

SpaceRank: Using PageRank to estimate location importance

Stefano De Sabbata and Stefano Mizzaro and Luca Vassena¹

Abstract. The recent advance of Web 2.0 and the rapid development of online social websites have created a large interest in the collection and analysis of social data. Although social data mining is usually done on the Web, social data do not exist in the virtual world only, but also in the physical one. Mining real people positions can allow to derive the importance, or popularity, of places in the real world. The collected data can be useful for many applications. In this paper we define SpaceRank, an approach to mining people positions, on the basis of the PageRank algorithm. Preliminary experimental results show that SpaceRank computes a notion of location importance that seems reasonable and is different from more classical indexes.

1 INTRODUCTION

Data mining of social data on the Web is much studied [1, 11]. However, social data do not exist in the virtual world only: they do also exist in the physical world. Example are the position of people, the usual route taken when driving to work, the usual path followed by people walking in the mountains, etc. An increasing amount of these data is becoming available because people usually take their mobile devices (phones, smartphones, PDAs, etc.) with them, and these mobile devices can (and indeed they do) leave a record of their past positions. Several technologies allow the generation of this huge amount of data: GSM, UMTS, HSDPA, etc. phones frequently exchange data with the network antennas; Bluetooth and Wi-Fi devices can see and be seen by other devices in close proximity; GPS and Galileo can determine the device position with good accuracy; triangulation can be used with all these technologies to provide a more accurate estimate; and so on.

Recently, a large amount of this kind of data has been collected and analyzed, deriving that it is generally more likely that a user will go in a place more related to his/her habits [6]. We propose to mine these data to derive the importance of the single locations. In this paper we propose a structured approach to estimate locations importance, given by either a single user or a community; our approach is based on the encoding of users behavior with graphs and the PageRank algorithm.

The paper is structured as follows. We first briefly survey related work in Section 2. Then, in Section 3, we describe our approach, named *SpaceRank*, in which the PageRank algorithm is exploited to mine people positions. In Section 4 we use our approach to analyze the data of two different kinds of experiments. At the end we present some conclusions and the future work.

2 RELATED WORK

2.1 Location importance

New mobile devices and new technologies allow to record the movements of single users. All these records are an interesting data source that can be used to determine past people positions; in turn, mining people positions can allow to derive the importance, or popularity, of places in the real world; and, finally, importance/popularity of locations in the real world can be very useful for several applications. For example, in traffic control [12], in itineraries planning [3], in network handover [13], and, in general, in location-aware applications [7], knowing which location is the more likely future destination would allow resources optimization, more efficient services, and even more features. Indeed, in the field of context-aware applications, although several factors are taken into account in order to define user's current context, location is usually the main one.

Past user (or users) behavior can be exploited in this respect: generally speaking, it is more likely that a user's position in the future will be a location where he or she has already been (or is used to go to) than a position where he or she has never been (or usually does not go to). Also, if one has to predict the user next position, it is generally more likely that a user will go in a place more related with his/her habits.

In order to define a location importance to a user, it is straightforward to think of three indexes:

- *#visits*: number of visits by the user in the location;
- *avgTime*: average time spent by the user in the location;
- *totTime*: total time spent by the user in the location.

These indexes can be computed on the basis of the data collected during a certain period of time (one day, one week, one month, etc.). This procedure can be tailored either to a single user or to a community: thus, the results will be related to the single user habits or to the community behavior. This approach is followed in several studies [2, 4, 5]. In particular, in [5] the prevision of the destination is computed by means of a Markov model; this suggested us to estimate the importance of the locations using another algorithm based on Markov chains, PageRank.

If these indexes are considered separately, they just give a partial vision of the user's behavior, and for some applications this could not be enough. For example, if we consider *#visits* only, a location where the user passes by without stopping and a location where the user stops for a long time would be indistinguishable. Similarly, using *avgTime* could put at the same level locations with rare visits and a location with frequent visits. The last index, *totTime*, could be a good importance index since a location should be visited frequently or for long periods to have a high *totTime* value. However, for example, we

¹ Department of Mathematics and Computer Science, University of Udine, Italy, email: stefano.desabbata@gmail.com, {mizzaro, vassena}@dimi.uniud.it

would like to give different importances to locations with the same *totTime* value, on the basis of one or both of the other two indexes.

