Identification of new road segments using a modified version of the k-means algorithm

G. Greco[1], C. Lucianaz[2], E. Vittaz[1], S. Bertoldo[2], O. Rorato[2], G. Perona[3] & M. Allegretti[1]
[1]Envisens Technologies s.r.l., Italy
[2]Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, Italy
[3]Consorzio Interuniversitario Nazionale per la Fisica delle Atmosfere e delle Idrosfere (CINFAI – Unità locale Politecnico di Torino), Italy

Abstract

The present work explains how to identify transformation and new road segments in existing electronic maps by using data coming from devices carried on board on different vehicles, by using a modified version of a standard clustering algorithm called k-means.

The working dataset appears as sparse clouds of points with their centres over the road segments, including other different data. Because of the spatial distribution of the points and in order to allow the algorithm to converge in a short number of steps, a simple modification has been implemented. With respect to the standard k-means algorithm which works with a fixed number of clusters, the present modified version works with a large initial number of clusters, with sizes defined a priori on the basis of the smallest possible road segment. The number of clusters is progressively reduced considering, for each steps, only the clusters including a number of points above a specific thresholds.

At the end of the algorithm, all the identified clusters are superimposed over a common map in order to validate if a new road segment is identified. The algorithm has been applied on different datasets acquired on the road network of Turin with good results allowing the identification of new road segments not present in the reference map and one-way roads (change in travel direction).

Keywords: ITS, transport system, modified k-means algorithm, traffic problem.

doi:10.2495/SC140691

1 Introduction

The importance of electronic road maps and their maintenance are increasing due to the widespread use of services such as route finding. Many people now completely rely on the satellite navigation system, not only to reach locations of which they do not know the position, but even to find the best route according to length, cost, traffic statistics, or current traffic conditions.

The route graph has therefore to be represented by a data collection more and more precise, faithful and, above all, updated frequently and in the shortest time possible.

For the companies that supply route maps, the digitalization of new areas and the maintenance of the old ones have a very high cost, which heavily affects the quality of the service. In this article we present a way to identify new road segments or other circulation changes (circulation sense, closed road, etc.) from the offline data of the flat management GPS trackers installed on board of thousands of vehicles.

2 Infrastructure

The infrastructure is based on car black box, featured with three basic elements: the GPS unit, the accelerometer for crash detection and the GSM/GPRS module for data transmission and communication with the call centre in case of crash or breakdown.

The back end of the system is a server collecting all the data in a geographic database.

The black boxes are installed in hundreds of thousands of vehicles for various purposes: tracking a vehicle's user, finding a stolen car, car accident investigation, saving on insurance rates. The equipped vehicles are of different types, indeed the black box is installed on heavy-goods vehicles, business cars and private cars.

The advantage is that a huge amount of data is obtained from different vehicles driven by users with different habits. For this reason, these data cover a wide area and represent a good statistics of the traffic existing on roads.

Considering the growing market of mobile devices installed on vehicles, the collected data will be more and more representative of the traffic condition and infomobility applications are emerging.

3 Data organization

Algorithms and procedure described in the following sections seek for the associations between set of data coming from different sources (e.g. the GPS points from the vehicles and the road map), in order to elaborate statistics and generate additional information. The starting point is the creation of a spatial database, where different sets of data are organized in tables. The spatial database is then optimized to store and query data representing objects defined in a geographic space.

Objects like streets, roundabout, bypass, highway, are decomposed in segments that describe their physical properties and represent the base element of the road graph. This dataset is considered static for a given time and form the *map data*. The other set of data, called in the following *raw vehicle data*, is composed by thousands of data that every minute are coming from the devices installed on board of vehicles, with different information, such as position, velocity, direction.

3.1 Map data

The entire road graph is composed by individual road segments. The record describing a segment is univocally identified by an index and contains the geographic coordinate of the starting point and the end point in WGS84 coordinate system, and the travel direction using an integer code (1=forward, 2=backward, 3=both directions).

Table 1: Example of road segment record.

