CLUSTERING THE MOBILE PHONE POSITIONS BASED ON SUFFIX TREE AND SELF-ORGANIZING MAPS

*Jan Martinovič, Tomáš Novosád, Václav Snášel, Peter Scherer, Petr Klement**, *Roman Šebesta*†

Abstract: In this article we present a novel method for mobile phone positioning using a vector space model, suffix trees and an information retrieval approach. The algorithm is based on a database of previous measurements which are used as an index which looks for the nearest neighbor toward the query measurement. The accuracy of the algorithm is, in most cases, good enough to accomplish the E9-1-1 standards requirements on tested data. In addition, we are trying to look at the clusters of patterns that we have created from measured data and we have reflected them to the map. We use Self-Organizing Maps for these purposes.

Key words: *Mobile phone positioning, GSM, suffix tree, Self-Organizing Maps, clustering*

Received: September 1, 2011
Revised and accepted: August 2, 2012

1. Introduction

Nowadays there is a wide variety of methods to determine mobile station location. We can broadly divide the localization systems into three main categories – network based, handset based and hybrid systems. The network based systems use the measurement of signals, and these can be applied to any cellular system such as GSM. The most important methods are based on measurement of signal propagation time (i.e. Time Of Arrival – TOA, Fig. 1), Time Difference Of Arrival (TDOA), Angle Of Arrival (AOA) and carrier phase, etc. [4]. The position of the mobile station is then calculated by techniques like trilateration, multilateration

*Jan Martinovič, Peter Scherer, Tomáš Novosád, Petr Klement, Václav Snášel
Faculty of Electrical Engineering and Computer Science, Department of Computer Science, VŠB–Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic, {jan.martinovic, tomas.novosad, vaclav.snasel, peter.scherer, petr.klement}@vsb.cz
†Roman Šebesta
Faculty of Electrical Engineering and Computer Science, Department of Telecommunications, VŠB–Technical University of Ostrava, 17. listopadu 15, 708 33 Ostrava – Poruba, Czech Republic, roman.sebesta@vsb.cz

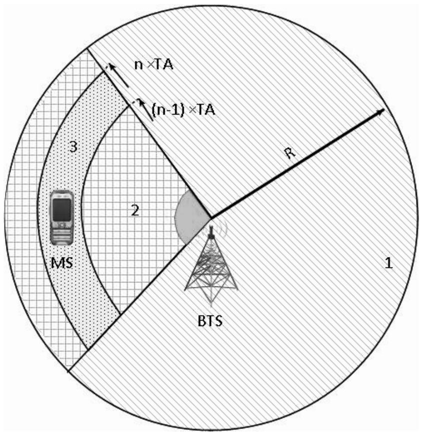

Fig. 1 *Basic localization method in GSM network.*

and triangulation. Several other methods based on previous ones also exist and some of them have become standard [40, 31]. The main disadvantage of all of these approaches is the additional hardware requirements to wireless network infrastructure such as Location Measurement Unit (LMU) devices or Angle Measurement Unit devices (AMU). These electronic devices must be installed on every Base Transceiver Station (BTS) and are also very expensive. On the other hand, these techniques are very accurate and successfully meet all of the E9-1-1 standards.

In handset based systems the position of a mobile phone is recognized upon the parameters which the mobile station receives from the wireless network. In GSM networks, these parameters include Cell IDentity code (Cell ID), Base Station Identity Code (BSIC), Received signal Level (RxLev), Timing Advance (TA), Broadcast Control CHannel (BCCH), etc. The mobile station receives the parameters from the serving station and is able to monitor up to next six neighboring stations as well. One of the main advantage of this system is that it does not require any additional hardware. It has been developed in many research projects in this field and it still remains a great challenge in these days in computer science. For example in [24, 25] the authors used artificial neural networks and radio signal strength to locate the mobile device. Another popular approach is space segmentation [26], application of fuzzy logic [16, 29, 12], hidden Markov models and pattern recognition methods [10, 11], data mining techniques [1, 18], probabilistic methods [27, 28], and many others.

The last category of the three localization systems consists of hybrid positioning systems which use a combination of network-based and handset-based technologies for location determination. Currently the most frequently used would be probably Assisted GPS (A-GPS) [2], which uses both GPS and network information to compute the location of the mobile station. Hybrid-based techniques give the best accuracy of the three but inherit the limitations and challenges of network-based and handset-based technologies as well. Different hybrid positioning systems are currently being developed and used in services from Navizon, Xtify, PlaceEngine, SkyHook, Google Maps for Mobile for applications in smart-phones including Apple iPhone, HTC Desire, etc. The main disadvantage of these systems is a necessity of a GPS module installed in the mobile device, which is not a common standard in these days yet.

