

# NAVIGATION THROUGHOUT OBJECT SPACE ON WIRELESS TERMINAL

Dorian Gorgan, Tudor Borlea

Computer Science Department, Technical University of Cluj-Napoca, Romania  
dorian.gorgan@cs.utcluj.ro, b\_2dor@yahoo.com

Commission IV, WG IV/6

**KEY WORDS:** Spatial Information Sciences, LBS, GIS, mapping, user interface, mobile, object space navigation.

## ABSTRACT:

Application running on wireless terminals and providing interaction techniques through a graphical user interface requires a high communication rate. Commonly such a user interaction involves just navigation within image space over a raster image. The interaction is supported mainly by the interface objects such as the usual buttons and menus. The paper presents the research and the experimental results on the interaction and navigation techniques through the application space, which is actually the real object space. The experiments concern with the interaction by street network at various levels of details, and the adaptive selection accuracy of the user cursor. The work is an extension of the TUGIS platform to support the development of LBS applications. The purpose is to provide all the functionality of the TUGIS system, adapted to the very specific and restricted context of the wireless mobile platforms. For this purpose, new interaction modes have been researched, as well as ways to transfer data more efficiently and economically. Some practical tests on smart phone are presented as well.

## 1. INTRODUCTION

This paper highlights the results of the research in the field of GIS Applications on the wireless hand held devices. In mapping processing and display on such devices one should take in account the limited computation capability on the wireless terminal, and the costs of the data communication to and from the server. There are at least two reasons to minimize the size of data required to communicate in mapping on wireless terminals. The first one is the data size based billing in mobile networks, and the second one is the limited communication bandwidth. Hereby this considerations, the vector based graphics emerges as an efficient solution. For instance, the SVG (Scalable Vector Graphics) encoded maps have relatively reduced dimensions and require low costs for data transmission (Eisenberg, 2002). The map visualization is supported commonly by a local viewer appropriate for the restrictive wireless platforms SVG-Tiny (SVG-Tiny, 2006). The usual operations available in the user interface are *move up/down, left/right, and zoom in/out*. This paper concerns with the development of user interaction techniques, especially through the object space (e.g. selection and navigation from one to another street within the map).

This paper is structured as follows: Section 2 highlights some works related with user interaction techniques, mapping and wireless applications. Section 3 describes the TUGIS Platform that is basically the technological support of this research. Section 4 defines the main issues of the map based interaction and the section 5 sketches the solutions. Section 6 is the main part of the paper. It describes the design solutions and the related experiments. Finally, section 7 concludes the results.

## 2. RELATED WORKS

The solution proposed by this paper is based on the Java technology and particularly on J2ME (Java 2 Micro Edition). A similar achievement is the Mobile Europe system of the Route66 Company (Mobile, 2007), developed by the Symbian

technology, and based on the C++ language (Symbian, 2007). It is a static solution, by which the map is entirely stored on a local support (e.g. flash disk), and does not require a data transfer through the mobile network. The map data needs a large storage space, and uses a proprietary non-standard format, instead of SVG. As location input, the Mobile Europe system uses a GPS (Global Positioning System) device, which is actually the single interaction mode. In fact the Mobile Europe system is appropriate for on-board car navigation applications. The system described by this paper presents user interaction techniques appropriate for the pedestrian traffic applications. Wayfinder (Wayfinder, 2007) offers maps, directions and GPS navigation services accessible by mobile phone. The user may access content services, including digital city guide books,

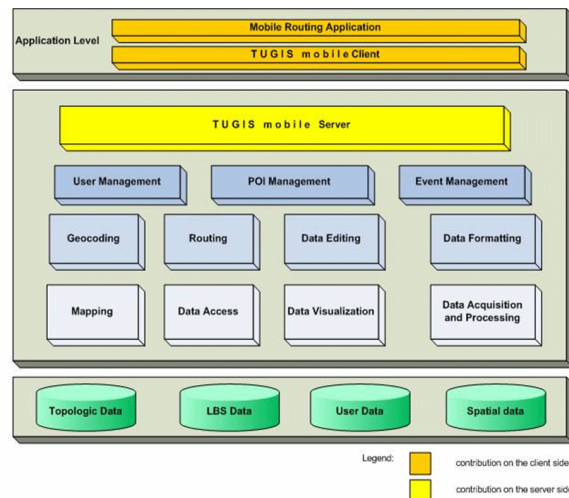


Figure 1. TUGIS Platform. Mobile client and mobile server layers extends the TUGIS wireless related functionality.