One approach could be to use a combination (e.g., a linear combination) of the three indexes, or of some of them. However, this would rise the problem of the choice of both the indexes and the weights to assign to each of them. We propose a novel approach, based on the PageRank algorithm, briefly described in the next subsection.

2.2 PageRank

PageRank [10] is a well known algorithm for the analysis of links on a hyperlinked set of documents, whose goal is to measure the importance of each document within the set. PageRank assigns a numerical weighting to each element in the set and so represents the likelihood that a person will arrive at that page, randomly clicking. PageRank is used, together with thousands of other features, by search engines to rank the pages retrieved after a user query.

The analysis of the Web pages by PageRank starts from the adjacency matrix W of links between pages, and from a random walk on it (i.e., a model of a user randomly clicking on links and thus browsing the Web). Then a teleport factor is considered: it defines the possibility of not following the links in the page during the navigation and to jump from a page to any other page in the Web graph. This is modeled by the teleport matrix T , that encodes a graph where there is a direct edge between every possible couple of nodes and where all the edges weights are equal. This graph can be interpreted as a Markov chain where we have the same probability of performing a transition between any couple of Web pages. Then the PageRank algorithm is applied to the matrix:

$$PR = (1 - d) \times W + d \times T, \quad (1)$$

where d , called dumping factor, is the factor that indicates the probability, defined a priori, that the user teleports. Usually the dumping factor is 0.15. The teleport matrix T allows to have an ergodic Markov chain (provided that $d > 0$), and therefore to have a convergent PageRank computation (which would not be guaranteed on the original graph/matrix W).

The results of PageRank evaluation are encoded in a *pr* vector, which is the dominant eigenvector of PR matrix.

3 SPACERANK

We propose to build a matrix on the basis of both the geographical properties of the locations and the movements of the users, and then to apply the PageRank algorithm on the obtained matrix. We name our approach SpaceRank. The process of building the matrix is composed by three steps: creation of the transition matrix, creation of the matrix based on the registry of the historical data, combination of the two previous matrixes. We start by a more precise problem formalization.

3.1 Problem formalization

A first problem formalization is the following. The aim is to determine each location importance on the basis of past data; past data can concern either a single user (private importance) or a group of users (social importance). We start by concentrating on a delimited movements area, that we divide into a finite n contiguous sub-areas $L = \{l_0, l_1, \dots, l_{n-1}\}$, defined as *locations*. Locations could be of any size, although in this paper, for the sake of simplicity, we only deal with locations that have the same size. We suppose to record one

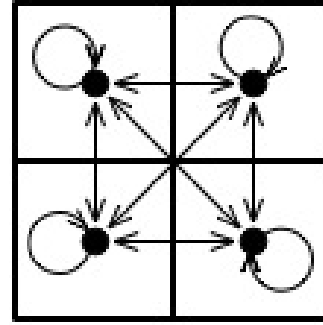


Figure 1. An example of a basic transitions graph (without edges weights) with four square locations.

or more users' movements, sampling at regular intervals position and speed. For each single user u we can create a registry with all the user's positions $R_u = \{r_0, r_1, r_2, \dots\}$, where each element in the registry is a triple $r_i = \langle t, l, s \rangle$ with the timestamp t , the location l , and the movement speed s . Then we define $R = \{R_0, R_1, \dots, R_m\}$, the set of all registries for the m users. We estimate the importance of the single locations on the basis of these data, namely: the locations L , their geographic features (essentially, if they are contiguous or not), and the registries R .

3.2 Transition matrix

We start by defining the transition matrix B that encodes neighborhood among locations. B is an $n \times n$ matrix, where n is the cardinality of L (the number of locations in the area of interest). This matrix encodes a graph where, given a node representing a location l , every outgoing edge directed to nodes representing locations close to l (l included) has the same weight. The transition matrix B in SpaceRank corresponds to a teleport matrix, used in PageRank, limited by real world two dimensional movements' constraints.

We suppose that the locations are contiguous and there is no disconnected location. The number of neighbors for each location is dependent on how the area is divided into locations; in our experiments we use square locations of the same size.