SEGMENT_ID	PATH	DIRECTION
i	45.08612, 7.58123; 45.08623, 7.58134	1/2/3

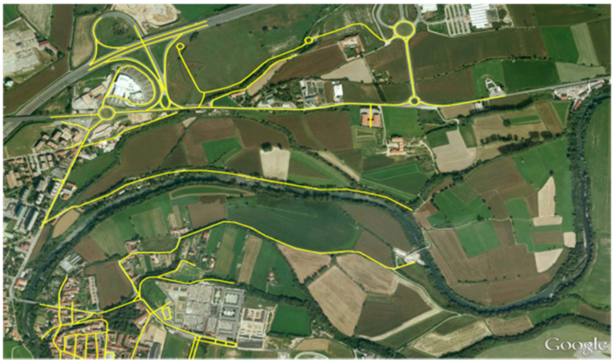

Figure 1: Graphic view of “map data”.

3.2 Raw vehicle data

The devices installed onboard on the vehicles, periodically send to the central server a record containing the following information:

- the timestamp: GPS time at which the record refers to;
- GPS coordinates: geographic coordinates in WGS84 coordinates system;
- the parameter DOP (Dilution Of Precision) computed by the on board GPS and useful to quantify the precision of the measurements;
- velocity vectors: North and East velocity vectors.

Table 2: “Raw vehicle data” table.

TIME	GPS coord (lat, lon)	DOP	Vx (m/s)	Vy (m/s)
gg/mm/aa-hh.mm:ss	45.086123, 7.581234	7	22	11

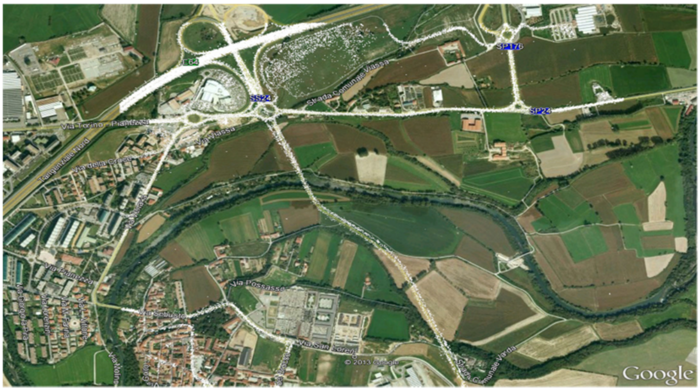

Figure 2: Graphics view of “raw vehicle data” table, each single white point (not singularly visible in the figure) represents a raw datum.

4 Procedures

In order to reach the expected results some procedures are implemented.

4.1 Raw data validation procedure

In a dense urban environment or in narrow valleys the Navstar GPS constellation coverage is not sufficient to guarantee a reliable reception at the GPS receiver. Moreover some multipath effects occur causing large positioning errors. Due to these errors, some records are not reliable. Since the DOP field, computed by GPS receiver, is a measure of positioning accuracy, it is possible to consider this value

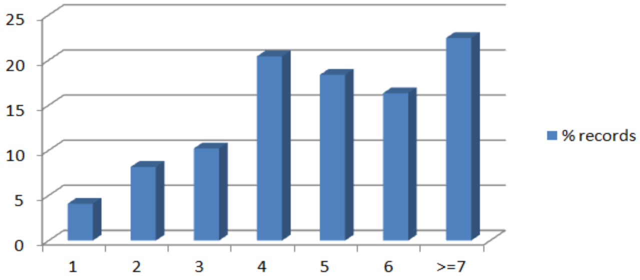

Figure 3: DOP quality distribution.

as a metric to accept or discard records, according to a specific threshold. In order to grant a high quality of the data used in statistics, the records where the DOP value is greater than 4 are discarded

Due to the nature of data sources, some records are not located in road segments, but may be located in public or private parking and they can add noise affecting the performance of the entire procedures. The solution to solve this problem is to select data according their associated speed value, assuming that moving cars are always in streets. Accordingly all data with speed under 5 km/h will be removed.