In this article we introduce a novel method for determining mobile station location in handset based systems. We use values of parameters which the mobile phone receives from the wireless GSM network (from the serving station and six neighboring base transceiver stations) as an input to our algorithm – namely base station identity code, cell identity code, broadcast control channel and timing advance. The method is based on a database of such measurements collected in the desired location with GPS positions for each such measurement. All measurements in the database are indexed by suffix tree data structure. After that we extract the attributes from the suffix tree and build a vector model. For evaluation of our method we use measurements which were not previously indexed. The result of the query in the vector model (determined location of a mobile phone) is simply the first nearest neighbor measurement returned by the system. In addition, we are trying to look at the clusters of patterns that we have created from measured data and we have reflected them to the map. We use Self-Organizing Maps for these purposes.

2. Experimental Data

The data for our experiments were collected in the Czech Republic in the city of Ostrava, location Ostrava – Poruba and adjacent city parts. The scanned area is about 16 km^2.

First of all, we need to record the parameters which are received by mobile station from the wireless network and exact GPS positions. This task is done by the device we call the NAM Logger from NAM Systems company. The NAM Logger is a device for measuring GSM parameters from the wireless network and can determine exact GPS position at a given (measured) place, actual velocity, time of measurement, altitude, identification of country and mobile operator – Mobile Country Code (MCC) and Mobile Network Code (MNC). Next it gives us the following parameters from the serving BTS:

- Cell ID – Cell (antenna) IDentity code
- BCCH – Broadcast Control CHannel (It represents the ARFCN that shows the Absolute Radio Frequency Channel Number, which identifies the BCCH carrier.)
- RxLev – Received signal Level
- BSIC – Base Station Identity Code
- TA – Timing Advance

and the following parameters from 6 neighboring BTS:

- BCCH – Broadcast Control CHannel (It represents the ARFCN that shows the Absolute Radio Frequency Channel Number, which identifies the BCCH carrier.)

- RxLev – Received signal Level
- BSIC – Base Station Identity Code

Thanks to this device we are able to measure the parameters above at a given place with an exact GPS position. NAM Logger uses external GPS and GSM antennas. As the end, terminal representing Mobile Station (MS) is used as a common mobile phone. The active call is performed from the site of MS to NAM Logger (see Fig. 2).

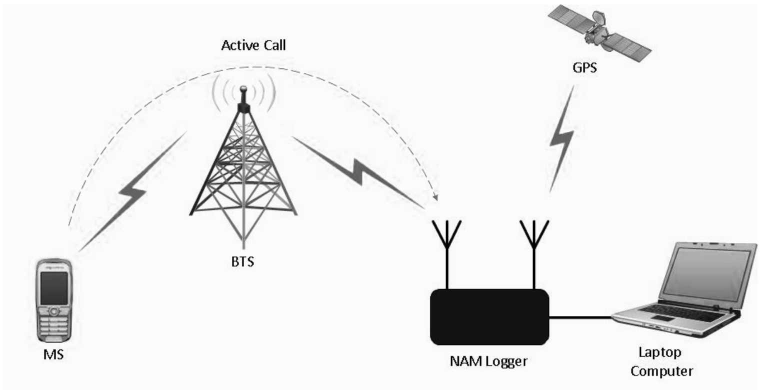

Fig. 2 *Measurement tools for data logging.*

Experimental data for this article were recorded by the process described above. Data were separated into a so-called training set and testing set. The training set does not contain any samples of the testing set and vice versa. The training set consists of 7855 measured samples. We have experimented with different combinations of measured parameters as an input to our algorithm. After a series of experiments we have found the best attributes for indexing are:

- BSIC – Base Station Identity Code from serving and neighboring BTS
- BCCH – number of broadcast channel from BTS, which provides the signaling information required by the mobile phone to access and identify the network
- TA – Timing Advance from serving BTS
- Cell ID – Cell IDentity code from serving BTS

2.1 Data encoding for indexing

The objective of this stage is to prepare the data for indexing by suffix trees. The suffix tree can index sequences. The resulting sequence in our case is a sequence of non-negative integers. For example, let us say we have two samples of a measurement (two sequences of words) from a NAM Logger device:

{34 29 758F 4 34 29 01 60 }
{34 29 76B4 3 34 29 FF 57 }

After obtaining this sequence of 16 words in this case (measured parameters), we create a dictionary of these words (each unique word receives its own unique non negative integer identifier). The translated sequence appears as follows:

{1 2 3 4 1 2 5 6 }
{1 2 7 8 1 2 9 10 }

In this way, we encode each sample of measurement from a NAM Logger device. This task is done for training set samples as well as for every sample from test sets. Now we are ready for indexing training samples using suffix trees.

3. Background

In this section we describe theoretical background required to understand our algorithm. It consists of high level description of Vector Space Model, Suffix Trees and Self-Organizing Maps as well.

3.1 Vector space model

The vector model [3] of documents was established in the 1970s. A document in the vector model is represented as a vector. Each dimension of this vector corresponds to a separate term appearing in document collection. If a term occurs in the document, its value in the vector is non-zero.

