

Virtual reality for blind users

Roman Berka and Pavel Slavík

Department of Computer Science and Engineering, Czech Technical University Prague,
Karlovo nám. 13, 121 35 Prague, CZECH REPUBLIC

berka@sgi.felk.cvut.cz, slavik@cslab.felk.cvut.cz

ABSTRACT

The paper describes a solution to the problem how blind users can work with the three dimensional information available in the computer environment. This serious problem gains more and more importance as the use of three dimensional information is permanently growing. An experimental system that allows such communication has been designed and implemented. This system can be used as a sort of kernel for applications of various kinds that deal with the problem of communication between a blind user and a three dimensional model.

1. INTRODUCTION

Virtual reality has gained more and more importance in recent years. This term covers a large spectrum of meanings which means that its interpretation is rather complicated in some cases. In this paper we will deal with the meaning that is used in the web environment and in similar applications. These applications cover for example virtual walkthroughs in virtual scenes. Both these applications and the web varieties use the 3D model of a scene the user works with. Unfortunately there is a large group of users that can take no benefit from the possibilities offered by virtual reality. This group of users consists of blind users [1, William A. Barry, 1994],[3, D. Hermsdorf, www]. As the share of graphical information (and 3D information in particular) in various media (especially on the web [2, R. Berka, 1996]) steadily increases, this group of users is more and more limited in the usage of information resources, the importance of which is permanently growing. This situation requires an immediate solution. It is necessary to develop tools that will allow the blind user to obtain information about a 3D scene (or its model) in a form that will be suitable for such a user. Such a user can not perceive the scene in a global way as normal users can. The solution to this problem is to develop a special type of communication that will make it possible to ask questions of various kinds. The answers to these questions will provide the user with information both structural (e.g. which object is to the left of the current object) and geometrical (e.g. how far from the current position is the object B). Moreover, it is necessary to have some statements at hand by means of which the movement of the user from one position to another could be described. The system that would satisfy all these needs will be based on a special language, the statements of which will allow communication between the user and the system working with a 3D scene.

The communication with the 3D model can have various frameworks. Let us discuss some typical situations:

1. Navigation in a 3D virtual environment. As an example we can give the situation when a blind user tries to learn about some building he/she is going to visit the next day (bank, hospital, railway station etc.). By means of the system the user can obtain information about the structure of the building and thus some idea how to get from point A (let us say the entrance) to point B (let us say the wash-room). The user can train his/her movement in an unknown environment. The communication between the user and the system can be performed by means of standard input and output devices.
2. The extension of the previous example can be done in such a way that the user will use the model for real navigation in unknown environment. This will require some additional equipment that will allow him/her to establish their exact position in space. As the user moves in a building, where a lot of obstacles can be, it is necessary to establish the position with very high accuracy. One solution would be to have some signal sources (ultrasound, infrared, radio etc.) that could be received by the appropriate equipment attached to the computer carried by the user. Processing such a signal would establish the exact position of the user. Then the user can either ask questions about his/her position or can be automatically navigated to the final position. The situation described corresponds to the use of wearable

computers. The information obtained has a very close link to reality and the information from the 3D model can be projected to the real environment.

3. A modification of the above given approach can lead to the development of a special 3D browser for web application. This browser will be oriented to blind users. The idea is that the 3D scene description will be in VRML and the user can ask questions about the situation in the immediate vicinity of the current position, etc. In such a way the blind user can get information about the 3D scene. This information was not available to him/her when using the web in a standard way (the user could work mostly with textual information only).

2. THE TRAINING AND NAVIGATION SYSTEM EYE

Within the framework of our research, we have implemented the EYE system. This system satisfies the requirements stated above. Using the current implementation of this system a blind user can train his/her movement in an unknown environment (e.g. railway station halls, airport halls and other public accessible areas).

Based on a 2D map, the EYE builds a virtual scene and locates the user in this scene. The user can freely walk through the scene and the system informs him/her about important places, objects and obstacles. The system has two data sources: end-user interaction and the 2D map created by user-author. This approach is based on three components:

- a digital model of the scene
- the way the user communicates with the system
- the way the system communicates with the user

The digital model of the scene is reduced to a 2D form since it contains all information necessary to solve our task. In addition, the 2D solution is much easier than the use of the 3D algorithms used in virtual reality. The map represents necessary information including obstacles, significant objects and important paths among these objects.

