# GPS Based Naval Navigation System on Pocket PC Platform

Miroslav Lakotić, and Željka Mihajlović, *Member, IEEE*

***Abstract*-- This paper presents a GPS based system for naval navigation and the development process of such system. In this paper path finding algorithm for naval navigation is proposed that is appropriate for implementation on Pocket PC device. The implementation of the proposed system requires some specific adapting of a number of other algorithms and methods to run well on embedded systems like Pocket PC device. Experimental results are presented using developed configuration. Adriatic coast is used as surroundings, because it is highly indented, so it presents one of the most demanding challenges.**

***Index Terms*-- naval navigation system, Pocket PC device, path finding algorithm.**

## I. Introduction

PERSONAL Digital Assistants (PDA-s) and mobile phones are today's most common commodity, but software development on such systems still presents field of study that has not been completely explored. Development of software for embedded systems grows every day [13], but there is still place for in-house development of high quality solutions. Limited memory and resources, sets limits to the complexity of the developed software, and innovations. The old-style programming, manual memory management, optimized algorithms and space conservation, is more than recommended. One of the goals of this paper is presentation of real, fully functional software that has been developed with minimum investment in hardware platform while on the other hand supports mobility, usefulness and comfort.

Geographic information systems, more than all require mobility. This mobility has become available at small cost. Location based services support PDA-s and mobile phones, and are available in forms of GPS equipment, ground based services etc.

## II. Choice of Technologies and Platforms

Before the actual work, some research is required. First there was a choice of many forms of small GIS systems. Most of those are specializations for some area, like roadmaps, hiking support, land measurement, and naval navigation.

Naval navigation was the choice because of the deficit of this field of GIS in PPC and Smartphone development. The simplicity of development and optimal set of capabilities leaded to the choice of Pocket PC platform as a suitable platform. Although Pocket PC 2003 platform is quite old in terms of technology, it is highly compatible, so the project will work on Windows Mobile platforms as well. Because of the similar technologies, PPC project can be easily migrated on Smartphone platform.

The speed and memory requirements can be critical for such system, so it should be taken into consideration as well. At the time when project was started, the .Microsoft .net compact framework was still new technology, and the memory management capabilities of C++ programming language and MFC libraries proved to be perfect balance between resource consumption and development quality.

The entire set of technologies and development tools is available online, and can be acquired by anyone. MFC is well documented, supported and compatible.

### *A. Additional Equipment*

The use of GPS system is mandatory in every today's GIS system so the additional GPS equipment was acquired. Today's GPS equipment is quite cheap, correct, and easy to integrate. GPS Equipment can be connected in various ways, so there are Bluetooth, serial, and even embedded GPS modules. Most of the modules conform to standards like NMEA (National Marine Electronics Association) and SiRF [1]. Although the entire development can be done using emulators, the connection with GPS is not easily emulated, so the actual Pocket PC device was used.

## III. Specific Requirements and Key Aspects for System Implementation

GIS systems require extensive knowledge and development experience to be executed because of the whole array of multidisciplinary problems. During development of such project one will have to use knowledge from fields of computer graphics [7], robotics [14], location services [8], [12] and embedded development [10]. Those areas are result of project requirements.

The following problems had to be solved.

1. Charts presentation using vector data and one of the popular data formats, or its modifications.
2. Getting position using GPS system and GPS equipment.
3. Path-finding for naval navigation. Finding the algorithm for safe path on indented coast.
4. Integration of user data in charts.

This work has been founded by Ministry of Science, Technology and Informatics of the Republic of Croatia.
M. Lakotić, was student at the Department of Electronics, Microelectronics, Computer and Intelligent Systems, University of Zagreb, Faculty of Electrical Engineering and Computing, Unska3, 10 000 Zagreb, Croatia, (e-mail: miroslav@codeit.hr),
Ž. Mihajlović is with the Department of Electronics, Microelectronics, Computer and Intelligent Systems, University of Zagreb, Faculty of Electrical Engineering and Computing, Unska3, 10 000 Zagreb, Croatia, (e-mail: zeljka.mihajlovic@fer.hr).