For example, the area in Figure 1 contains four square locations. In this case, each location will have as neighbors all the four locations in the area. The corresponding matrix B would be a 4×4 matrix containing the 0.25 value in each cell. In the example in Figure 2, we can tag the bottom-left location as l_0 , the bottom-center as l_1 , the bottom-right as l_2 , the middle-left as l_3 and so on. Here, the location l_0 , which is in a marginal position with respect to the other locations, will have only four neighbors, including itself, but the central location l_4 will have all the nine locations as neighbors.

In a more formal way, we define:

- N_i as the set of all neighbors of location l_i , i.e., the locations adjacent to the location l_i (included l_i itself);
- c_i the cardinality of N_i , for each N_i related to a location l_i belonging to L ;
- N the set of all N_i .

The matrix B is then defined as:

$$B[i, j] = \begin{cases} (\frac{1}{c_i}) & \text{if } l_j \text{ belongs to } N_i, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

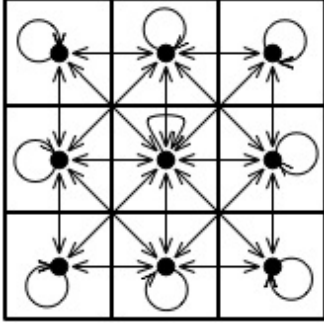


Figure 2. An example of a base transitions graph (without edges weights) with nine square locations.

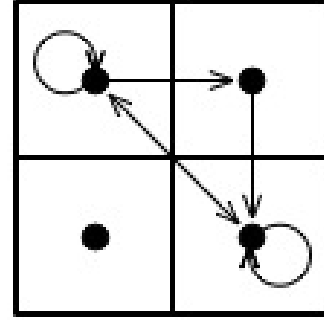


Figure 3. An example of an historical data registry based graph (without edges weights) with four square locations.

By considering edge weights as probabilities of users' movement between locations, $B[0, 4] = \frac{1}{4}$ will be the probability of a movement from l_0 to l_4 and $B[4, 0] = \frac{1}{9}$ will be the probability of a movement from l_4 to l_0 .

This initial graph can be considered an ergodic Markov chain, because each node/state is:

- aperiodic: its period is 1, so the state must not occur in multiples of a certain number of time steps;
- positive recurrent: there is a zero probability that we will never return back to this state in an infinite random walk and the number of steps between first visit and first return time is finite.

Let us remark that, unlike PageRank on the Web where there is the possibility to jump to any other page, B encodes is a proximity notion based on the conformation of the physical space. Indeed, although teleport does model something meaningful in the virtual world (i.e., a user going to a specific URL besides those listed in the links on the page), it would not be reasonable in the physical space.

3.3 Matrix based on the historical data registry

We now propose a general definition of the matrix A that encodes the historical data registry. Some variants are discussed below.

The matrix A , that encodes the users habits, is an $n \times n$ matrix, where n is the cardinality of L (the number of locations in the area of interest). It can be defined incrementally: let us start with $A[i, j] = 0$ for each i and j value between 0 and $n - 1$. Then, given the set R , for each registry R_u belonging to R , for each pair of temporally subsequent elements r_x and r_{x+1} in R_u , the update $A[i, j] = A[i, j] + 1$ is performed, where l_i is the location in the triple r_x and l_j is the location in the triple r_{x+1} .

This update, given a user's two temporally subsequent locations (recorded with an interval time equal to the temporal granularity used for the data recording), increases the probability of the transition between the two locations. If the two elements of the historical registry are related to the same location ($i = j$), the probability that the user will remain in that location will increase. The underlying idea is to adapt the edges weights to the habits of the user(s). To interpret the edge weights in the graph as the probability transitions in the corresponding Markov chain, the obtained matrix has to be normalized, so that the sum of the values of a single row will be equal to 1. The simplest, and probably most reasonable, normalization is to divide each value for the sum of the values of its row. In Figure 3 we have an example of a result graph, without edges weights.

3.4 Combination of the matrixes

During the navigation between locations, the user cannot move to any location in the interest area, but he can just move to the location adjacent to his actual one. With respect to the collected data, the user can move from a location to another adjacent even if the transition has never been done, or even if last location has never been visited before. The matrix B encodes this kind of behavior not congruent with the observed habits. We consider d the factor of "not regular" behavior. The matrix to which PageRank is applied to compute the importance of the locations is:

$$SR = (1 - d) \times A + d \times B \quad (3)$$

At first, the value $d = 0.15$, can be taken into consideration as factor to combine the two matrixes. A lower value gives more importance to the observed user habits, while a higher value gives more importance to a not regular behavior. With $d = 1$ we have $SR = B$, a behavior totally independent from the collected data, while with $d = 0$ we have $SR = A$, a behavior based totally on the data collected. In this last case, the Markov chain could be not ergodic, because some nodes/states could be not positive recurrent or periodic. Therefore, our SpaceRank evaluation for all locations can be encoded in an sr vector, which is the dominant eigenvector of SR matrix.