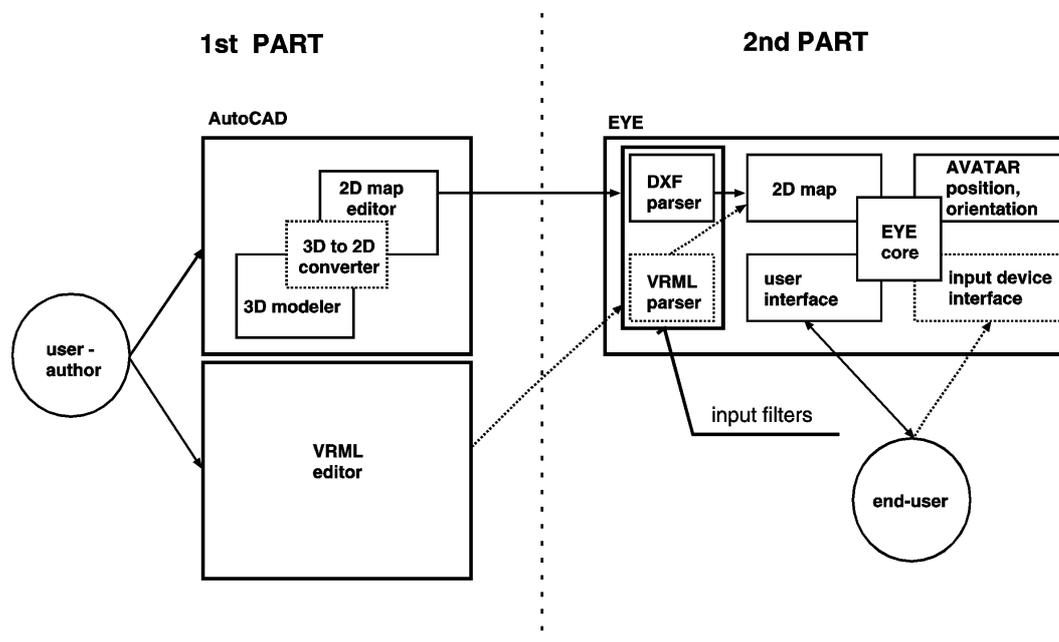


Figure 1: The data flow through the system

One of the most important parts of the system is the data preparation module. Since the data structure imported as a 2D map has to meet special requirements it is impossible to work with a general 3D model. There are two ways to obtain a suitable 2D data. First, the 2D map is generated by conversion from any 3D model and certain additional information is then appended. Second, the 2D map is generated using a special software tool. We used the second way in our approach in which a special application for map creation was developed.

The communication between the system and the user and vice versa represents a set of problems due to the fact that the information will be presented to the blind end-user in a non-visual form using some screen-reading tools or a synthetic voice. In addition, it is expected that the set of potential users will be wide and that every user will have to learn to work with the system. Therefore, there are demands on the communication protocol between the user and the EYE. First, the data flow coming in direction from the system to the user has to be minimal as the user must be able to absorb all information given by the EYE. Second, the system is controlled by a set of commands in which the minimal necessary subset needed to work with the system is reduced to 3 commands. All commands have to be very simple as they must be easy to memorise.

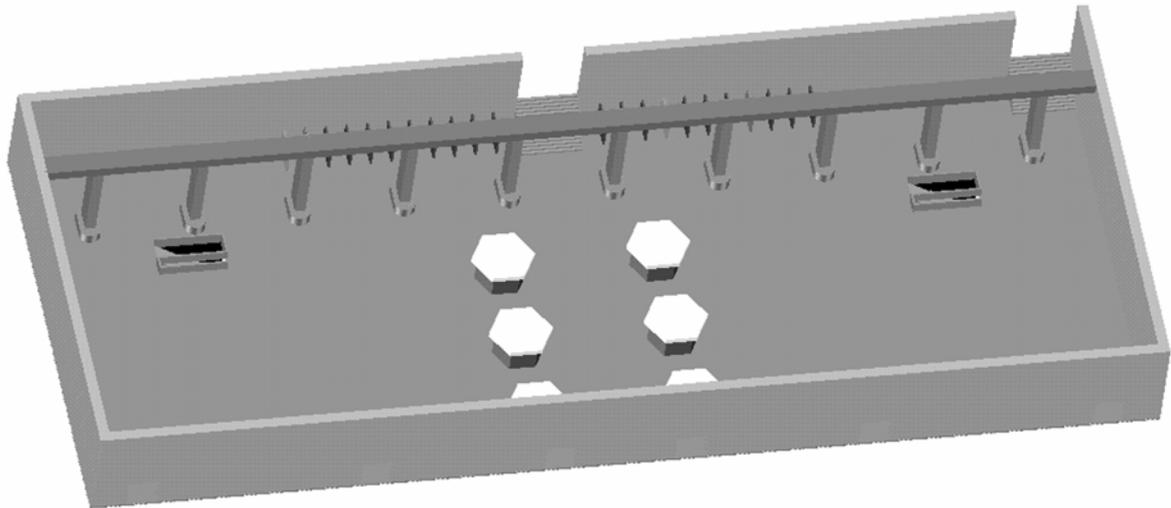


Figure 2: 3D model of Wilson railway station hall in Prague

2.1 Two Parts of the System.

From the view of an author who generates input data, the system consists of two parts (see Fig. 1). The first is a data preparation process during which the scene description is created using text and geometrical data. The second part consists of importing a scene description to the system EYE and of the training process. Both parts are independent and are not necessarily parts of one program. The end-user then works with the second part only.