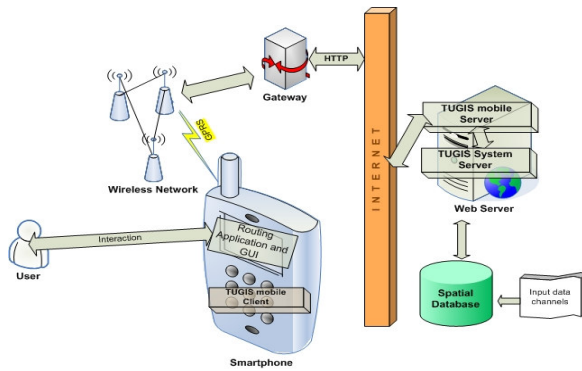


Figure 2. Conceptual architecture of the map space navigation application.

traffic information, currency conversion and weather reports. There is an increasing number of mobile guide systems, which provide users with Location Based Services (LBS) and navigational assistance usually by means of mobile maps and position determination (Baus, 2005; Patalaviciute, 2005). Reichenbacher proposed in (Reichenbacher, 2005) the adaptation of maps to the user's context such as remote natural area rather than in a more static urban setting.

### 3. TUGIS PLATFORM

The solutions presented by this paper are based on the TUGIS system, which is a GIS/LBS platform developed within the Technical University of Cluj-Napoca (Muresan, 2006a; Muresan, 2006b; Muresan, 2004, Gavrea, 2004.). Throughout the development of this framework, the common steps required in building a GIS platform have been covered with high emphasis on extensibility and adaptability (Figure 1). This includes the data acquisition from different formats, the implementation of means of transparently accessing spatial data, the generation of maps with various parameters and the interrogation and visualization of GIS data. A special emphasis has been given to supporting a wide range of potential users. In

order to address this requirement the framework provides both vector and raster map generation capabilities and supports map publishing by using such technologies as Servlets or Web Services.

This paper concerns with the extension of the TUGIS Platform by the wireless related functionality, and presented initially in (Borlea, 2005a; Borlea, 2005b; Gavrea, 2006, Muresan, 2006a). The extension consists of three modules: TUGIS mobile server, TUGIS mobile client, and the end user application: TUGIS mobile routing application (Figure 1 and Figure 2). The routing algorithm is implemented in the TUGIS Platform as it was presented in (Gavrea, 2004) and according with (Abuja, 1993, and Zhan, 1996). The TUGIS mobile server module is resident on the server side and handles the requests received from clients, by creating a unique environment for each client, and communicates with the lower layers of the TUGIS system to obtain the required answer for a given client. The TUGIS mobile client module is located on the user's wireless terminal. It acts as a buffer between the routing application and the server module, accepting input and forwarding processed data in both directions. The TUGIS mobile routing application also resides on the user's wireless terminal, accepting input from the user, and making the data available for him in an intuitive manner. The routing application module is the most specialized of the three and therefore the least portable. It has to be specifically designed to take advantage of features specific to each terminal model depending on its class, manufacturer and so on. Nevertheless, the particular application developed as an example should be usable on most Java enabled smartphones, and has been successfully tested on a Nokia Series 60 smartphone.

