*Poster Proceedings of ACDM 2002*
*University of Exeter, UK*

# INSTRUCTION-BASED LEARNING FOR MOBILE ROBOTS

Guido BUGMANN, Stanislao LAURIA, Theocharis KYRIACOU, Johan BOS*, Ewan KLEIN*

Centre for Neural and Adaptive Systems, School of Computing, University of Plymouth
Drake Circus, Plymouth PL4 8AA, United Kingdom.
*Institute for Communicating and Collaborative Systems, Division of Informatics, University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, Scotland, United Kingdom.
http://www.tech.plym.ac.uk/soc/staff/guidbugm/ibl/index.html

## ABSTRACT

Programming by natural language will be a key method enabling computer language-naïve users to instruct domestic robots. Its main advantages over other learning methods are speed of acquisition and ability to build high level symbolic rules into the robot. This paper describes the steps towards the design of a practical system that uses unconstrained speech to teach a vision-based robot how to navigate in a miniature town. The robot has a set pre-programmed primitive navigation procedures that the user can refer to when explaining how to traverse an unknown route. A particularity of this project is that the primitive procedures are determined by analysing a corpus of route instructions. It is found that primitives natural to the user, such as “turn left after the church” are very complex procedures for the robot, involving visual scene analysis and local route planning. Thus, to enable natural user-robot interaction, a high-level of intelligence needs to be built into “primitive” robot procedures. Another finding is that the set of primitive procedures is likely not to be closed. Thus, on time to time, a user is likely to refer to a procedure that is not pre-programmed in the robot. How best to handle this is currently investigated. In general, the use of Instruction-Based Learning (IBL) imposes a number of constraints on the design of robotics systems and knowledge representation, and has the potential for improving their robustness.

## INTRODUCTION

Future domestic robots will need to adapt to the special needs of their users and to their environment. This includes learning the name and location of objects, and learning specialised tasks, for instance for the care of elderly.
Most potential users are computer-illiterate and will not be able to use standard programming methods to modify and extend the program of their robot. For that reason, we are investigating Instruction-Based Learning (IBL), a methodology for enabling naive users to program their robots by speaking to them using natural language. This paper describes the steps towards the design of a practical system that uses unconstrained speech to teach a vision-based robot how to navigate in a miniature town. In principle, the developed methodology should enable to transfer human knowledge into any computer system.

## THE IBL CONCEPT

An IBL system (Fig. 1) operates according to following scenario. First, a user asks the robot to perform a given task. His/her spoken words undergo a series of conversions: speech recognition, semantic analysis, and finally the extraction of a functional representation of the requested task (more details in Lauria et al., 2001).
If the robot knows how to perform the task, the corresponding program will be retrieved from a library of procedures and executed. If the task is unknown, the robot will ask the user to explain how to perform the task. The user then explains the task step by step. At the end of this learning process, the robot will have built a new procedure that becomes part of its knowledge base.

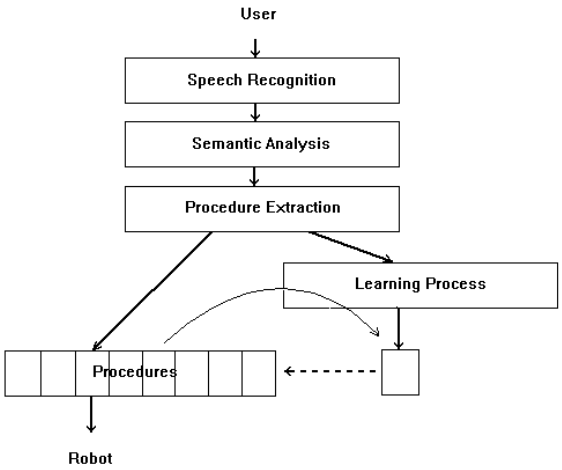



**Figure 1**. Instruction-Based Learning concept

New procedures can only be built using other procedures previously known to the robot. So there is a need to build into the system an initial set of "primitive" procedures.
To determine the minimal necessary set of primitive procedures, the method followed here is to invite a number of subjects to speak freely to the robot, and then to analyse what functions they tend to refer to. This was done using a miniature town in which a miniature robot (Figure 2) is explained how to navigate between different pairs of landmarks.
The recorded corpus of 144 instructions revealed that subjects tend to refer to around 15 different actions in this navigation context. One is for instance "take the $n^{th}$ turn left/right" or "move forward until a given landmark" (table 1). These do not necessarily