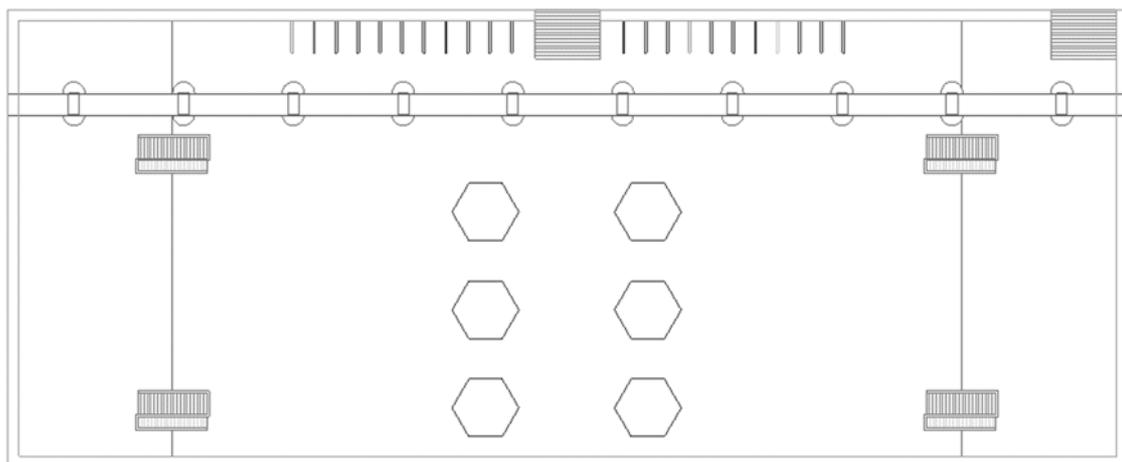


Figure 3: Top view to model of Wilson railway station

A special application called a "map editor" (see Fig. 1 - the first part) is used to create 2D maps according to rules which make the data readable by the EYE system. Generally, the map editor loads any geometrical description of the scene, converts it to the acceptable form and appends additional information. This

information includes possible paths among important places of the scene and also the special attributes of all objects. These attributes have a text form. Every object in the scene has a set of predefined attributes with the meaning of names and short descriptions of the object (object functionality - e.g. "newspaper stand").

The above mentioned fact means that some part of the input information is removed as e.g. the third dimension of objects in the scene and the other information is added using a map editor. This editor makes it possible to add some attributes in text form. These attributes, in most cases describing the objects, are later included into messages sent to the user during conversation. Next, using the map editor the author of the map defines all possible paths between every two potential goals of the user. These paths are generated automatically by the map editor. Later, the author can edit generated paths by hand and correct them.

Fig. 2 shows a 3D view of part of Prague's Wilson railway station hall and Fig. 3 shows a view of the same scene from the top. Fig. 4 shows a 2D map corresponding to the previous view. This 2D map already includes added information.

All these pictures demonstrate the two individual steps of the data preparation process. During the first step, 3D input data are converted into the 2D description according to the special rules. In the next step, additional information is appended as described in the previous two paragraphs.

