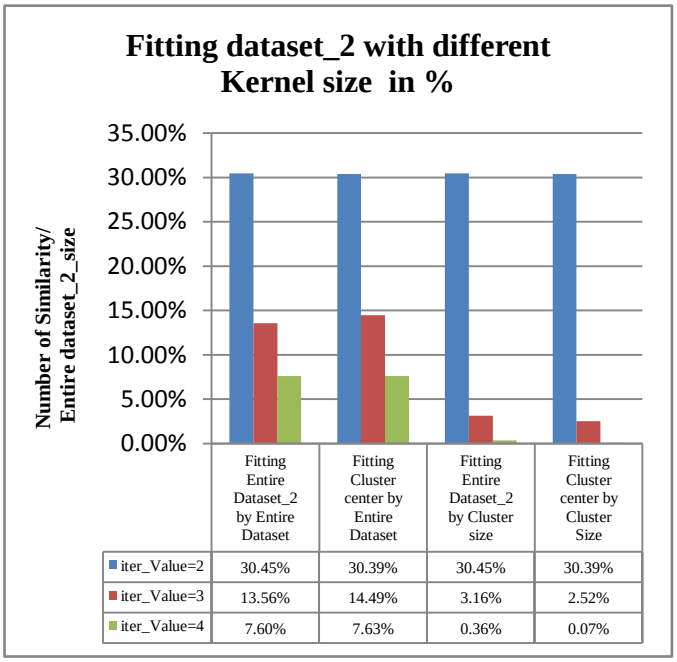

Fig. 5. Fitting Dataset_2 with different kernel size in %

When the kernel size is as same as size of the entire dataset (9885) will yield 100% the similar center values as that of entire datasets. But this number of dataset will take more computation and for tracing the direction from unknown to known position will be so tedious in computation on implementation in small computer.

In the Fig. 6, it is clearly shows that the optimum measure is obtained on fitting cluster centers with an entire dataset_1. In that iter_value at 3 is chosen, since the number of similar cluster centers with respect to acutal dataset_1 is about 13.02%, which will be optimum in measuring elevation distance in the enitre forest with easy computation

TABLE II. FITTING DATASET_1 WITH DIFFERENT KERNEL SIZE.

Kernal Size	Actual Dataset =2451		Similar Cluster center value as same as actual datasets	
	Iter_Value	*KernalSize = Cluster center size*	*Fitting with entire Dataset*	*Fitting with cluster centers*
$\frac{Total\ Dataset\ Size}{Iter\_Value}$	2	1225	676	678
	3	817	265	319
	4	612	139	158
$\frac{Cluster\ center\ Size}{Iter\_Value}$	2	1225	676	678
	3	408	52	58
	4	102	3	1