### 4. MAP OBJECTS BASED USER INTERACTION

The paper concerns with the solutions for user interaction techniques on wireless devices based on the SVG-Tiny standard (SVG-Tiny, 2006), in GIS applications. Considering the client-server architecture of a GIS system (see Figure 2), by which the user interacts through a restrictive wireless platform, and asks to the server the maps in the SVG-Tiny format, let us analyze solutions for an efficient client-server communication, in terms of speed and cost. Actually the client sends to the server a requirement such as "I would like to travel from the street A to the street B". The server processes the requirement, searches the two streets in the spatial database, and computes a route from A to B. If the server finds the route, it generates a SVG file for the rectangular surrounding area (i.e. *bounding-box*), and returns the file to the client. By this solution the communication between the client and the server consists in both directions just by text format of the request and respectively the SVG format of the answer. The main problem could be the processing of data on the *fat-client*. It is a quite fragile equilibrium between the limited processing capability of the client and the constraint of a low cost and fast data transmission. On the client side the map consists of line based graphics primitives. The client challenges two significant issues: (a) text rendering, and (b) user interaction techniques.

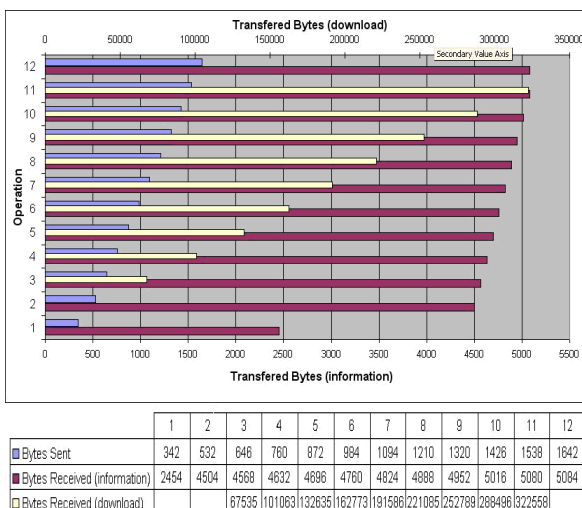


Figure 3. Comparative graph of variation of data transfer size with operation in raster map mode, grouped by data category.

#### 4.1 Text Rendering

The SVG-Tiny format supports just the horizontal direction of the text, rather than the *textPath* feature of the complete SVG. This is a significant withdraw of the mobile platforms which cannot display the maps in a similar manner of the printed maps.

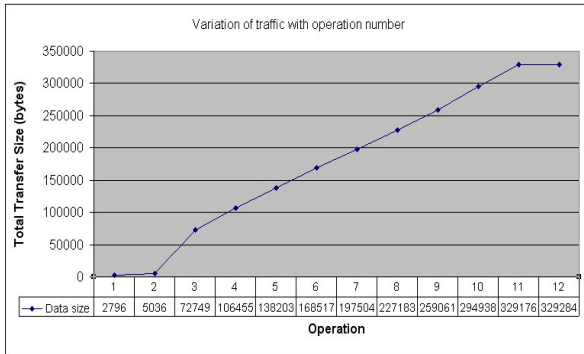
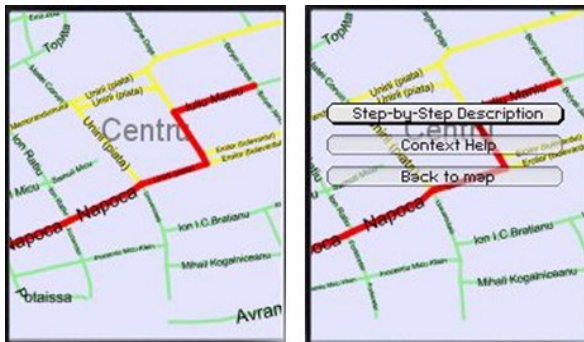


Figure 4. Raster based graphical user interface and the related traffic of data communication between client and server.

#### 4.2 User Interaction

The SVG-Tiny format provides a minimal user interaction by operations such as *pan*, *zoom* and *link*. In the case of the map based navigation it is useful indeed to navigate within the street space, and perform operations such as searching, selection and highlighting of some street locations and names. Actually it is necessary interaction techniques that provide the user with simple perceptions of the map without high data transfer rate and expensive computation.

In order to achieve these objectives it should consider the technological constraints of the mobile platform, which are the low resolution screen, limited keyboard (e.g. just numerical, and a few function keys), and reduced memory resources. The experiments have been developed on the Nokia Series 60 Smartphones that provide the minimal required characteristics.