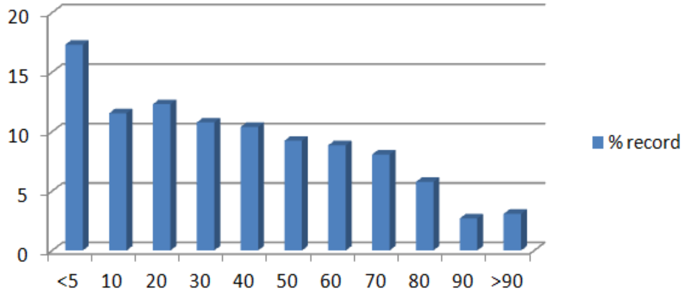

Figure 4: Speed distribution.

According to the raw data validation procedure described above and to the distributions reported in figures 3 and 4, it is possible to note that a large amount of data is discarded. However, this fact does not compromise the algorithms but increases their performance, because they work on a more statistically robust dataset.

4.2 Segment-matching procedure

The Segment-matching procedure assigns any data points of the GPS raw-vehicle-data table to a segment of a graph that represents the road network. The point is assigned to the nearest segment with the similar direction only if the distance is below a certain threshold value.

At the end of this procedure all records respecting the conditions in following eqn (1) and eqn (2) are signed with the identifier of the related segment in the field MATCHED of the correspondent record.

$$d < d_{MAX} \tag{1}$$

$$\theta < \theta_{MAX} \tag{2}$$

At this point two new subsets of data are defined:

- Matched points: all GPS points located in the road graph.
- Unmatched points: GPS points rejected because they parameters do not match the conditions.

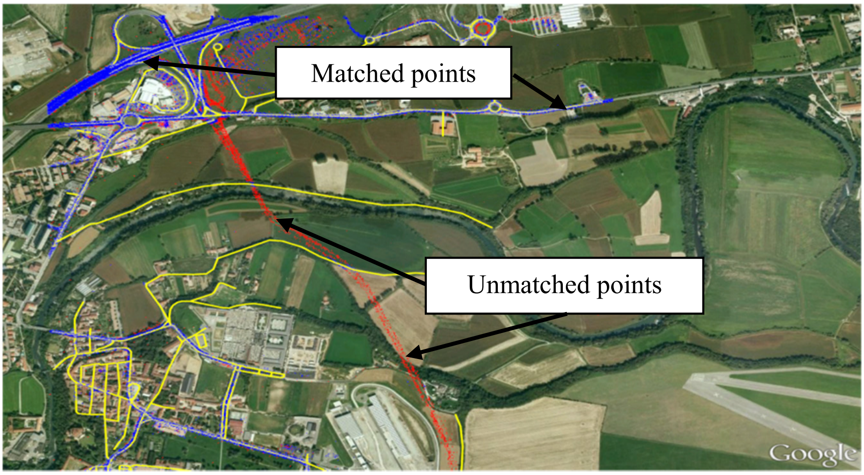

Figure 5: Example of matched points and unmatched points.

4.3 Data aggregation procedure

The goal of this procedure is to have a statistical information about the number of passages and the average speed in both directions, so it is possible to analyze week by week, month by month the trend of any particular road segment. To reach the target, a new data table is created and particular procedures are developed.

4.3.1 Aggregated data

Aggregated data is the result of the assignment of the raw data to a specific segment of the map. It represents, for each road segment, the statistical value of frequency of daily passages and average speed for each travel direction.

Table 3: Aggregated data table.

SEGMENT_ID	N_1	N_2	SPEEDAVG_1 (m/s)	SPEEDAVG_2 (m/s)
I	122	132	34	33

4.3.2 Aggregation procedure

For any point matched with segment *i* the direction is computed as in eqn (3) and eqn (4):

$$\partial = \text{atan2}(y,x) \tag{3}$$

$$\varphi = \partial - \vartheta i \tag{4}$$

where ϑi is the direction for segment *i*, so if the following eqn (5) is satisfied the direction is considered forward, otherwise backward:

$$|\varphi| < {}^{\pi}/_{2} \tag{5}$$

Any point matched with a segment can now be aggregated with corresponding points to create a statistics dataset from which an average speed (X_{avg}, Y_{avg}) for both directions is derived according to eqn (6):

$$\mathrm{SPEED}_{\mathrm{AVG}} = \frac{\sum_{k=0}^{n}(\mathrm{SPEEDi})}{\mathrm{n}} \tag{6}$$

where *n* is the number of points aggregated for a specific segment in a given direction.