We use m different terms $t_1, \dots, t_m$ for indexing N documents. Then each document d_i is represented by a vector:

$$d_i = (w_{i1}, w_{i2}, \dots, w_{im}) \,, \tag{1}$$

where w_{ij} is the weight of the term t_j in the document d_i. The weight of the term in the document vector can be determined in many ways. A common approach uses the so called $tf \times idf$ (Term Frequency $\times$ Inverse Document Frequency) method, in which the weight of the term is determined by these factors: how often the term t_j occurs in the document d_i (the term frequency tf_{ij}) and how often it occurs in the whole document collection (the document frequency df_j. Precisely, the weight of the term t_j in the document d_i is [15]:

$$w_{ij} = tf_{ij} \times idf_j = tf_{ij} \times \log \frac{n}{df_j}, \tag{2}$$

where idf stands for the inverse document frequency. This method assigns high weights to terms that appear frequently in a small number of documents in the document set.

An index file of the vector model is represented by matrix:

$$D = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1m} \\ w_{21} & w_{22} & \dots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \dots & w_{Nm} \end{pmatrix}, \tag{3}$$

where i-th row matches i-th document, and j-th column matches j-th term.

The similarity of two documents in vector model is usually given by the following formula – Cosine Similarity Measure:

$$sim(d_i, d_j) = \frac{\sum_{k=1}^{m} (w_{ik} w_{jk})}{\sqrt{\sum_{k=1}^{m} (w_{ik})^2 \sum_{k=1}^{m} (w_{jk})^2}}, \quad (4)$$

For more information, please consult [19, 22, 3].

3.2 Suffix trees

A suffix tree is a data structure that allows efficient string matching and querying. Suffix trees have been studied and used extensively, and have been applied to fundamental string problems such as finding the longest repeated substring [36], strings comparisons [5], and text compression [23]. Following this, we describe the suffix tree data structure – its definition, construction algorithms and main characteristics.

3.2.1 Definitions

The following description of the suffix tree was taken from Gusfield's book *Algorithms on Strings, Trees and Sequences* [8]. Suffix trees commonly deal with strings as sequence of characters. One major difference is that we treat documents as sequences of words, not characters. A suffix tree of a string is simply a compact tree of all the suffixes of that string. Citation [39]:

Definition 1. *A suffix tree T for an m-word string S is a rooted directed tree with exactly m leaves numbered 1 to m. Each internal node, other than the root, has at least two children and each edge is labeled with a nonempty substring of words of S. No two edges out of a node can have edge labels beginning with the same word. The key feature of the suffix tree is that for any leaf i, the concatenation of the edge labels on the path from the root to leaf i exactly spells out the suffix of S that starts at position i, that is it spells out $S[i \ldots m]$.*

In cases where one suffix of S matches a prefix of another suffix of S then no suffix tree obeying the above definition is possible since the path for the first suffix would not end at a leaf. To avoid this, we assume the last word of S does not appear anywhere else in the string. This prevents any suffix from being a prefix to another suffix. To achieve this we can add a terminating character, which is not in the language that S is taken from, to the end of S.

Example of suffix tree of the string *"I know you know I know you#"* is shown in Fig. 3. Corresponding suffix tree of the string *"I know you know I know you#"* is presented in Fig. 4. There are seven leaves in this example, marked as rectangles and numbered from 1 to 7. The terminating characters are also shown in this Figure.

In a similar manner, a suffix tree of a set of strings, called a generalized suffix tree [8], is a compact tree of all the suffixes of all the strings in the set [39]:

Definition 2. *A generalized suffix tree T for a set S of n strings S_n, each of length m_n, is a rooted directed tree with exactly $\sum m_n$ leaves marked by a two number tuple (k, l) where k ranges from 1 to n and l ranges from 1 to m_k. Each internal node, other than the root, has at least two children and each edge is labeled with a nonempty substring of words of a string in S. No two edges out of a node can have edge labels beginning with the same word. For any leaf (i, j), the concatenation of the edge labels on the path from the root to leaf (i, j) exactly spells out the suffix of S_i that starts at position j, that is it spells out $S_i[j \dots m_i]$.*