#### 5. MAP VIEWS AND INTERACTION TECHNIQUES

The first challenge is the text rendering. The SVG-Tiny language supports just the horizontal direction of the text. The extensions of the SVG-Tiny language by tags that support a curve based direction require the implementation of a new language interpreter and an unacceptable expensive computation. The second solution could be to display the street name near the street itself. In fact, this solution makes crowded maps, with a high rate of the street identification errors. It is a viable solution just for high level of details when the screen displays a few streets. The solution proposed by the authors displays the name of the selected street through a separated status bar. In this case the street network is displayed just as lines based graph. In a step by step routing the application could highlight the current street through which the user is moving on.

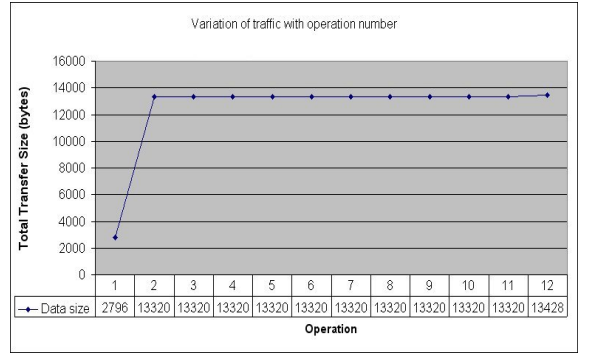
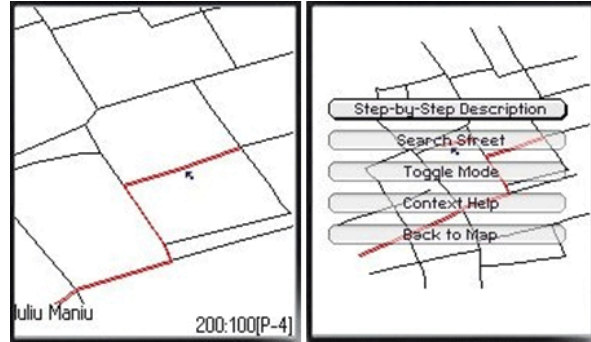


Figure 5. Vector based graphical user interface and the related traffic of data communication between client and server.

The user visualizes the current position street, and moreover may navigate over the map around the current location in order to get information about the neighbour streets.

The second challenge in the GIS applications is the navigation within the street space. Actually this feature is not supported by the native SVG-Tiny standard. The first solution of extension by specialized tags is unacceptable for incompatibility. Therefore, the authors have been decided to transfer the new functionality to the parser and the renderer of the SVG-Tiny viewer. Hereby the viewer is a particular solution which supports the street space navigation, keeping the standard format of the SVG-Tiny file. By this solution the same map could be visualized as well by any other standard viewer, but unless supporting the street navigation technique.

#### 6. IMPLEMENTATION AND EXPERIMENTS

A route computation application has been developed in Java and J2ME in order to prove the previously described features (Gavrea, 2004; Gavrea, 2006; Muresan, 2006a). The kernel of the client platform is the SVG-Tiny viewer that provides the user with the street navigation by a cursor similar with one of the desktop applications. The cursor movement is adapted to the very specific goal of navigation within the map space through the screen of the mobile device. Actually the cursor's movement in the screen space is mapped into the object space of the map (i.e. topological space).

The user controls the cursor's movement by the joystick key of the mobile keyboard. When the cursor is moving near a street this is highlighted and its name is displayed in the status bar. It seems to be quite simple, but a few issues must be considered: real time client-server communication, proximity detection,

adapted cursor accuracy, control of the cursor movement at various levels of details, fast movement of the cursor, key press optimization etc.

### 6.1 Real time client-server communication

The research concerns as well with the quantity of data and the time required to support real time navigation in the map space, and satisfactory user interaction techniques. In order to design the appropriate map dimensions, map format and dynamical parameters of the map based navigations two experiments have been carried out.