5 Clustering procedure: the modified k-means algorithm

As shown in the previous Figure 5, the unmatched point rejected in the segment matching procedure highlight a new road. The goal of this algorithm is to identify new map segments, starting from a base algorithm known as k-means [1–4] and adding some heuristics about the context.

The next paragraphs describe the new developed algorithm.

5.1 Classes definitions

Two classes have been defined in the algorithm:

- Cluster: the class Cluster represent a group of points identified through latitude and longitude coordinates, direction and a Point3D list.
- Point3D: the class Point3D represent a group of raw data identified through latitude and longitude coordinates and direction.

5.2 Parameters definitions

Some parameters have been defined in the algorithm:

- DISTMAX: the radius of clusters.
- TETHA: the maximum angle identifying two points in the same direction.
- NCLUSTER: the number of clusters surviving through iterations. It is the acceptable final number of clusters. This number must be bigger than the number of possible new road segments.

5.3 Functions definitions

The modified k-means algorithm is based on two main functions:

- *int GetDistance(Cluster c, Point3D p):* return the geometric distance between the cluster *c* and the point *p* if:

  p.direction-c.direction<TETHA otherwise return ∞.
- *Cluster GetNearestCluster(ClusterList clusterslist, Point3D point):* return the nearest cluster from *clusters* to *point.*

5.4 Modified k-mean algorithm

The modified version of the k-mean clustering algorithm uses as a boundary condition the maximum number of clusters (NCLUSTER) that we want to create and the maximum number of iterations the algorithm goes through.
In order to better explain the algorithm the following pseudo-code is used:

```
1.  Copy all UNMATCHED points to startingPoint3DList
2.  Pull a record pi from startingPoint3DList
3.  Get the cluster nearest to pi:
    nearestCluster = GetNearestCluster(clusterslist, pi)
4.  If GetDistance(nearestCluster, pi)<DISTMAX
    then
            add pi to the Point3D list of nearestCluster
    else
            create new Cluster with same coordinates and direction of pi and
            add the cluster to clusterslist
5.  If startingPoint3DList is not empty go to step 2
6.  Sort clusterlist by size of point3D list from biggest to smallest
7.  Keep in clusterlist only the first NCLUSTER items
8.  For any cluster in clusterlist calculate the centroid of the cluster as the
    average of his point3D items as:

            $\text{cluster position} = \frac{\sum_{i=0}^{n} \text{point3D.position}_i}{n}$        (7)

9.  In the same way compute the mean direction as:

            $\text{cluster direction} = \frac{\sum_{i=0}^{n} \text{point3D.direction}_i}{n}$        (8)

    Now clusterlist is the representation of clustering at iteration j.

10. If clusterlist(j)=clusterklist(j-1)
    then
            the algorithm terminate
    else
            go to steps 1.
```

The final result is a list of clusters of NCLUSTER items with a position and a direction as well as a list of points. However in this way too many false positives occur. To improve the situation some filtering is performed on the basis of thresholds defined by number of points, variance of positions, variance of directions, in this way effectively limiting the false positive rate of suggested possible new road segments.

5.5 Results

In the next figure, on the left, is reported an example of a successful identification of new road segments. In an area near Turin analyzed one year ago, the procedure put in evidence new segments not present, at that time, in the available street map; however such roads are now visible on the most recent aerial photo on the right.

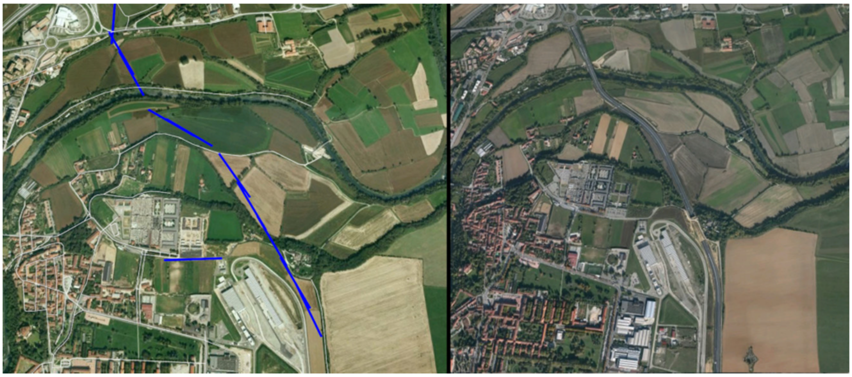

Figure 6: Graphic view of clusters discovered by algorithm compared to an updated view of the area.