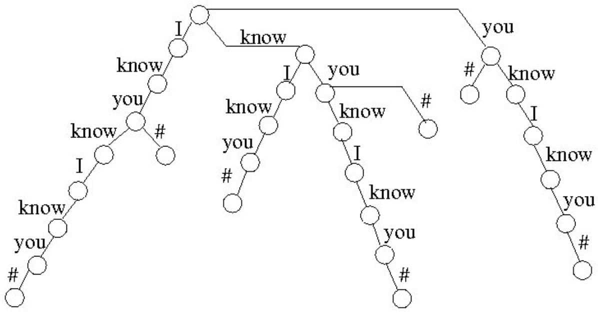

Fig. 3 *Simple example of suffix tree.*

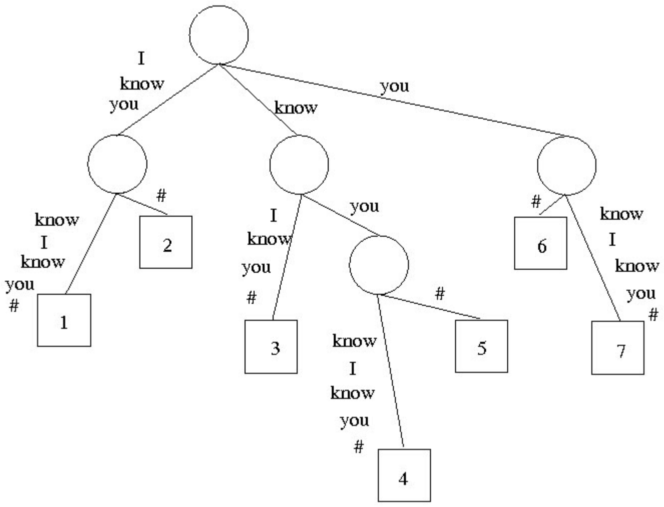

Fig. 4 *Simple example of suffix tree.*

Fig. 5 is an example of a generalized suffix tree of the set of three strings – *"Tom knows John #1"*, *"Paul knows John too #2"* and *"Tom knows Paul too #3"* (#1, #2, #3 are unique terminating symbols). The internal nodes of the suffix tree are drawn as circles, and are labeled from a to f for further reference. Leaves are drawn as rectangles. The first number $d_i = (d_1, \dots, d_n)$ in each rectangle indicates

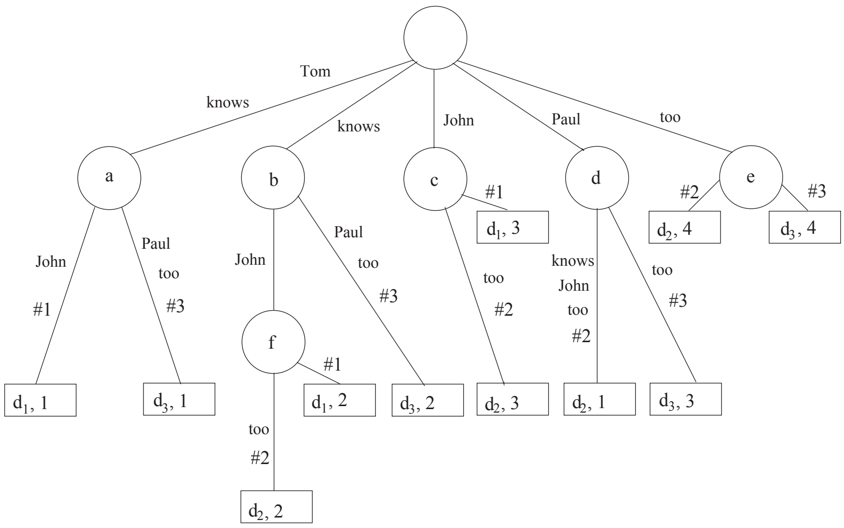

Fig. 5 *Example of the generalized suffix tree.*

the string from which that suffix originates – a unique number that identifies the string. The second number represents the position in that string d_i where the suffix begins. Each string is considered to have a unique terminating symbol.

3.3 Clustering and Self-Organizing Maps

Cluster analysis groups objects (data records) into classes (clusters) in such a way that objects in the same cluster are very similar, while objects in different classes are quite distinct. One of the possible clustering methods is competitive learning [7]. Given the training set of objects, competitive learning finds an artificial object (representative) most similar to the objects of a certain cluster.

A commonly used application of competitive learning is the Kohonen Self-Organizing Map [13], or SOM, described by Teuvo Kohonen in 1982. SOM is inspired by the cortex of the human brain, where information is represented in structures of 2D or 3D grids. Formally, SOM is a type of artificial neural network [9] with two fully interconnected layers of neurons, the input layer and the output or Kohonen layer.

The first step of Kohonen learning is competition. Given the training vector on the network's input and weight vector for each neuron of the Kohonen layer, the neuron with the minimal (usually Euclidean) distance between weight and input vectors is excited or selected as the winner of the competition [7, 9]. The second step is adaptation. The neurons of the Kohonen layer are organized in a one-, two-, or three-dimensional lattice, reflecting its biological inspiration.

A topological neighbor-affecting function is defined on the Kohonen layer, assigning a degree of participation in the learning process to the neurons neighboring the winning neuron. In every learning step the weight vectors of the winning neuron and its neighbors are adjusted to move closer to the input training vector.

In the batch version of the SOM algorithm [33], equivalent to Lloyd's vector quantization [17], the winning neuron weights are not adapted immediately after the competition step. When all the training set is consumed, the weight vector of the output neuron N_i, $i = 1, \ldots, n$ is replaced by the weighted mean value of the training cases assigned to the clusters represented by the neuron N_i and its neighbors, using the neighbor-affecting function as the weight function for the mean calculation.

The trained network finally sets its weights in such a way that the topologically near neurons represent similar training cases while distant ones reflect different cases. This is analogous with the cortex of the human brain, where similar knowledge is represented by adjacent parts of the cortex. The topology of a trained SOM forms an inherently useful base for clustering.

To get a satisfactory approximation of a data set with higher variance, the number of neurons in the static SOM exceeds the number of potential clusters. Agglomerative clustering [37] is, therefore, used over the trained SOM.