3.5 Variation in the encoding of the collected data

In the formalization of the problem, we decided to record users' movements by sampling at regular intervals position and speed. However, in the previous analysis, speed has not been taken into consideration. We now show how this factor can be exploited in our formalization.

Speed can be used to discriminate if the user is moving or if he is standing still in a location. For example, assuming that we have registered two temporally subsequent elements with the same location, $speed > 0$ suggests that user is moving, while $speed = 0$ suggests that the user is standing still in the location. This last situation is encoded in the location graph as direct edges from a location to the same one (loop).

In the creation of the matrix A , if we consider as permanences (i.e. loops on the same location) only the records with $speed = 0$, the importance of locations where the user has stopped is increased, while the importance of locations where the user has just passed through is decreased. In the same way if we consider only the records with $speed > 0$, the results, obtained with the computation of the importance, will highlight the areas that require a high amount of time to

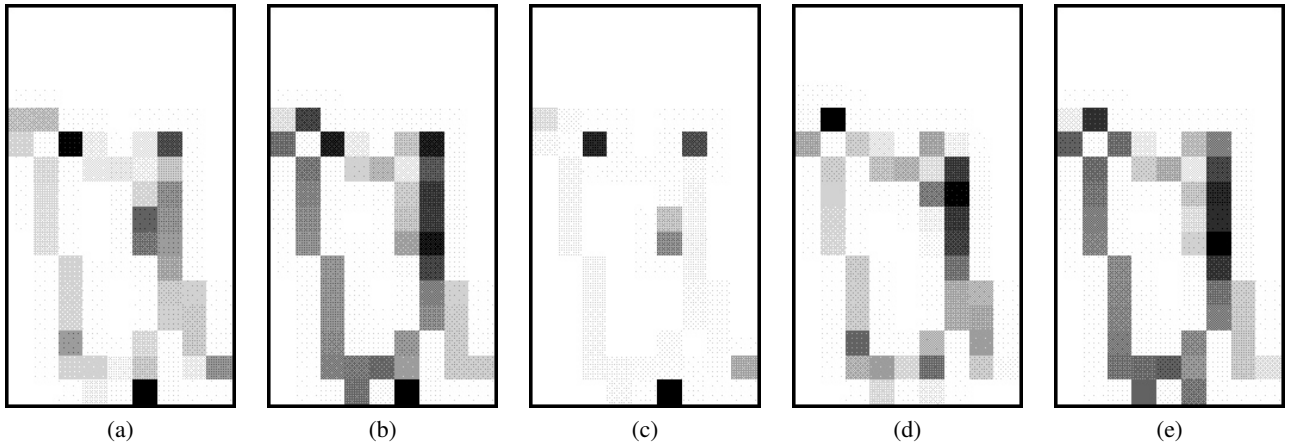


Figure 4. SpaceRank results for the small data set (a); considering: only visits with zero speed as stays (b); the total time of visits with zero speed as stays (c); the total time of visits with non zero speed as stays (d); and without any kind of stays (e).

be crossed, reflecting locations with high traffic density because of traffic jams.

4 PRELIMINARY EXPERIMENTS

We applied SpaceRank to two different data sets: a small data set based on the habits of one of us, and a larger data set derived from <http://www.ecourier.co.uk/>. We now briefly analyze the data of the two experiments.

In following experiments we use $d = 0.05$ in order to give more importance users' habits.

4.1 A small data set

In this first small example, the log data concern a typical week of one of the authors of this paper. The relevant area, Udine town and his surroundings, is divided into 9×16 squared locations, each of them having a surface of 700 square meters. The SpaceRank indexes for these locations are obtained using $d = 0.05$ and are shown in Figure 4(a). The most important location (darker squares in the location, in the bottom part of the figure) is the home of the user; another high importance location is University (in the top left hand side), where the user stays about 9 hours per day 5 days per week. Other locations with a somehow high importance have been visited for hours by the user. The lighter gray level locations are the roads where the user drives usually. White squares locations have never been visited.

