A Context-Aware Locomotion Assistance Device for the Blind

Christophe Jacquet[1,2], Yacine Bellik[2], and Yolaine Bourda[1]

[1] Supélec, Plateau de Moulon, 91192 Gif-sur-Yvette Cedex, France
FirstName.LastName@supelec.fr
[2] LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France
FirstName.LastName@limsi.fr

Abstract. In this paper, we present a study which aims at designing a locomotion assistance device that can deliver semantic information about its surrounding environment at any time. As a first step towards this goal, we introduce an original model suited for the description of building structure, and we present an algorithm that exploits these descriptions. Then, we explain how it is possible to link semantics to structure. Finally, we expose some research directions for user positioning and human-computer interface design.

Introduction

Over the past few years, the LIMSI (Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur) and the LAC (Laboratoire Aimé Cotton) have been developing the *Teletact* (see Fig. 1), a locomotion assistance device for the blind [1, 2]. The system uses a laser telemeter to measure the distances to obstacles and transforms them into tactile vibrations or musical notes (the higher the tone, the closer the obstacle). A project to improve the system is under way in collaboration with Supélec: we want to give it the ability to provide *symbolic information* about pointed objects.

Fig. 1. Photo of the current *Teletact*.

After giving a short overview of the system, this paper discusses our preliminary results. We present a model for the description of architectural environments in buildings. Then, we introduce an algorithm capable of determining which default information is most relevant to the user. Next, we introduce semantic representations, and we show how to link them to building structure descriptions. This represents the work achieved so far. The last part of the paper presents future research directions for user positioning, description acquisition and user interaction.

1 System Overview

The system will try to determine its position thanks to a GPS (Global Positioning System) receiver where GPS reception is possible, and otherwise thanks to an inertial unit. The position calculated from these devices will then be matched against structural and semantic information embedded in the environment description retrieved from the ambient network (see section 5.1), so as to increase precision and compensate for positioning errors (see Fig. 2).

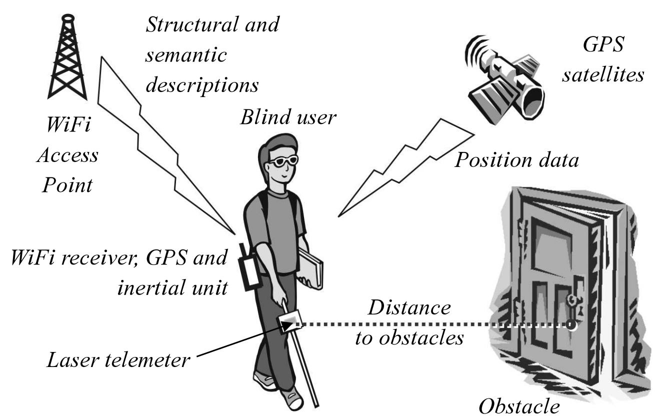

Fig. 2. Overview of the system.

Context-awareness will be enhanced thanks to telemeter data (as in existing devices the new system builds upon) and to light sensors that can provide additional information about light sources (sunlight, artificial light).

When the position of users has been determined, the system will give them context-related semantic information. And when they point at some specific object or location, the system will provide them with information *about* this object or location.

For instance, when a user points at their boss' door, current devices are able to tell them that "there is an obstacle three meters ahead". The device we are describing here will be able to add that "this obstacle is a door", and that "this door leads to the boss' office".

In everyday life, such a system can significantly improve blind people's lives, by giving them precise information about their environment.

2 Structure Modeling

To build this system, we must be able to model users' environments. To date, we have worked on building descriptions only. But of course, our model will eventually cover the full range of environment descriptions.

2.1 Existing Description Formats

Current formats for 3D-scene description (such as VRML [3] or X3D [4]) focus on describing the mere visual *appearance* of environments. Indeed, they target *sighted people* and try to render scenes with as many visual details as possible.

In contrast, we target *blind people* and thus we need to model the *structure* and *semantics* of environments.