Initially, each of the SOM neurons represents a separate cluster. In each iteration a distance function is computed for every couple of clusters and those with the shortest distance are merged together to form a new cluster. The iteration process stops when the specified number of clusters is reached. Examples of distance functions being used with SOM are the overall variance of the map [37], the Ward and the SOM-Ward distance [33].

In our work, the SOM algorithm with the clustering extension performs unsupervised cluster analysis over the training set. If there is a similarity hidden in the training set of records, we suppose that it would be detected by some subset of records creating an isolated cluster, which could be useful in the localization process.

In our case, SOM realizes the transformation of the relations of the objects from the m-dimensional input space into the two-dimensional map of nodes (neurons) of the resulting Kohonen network. The complexity of the input space is reduced significantly and, in conjunction with coloring the nodes of the resulting network, data clusters can be effectively visualized.

SOM networks are especially suitable for hidden knowledge presentation. Both the structure of data clusters and query result can be easily visualized. For the overall view of learned data, we use the so-called *Unified distance matrix* (U-matrix), which records the values in clusters and cluster boundaries. The values are assigned to the neuron which wins competition for them, and the distances between neighboring neurons are recorded with greyness level. Darker colors usually mean greater distance. On the other hand, close data can be colored with similar colors, in this case the boundary between clusters is shown as a steep change in color hue.

4. Mobile Phone Positioning Algorithm

In this section we describe the algorithm for determining the mobile station location based on database of previously measured samples. A brief description of the algorithm follows:

1. Prepare the data as were discussed in Sect. 2.

2. Insert all encoded measurements of the training set into the generalized suffix tree data structure.

3. For each query sequence (measurement from training set) construct a vector model.

4. For each query sequence find nearest neighbor from training set.

4.1 Inserting all measurements into the suffix tree

At this stage of the algorithm, we construct a generalized suffix tree of all encoded measurements from the training set. As mentioned in Sect. 2., we obtain the encoded forms of measurement samples – sequences of positive numbers. All of these sequences are inserted into the generalized suffix tree data structure (Sect. 3.2).

4.2 Build vector model for query sequence

In this section we describe the procedure for building the matrix representing the vector model index file (Sect. 3.1) for each query sequence. In a classical vector space model, the document is represented by the terms (which are words) respectively by the weights of the terms. *In our model the document is represented not by the terms but by the common phrases (nodes in the suffix tree)!* – the term in our context is a common phrase, i.e. node of the suffix tree.

To be able to build a vector model for query measurement, we have to find all nodes in the suffix tree which match the query sequence – common phrases. Recall the example given in Fig. 5: the phrases can be e.g. *"Tom knows John #1"*, *"knows John #1"*, *"John #1"*, etc. (just imagine that *"Tom knows John #1"* is equal to *"34 29 758F #1"*). The *phrase* in our context is an encoded measurement or any of its parts. The document in our context can be seen as an encoded measurement.

For all suffixes of the query sequence we simply traverse the suffix tree from the root and find the longest matching path. All the nodes on the path from the root to the position where a match does not occur represent the common phrases between query sequence and sequences in the training set. In this step we identify the attributes for the vector model (common phrases) as well as documents (measurements) which match the query sequence.

The node in the generalized suffix tree represents the group of the documents sharing the same phrase (group of measurements sharing the same subsequence). Now we can obtain the matrix representing the vector model index file directly from the generalized suffix tree. Each document (measurement) is represented by the nodes of the generalized suffix tree in which it is contained. For computing the weights of the common phrases (nodes matching the query sequence), we are using a $tf \times idf$ weighting schema as given by Eq. (2).

Simple example: Let us say that we have a node containing documents d_i. These documents share the same phrase t_j. We compute w_{ij} values for all documents appearing in a phrase cluster sharing the phrase t_j. This task is done for all the nodes identified by the previous stage of the algorithm.

Now we have a complete matrix representing the index file in a vector space model (Sect. 3.1).

Test Set	No. of Samples	>500 m	Mean Average Error
TS1	23	1	271.78 m
TS2	3753	101	122.87 m
TS3	1564	496	407.97 m
TS4	241	16	220.24 m
TS5	593	35	148.13 m

Tab. I *Experimental Results – Classic Vector Model.*

4.3 Finding nearest neighbor for query sequence

This step of the algorithm is very simple. From the previous stage of the algorithm we have also constructed a query vector – all the nodes found (common phrases). This query vector contains no null element toward the vectors representing the measurements in the index file. Now we need to sort all the documents in the constructed index file against the query document (query measurement). We compute the similarity score residing in Eq. (4) between all the documents in the index file and query sequence and sort them. The highest scoring document from the index file is returned as a result of the query – the first nearest neighbor.