If we consider as real stays only those with speed equal to zero, the locations having importance greater than zero are the same as above, but without the locations used only as paths (like roads) to reach some destination, whose importance is zero like never visited locations. However, these results hold only when we compute the number of stays using *totTime* (Figure 4(c)). Considering *#visits* as number of stays (Figure 4(b)), we obtain a sensible increment of the importance values for those locations used as paths.

If, conversely, we consider only stays with non zero speed (Figure 4(d)), the higher importance is obtained for the locations that needed more time to be traversed. Finally, by not taking into account the permanence in a location (i.e., without loops in the graph), the higher importance locations are transit locations and joints for the various trajectories followed by the user (Figure 4(e)).

4.2 eCourier data set

In this second case study, we compute SpaceRank on the GPS data gathered by the eCourier couriers and made available by API on the <http://api.ecourier.co.uk/> URL. Data concern one typical workday. The area taken into account is divided into 50×50 square locations having sides with length of about 350 meters. We used $d = 0.05$. The results are shown in Figure 5(a). We also compared the obtained SpaceRank values with the other indexes: *#visits* in Figure 5(b), *avgTime* in Figure 5(c), *totTime* in Figure 5(d), and a linear combination (with equal weights) of *#visits* and *totTime* in Figure 5(e). Table 1 shows the correlation values among the indexes, that are quite low for SpaceRank. We can conclude that although SpaceRank computes a notion of location importance that seems reasonable (see the figures), it can't be approximated accurately with any of the other indexes (see the table).

	<i>avgTime</i>	<i>#visits</i>	<i>Lin. comb.</i> <i>time-visits</i>	<i>SpaceRank</i>
<i>totTime</i>	0.99	0.98	0.99	0.35
<i>avgTime</i>		0.97	0.99	0.34
<i>#visits</i>			0.98	0.34
<i>Lin. comb. time-visits</i>				0.35

Table 1. Linear correlations among the various indexes.

5 CONCLUSIONS AND FUTURE WORK

We have emphasized the importance of mining social data in the real world and we have proposed a procedure to compute a new index for location importance assessment. The procedure, named SpaceRank, is based on computing PageRank on a matrix obtained on the basis of location proximity and users habits. Preliminary experimental results show that SpaceRank computes a notion of location importance that seems reasonable and is different from more classical indexes.

In the future, we plan to use SpaceRank as a basis for some destination prediction algorithms. In [4], SpaceRank results could be used in place of visits frequency in *Location Criterion*, a part of the proposed algorithm. In [8], SpaceRank results could be used in place of