5. Acceptable user interface.
6. Acceptable execution speed on most Pocket PC 2003 device.

### A. Charts Presentation

The simplistic approach is the best solution because it will conserve resources and processor time of the device. ESRI SHP file format is a simple set of vector data that can store points, lines and polygons [2]. PPC platform and MFC supports methods for point, line and polygon drawing, making implementation of chart drawing quite simple and fast. Nonetheless some optimization has to be done in order to present different areas of charts. For this purpose the implementation of simple LOD (Level of Detail) algorithm is implemented. LOD algorithm simply ignores the points of lines that are not visible for the current zoom value. The line is not visible if it's length is less than one pixel on the graphics device. Algorithm simplifies lines by calculating distance between current and previous line. If that distance is shorter than given limit for the current zoom value, the current point is ignored in the result set. Fig 1. presents example, where different LODs are applied.

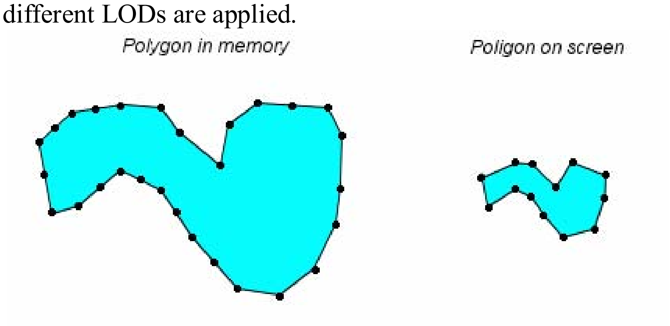


Fig 1. Two level of details LOD are shown: a) Left - Original polygon (in device memory) and b) Right - simplified polygon are presented.

### B. Getting GPS Position

GPS modules can be connected to a PPC device using few interfaces. Although physical interface varies, the data can be accesses through standard serial port programming. Most GPS modules support NMEA 0183 industrial protocol for data transfer. NMEA 0183 was originally created as a standard for naval equipment buses [3]. GPS data is just small portion of the full NMEA protocol. The protocol uses textual strings that can be parsed to get required data. GPS module refreshes information in regular intervals by broadcasting NMEA sentences.

Advanced control of GPS module is possible, but in most cases it requires implementation of different protocols and is different for different manufacturers.

The precision of the position lock depends on the environment and GPS module. At least three satellites have to be in range of the GPS module to get a position fix, and as the number of satellites grows, so does the precision [6]. Today very crucial characteristic of the module is also the initial locking time that can be from only few seconds up to few minutes.

### C. Expanding Polygons

The algorithm has to be able to plot the optimal, safe course between polygons, considering minimal distance from coast. To support the path-finding method, number of other algorithms had to be implemented. First necessary algorithm is the expansion of every polygon on the chart. The problem is because most of the polygons on the chart are concave, and can't be easily expanded (Fig 2). Few algorithms were taken under consideration including the decomposition of concave polygon into convex ones, but such algorithm takes too much time on limited PPC device. Finally the hybrid method is used to solve the problem with acceptable time and quality of result. The method first expands the concave polygon just as it was for the convex one, and then removes all intersecting curves as on Fig 2. The polygon is also simplified, so it shortens the time needed to calculate the route.

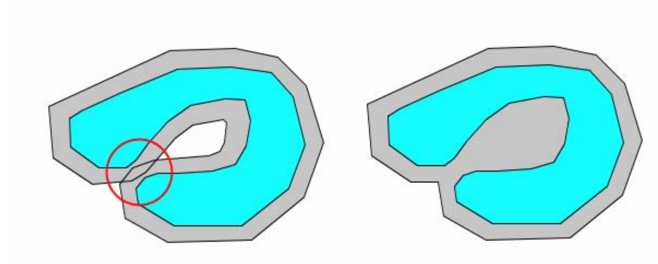

Fig 2. Expansion and simplification of concave polygons.

Expansion is executed by pushing every point of the polygon outward. The direction of the expansion fallows the normal vector drawn on the line through neighboring points (Fig 3). Red, blue and purple lines and respective doted lines represent directions of expansion of cyan polygon. The path-finding will use expanded polygons taking into consideration minimal distance from the coast line.