4.4 Evaluation

In this section we present the evaluation of the algorithm for mobile phone positioning. Five test sets were evaluated and compared with a classic vector model. In our previous study we developed a classical vector model approach to document indexing [20]. The key difference is that in a classical vector model the terms are single words whereas in this new approach the terms of the vector model are common phrases – common parts of measurements. We use GPS readings to evaluate our algorithm. For each measurement from test sets (each query to vector model) the distance in meters is computed with a result of the query, which is the first nearest neighbor found in the training set. Five test sets were analyzed by our algorithm. The total number of training samples is 7855. Following Tab. I and Tab. II present the results of our experiments. Descriptions of Tabs. I and II are as follows: column *Test Set* stands for a label of test set used, column *No. of Samples* means number of samples in a given test set, *column* > 500 indicates the number of tested samples for which the distance against the first nearest neighbor was more than 500 meters, and the column *Mean Average Error* stands for average measured error between all query samples from test set (mean distance of all nearest neighbors).

5. Clustering Mobile Position Based on SOM

In this section there are described the experiments that use SOM to visualize the clusters of measured samples in the map of Ostrava – Poruba.

The aim of the experiments was to determine, whether the neurons in the SOM form clusters of the measured samples with GPS position in the same area. For

Test Set	No. of Samples	>500 m	Mean Average Error
TS1	23	1	175.30 m
TS2	3753	69	104.74 m
TS3	1564	20	84.44 m
TS4	241	14	201.79 m
TS5	593	20	134.05 m

Tab. II *Experimental Results – Suffix Tree Vector Model.*

the learning we have used measurements collected from Ostrava – Poruba. The description of data and maps of the SOM is as follows:

- number of the input vectors was 7855,
- number of the attributes of the vector was 3443,
- size of the SOM was 20×20,
- number of the learning iterations was 100.

For the experiments we have used a parallel implementation of the SOM that was described in the articles [14, 34], and we have used software SOMToolbox [30] for the U-Matrix visualization of the SOM.

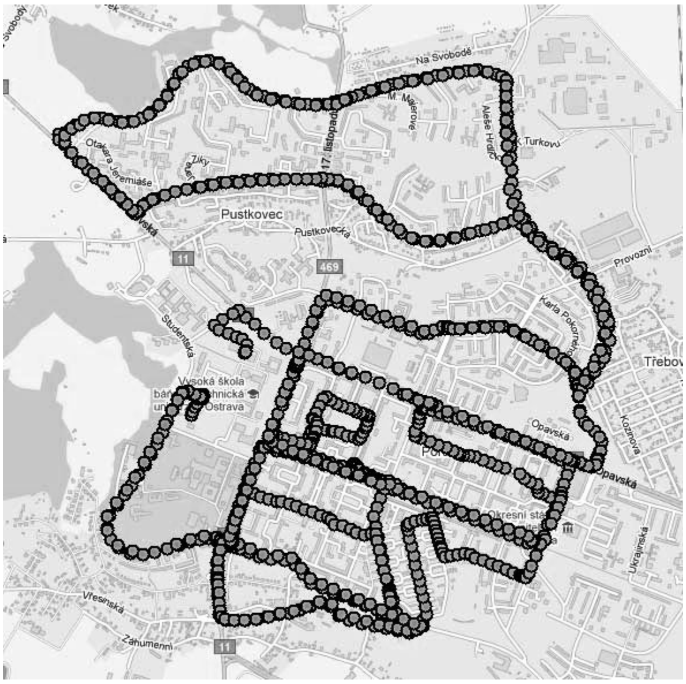

Fig. 6 *Application LoMo Map Viewer for visualization of the measured network parameters.*

The application LoMo Map Viewer was developed to generally visualize all parameters of the measured network. It allows visualization of the individual values measured together with their GPS position (see Fig. 6.). This visualization allows further analysis of acquired network parameters. LoMo Map Viewer consists of several layers. Over the base map layer there is applied a layer consisting of individual measurements and other objects. There is also an ability to save and load the internal database as well as the import of several file formats.

The LoMo Map Viewer application has been extended by the function for visualization of the U-Matrix and by the function for the selection of the neurons or the regions of neurons. U-Matrix has been created by the application SOMToolbox.

LoMo Map Viewer application displays different samples in the map after selection of a neuron or set of neurons from the U-Matrix. For the sample visualization a red ring has been used. The result of this experiment is shown in Fig. 7.

Fig. 8 shows the selected neuron from U-Matrix with the corresponding samples. Furthermore, the selected area has been extended to the neighboring neurons (Fig. 9). The second example with the similar output is on Fig. 10 and Fig. 11. Unlike the previous example, there are the two completely separate clusters. This can be seen from the measured output data as well, where the data are located into two separate areas in the map.