This section presents the connectivity related tests performed on the TUGIS mobile Client. These are traffic related comparative tests that show the difference between requesting a raster map format, and a vector map format. The main difference between the two, from a connectivity point of view, is the amount of data that needs to be transferred. In order to test and compare the data volume transferred in each of the two map modes, a scenario was composed, where the user would perform 12 steps. These steps would take him through the entire lifecycle of the application, from entering the routing parameters, to viewing the map, interacting with it, and then exiting. The first three steps in the communication process are represented by the exchange of routing information between client and server:

1. The client sends the routing parameters, and receives the available options for the second step;
2. The client sends the user selected options and receives the available detailed point options;
3. The client sends the detailed point selection and receives of the step by step text description, if previously requested.

During the third step the requested map has been generated on the server in the corresponding environment of the accessing client. The fourth step on the client-side consists in the download of the initial map representation. In the case this is a raster map representation, further interactions of the user with the map (pan, zoom) will have as effect the server-side generation of new raster images, which will then be downloaded by the client. In the case the map was requested in vector format, the entire interaction takes place on the client side, no further communication with the server being required until the last step. The last step always consists in the client sending a “good-bye message” to the server, as a signal that it has disconnected, so the server can remove any temporary files created (actually the server removes the entire environment for the respective client).

Figure 3 shows comparatively the detailed graph of the 12 step communication in raster map mode between the client and the server. On the vertical axis is represented the operation number, on the bottom horizontal axis the size of data transfer for the control information passed (in bytes), and on the top horizontal axis the size of data transfer for the downloaded images from server. The double numbering of the horizontal axis had to be introduced in order to be able to visualize the relative growth of the three types of data transfer on the same graph. That would not have been possible otherwise, as the download size order exceeds by far the size order of the control information. As it can be seen, the top values for the horizontal axis vary in the range [0, 350000] while the bottom range is only [0, 5500]. So the yellow bars should be interpreted according to the top horizontal axis, while the blue and magenta bars should be interpreted according to the bottom horizontal axis. However,

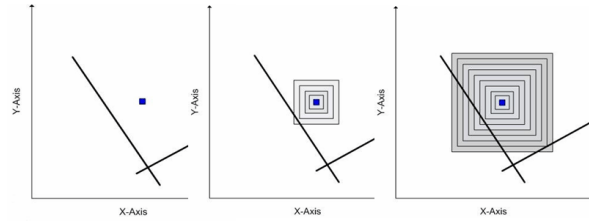


Figure 6. Adaptive cursor sensitivity according with the level of details, in order to avoid multiple street selections.

the visual information that can be extracted from the graph is the following: the general tendency is one of linear growth.

Figure 4 shows the actual numbers involved in each step, grouped by category. The figure shows the evolution of the total amount of data transferred (in bytes), for the raster map format. As it can be seen, after the first two steps and up to the last, the growth is linear. That is, the amount of data transferred between client and server grows linearly with the number of operations performed on the map. In the particular example presented, the total amount of transferred data is 329284 bytes (about 300kB) after 12 steps. This is in the context of a normal, average usage of the map (about 5 zoom operations and 5 pan operations for a relatively simple route). The costs involved in a real life situation would be relatively modest. However for an intensive use of the system these costs would add up to levels that could become unacceptable for the average user.

This problem can be relatively easily solved by requesting the map in vector format that would produce much less traffic as shows the Figure 5. Actually the graph shows that when requesting the map in vector format, the total amount of data transferred remains constant after the first two steps, up to the last (when the “good-bye message” is sent, generating additional traffic).

### 6.2 Proximity Detection

The notion of proximity is quite evident for paper maps and human operator’s eye, but in the computer based spatial processing it should be defined in terms of model objects: positions, streets and cursor. Obviously, the notion of proximity



Figure 7. Street selection at low level (100%, sensitivity of 4 pixels) and higher level (320%, 9 pixels) of details.

corresponds to the distance between the cursor's pointer and the street. It looks like the cursor needs to feel the street vicinity. For this reason the cursor has a characteristic called *sensitivity* (Figure 6). The sensitivity is defined as the maximum distance at which the cursor can detect a line. This characteristic is specific for the navigation around the low resolution screens and limited control keys. It would be not necessary in a desktop environment where the cursor accuracy is higher and the user can move the cursor and select the street by mouse in a very fast and easy manner.