The *structure* of architectural environments determines how architectural elements are organized to compose buildings. Structure is either visible (e.g. walls) or invisible (e.g. frontiers dividing a room in several zones, for instance smoking ones and non-smoking ones).

Symbolic data bring semantics to the structure they are associated with. For instance, these data may contain information about the owners of rooms in a building, access restriction schemes, fire instructions, etc.

For data storage, geographical databases [5] represent an interesting framework. However, they do not allow the representation of strong structure, so we will not use them for building description.

To describe both the *structure* and *semantics* of environments, we need therefore to define our own formalism.

2.2 Modeling Building Structure

A Three-Tier Model. We introduce a three-tier approach to describe building architectures:

– *first tier:* we call *lexical elements* the simple (elementary) architectural elements, such as walls, doors, flights of stairs, and so on;
– *second tier:* so-called *syntactic elements* are complex (composed) architectural elements, constituted by putting together several lexical elements. For instance, a room is defined by its walls, a stairway is defined by several flights of stairs and landings, and so on;
– *third tier:* syntactic elements are further aggregated in what we call *aggregation elements*. For instance, several offices can be gathered in a *cluster* called, say, "sales department".

The concepts (or classes) used to build descriptions take place in a concept hierarchy whose main branches correspond to the three families of objects.

Building a Description. Objects from the three tiers are bound together by two kinds of edges (cf. Fig. 3):

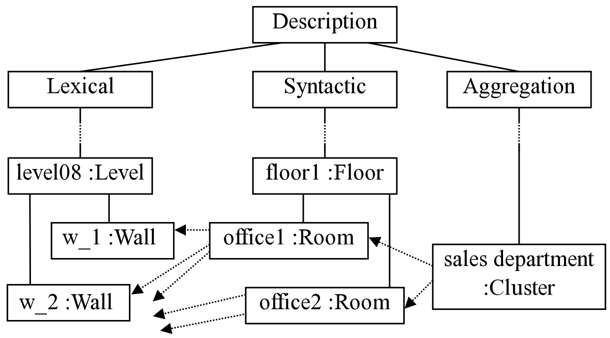

Fig. 3. Excerpt of an example description. The *sales department* is composed of two offices, in turn defined by some walls, while being included in a floor.

– *inclusion links* (solid lines) represent inclusion between elements within a given tier;
– *composition links* (dotted lines) enable objects of tier n to be composed of elements of tier $n-1$.

3 Relevant Information

3.1 The Problem

Using these architectural descriptions, the system will be able to determine where the user is, and what object or location he or she points at.

However, we still do not know the *level of detail*, i.e. the *granularity* of information needed by the user.

Indeed, too general information is useless, and too detailed information might not be understandable if the user does not know the associated context. To illustrate this, let us look at an example (see Fig. 4).

3.2 Proposed Algorithm

Suppose that the user is located in u, on the second floor of the Computer Science (CS) Laboratory. He or she points through the window at p, an office on the first floor of another building, the Library building, located next to the CS building. What information shall we return? Information attached to the room, the floor, the building, the campus...?

To solve this problem, we represent the scene as a tree. First, let us find the deepest node that is common to both the path leading from u to the root, and the path leading from p to the root. This node is labeled c on Fig. 4.

What happens if we return information located on c or above? Such information is too general, because it covers p as well as u. Thus, it will probably be useless for the user, because being in the CS building they already know that they are on the campus.

In consequence, we should return information located *below* c. Therefore, this information will be on the sub-path leading from p to c since it must describe p. But it must be within the context of the user, because they could not understand it otherwise.

Therefore, we return information located in i, i.e. the most general information on the sub-path leading from p to c. This corresponds to a *default granularity level*. However, the user may wish to retrieve information located at another level. For this reason, the final user interface will offer some means of climbing up and down the tree.

