# Integration of Speech and Vision in a small mobile robot

Dominique ESTIVAL
Department of Linguistics and Applied Linguistics
University of Melbourne
Parkville VIC 3052, Australia
D.Estival@linguistics.unimelb.edu.au

## Abstract

This paper reports on the integration of a speech recognition component into a small robot, J. Edgar, which was developed in the AI Vision Lab at the University of Melbourne. While the use of voice commands was fairly easy to implement, the interaction of the voice commands with the existing navigation system of the robot turned out to pose a number of problems.

## Introduction

J. Edgar is a small autonomous mobile robot developed in the AI Vision Lab at the University of Melbourne, which is primarily used as a platform for research in vision and navigation. The project which we describe in this paper consists in the addition of some language capabilities to the existing system, in particular the recognition of voice commands and the integration of the speech recognition component with the navigation system.
While the vision and navigation work is mainly carried out by Ph.D. students in Computer Science, adding speech and language capabilities to the J.Edgar robot has been a collaborative project between the two Departments of Computer Science and of Linguistics and Applied Linguistics, and the work has been performed by several linguistics students hosted by the Computer Science department and working in tandem with CS students.
The paper is organized as follows: section 1describes the capabilities and restrictions of the robot J. Edgar, section 2 is an overview of the speech recognition and language understanding system we have added to the robot, section 3 goes through the different stages of the integration and section 4 briefly describes the generation component.

## 1 Description of J. Edgar

### 1.1 Moving around

The J.Edgar robot is rather limited in the types of movement it can perform. Its twin wheels allow it to move forward in a straight line, and to turn around, either right or left, up to 360°, but it cannot move backwards. Its speed can be varied, but is usually kept very low to avoid accidents.

### 1.2 Vision and Navigation

#### *1.2.1 Vision*

The vision system of J.Edgar consists in a one-eye monochrome camera mounted on a small frame with two independent drive wheels and a pan head. Its spatial representation is two-dimensional and relies on edge detection. More specifically, it interprets discontinuities as boundaries between surfaces, which constitute obstacles.

#### *1.2.2 Navigation*

The J.Edgar robot uses MYNORCA, a vision-based navigation system developed in the University Melbourne AI Vision Lab (Howard and Kitchen, 1997a, 1997b). This navigation system is divided into two levels:

- The local navigation system uses visual clues for obstacle detection and to form *local maps*. It allows the robot to navigate in its immediate environment and to reach local goals without colliding with obstacles. Most solid objects are recognized as obstacles, but obstacles can also be recognized as walls, corners or doorways (see section 3.3).
- The global navigation system detects significant *landmarks* and uses a global map to determine its location in the environment. It allows the robot to reach distant goals specified according to the global map. The detection of landmarks also requires a level of object recognition

and the interpretation of visual cues needed at the local level.

Figure 1 shows a series of snapshots for the local and global navigation systems during a given time period. Both systems are based on the production of *occupancy maps* generated by a visual mapping system based on the detection of boundaries.

This project has so far been able to interface only with the vision-based navigation system at the local level, but we hope we will soon be able to extend it to the object recognition aspect and interact with the global level.