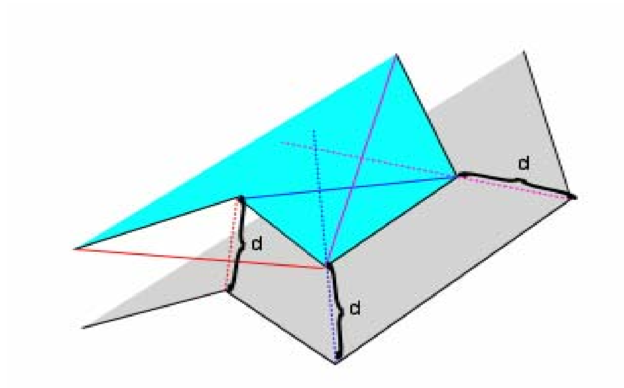


Fig 3. Expansion of points using neighboring points.

### D. Minimal Distance Calculations

After polygon expansion, additional problems arise. Two of these problems are closed areas and intersecting polygons. Every polygon, that is closer to other less than two times minimal distance, will cause intersection. In order to solve this, some data need to be calculated before execution of the program. This data show whether some point of the polygon has another polygon close by. If the polygons intersect, the path-finding algorithm will treat them as one polygon.

The problem with closed area is solved by limiting the depth of the path-finding recursion.

*E. Path-Finding Algorithm*

Implementation of the path-finding algorithm was the most critical part of solving the current project. Although many path-finding algorithms exist [14], [15], the one that has to chart path on indented coast with big number of polygons and points was impossible to find, so the new solution was in order.

Requirements for the path-finding algorithm are next:

1. Algorithm works with up to 100 000 points.
2. Algorithm converges in finite time.
3. Resulting route must be optimal, or close to optimal
4. Minimal distance from shore must be taken into account.
5. Algorithm works with concave polygons.
6. Algorithm must be robust and resistant to errors in data.
7. Resulting route must be level of detail independent.
8. Calculation time must be acceptable.

The path-finding algorithm is based on line intersection method and is supported by number of mathematical and geometrical equations and previously described algorithms.

Steps of the path-finding algorithm:

1. The starting and ending points are set
2. Line is set through starting and ending point, and the algorithm finds polygons that intersect that line (Fig 4. and Fig 5.).
3. Algorithm locates points on the intersecting polygons that are most distant from the intersecting line. If the polygons are too close, they are treated as one, as in Fig 5.

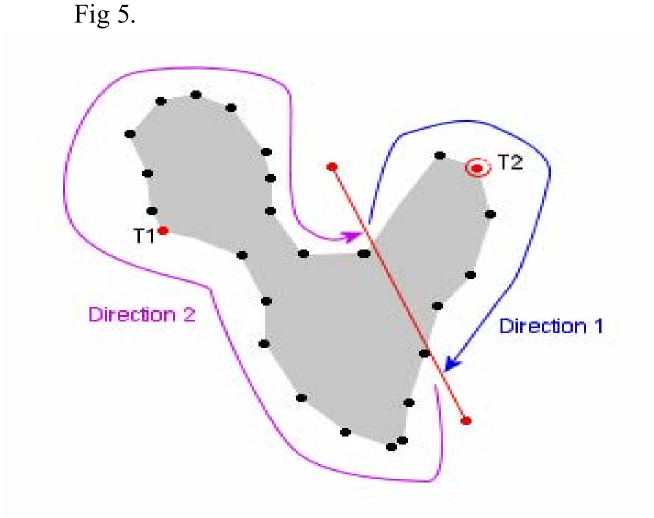


Fig 4 . Search for the most distant point on the intersecting polygon.

Those points are found by search around the polygon in both directions (Fig 4). There is always even number of resulting points. Fig 5. presents two polygons that are too close and most distant points $T_1$ and $T_2$.

4. The routes are constructed through every point individually, and are put on the tree of routes (Fig 7). Every point from previous step will have its own resulting route. The next step is executed on the best (shortest) route from this step, and will be put on the tree as children of the current one. Only leafs of the route tree can be taken in the next step.
5. Algorithm ends if any leaf route does not intersect polygons. The resulting route is the shortest one.

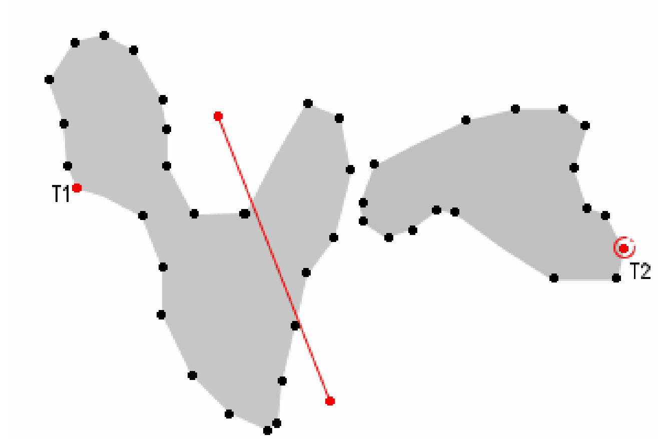


Fig 5. Search for the most distant points in case the distance between two polygons is less than two times minimal distance from shore.