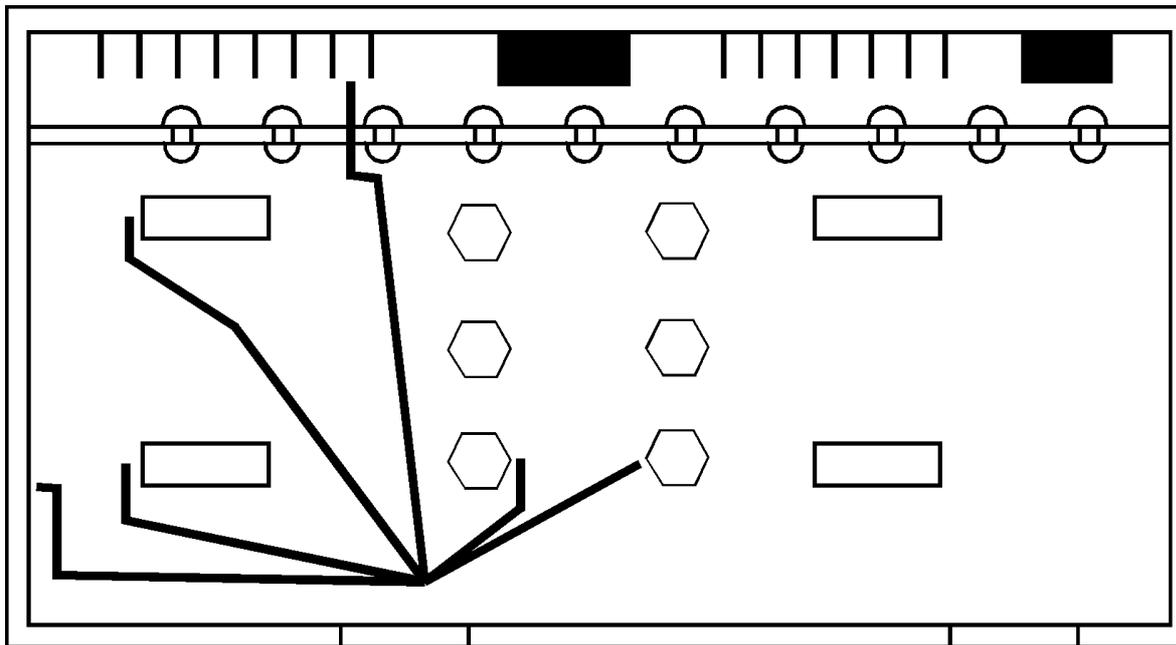


Figure 4: *Processed 2D map of Wilson railway station hall in Prague*

The second part is represented directly by the EYE system as shown in Fig. 1. It consists of modules communicating via a core which processes all data flows in the system. The first activity of the EYE after start up is reading the map. The input data representing the map is entered through input filters which transform this data into internal data structures. These filters should be specialised for any specific input data format. Our current version includes only one input filter for DXF (AutoCAD Drawing eXchange Format) used usually in area of CAD systems. To extend the possibility of processing another format (e.g. VRML), a new filter has to be implemented and linked to the system as shown in Fig. 1.

When the map is stored in the desired structure communication with the user can start. First of all, the initial localisation of the user in the scene should be made. Information about the position and orientation of the user are stored in an object representing the user in the scene. This object will be called AVATAR (according to terminology used in the area of VR). The AVATAR has a circular shape with a size corresponding to the projection of the human body to the ground. The core uses this representation to compute distances of the AVATAR from the goal position and to detect AVATAR's collisions with other objects in the scene.

The user interface module communicates with the user by means of natural language using short formulations in text form. The user's communication with the system consists of a small set of commands. These commands are transformed into internal instructions, in most cases changing position or orientation of the AVATAR. While the localisation of the AVATAR is based on information given by the user it is also

possible to use a special input device interface which gets this information from devices measuring orientation and distances. Such a solution could also be used as navigation system in the real environment.

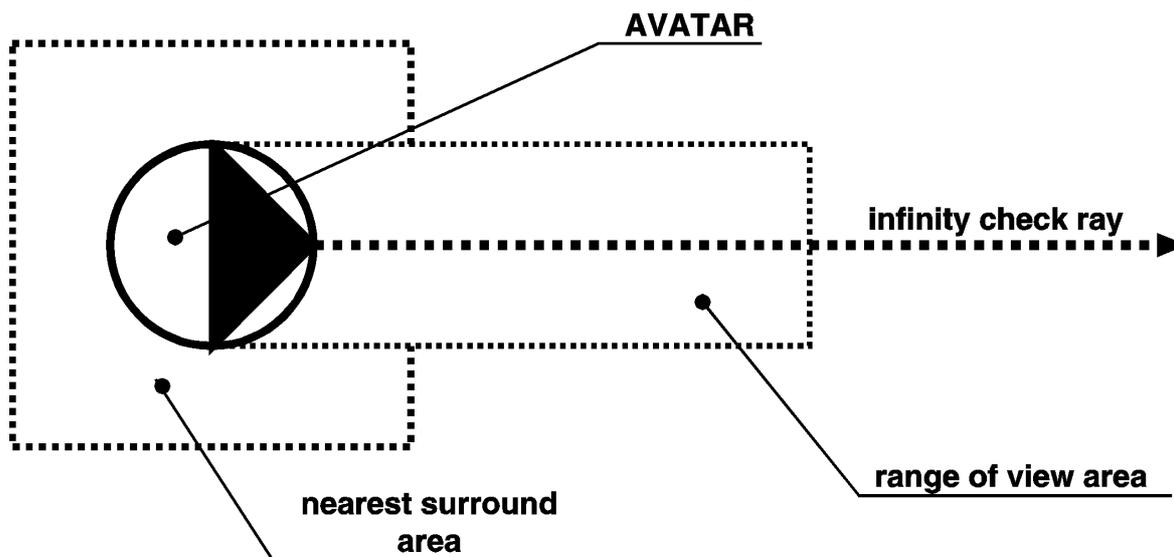


Figure 5: The AVATAR and its sensitive areas

2.2 The System - User Communication

All objects in the scene are potential obstacles for the user. Since the system is proposed for use by blind people the EYE recognises six classes of objects with respect to the danger level of collision with them (e.g. a door represents another class than a hole).