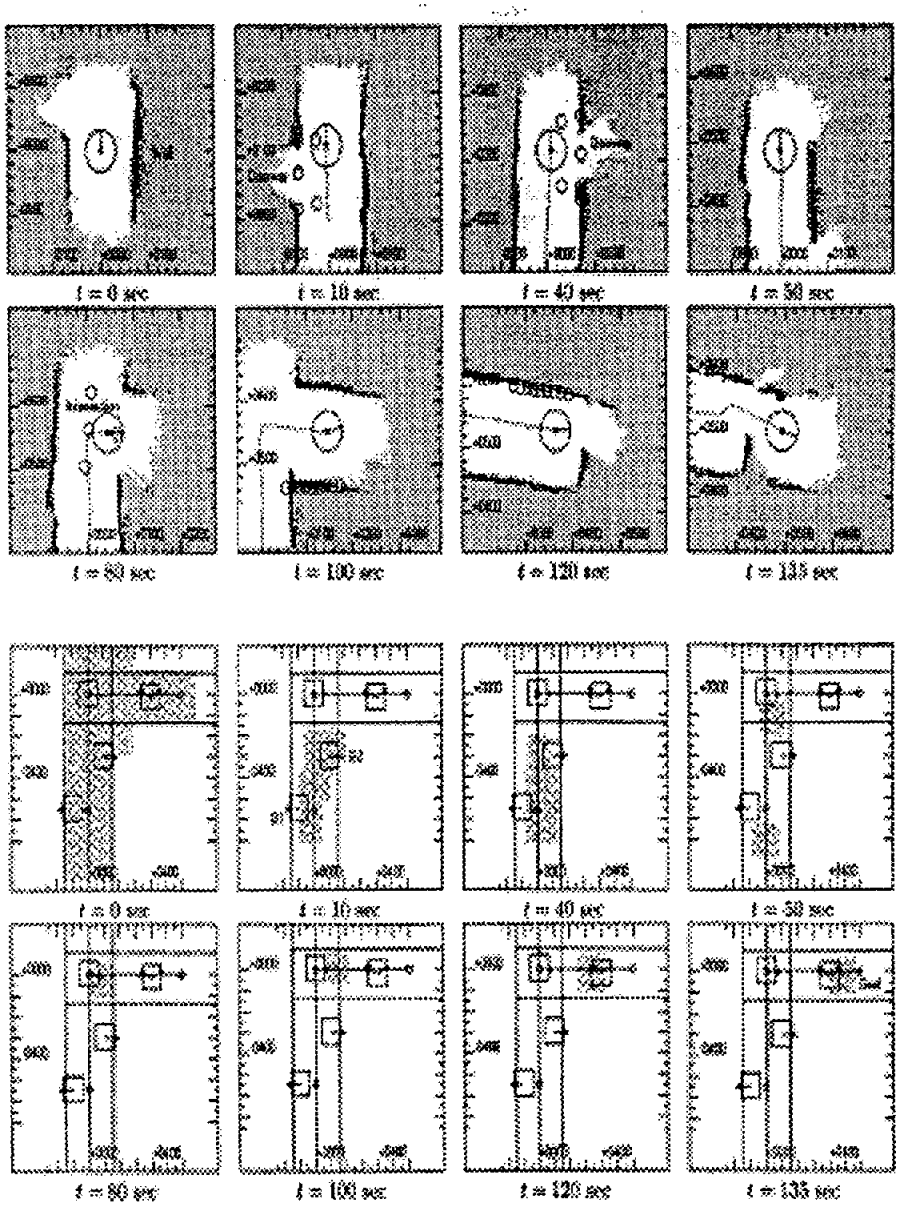

Figure 1: The upper set of images is a series of snapshots of the local occupancy map indicating the robot's current location and path. The lower set of images is a series of snapshots showing the evolution of the estimated global position (global pose estimate). The cross-hatched region indicates possible robot locations in the global model. [from Howard & Kitchen 1997a].

The vision and navigation systems are installed on a base-station which communicates via a UHF data-link with the on-board computer. The on-board computer performs the low-level hardware functions and supports the obstacle detection agent (see below).

## 2 Speech and Language

The first step towards integrating some sort of Natural Language capabilities into the robot was to install a speech recognition component. The second step was to develop a grammar to analyze voice commands and to map those commands onto the actual actions which the robot can perform.

In the next stage of the project, we are now working towards the development of a dialogue system, with which J.Edgar can respond according to its internal status and make appropriate answers to the voice commands it recognizes. Until the speech synthesizer component is fully incorporated into the system, we are using canned speech for the answers.

The speech recognition system is installed on the base-station and communicates with the robot via the UHF modem.

### 2.1 Speech Recognition

The main factor taken into consideration in choosing an off-the-shelf speech recognition system was the possibility of building an application on top of it, and the IBM VoiceType system was first chosen because of the availability of development tools. Despite some initial problems, these tools have proven useful and have allowed us to develop our own grammar and interface with the robot. We have now migrated to the IBM ViaVoice Gold system, which provides better speech recognition performance and the same development tools as VoiceType. In addition, ViaVoice includes a speech synthesizer, which we are currently incorporating in our system. In the remainder of this paper, I will describe the work that has been carried out using the IBM VoiceType system and ported to the ViaVoice system.

The system is speaker-independent and so far has been trained with more than 15 people. Care has been taken not to overtrain it with any one particular person in order to maintain speaker-independence.

In general terms, the lexicon used in the system maps onto the actions which the robot can perform and the entities it can recognize. The lexicon is thus as limited as the world of the robot, but it includes as many variant lexical items as might be plausibly used (e.g. *turn, rotate, spin* etc. for TURN). These actions and entities are described in section 3.

The IBM VoiceType or ViaVoice system can be used either as a dictation system with discrete words, or in continuous speech mode. Taking advantage of the grammar development tools, we are using it in continuous mode, and the voice commands are parsed by the grammar described in section 2.2.

### 2.2. Commands Grammar

In addition to the baseline word recognition capability, the development tools in the IBM VoiceType or ViaVoice systems all the developer to write a BNF grammar for parsing input strings of recognized words. We have thus developed a grammar mapping voice commands to the actions J. Edgar is capable of performing.

#### *2.2.1. Semantics*

Each item in the lexicon is annotated with an "annodata", which can be thought of as its semantic interpretation for this domain. Recognized input strings are thus transformed into strings of "annodata", which are further parsed and sent to the communication protocol. A command such as (1) will be recognized as (2) and the string of annodata (3) will be then parsed to produce the sequence of commands (4).