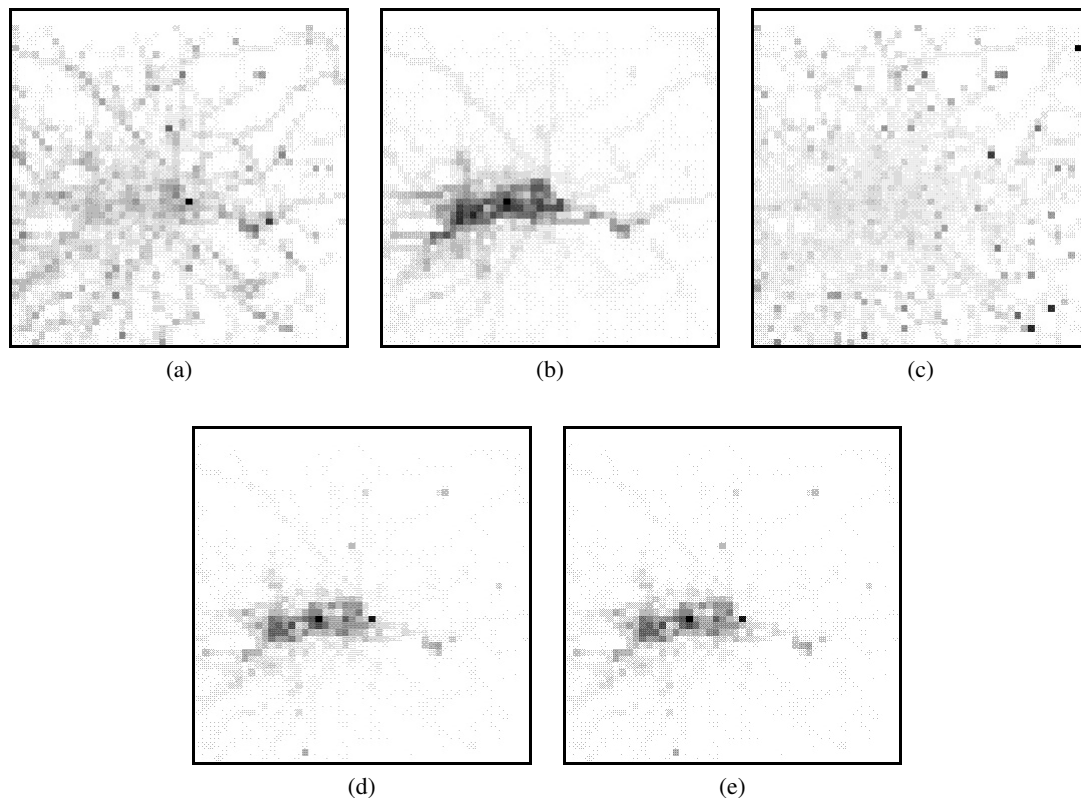


Figure 5. SpaceRank results for the eCourier data set (a); the same results using $\#visits$ (b), $avgTime$ (c), $totTime$ (d), and a linear combination (with equal weights) of $\#visits$ and $totTime$ (e).

the *Ground Cover Prior probability distribution* as a-priori location importance. In the location-aware application field, in [9], SpaceRank could be used as an additional filter for retrieved information. We are also developing a novel destination prediction algorithm on the basis of metaphors from physics, and we will apply SpaceRank to it. We also plan to test this index in several social data mining environments, both with respect to online 3D games and the real world, and in the general context-aware application world. Finally, we will use Kleinberg's HITS algorithm, besides PageRank, for location importance assessment.

References

- [1] Brian Amento, Loren Terveen, Will Hill, Deborah Hix, and Robert Schulman, 'Experiments in social data mining: The TopicShop system', *ACM Transactions on Computer-Human Interaction*, **10**(1), 54–85, (2003).
- [2] Daniel Ashbrook and Thad Starner. Using GPS to learn significant locations and predict movement across multiple users, 2003.
- [3] Michel Bierlairea and Emma Frejinger, 'Route choice modeling with network-free data', *Transportation Research Part C: Emerging Technologies*, **16**, 187–198, (2008).
- [4] J. Chan, S. Zhou, and A. Seneviratne. A QoS adaptive mobility prediction scheme for wireless networks, 1998.
- [5] Nathan Eagle and Alex (Sandy) Pentland, 'Reality mining: sensing complex social systems', *Personal and Ubiquitous Computing*, **10**(4), 255–268, (2006).
- [6] Marta C. González, César A. Hidalgo, and Albert-László Barabási, 'Understanding individual human mobility patterns', *Nature*, **435**(7196), 799–782, (2008).
- [7] Mike Hazas, James Scott, and John Krumm, 'Location-aware computing comes of age', *Computer*, **37**(2), 95–97, (2004).
- [8] J. Krumm and E. Horvitz, 'Predestination: Inferring destinations from partial trajectories', in *Lecture Notes in Computer Science*, ed., Springer, volume 4206, pp. 243–260, (2006).
- [9] David M. Mountain, 'Spatial filters for mobile information retrieval', in *GIR '07: Proceedings of the 4th ACM workshop on Geographical information retrieval*, pp. 61–62, New York, NY, USA, (2007). ACM.
- [10] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, 'The PageRank citation ranking: Bringing order to the web', Technical report, Stanford Digital Library Technologies Project, (1998).
- [11] Toby Segaran, *Programming Collective Intelligence*, O'Reilly, 2007.
- [12] Michael A. P. Taylor, Jeremy E. Woolley, and Rocco Zito, 'Integration of the global positioning system and geographical information systems for traffic congestion studies', *Transportation Research Part C: Emerging Technologies*, **8**, 257–285, (2000).
- [13] Z. Zaidi and B. Mark, 'Real-time mobility tracking algorithms for cellular networks based on kalman filtering', *IEEE Transactions on Mobile Computing*, **4**(2), 195–208, (2005).