After determination of points on intersection polygons, new paths are generated and evaluated. The problem arises on oddly shaped polygons that are close to each other as in Fig 6. Those polygons are treated as one only if there is point that is closer than two times minimal distance from shore. The entire neighboring polygon is included in the round search for most distant point. The algorithm is applied recursively, and the tree of routes is constructed like in Fig 7.

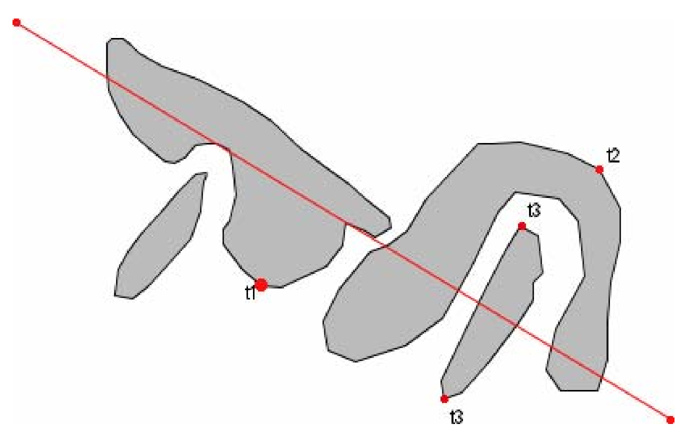


Fig 6. First step of the path-finding algorithm.

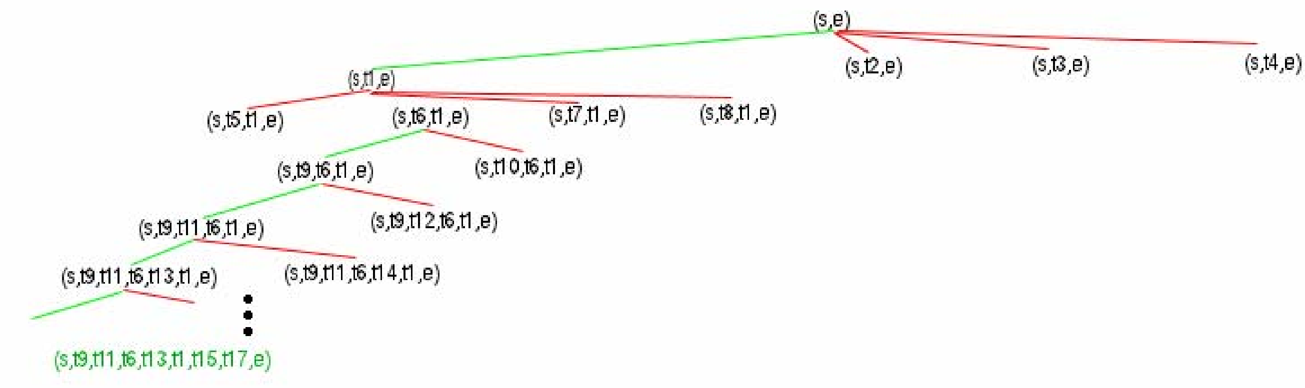


Fig 7. Tree of routes is constructed through iterations of the path-finding algorithm.

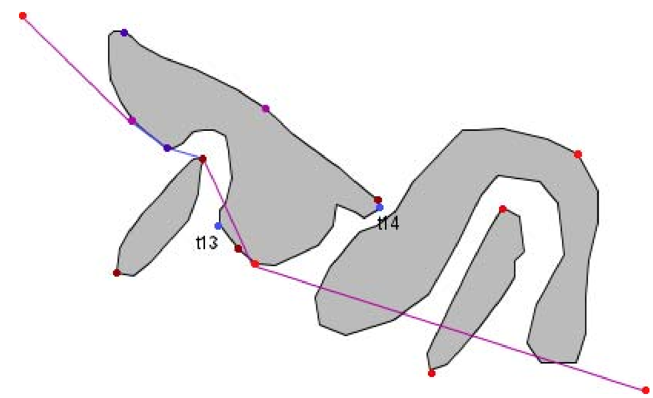


Fig 8. Intermediate step of the path-finding algorithm.