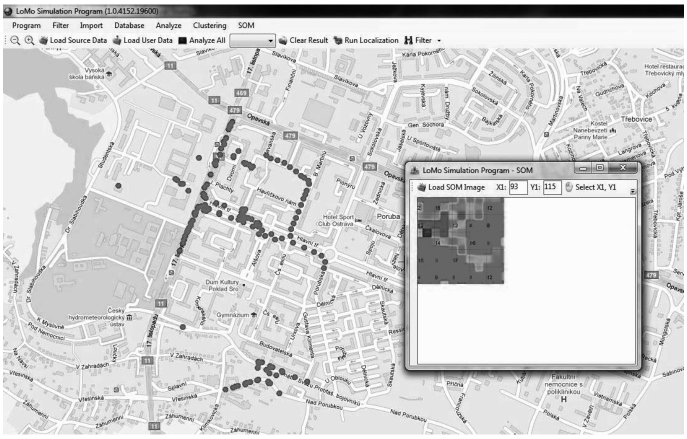

Fig. 7 *LoMo Map Viewer – SOM visualization.*

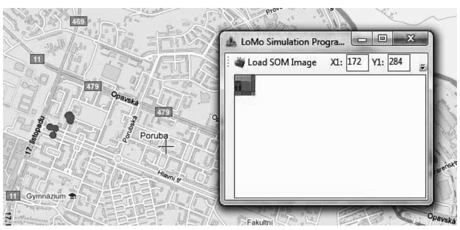

Fig. 8 *Example 1 – Selected neuron from SOM.*

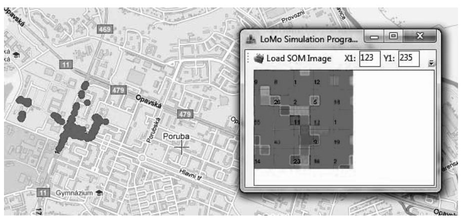

Fig. 9 *Example 2 – Selected neuron from SOM and its neighbors.*

Fig. 10 *Example 3 – Selected neuron.*

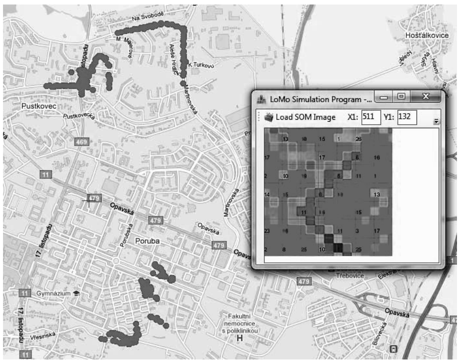

Fig. 11 *Example 4 – Selected neuron from SOM and its neighbors.*

Fig. 12 *SOM U-Matrix moved to the map.*

The last type of visualization (Fig. 12), which we have selected, has been mapping neurons from the SOM to the base map for the location of Ostrava – Poruba. This visualization has been created by counting the centroid distance of the all neuron samples.

6. Conclusion

In this work we have proposed a novel method for mobile phone localization. The algorithm works with parameters which can be obtained from every common mobile phone, and does not require a GPS module to be installed. We have shown that the results are very accurate and in most cases of test measurements accomplish the E9-1-1 standards requirements in all of the test sets. We have also found the use of suffix trees very useful and realized the suffix tree vector model outperforms the classical vector model. This leads us to the idea that the measured parameters are dependent on the sense of ordering. Also we have found the use of common subsequences very useful toward the use of simple single terms.

In the final map we can see correspondences between neighboring nodes of the trained SOM and geo-coordinates system in the real map.

Acknowledgement

This paper was supported by the grant IT Cluster 2009-2012 No. 5.1 SPK01/029 which is funded by the European Fund for Regional Development and the Ministry of Industry and Trade.

References

[1] Ashkenazi I. B., Moshe B.: Radio-maps: an experimental study. Report, Ben-Gurion University, 2005. http://www.cs.bgu.ac.il/ benmoshe/RadioMaps (cited 6th April 2012).

[2] Jarvinen J., DeSalas J., LaMance J.: Assisted GPS: A Low-Infrastructure Approach. GPS World, March 1, 2002. http://www.gpsworld.com/gps/assisted-gps-a-low-infrastructure-approach-734 (cited 6th April 2012).

[3] Baeza-Yates R., Ribeiro-Neto B.: Modern Information Retrieval. Adison Wesley, 1999.

[4] Drane C., Macnaughtan M., Scott G.: Positioning GSM telephones. IEEE Communication Magazine, **36**, 4, Apr. 1998, pp. 46–54, 59.

[5] Ehrenfeucht A., Haussler D.: A new distance metric on strings computable in linear time. Discrete Applied Math, **20**, 3, 1988, pp. 191–203.

[6] The FCC. Fact Sheet-FCC Wireless 911 Requirements, FCC, Jan. 2001.

[7] Gan G., Ma Ch., Wu J.: Data Clustering: Theory, Algorithms and Applications, SIAM, Philadelphia, 2007.

[8] Gusfield D.: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, 1997.

[9] Haykin S.: Neural Networks: A Comprehensive Foundation, 2nd edition. Upper Saddle River, NJ: Prentice-Hall, 1999.

[10] Kennemann O.: Pattern Recognition by Hidden Markov Models for Supporting Handover Decisions in the GSM System. In: Proc. 6th Nordic Seminar Dig. Mobile Radio Comm., Stockholm, Sweden, 1994, pp. 195–202.

[11] Kennemann O.: Continuous Location of Moving GSM Mobile Stations by Pattern Recognition Techniques. In: Proc. 5th Int. Symp. Personal, Indoor, Mobile, Radio Comm., Den Haag, Holland, 1994, pp. 630–634.