PORTAL - represents doors

ACCESSIBLE - represents objects accessible without danger (e.g. grass plot along the path)

WALL - represents general obstacles

HANGING - represents obstacles hanging in given height over the ground and user's head could be hit by the obstacle

HOLE - represents dangerous obstacles as holes in the floor

UNKNOWN - represents obstacles without specific description

All classes differ in reaction from the side of the system when the user collides with the obstacle (e.g. a collision with the door will evoke entering to an other room but on the other hand a collision with a wall evokes a warning message).

The AVATAR is equipped with sensors which are sensitive to the existence of obstacles in specified areas around him. As the AVATAR is walking through the virtual scene the EYE system informs him/her about obstacles which appear in his/her surroundings. Fig. 5 shows the AVATAR and his/her sensitive areas. The AVATAR has three active sensors. The first one checks the area in front of the AVATAR and corresponds to a healthy human's range of view. The second sensor works as a ray sent by the AVATAR into the space. The system then returns information about objects intersected by the ray. This sensor is equivalent to the vision to large distances needed for orientation in large space. The last sensor informs the AVATAR about objects in the nearest area surrounding him. The two last sensors are needed for orientation and they are evoked on request of the user. The first sensor is activated every time the AVATAR changes its position or orientation and gives information concerning which objects would be potential obstacles for the walking user/AVATAR.

As the AVATAR moves in the scene the system checks his/her collisions with all other objects and informs the user about objects in the scene and about the possibility of colliding with them. Since the potential user will communicate with the system using special peripheries for blind people the messages returned by the EYE should be as short as possible. This requirement is important for the data preparation process where most of the attributes have a text form and they are used as parts of messages sent to the user.

The user gives the system very simple commands from a small set (see this complete set in APPENDIX A). The basic two commands are used to change the position or orientation of the AVATAR. Using the other two commands the user can check either the immediate surroundings or distant area in front of him/her. The

other commands are used to control the EYE system where the user can set the length of his/her step or the range of the area checked by the "range of view" sensor.

For easier communication the system uses special units which correspond to terminology used by blind people. The distances are measured in steps in which the step length is optional. The direction is given in a natural form in which the direction is derived from a clock-arm position relative to the AVATAR. Fig. 6 shows all possible directions with respect to AVATAR's orientation. Here, for example 3 hours equals east.

The initial communication of the user in the virtual scene starts with a list of possible goals in the current scene (see APPENDIX B). This list is offered by the system. After the user answers he/she enters the room and the system gives him/her basic information about the room, about his/her immediate surroundings and gives him/her directions to the chosen goal.

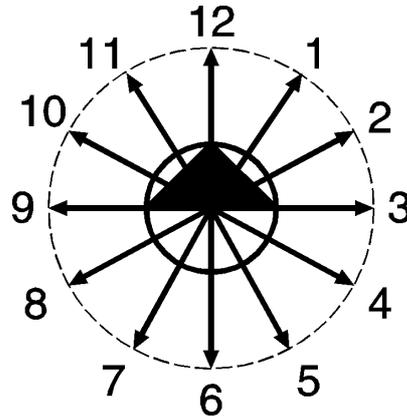


Figure 6: Possible directions measured in clock-arm positions

The user can freely walk through the scene and the system navigates him/her to the right path. When the user reaches the goal the system stops the navigation process and gives him/her a new list of possible goals.