The final phase of the algorithm is always the best leaf. The only difference between candidate for next iteration and final path is the number of intersecting polygons. Intermediate step of the path-finding algorithm is shown in Fig 8. where intersections of polygons still exist. Final path intersects no polygons, and the candidate for next iteration does (Fig 9.).

The tree of routes consists of all resulting paths for all resulting points, and it's depth should be limited.

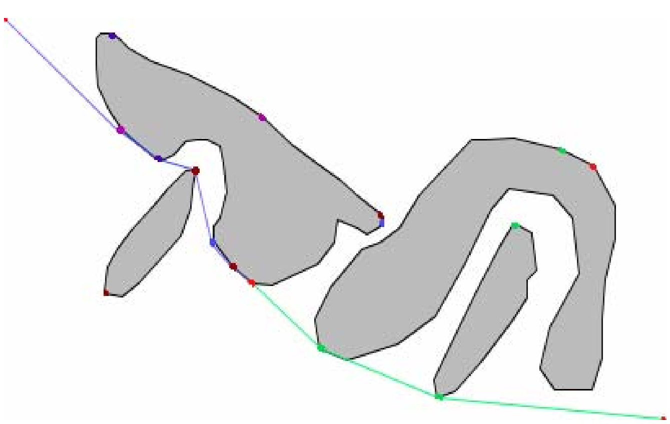

Fig 9. Final step of the path-finding algorithm.

### *F. Numerical Considerations*

The algorithm is initialized with starting and ending point (Fig 6), which are presented in tree of routes (Fig 7.) as pair $(s, e)$. Straight line is calculated for such pair. Implicit equation of line is appropriate:

$$a\,x + b\,y + c = 0\,, \tag{1}$$

because when it is normalized with $\sqrt{a^2 + b^2}$ the calculation of distance between point and line requires only two multiplication and two additions. For each point on polygon distance is calculated by substitution of its coordinates in equation (1) and the most distant point is found. The most distant points are placed in tree of routes and according to shortest route next step is defined. This is hill climbing optimization technique that is more appropriate than A$^*$ algorithm within given constraints of limited performances of the Pocket PC platform.

### *G. Data Integration*

Final project need's to present data about areas of interest and potential dangers and warnings. This data is embedded with point vector data, and can have different types. In current project, data is available on special navigation marks, cities, depths of the sea, marines, hotels, restaurants, gas stations etc. Data can't be binded with lines or polygons, and is not ESRI SHP compatible.

### *H. Distance Transformation*

All points inside charts are represented using WGS84 format, so some calculation is required to get an adequate distance calculation. Longitude $lon_1$, $lon_2$ and Latitude $lat_1$, $lat_2$ are angles on a sphere, and to get a distance in miles or kilometer, the spherical or ellipsoid model. In current project approximation of spherical model is used.

$$\begin{aligned} d = a \cdot \operatorname{acos}(\sin(lat_1) \cdot \sin(lat_2) + \\ + \cos(lat_1) \cdot \cos(lat_2) \cdot \cos(lon_1 - lon_2)) \end{aligned}, \tag{1}$$

where *a* is the radius of earth in miles (3,963 miles). Equation (1) gives distance *d* on sphere in miles.

Spherical model is not the best model for distance calculation, and can be used only for smaller distances. More precise distance can be calculated using ellipsoid model and map of referent null points. This map takes into account imperfections of Earths crust.

### I. User Interface

Final user interface proved to be satisfactory, and through tree different view user can fully control the system, plot routes, view data and view charts.

Wide view enables user to use the entire screen to view map and data (Fig 10). Only limited set of options is available in this view.


Fig 10. Wide view of the chart on Pocket PC.

Standard view is shown in Fig 11. and it presents all buttons and options, as well as the zoom buttons that can be used to change the area of the chart that is presented.

There is also GPS data view shown on Fig 12. that enables user to observe detailed GPS data like position, speed, direction, locked satellites etc. Database view shows data that is marked as point of interest for the user. Such objects are cities, gas stations, marines etc.