In our experiments we have considered a spatial data structure which supports fast operations such as indexing, searching, replacement, spatial analysis, proximity detection etc. The main issue was the limited resources rather than the conceptual design.

### 6.3 Adaptive Selection Accuracy

The selection *accuracy* is the cursor's selection ability of the objects in the context of the application (Figure 7). When the zoom level is low, the cursor sensitivity need not be high, or several streets might be detected and highlighted at the same time. When the zoom level is high, and there are a few streets displayed, with large gaps among them, the sensitivity of the cursor should be increased, as the risk of selecting more than one street segment at once is diminished.

### 6.4 Cursor Based Navigation

In the context of a restricted wireless platform, like a mobile phone, where the control of the cursor is made only with the aid of the directional keys, or the equivalent joystick, the speed and accuracy the user expects when handling the cursor are still similar to those obtained on a desktop computer. To achieve that, the movement of the cursor must be done in steps that are big enough to create the impression of speed similar to that of the mouse cursor movement, but small enough so that the user can easily select exactly the desired street. The effort of the user from the moment he chooses in his mind a street to select and to the moment that street is selected on the screen, should be minimal. Minimal effort translates to minimum number of key presses/ joystick movements.

### 6.5 Experiments on Cursor Movement Control

The experiments and the study of the cursor movement in the practical context of navigating with the cursor on the screen of a mobile are presented below. The Nokia Series 60 phones have a screen of 176 pixels wide and 208 pixels high. For the user to move the cursor from top left corner to bottom right, using steps of 1 pixel, it would take  $176+208=384$  key presses, which is simply unacceptable for any user. Even by using steps of 5 pixels for cursor movement it would still require  $384/5=77$  clicks, much better, but still impracticable by most user standards.

As a result, a new cursor control method is required, one that would diminish the number of key presses even further, to a practicable number. The solution proposed by this paper is related to the observation that on the vast majority of the available phone models, the numeric keys of 1 to 9 are laid out in such a way that they form an imaginary 3x3 grid (3 rows and 3 columns), each key being a cell (see Figure 8). One solution could be to use these keys for an intuitive mapping from key space to screen space. That is, the screen can also be



Figure 8. Numeric key based navigation to nine screen areas.

imaginarily divided into a 3x3 grid, and the user could immediately make the association between a key's position in the grid of keys and the cursor position within an area of the screen. By applying this idea, when the user presses one of the numeric keys of 1 to 9, the cursor will "jump" to the middle of the corresponding rectangular area on the screen. The number of key presses thus required to reach from one corner of the screen to the opposite becomes  $1+(176/3)/2 + (208/3)/2 = 1+63 = 64$  for a 1 pixel step, and  $1+(63/5) = 1+13 = 14$  for a 5 pixel step. The last solution, of using the 5 pixel step, is definitely acceptable for the traversal of the screen. It must be noted that it represents the maximum number of key presses necessary, as not always will be the case to cross the screen from one corner to the opposite. In the previous arithmetic calculus the 1 represents the first key press necessary for the cursor jump, and the 63 (or 13 respectively) represents the number of key presses necessary for the following sequential moves of the cursor.

## 7. CONCLUSIONS

The aim of this research was to create a SVG-Tiny based extension to the TUGIS framework, namely a module that would handle communication with restricted wireless platform clients. The maps are available in both raster and vector formats. The advantages and disadvantages of both formats were discussed, and a study regarding the amount of data transferred in each case was performed. New user interaction techniques in map space, over the hand held devices have been developed. The user may navigate in the map space and select by an adaptive selection accuracy of the cursor a street any level of details. A data model has been experimented on the client side to support in real time the user interaction techniques. The experiments and the results have proven the possibility to use in wireless GIS application, the real time user navigation within the street network space.