[12] Kim S. Ch., Lee J. Ch., Shin Y. S., Cho K. R.: Mobile Tracking Using Fuzzy Multi-criteria Decision Making. MSN 2005, LNCS 3794, 2005, pp. 1051–1058.

[13] Kohonen T.: Self-Organizing Maps. Springer-Verlag, Berlin, 1995.

[14] Klement P., Snášel V.: Using SOM in the performance monitoring of the emergency call-taking system. Simulation Modelling Practice and Theory **19**, 1, 2011, pp. 98–109.

[15] Lee D. L., Chuang H., Seamons K. E.: Document ranking and the vector-space model. In: emphIEEE Software, 1997, pp. 67–75.

[16] Lee J., Yoo S. J., Lee D. Ch.: Fuzzy Logic Adaptive Mobile Location Estimation. NPC 2004, LNCS 3222, 2004, pp. 626–634.

[17] Lloyd S. P.: Least Squares Quantization in PCM. IEEE Transactions on Information Theory, **28**, 2, 1982, pp. 129–137.

[18] Manzuri M. T., Naderi A. M.: Mobile Positioning Using Enhanced Signature Database Method and Error Reduction in Location Grid. WRI International Conference on Communications and Mobile Computing, **2**, 2009, pp. 175–179.

[19] Manning C. D., Raghavan P., Schütze H.: Introduction to Information Retrieval. Cambridge University Press, 2008.

[20] Martinovič J., Novosád T., Snášel V.: Vector Model Improvement Using Suffix Trees. IEEE ICDIM, 2007, pp. 180–187.

[21] McCreight E.: A space-economical suffix tree construction algorithm. In: Journal of the ACM, 1976, pp. 262–272.

[22] van Rijsbergen C. J.: Information Retrieval (second ed.). London, Butterworths, 1979.

[23] Rodeh M., Pratt V. R., Even S.: Linear algorithm for data compression via string matching. In: Journal of the ACM, **28**, 1, 1981, pp. 16–24.

[24] Salcic Z., Chan E.: Mobile Station Positioning Using GSM Cellular Phone and Artificial Neural Networks. Wireless Personal Communications, **14**, 2000, pp. 235–254.

[25] Salcic Z.: GSM Mobile Station Location Using Reference Stations and Artificial Neural Networks. Wireless Personal Communications, **19**, 2001, pp. 205–226.

[26] Simic M. I., Pejovic P. V.: An Algorithm for Determining Mobile Station Location Based on Space Segmentation. IEEE Communications Letters, **12**, 7, 2008.

[27] Simic M. I., Pejovic P. V.: A probabilistic approach to determine mobile station location with application in cellular networks. Annals of Telecommunications, Publisher Springer Paris., **64**, 9-10, 2009, pp. 639–649.

[28] Simic M. I., Pejovic P. V.: A comparison of three methods to determine mobile station location in cellular communication systems. European Transactions on Telecommunications, **20**, 8, 2009, pp. 711–721.

[29] Song H. L.: Automatic Vehicle Location in Cellular Communication Systems. IEEE Transactions on Vehicular Technology, **43**, 1994, pp. 902–908.

[30] SOMToolbox: `http://www.ifs.tuwien.ac.at/dm/somtoolbox/somViewer.html` (cited 6th April 2012).

[31] Sun G., Chen J., Guo W., Liu K. J. R.: Signal processing techniques in network aided positioning: a survey of state-of-the-art positioning designs. IEEE Signal Processing Mag., **22**, 4, July 2005, pp. 12–23.

[32] Ukkonen E.: On-line construction of suffix trees. Algorithmica, **14**, 1995, pp. 249–260.

[33] Viscovery®. SOMine 5.0. Copyright 1998-2007 by Viscovery Software GmbH, `http://www.viscovery.net` (cited 6th April 2012).

[34] Vojáček L., Martinovič J., Slaninová K., Draždilová P., Dvorský J.: Combined Method for Effective Clustering based on Parallel SOM and Spectral Clustering, DATESO 2011, Písek, Czech Republic, April 2011, pp. 120–131.

[35] Wang S., Min J., Yi B. K.: Location Based Services for Mobiles: Technologies and Standards. IEEE International Conference on Communication (ICC), Beijing, China, 2008.

[36] Weiner P.: Linear pattern matching algorithms. In: The 14th Annual Symposium on Foundations of Computer Science, 1973, pp. 1–11.

[37] Yang M. Y.: Extending the Kohonen self-organizing map networks for clustering analysis. In: Computational Statistics & Data Analysis, 38, 2001, pp. 161–180.

[38] Zamir O., Etzioni O.: Web document clustering: A feasibility demonstration. In: SIGIR'98, 1998, pp. 46–54.

[39] Zamir O.: Clustering web documents: A phrase-based method for grouping search engine results. In: Doctoral dissertation. University of Washington, 1999.

[40] Zhao Y.: Standardization of mobile phone positioning for 3G systems. IEEE Communications Mag., **40**, 7, July 2002, pp. 108–116.