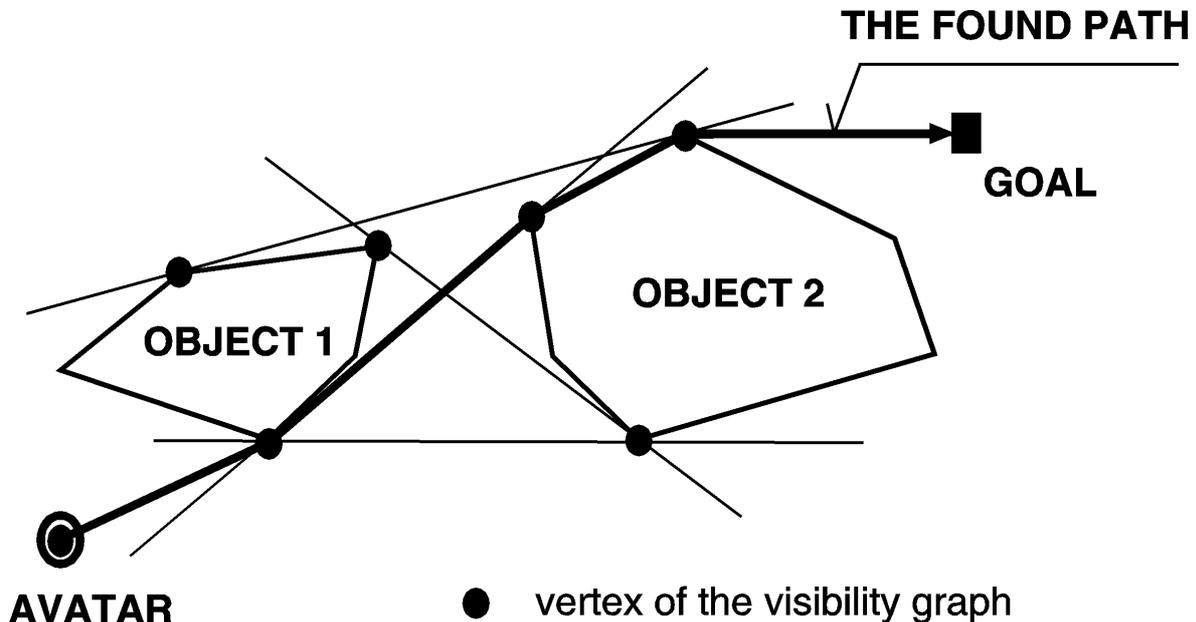


Figure 7: Example using the tangent algorithm

3. IMPLEMENTATION DETAILS

The system is realised in C++ programming language using object oriented technology where individual objects represent real objects in the system as described above. This approach is suitable for most similar problems solved in virtual reality. The following three main problems had to be solved in order to implement the EYE.

- integration of graphical and non-graphical data in one data format
- navigation algorithms
- communication peripherals

The first is not only a file format problem. It is a problem of data import from standard graphical systems. Since the data read by the EYE is required to include special information it was necessary to implement a special application under any standard graphical system which is able to produce data directly in the required format.

We have chosen AutoCAD as the drawing system and the application is now implemented using an ADS development extension. The DXF format meets the requirements specified above and therefore we used it as a format for our data. See Fig. 1 which shows the data flow through both parts of the system and between the system and peripherals.

The navigation algorithms use polylines representing the paths included in the 2D maps during data preparation process. Every polyline has a set of attributes. Two of these attributes are labels at both end-points of every path. These labels correspond to the names of places or objects where the path is terminated. They are also used for user orientation in the scene. The user is navigated by the system using vertices of the polyline. At any time the user is informed about the direction to the visible vertex lying nearest the goal on the specified path.

During the data preparation process, the path can be created by hand or automatically using the shortest path searching algorithm implemented as a subpart of the first part in Fig. 1. The path searching algorithm uses a visibility graph method [4, F. Kuo-Chin, 1994] where the graph is created by a "tangent algorithm". The basic idea of the tangent algorithm assumes that there exist exactly four tangents between two non-overlapping 2D objects in a plane. Creating tangents between every two objects we obtain sets of vertices and edges (including edges from original objects and tangents) which define the desired visibility graph. The path between every two given points in the scene is then found as the shortest path in the visibility graph. The Fig.7 demonstrates the tangent algorithms on two general objects. The marked vertices of both objects are also vertices of the visibility graph. The some edges of the objects and parts of tangents are the edges of the graph.

4. FUTURE WORK

The system is still under development. In the near future all parts that are not yet finished will be completed and the system will be fully operational. In addition there are many possibilities for extending the system. These extensions would allow us to use the system and the results of the research performed in specific applications. The extensions will cover:

- the use of voice input and output. This will significantly ease the use of the system for blind users - especially in the case when navigation in a real environment takes place. The overall cost of the system will not increase significantly as the voice cards and speech synthesizers can be purchased at a reasonable price.
- the use of auxiliary sound navigation. This feature is like navigation in real 3D space when blind persons generate sounds when walking (e.g. the sound of their steps, clicks with their white stick etc.). Such a sound typically generates an echo that gives an idea about the size of the room in which the person is and also about its basic configuration. There are currently methods under development that will generate 3D sound in a virtual model described by means of VRML (or in another format). The development also takes into account the materials which the walls, ceiling, objects in the room, etc. are made from. This feature is of interest in case the user trains his/her movement in the training environment.
- the research performed up to now was also targeted to the development of real 3D browser for the blind users. Besides system modules already existing it would be possible to include both "sound" modules described above. Such an approach will allow blind users to work to a certain extent with 3D information available on the web. This kind of information is not currently available for them to a satisfactory extent. Such a development will ease the use of the web considerably. This activity is linked with other research performed in our department where pictorial information is transformed into structured textual information what will allow the user to work with pictorial information as well. Integration of these two approaches will allow the development of special modules that will enhance

current web browsers. These modules will compensate in certain ways for the handicap of the given class of users.