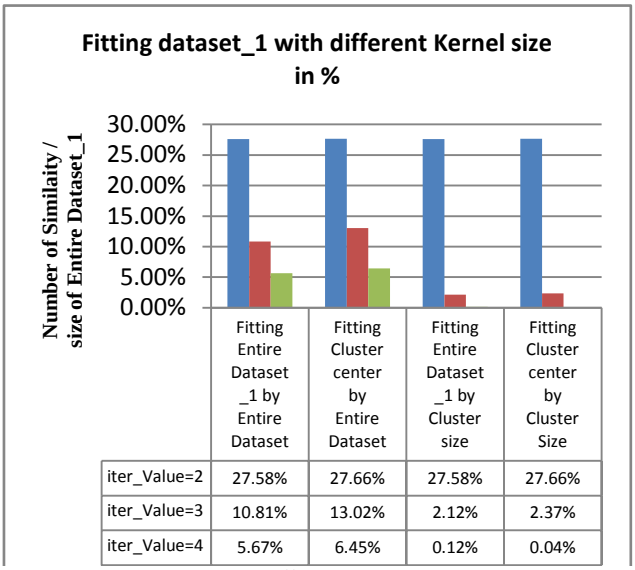

Fig. 6. Fitting Dataset_2 with different kernel size in %

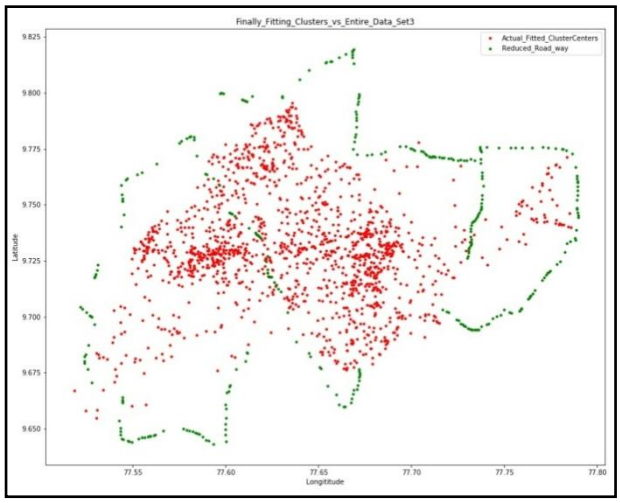

Fig. 7. Reduced Dataset by fitting dataset with kernel size divided by 3

In Fig.7, displays the scatter plot of reduced dataset_2 of latitude, longitude and altitude which is plotted in red color. The total number of data in the reduced dataset_2 as similar to that of actual dataset_2 is (1432,3) sets.

B. Steps of Proposed Algorithm

Step1: *Extraction of datasets and preprocessed into required format of computation*

Step2: *Reduction of dimension in the dataset vectors by using k-means clustering as discussed above section.*

Step3: *Obtain the target Latitude and Longitude point by using (5), in which considering the datasets latitude, longitude points of index at minimum distances.*

Step4: *Filter the dataset_2 set at desired direction of target point by go to Step 7.*

Step5: *Append the points from unknown points till reaching the target point.*

Step6: *Goto step 3 until dataset_2 is empty*

Step7: *Diff_Lat,Diff_Lon = abs(X-Y)*
X = UnknownPoint
Y = Target Point

Step8: *if(X>0)&&(Y>0):3rd Quadrant of dataset is filtered*

Step9: *if(X>0)&&(Y<0):4th Quadrant of dataset is filtered*

Step10: *if(X<0)&&(Y>0):2nd Quadrant of dataset is filtered*

Step11: *if(X<0)&&(Y<0):1st Quadrant of dataset is filtered*

Step12: *Go to Step 4.*

C. Target Latitude and Longitude Points from unknown Points

The target point is measured using (5) and top 10 minimum distance to reach the roadway has been listed below.

Top 10 Roadway points are [[9.7723377.67851],[9.7722777.67602],[9.7727377.68377], [9.7723677.6706],[9.7736277.68589], [9.7032277.71693], [9.7749377.6687],[9.702377.71896], [9.7759377.69366], [9.7019477.7199]]

Distance in meter are [4714.99, 4725.69, 4774.63, 4826.95, 4898.57, 5030.54, 5152.94, 5270.88, 5328.01, 5378.05]

Indexing 0th column refer to Latitude and 1st Column refer to Longitude of the roadway and the distance to travel each points are listed below in meter. In this first element is chosen as the target point.

D. Filtered Cluster center points towards target

Out of 1432 points of cluster centers, only certain points will be consider towards the direction of target point with respect to unknown points. The planar(flat) distance can be measured using Haversine formula(5) , but in the forest region, the surface will not be always be planer and hence elevation distance is to be measured by using Pythagoras theorem with the altitude from dataset_2 and the planar distance using (5) with respect to unknown point.

Test Case I:

The input of unknown point from GPS reading is taken in the format of latitude, longitude, altitude in DMS format and altitudes are measured in meter

Unknown point = [9.7,77.67,100] is given as input and nearest target point to reach is measured to be [9.67647 77.67173] is obtained using (5).

List of Points in between Unknown Points and Target Points are [[9.7, 77.67, 100][9.69688333, 77.67262, 221.7266667] [9.69685, 77.67449, 231.1575] [9.69186, 77.6767425, 181.4725] [9.69099, 77.68067, 217.87] [9.68923, 77.68385, 368.85] [9.68757, 77.68612, 226.51]] and displayed in Fig.8.