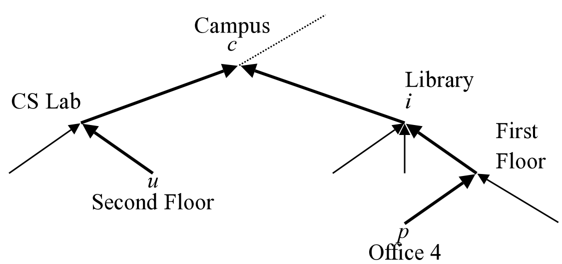

Fig. 4. A user points from one building to another one.

4 Beyond Structure: Semantics

4.1 Motivation

What are we able to do now? When the user points at an architectural element, the system is able to find it in its cartography. For instance, if the user points at a door, the system *knows* that it is a door, and that there is, say, an office behind.

However, our ultimate goal is to provide the user with *semantic* information. In the above example, the system would not only state that the user is pointing at a door leading to an office: it would also return the office owner's name, the office function, and so on.

To do this, we associate semantic information to the structure description. More generally, such information can be used:

- to add *normative* information to the structure, for example in order to tag restricted areas in a building,
- to identify objects, rooms, and zones;
- to represent connexity information;
- to add specific information to certain kinds of objects; for instance information about painters could be associated with sculptures in a museum.

4.2 Technical Solutions

To represent semantic information, the Resource Description Framework (RDF), a W3C standard quite close to the theory of conceptual graphs [6], is rapidly becoming commonplace. It allows us to express relationships between objects of interest. These objects are instances of classes that are defined in a class hierarchy expressed in OWL (Web Ontology Language, [7]).

Up to now, we have defined a basic ontology for structure description, that defines :

- object classes, divided in three categories: locations, people and places;
- properties used to specify relations between class instances;

Actually, locations appear both in structure and semantic description. Indeed, they allow semantics descriptions to be *anchored in* the underlying structure. In practice, this is achieved through the use of a common identifier.

5 Future Work

5.1 Tracking User Position

The whole system depends on its ability to track the position of the user. *Where reception is possible*, GPS will provide a reliable and *absolute* positioning solution. However, in places where GPS reception is unavailable or inaccurate (e.g. inside buildings and in dense urban areas subjected to *canyon effect* [8]), the device will compute its position by means of *dead reckoning* using an embedded inertial unit. This way, new positions are determined in a *relative* fashion, by estimating how far the user has moved since the last GPS-acquired position.

Other systems like the Cyberguide [9] use beacons (infrared beacons, bluetooth devices, etc.) to take over GPS inside buildings, and then compute their position through triangulation. We are not considering this option, because we want our system to work anywhere, not only in specially-equipped places.

Unfortunately, dead reckoning is very much error-prone [10] : computed positions are likely to deviate from real positions because of cumulative errors. To overcome this shortcoming, the *map-matching* method [11] suggests to restrict the movements of people along well-defined paths on a map. Hence, it is possible to reduce deviation errors by computing the most probable position of the user *along a path* and not in every possible direction.

This works well for car drivers that are forced to follow roads, but in case of pedestrians movements are less predictable (especially outside buildings). However, we think that knowledge about the structure and semantics of the environment is likely to help determine users' positions. Therefore, a method of *semantic map-matching* will be further investigated in the future.

5.2 Acquisition of Descriptions

Until now, we have assumed that we had environment descriptions at our disposal, but they actually need to be constructed. We have listed three ways of obtaining them:

- to write them from scratch, for example using a graphical editor;
- to perform a conversion from existing description languages, either automatically or semi-automatically;
- to scan environments with the device, and label objects on the fly.

The last method would allow blind people to use their locomotion assistance devices even in places where there is no available description. Visually impaired people would tag the environment when first visiting a new place accompanied by some sighted person (as they usually do). From these data, the system would compute a partial model (cf. *map learning*, [10]) that could be re-used and refined next time. Partial models could even be published for others to use them and improve them in turn.