- a very important field of application will be the use of modules developed in the environment of networks. Currently there exists a lot of applications where virtual worlds have been created (e.g. WorldChat). The users virtually meet in these worlds and communicate in a different way. As the 3D environment is described by means of standard formats (like VRML) it would be possible to enhance existing software with modules that will allow blind users to use these systems as well. Another aspect of communication in networks will be the navigation in actual space.

When speaking about the application of the system developed in the framework of wearable computers there is an important feature that could substantially extend the field of use of such a system. Up to now we have assumed that the 3D model is stored permanently in the computer (on the hard disk) or on a floppy disk with the model which will be used. The direction in which the system can be developed is the use of the system in the network environment. In the basic version of such a system the situation will be characterised by having the possibility of acquiring the 3D model via network. This will give an opportunity to dynamically obtain information about the environment where the user currently is. This mode of use lies in fact in the field of mobile computing where the connection between the user and database (where 3D models are stored) can be accomplished by means of GSM or similar technologies. This approach will allow the user to get very detailed information about the environment even in very complicated cases (multi-storey buildings, etc.).

5. CONCLUSION

The first two versions of our new training and navigation system EYE were developed with the goal of making a large amount of 3D data accessible to blind people. This data can be used by blind users learning to move in unknown environments. Every user can communicate with the system using a few simple basic commands and the EYE system answers with short simple sentences. Therefore, the communication should be easy to learn for most potential users. We tested the system with a blind user who found all the desired objects in the scene. The first results of our experiment indicate the strategy of navigation in the virtual scene is correct and can be used in the future development of the system.

The system developed can be used as a kernel for various applications (3D browser, navigation in a real environment, virtual walk-throughs etc.). Some of the results achieved were obtained in the framework of national project GACR 201/96/0706: Graphical Communication Between Human and Computer for Blind People [5, P. Slavík, 1997].

6. REFERENCES

- [1] William A. Barry and John A. Gardner. Accessibility to Scientific Information by the Blind. <http://dots.physics.orst.edu/csun94.html>, 1994. Department of Computer Science, Cornell University.
- [2] R. Berka, M. Brachtl, A. Holecek, M. Novotny, J. Prikryl, P. Slavik, and J. aZra. Three-dimensional Information System on the World Wide Web. In *Multimedia '96*, pages 70-78, Wien, 1996. Eurographics, Springer-Verlag.
- [3] D. Hermsdorf. Teleworking for Disabled People. <http://www.yuri.org/webable/tedis.html>. The German National Research Center for Computer Science: TEDIS, Sankt Austin, Germany.
- [4] F. Kuo-Chin and L. Po-Chang. Solving Find Path Problem in Mapped Environments Using Modified A* Algorithm. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(9):1390-1396, september 1994.
- [5] P. Slavik and R. Berka et al. Graphical Communication Between Human and Computer for Blind People. Technical report, Czech Technical University Prague, 1997. national project GACR 201/96/0706, pp. 69.

APPENDIX A. THE COMMAND SET

The end-user can communicate with the EYE using ten simple commands. These commands are divided into three categories. The first category includes commands changing the position or orientation of the

AVATAR. The second category consists of commands requiring any information and the last class represents a group of system commands. Here is the list of all commands:

1. step [number of steps] - move a number of steps forward
2. turn [clock] - turn on the clock
3. back [number of iterations] - return a number of iterations back
4. look - give information about objects in the "range of view" area
5. describe - give summary information about the current room
6. infinity - give information about objects positioned far from the AVATAR
7. check - give information about objects in the immediate surrounding
8. range [number of steps] - sets the length of strip representing the "range of view" area
9. preset [distance in meters] - sets the user's step length
10. quit - terminates the program
11. ? - help

The user can type only the first character of every command for faster communication.