Map of the Croatian coast [9] is used as a test board for the proposed path finding algorithm. In Adriatic see there are altogether 1185 islands, cliffs and ledges. Length of sea coast is 5835km, 4058km of which are coasts of islands. The Croatian coast can be divided into Istria, Croatian Primorje and Kvarner. Coast line data was adapted for given constraints on Pocket PC and part of the Šibenik archipelago is shown in Fig 10. and Fig 11. As numerous of island are processed, task of path finding for a given starting and destination position is very challenging. Depending on local complexity, on the PPC, path finding requires form few seconds to few minutes for quite complex scenes.

## IV. Conclusion and Future Work

Although the described project is satisfactory for its original purpose, there is more than enough place for improvements. First planned improvement is making project more compatible and interoperable. This would include additional work on data integration, to fully support ESRI SHP format.


Fig 11. Standard view on Pocket PC.


Fig 12. GPS data view for detailed position and course information.

The second possible improvement is enabling charts exchange over services on the Internet. Mobile phones also present a great opportunity for further development because they enable communication with GSM network. This is excellent for implementing cheap tracking services.

Few more improvements present themselves in embedded development, like OpenGL-ES that can be used to present high quality 3D graphics on embedded devices. The entire project could be transferred to a 3D system, and with associated data formats could accommodate users like air glider pilots where third dimension is more than optional.

## V. Acknowledgment

We would like to thank all the people whose publications helped in making this project possible.

## VI. References

[1] SiRF Technology Inc.: „NMEA Reference Manual – Revision 1.3“, January 2005.

[2] Environmental System Research Institute Inc.: „ESRI Shapefile Technical Description“, July 1998.

[3] Klaus Betke: „The NMEA 0183 Protocol“, Standard for Interfacing Marine Electronics Devices, National Marine Electronics Association, Severna Park, Maryland, USA, August 2001.

[4] National Imagery and Mapping Agency: „Department of Defense World Geodetic System 1984.“, Third edition, June 2004.

[5] Peter H. Dana: „Geodetic Datum Overview“, from the Internet, http://www.colorado.edu/geography/gcraft/notes/datum/datum.html

[6] GPS satellites list, from the Internet, http://celestrak.com/NORAD/elements/gps-ops.txt

[7] Donald Hearn and M. Pauline Baker: „Computer Graphics with OpenGL“, third edition, Prentice Hall, 2003.

[8] Wikipedia, from the Internet, www.wikipedia.org http://en.wikipedia.org/wiki/GPS http://en.wikipedia.org/wiki/Trilateration

[9] Mozaik knjiga: „Veliki atlas Hrvatske“, 2002.

[10] Microsoft corporation: Microsoft Developer Network, msdn.microsoft.com

[11] Theresa Rowland: „Relativity and the Global Positioning system“, spring 2006. http://www.uncw.edu/phy/documents/Rowland_06.pdf

[12] "Position Determination with GPS", from the Internet, http://www.kowoma.de/en/gps/positioning.htm

[13] William G. Griswold, "Five Enablers for Mobile 2.0," *Computer*, Vol. 40, No. 10, pp. 96-98, October, 2007.

[14] Veeraswamy, A. and Amavasai, B. Optimal path planning using an improved A * algorithm for homeland security applications. In *Proceedings of the 24th IASTED international Conference on Artificial intelligence and Applications,* Innsbruck, Austria, February 13 - 16, 2006.

[15] V. Deved, Ed. International Association of Science and Technology for Development. ACTA Press, Anaheim, CA, 50-55.

[16] Steven S. Skiena, "The Stony Brook Algorithm Repository" (http://www2.toki.or.id/book/AlgDesignManual/WEBSITE/INDEX.HTM).

## VII. Biographies

**Miroslav Lakotić** received B.Sc. degree in computer science from Faculty of Electrical Engineering and Computing, University of Zagreb in 2007. The title of B.Sc. theses is "Presentation methods for navigation on embedded system". His areas of interest include geographic information systems, computer graphics and new technologies in web development.

**Željka Mihajlović** is an Associate professor at the Department of Electronics, Microelectronics, Computer and Intelligent Systems, of the University of Zagreb, Croatia. She received the B.S. degree in Electrical Engineering in 1988, the M.S. and Ph.D. degrees in Computer Science in 1993 and 1998 respectively, from the Faculty of Electrical Engineering and Computing, University of Zagreb. Her current research interest includes computer graphics and visualization algorithms, and reconstruction techniques.