6 Other results

The road graph described in section 3.2 and loaded in the GPS navigator to evaluate the route to destination is made up by road segments including their directions. Therefore if, in the course of time, one direction of the road is removed or, vice versa, is added, or the road itself is closed in both directions, the algorithm could be used to identify such changes.

It is just necessary to monitor the "aggregate tables" periodically. For example, let us assume that the number of transit in the field N_1 and N_2 of a segment *x* are approximately 200 every working day. Observing that the value collapse to 10 or 0, it means that this road is closed for some reason, (working in progress, crash, etc.); alternatively, let us assume that just N_1 is collapsed, corresponding to a specific direction. In both cases, consequently, the road graph can be updated.

In Figure 7 it was possible to observe a construction site obstructing one carriageway.

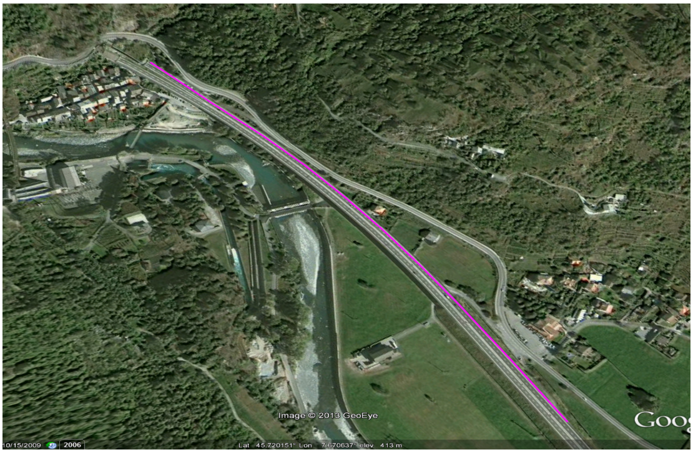

Figure 7: An example of direction of travel temporarily interrupted.

7 Concluding remarks

Due to the final algorithm conditions, the number of iterations could be infinite if there is no convergence. In order to avoid this problem, the solution the number of iterations is limited, and if the process exceeds this limit, no new roads are added to the road map.

The complexity of this algorithm is a consequence of the *GetNearestCluster* function. For each unmatched point it needs a full scan of the whole clusters list to get the nearest cluster to the geographical point. The theoretical worst case is given when the following eqn (9) is satisfied: it means that points have a uniform distribution in the area; in this case the complexity is O(n^2) [5].

$$\#cluster = \#\text{points} \tag{9}$$

When the number of points is too large, it is enough to split the area of survey in different smaller areas.

A critical point may be the speed of convergence of the algorithm: in our simulations and practical test cases the algorithm converges at most in 9-10 iterations.

Acknowledgements

The present work is the result of a cooperative project between CINFAI (Consorzio Interuniversitario per la Fisica delle Atmosfere e delle Idrosfere), Envisens Technologies s.r.l., Department of Electronics and Telecommunications of Polytechnic of Turin.

References

[1] Hartigan, J. A., Clustering algorithms, John Wiley & Sons Inc, 1975.

[2] Hartigan, J. A.; Wong, M. A., Algorithm AS 136: A K-Means Clustering Algorithm, Journal of the Royal Statistical Society, Series C, 28(1), pp. 100–108, 1979.

[3] MacQueen, J. B., Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press. pp. 281–297, 1967.

[4] Forgy, E. W., Cluster analysis of multivariate data: efficiency versus interpretability of classifications, Abstract in Biometrics, Vol. 21, Biometric Society Meeting, Riverside, California 1965.

[5] Vattani., A. k-means requires exponentially many iterations even in the plane, *Discrete and Computational Geometry*, 45(4), pp. 596–616, 2011.