***Table 1:*** *Functional primitives extracted from the corpus.*

| | Primitive | Description |
|---|---|---|
| 1 | **go**(description_1, landmark_1, preposition_1, description_2, landmark_2) | Instructs the robot to follow a known route (with known starting point and destination). |
| 2 | **location_is**(description_1, landmark_1, direction_1, preposition_1, description_2, landmark_2, description_3, landmark_3, ordinal_1) | Specifies a location. |
| 3 | **destination_is**(description_1, landmark_1, direction_1, , preposition_1, description_2, landmark_2, description_3, landmark_3, ordinal_1) | Indicates the destination landmark. |
| 4 | **go_until**(description_1, landmark_1, preposition_1, description_2, landmark_2) | Follow known route to a landmark until a specified location in the route. |
| 5 | **exit_roundabout**(ordinal_1, preposition_1, description_1, landmark_1) | Take a specified exit from a roundabout. |
| 6 | **turn**(ordinal_1, direction_1, preposition_1, description_1, landmark_1) | Take a specified turn off a road. |
| 7 | **follow_road_until**(preposition_1, description_1, landmark_1) | Move forward until a certain location. |
| 8 | **rotate**(direction_1, extend_1, around_1) | Rotate to a certain extend. |
| 9 | **exit_from**(description_1, landmark_1) | Exit from a place, usually used for the car park. |
| 10 | **cross_to**(description_1, landmark_1) | Instructs the robot to cross the road to a landmark. |
| 11 | **enter_roundabout**(direction_1) | Enter the roundabout in a specific direction. |
| 12 | **park**(preposition_1, description_1, landmark_1) | Park on, or close, to a certain landmark. |
| 13 | **take_road**(preposition_1, description_1, landmark_1) | Take a road in view. |
| 14 | **goto_side**(preposition_1, description_1, landmark_1) | Go round a landmark to one of its sides. |
| 15 | **fork**(direction_1) | Follow a one of the two branches of a fork (Y split). |

correspond to simple procedures to program, but they are needed to allow the user to speak without special training.

**Figure 2:** The robot is built on a 8cm x 8cm base. Its wireless color video camera sends images to a PC for processing, and the PC send motion commands by radio to the robot.

Analysing a corpus even of large size does not guarantee that the established list of primitive functions is complete (Lauria et al., 2001). Hence, a functional IBL system should also include so called "clarification sub-dialogues", that allow, for instance, a reformulation of the instruction in terms of known primitives. A similar observation concerns the lexicon or list of word and sentence structures that the robot needs to understand. About 340 different words are found in the corpus of 6600 words for this task domain, but it is highly likely that the system will be faced with unknown words in its instructions. Here lexical acquisition mechanisms need to be designed.

## WORK IN PROGRESS

### Natural Language Processing

On the natural language side, the current work is mainly focused on producing a proper translation chain from the user to the robot (Figure 1) using as speech input the recorded instructions from the corpus. This has required the design of several new software modules. The issues of clarification sub-dialogues and lexical acquisition will be handled in a second phase.