## REFERENCES

Ahuja, R.K., Magnati and T.L., Orlin, J.B., 1993. *Network Flows – Theory, Algorithms, and Applications*, Prentice Hall Inc.

- Zhan, F. B. and Noon, C. E., 1996. Shortest Path Algorithms: An Evaluation using Real Road Networks. *Transportation Science* vol. 32, no. 1, pp. 65-73.
- Baus J., Cheverst K., Kray C.A., 2005. Survey of Map-based Mobile Guides. In: Zipf, A. , Meng, L. and Reichenbacher, T. (eds.): *Map-based mobile services – Theories, Methods and Implementations*. Springer Geosciences. Springer-Verlag, Berlin Heidelberg New York.
- Eisenberg, J.D., 2002. *SVG Essentials*, First Edition, O'Reilly.
- Gavrea B., Veltan V., Gorgan D., 2006. Rutarea in Sisteme GIS Distribuite. *Prelucrarea grafica paralela si distribuita pe structura grid a datelor geografice si de mediu* (engl.: Routing in Distributed GIS Systems. Parallel and distributed graphical processing of geographical and environmental data), vol 2, ISBN: 978-973-713-092-1, Ed Mediamira, 2006, pp. 1204-213.
- Mureșan O., Gorgan D., 2006a. TUGIS – Servicii GIS în sisteme GRID, *Prelucrarea grafica paralela si distribuita pe structura grid a datelor geografice si de mediu* (engl.: TUGIS – GIS Services in Grid Systems. Parallel and distributed graphical processing of geographical and environmental data), vol 2, ISBN: 978-973-713-092-1, Ed Mediamira, 2006, pp. 196-203.
- Borlea T., Gorgan D., 2005a. *Interactiunea om-calculator in aplicatiile GIS pe platforme wireless* (engl.: User Interaction in Wireless Platforms Based GIS Applications). Volumul "Interactiune Om-Calculator 2005". Conferinta Nationala RoCHI 2005 Cluj-Napoca, ISBN 973-7973-24-0, pp. 71-74
- Borlea T., Gorgan D., 2005b. *Location Based Services for Wireless Platforms*. Research Report. Technical University of Cluj-Napoca, 2005, pp. 72
- Gavrea B., Gorgan D., 2004. *Routing in Location Based Systems*, Research Report. Technical University of Cluj-Napoca, 2004, pp. 99
- Muresan, O., Gavrea, B., Gorgan, D., 2006b. *TUGIS Platform for Location Based Services in Web Applications*, Proceedings of the IEEE-TTTC International Conference on Automation, Quality and Testing, Robotics AQTR 2006 (THETA 15), 25-28 mai 2006, Cluj-Napoca, Vol 1, and IEEE Computer Press (2006), pp. 436-441.
- Muresan, O., Gorgan D., 2004. *Core GIS in Location Based Systems*, Research Report. Technical University of Cluj-Napoca, 2004, pp. 109.
- Patalaviciute V., Dupmeier C., Freckmann P., 2005. Using SVG-based Maps for Mobile Guide Systems: A Case Study for the Design, Adaptation and Usage of SVG-based Maps for Mobile Nature Guide Application. 4th Annual Conference on Scalable Vector Graphics, Enschede, The Netherlands. <http://www.svgopen.org/2005/papers/MapsForMobileNatureGuideApplications/index.html>
- Reichenbacher T., 2002. SVG for adaptive visualisations in mobile situations, SVG Open Conference, Zürich (2002), [http://www.svgopen.org/2002/papers/reichenbacher\\_\\_svg\\_in\\_mobile\\_situations/](http://www.svgopen.org/2002/papers/reichenbacher__svg_in_mobile_situations/)
- Mobile Europe, 2007. Route 66, Symbian S60 smartphone. <http://www.66.com/route66/products>
- SVG Tiny, 2006. Scalable Vector Graphics (SVG) Tiny 1.2 Specification. <http://www.w3.org/TR/SVGMobile12/>
- Symbian C++, 2007. <http://seap.forum.nokia.com/main/resources/technologies/symbian/>
- Wayfinder, 2007. <http://www.wayfinder.com>