Distance required to reach each points in meter are [243,320,293,283,428,432]

Distance without elevation in meter: 2005.07

Distance with Elevation in meter: 1999.0

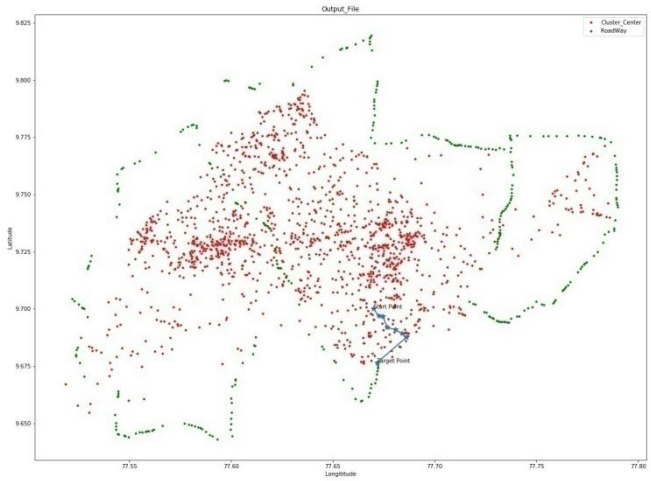

Fig. 8. Test Case output 1

Test Case II:

The input of unknown point from GPS reading is taken in the format of latitude, longitude, altitude in DMS format and altitudes are measured in meter

Unknown point = [9.75,77.7,100] is given as input and nearest target point to reach is measured to be[9.77138 77.71004833] is obtained using (5).

List of Points in between Unknown Points and Target Points are as follows and displayed in Fig.9.

[[9.75, 77.7, 100] [9.7671475, 77.7031725, 333.11] [9.76746, 77.7265575, 268.21]]

Distance required reaching points in meter are[347, 427.]

Distance without elevation in meter: 281.74

Distance with Elevation in meter: 774.0

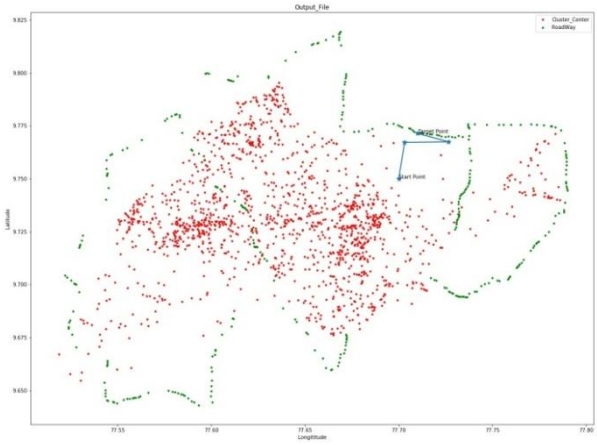

Fig. 9. Test Case output 2

VI. Conclusion and Future Scope

This paper was aimed to create an offline navigation system by the collections dataset and reducing to minimal dimension by using k-means cluster and measuring the shortest distance to navigate from any position where they are located to near by habitat location.

The work was implemented on python and in future have to evaluate the distance obtain by implementation on small computer like raspberry pi. The navigation has been designed specifically for Sathuragiri hills and it can be extended for many more region prone to sudden natural changes.

References

[1] Prathilothamai.M, PrashantR.Nair, R.Alakh.P.Singh, Aditya.R.N.S "Offline Navigation: GPS based Location Assisting system" Indian Journal of Science and Technology, Vol 9, pp 45., Dec 2016.

[2] Asmita Singh , Devendra Sowwanshi, "Offline Location Search using Reverse K-Means Clustering & GSM Communication" International Conference on Green Computing and Internet of Things,pp 1359-1364,Oct 2015.

[3] P.Pharpatara,B.Herisse, R.Pepy ,Y.Bestaoui, "Shortest Path for aerial vehicles in heterogeneous environment using RRT",IEEE International Conference on Robotics and Automation,pp 6388-6393,May 2015.

[4] Shi Na, Liu Xumin, Guan yong, "Research on k-means clustering algorithm - An improved k-means clustering algorithm", International Symposium on Intelligent Information Technology and Security Informatics, pp.63-67,2010.

[5] Youguo Li, Haiyan Wu, "A Clustering Method based on K-Means Algorithm",International Conference on Solid State Devices and Material Science,pp.1104 - 1109,2012.

[6] https://gis.stackexchange.com/questions/25494/how-accurate-is-approximating-the-earth-as-a-sphere#25580

[7] https://www.google.com/maps/d/

[8] http://www.enetplanet.com/

[9] http://scikit-learn.org/stable/modules/clustering.html