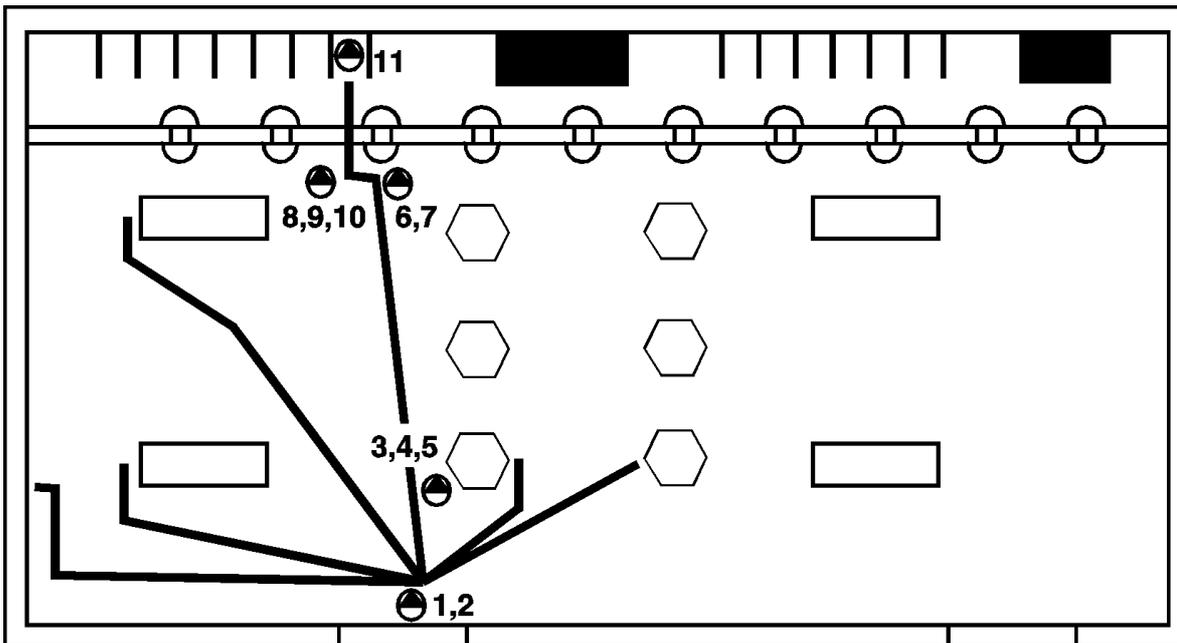


Figure 8: *The dialog corresponding with corresponding iterations between the user and the system*

APPENDIX B. EXAMPLE OF TYPICAL DIALOG

A typical dialog between an end-user and the system starts, as mentioned above, through the selection of a goal. Then the user gives the system commands changing its position until the goal is reached.

Fig. 8 shows the individual phases (iterations) of AVATAR's motion, and the corresponding sequence of commands follows. Fig. 8 describes situations after commands given by the user. The AVATAR is represented by symbols as on Fig. 5. Every symbol is marked by a set of numbers and every mark means a situation after a command with the corresponding number in parentheses.

The text of an example is written according to rules such that all messages given by the EYE are typeset by *italics* font and for all commands received from the user, the **bold** font is used.

This is EYE the Virtual Reality navigator for blind people version 2.0

Description of the room:

The Wilson railway station hall in Prague.

The door: north left entrance from park

Choose one path from the following list, type the corresponding number and confirm by pressing of ENTER, please.

0 ... ticket offices

1 ... underground toward Holesovice railway station

2 ... underground toward the center and South Town

3 ... post office

4 ... newspaper kiosk

5 ... fast food

path number: 0

from: north left entrance to: ticket offices

command (1): l

No objects detected in your range of view. The direction to your goal is 0 o'clock. The distance to the goal is 100 steps.

command (2): t 2

No objects detected in your range of view. The direction to your goal is 10 o'clock. The distance to the goal is 100 steps.

command (3): s 20

*objects in your range of view: fast food
The direction to your goal is 11 o'clock. The distance to the goal is 82 steps.*

command (4): c

objects in your immediate surroundings: fast food IN FRONT of you on your RIGHT hand.

command (5): t 11

No objects detected in your range of view. The direction to your goal is 0 o'clock. The distance to the goal is 82 steps.

command (6): s 65

*objects in your range of view: concrete column
The direction to your goal is 9 o'clock. The distance to the goal is 15 steps.*

command (7): t 9

No objects detected in your range of view. The direction to your goal is 0 o'clock. The distance to the goal is 15 steps.

command (8): s 5

No objects detected in your range of view. The direction to your goal is 3 o'clock. The distance to the goal is 13 steps.

command (9): t 3

No objects detected in your range of view. The direction to your goal is 0 o'clock. The distance to the goal is 13 steps.

command (10): i

far objects in infinity: ticket office

command (11): s 13

Your goal has been achieved, do you wish to continue to next possible goal?

command (12): n

Thank you for using the EYE. Goodbye.