5.3 Navigation Aid

Many navigation aids have already been developed for sighted people: for example, car navigation systems [12] are rapidly becoming commonplace. Thus, it is likely that future locomotion assistance devices for the blind will implement navigation aid functions as well.

If a device knows the structural description of its environment, it can assist users in planning their path to some target. It can list the possible ways, and even find out those with interesting properties, for example: the shortest, the least likely to be crowded, the more secure (from a blind person's point of view), etc.

Thus, a mere topological path planning algorithm will not be sufficient, because the algorithm will have to take into account not just structural information, but also semantic annotations that can influence the choice of a "best" path.

Usually, navigation aid interfaces require graphical output modalities, but since blind people cannot use visual interfaces, the details of user interaction with the device need thinking out, as the interface will determine how the system will be accepted among users.

Indeed, the user interface must remain very simple, but take maximum advantage of available modalities, thus permitting access to the full range of possibilities of the device.

The user interface is likely to use multimodal techniques, combining speech synthesis, musical notes, Braille displays and tactile vibrations.

Conclusion

Our project builds upon electronic travel aids that have been developed recently. Already useful, these devices are nonetheless capable of only indicating distances to obstacles, and not of giving higher-level information. To achieve our goal of being able to name objects and provide additional information, we have proposed solutions for two critical issues in this paper:

- defining a formalism to model the structure of visited buildings;
- designing a model to represent semantic information associated with the structure. When the system has the ability to constantly know its geographical coordinates, it will be able to determine candidate interesting information.

User position tracking, description acquisition and semantic data presentation will be among the topics of our future research work.

References

1. Farcy, R., Bellik, Y.: Comparison of various interface modalities for a locomotion assistance device. In Miesenberger, K., Klaus, J., Zagler, W., eds.: Computers Helping People with Special Needs, Springer (2002)
2. Farcy, R., Bellik, Y.: Locomotion assistance for the blind. In Keates, S., Langdom, P., Clarkson, P., Robinson, P., eds.: Universal Access and Assistive Technology, Springer (2002) 277–284
3. Web 3D Consortium: The Virtual Reality Modeling Language International Standard ISO/IEC 14772-1:1997. Technical report, (1997) Available at http://www.web3d.org/technicalinfo/specifications/vrml97/.
4. Web 3D Consortium: Extensible 3D (X3D) International Standard ISO/IEC FCD 19775:200x Final Committee Draft. Technical report, (2003) Available at http://www.web3d.org/fs_specifications.htm.
5. Hadzilacos, T., Tryfona, N.: An extended entity-relationship model for geographic applications. SIGMOD Record **26** (1997) 24–29
6. Sowa, J.F.: Conceptual graphs for a database interface. IBM Journal of Research and Development **20** (1976) 336–357
7. McGuinness, D.L., van Harmelen, F.: OWL web ontology language overview. Technical report, W3C World Wide Web Consortium (2003)
8. Chao, J., Chen, Y., Wu Chen, X.D., Li, Z., Wong, N., Yu, M.: An Experimental Investigation into the Performance of GPS-based Vehicle Positioning in Very Dense Urban Areas. Journal of Geospatial Engineering **3** (2001) 59–66
9. Long, S., Kooper, R., Abowd, G.D., Atkeson, C.G.: Rapid Prototyping of Mobile Context-Aware Applications: The Cyberguide Case Study. In: Mobile Computing and Networking. (1996) 97–107
10. Fusiello, A., Caprile, B.: Synthesis of Indoor Maps in Presence of Uncertainty. Robotics and Autonomous Systems **22** (1997) 103–114
11. Kitazawa, K., Konishi, Y., Shibasaki, R.: A Method of Map Matching For Personal Positioning Systems. In: Proceedings of the Asian Conference on Remote Sensing ACRS 2000, (2000)
12. Philip, O.: Car navigational systems: Futuristic systems. In: 2nd Annual CM316 Conference on Multimedia Systems, Southampton University, UK, (2002)