### Vision-based Robot Navigation

On the robotic side, work is progressing on the design of the navigation functions corresponding to the 15 primitives in table 1.
The robot uses vision for navigation. This is required by the content of the instructions that frequently refer to

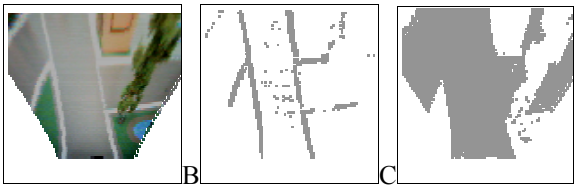
visual landmarks. A new template-based method has been proposed that enables the recognition of intersections in a "short-lived" internal map (Kyriacou, Bugmann, Lauria, submitted).

A short-lived map is a map of the immediate vicinity of the robot that is updated as the robot moves in its environment. The map records previously seen visual information which go out of view as the robot moves. The robot's position is always centred on the map, facing North (see e.g. fig. 4). As the robot moves, the map is translated and rotated to maintain this frame of reference. In the process, elements of the map that reach its edge will disappear. Thus the term "short-lived".

The purpose of constructing such a map is to compensate for the dead angles of the robot, to keep track of landmark locations and road layout features such as intersections and for resolving spatial relations between a landmark and the road (e.g. to define a road area "after" a building).

Two versions of the short-lived map are currently used, the first represents the position of the road surface and the second represents the position of road edges. These maps are constructed using road surface and road edge information filtered out from the top view of the scene. This view is produced by applying a perspective transform to the camera image. Figures 3B show the top view of 3A, and 3C and 3D show the extracted road edge image and road surface image respectively.

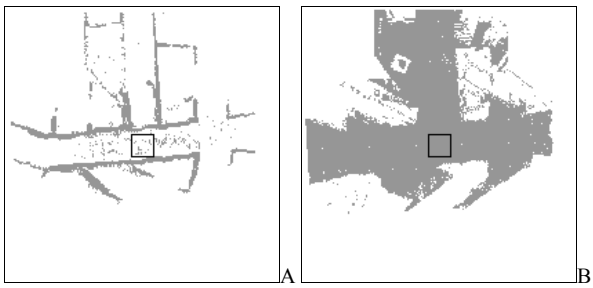

A



B C D

**Figure 3.** B: Top view obtained from a perspective transform of A. C: Extracted road edges. D: Extracted road surface areas.

These images are added to the existing short-lived map that progressively becomes a more complete map of the local area around the robot. See e.g. figure 4.



A B

**Figure 4.** Short lived maps of the road edges (A) and the road surface (B). The square represents the position of the robot.

The road-edge map 4A is used for aligning new images such as fig 3C. This then enables to paste the new road-image in the correct position, for instance 3D, onto 4B. The figure 4B is searched for intersections using templates described in (Kyriacou, Bugmann & Lauria, submitted). The search process is guided by the content of natural language instructions. For instance, if the robot is to "take the first right", only templates corresponding to right intersections would be used. This simplified greatly the analysis of the visual scene and leads to a robust detection even with a relatively noisy image and partially occluded regions.

## CONCLUDING COMMENTS

Natural language instructions are very under-specified. For example in: "take the second turn right", the absolute locations of the intersections, their orientations or shapes are not given. These pieces of information must be retrieved in-situ by the robot to successfully complete the task. Thus, robust artificial vision is required by an IBL system.

The work done so far indicates that IBL is a viable method of programming robots, provided that robust primitives are built into the robot, to compensate for the under-specification of natural language instructions. Repair mechanisms are also needed to deal with a number of possible errors in the translation process from the utterance of the user to the specification of a robot primitive.

In contrast to other more autonomous learning methods, IBL ensures that the user is in control of what the robot learns and how it is going to execute a new task. IBL also simplifies the problem of designing intelligent robots, as the robot can tap into his user's brain when faced with complex problems.

**Acknowledgement:** This work is supported by EPSRC grants GR/M90023 and GR/M90160.

**Reference:**

Stanislao Lauria, Guido Bugmann , Theocharis Kyriacou, Johan Bos, Ewan Klein (2001) "Personal Robot Training via Natural-Language Instructions" IEEE Intelligent Systems, 16:3, pp. 38-45.