(1) *J.Edgar before turning left and moving forward please turn around*

(2) *J.Edgar*:"INITIALIZE" *before*:"INIT2" *turning*:"TURN" *left*:"LEFT" *and*:"SEQUENCE" *moving* :"MOVE" *forward* :"FORWARD" *please* :"INIT1" *turn*:"TURN" *around*:"BACKWARDS"

(3) INITIALIZE INIT2 TURN LEFT SEQUENCE MOVE FORWARD INIT1 TURN BACKWARDS

(4) INITIALIZE INIT1 TURN BACKWARDS INIT2 TURN LEFT SEQUENCE MOVE FORWARD

*2.2.2. Syntactic analysis*

All commands to the robot are in the imperative. However, some structures for complex commands have been implemented. These concern mainly the coordination of commands and temporal sequence. As shown in the example above, conjunctions such as *before* and *after* will trigger the recognition of a temporal sequence and the possible reordering of the commands. Other recognized constructions include:

(5) IF .... COMMAND
*If there is a wall to your left, turn right and move forward.*

(6) WHEN .... COMMAND
*When you get to a all, go along it.*

## 3. Integration

### 3.1. Movements only

In the first stage of this project, the natural language system was only interfacing with the movement commands of the robot, and not with the navigation system (either local or global). That is, the robot was either performing in the voice command modality, or in the navigation modality. The main reason for this limitation was that the navigation system was still under development and not robust enough to ensure safe manoeuvering in case of voice commands leading to potentially damaging situations.

As a result, only commands relating to movements (MOVE or TURN), and their specifications (FORWARD, LEFT, RIGHT, and specific distances) were understood and there was no need for representing objects or entities.

### 3.2. Low-level vision

In the second stage of the project, we only integrated the language capabilities with the low-level vision system of the local navigation system. In practical terms this means that while the robot can both accept spoken commands and scan its environment, it can only recognize *local movement* commands and will only obey them if they do not lead to a collision.
Thus, this stage also did not require the addition of any semantic representation for objects. However, to avoid a collision with an obstacle, we need the local vision system for obstacle recognition. We use the "careForward" function, which overrides the default distance of 1 meter if there is an obstacle in the path of the robot and ensures that the robot will only move to a safe distance from it.

### 3.3. Local navigation

Further integration consists in issuing commands that involve locations and objects the robot knows about, as in (7):

(7) *Go down the corridor and go through the first doorway on the right.*

This stage involves referring to objects and entities recognized by the robot.

There are five types of *primitive objects* in the world which the robot can identify:

- WALL
  a straight line;

- DOORWAY
  a gap between two walls;

- INSIDE CORNER (*"in the corner"*)
  two lines meeting at an angle and enclosing the robot;

- OUTSIDE CORNER (*"around the corner"*)
  two lines meeting at an angle and going away from the robot;

- LUMP
  a bounded solid object.

From combining these primitive objects, the robot can also create representations for *complex objects*:

- INTERSECTION:
  two outside corners that form an opening;

- CORRIDOR:
  two parallel walls.

Both types of objects can be used as referents in commands and can be queried.

It is worth emphasising that obstacles are not recognized as a separate categorie, but are either walls, lumps, corners, or doorways which are not wide enough for the robot to pass through.

For instance, in Figure 2, the robot recognizes an opening in the wall on its right and might later recognize an outside corner to its left.

The white area corresponds to the area the robot has already recognized as being empty and the black areas to recognized walls.

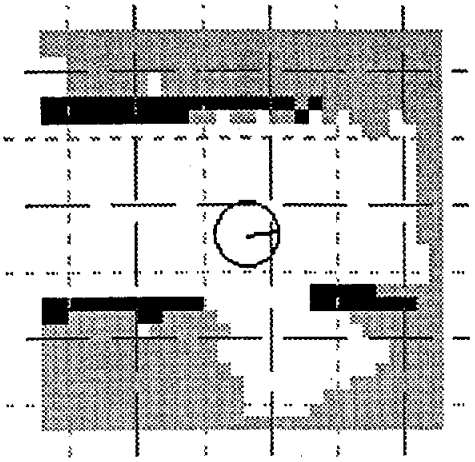

Figure 2: Obstacle detection

### 3.4. Global navigation

The next stage of the project is the integration with the whole navigation system, including the recognition of objects and locations. In this mode, the robot will not only stop when there is an obstacle, but will be able to decide whether to try to go around it. The objects to be used as referents will include locations such as *Office 214*, *Andrew's office*, or *Corridor A*, which have specific coordinates on the robot's global map. This is on-going work and we hope to have achieved this level of integration in the next few months.

## 4. Generation

In the meantime, the robot can return information about its perception of the environment, including the obstacles which were recognized, and can ask for further instructions. We have identified four situations for the generation of questions by the robot:

1. when a command is not recognized,
2. when a command is incomplete,
3. when a command cannot be completed,
4. when an object referred to in a command cannot be located.

The first and second situations only require input from the speech recognition system, including the mapping to robot commands. However, the third situation requires access to the local navigation system, or at least to obstacle detection, and the fourth situation requires access to either the local or global navigation system, depending on whether the object is a primitive object or whether it requires coordinates on the global map. In these last two situations, the generation of questions by the robot involves a mapping between the robot's internal representations of the recognized environment and the actual expressions used both in the commands and in returning answers.

## Conclusion

While this project has been a successful collaboration between vision-based navigation and natural language processing, the J.Edgar robot is still far from having achieved a convincing level of speech understanding. Some of the challenges of such a project reside in the successful communication between the speech recognition system and the robot, but the more interesting aspect is that of the correspondence between the entities used by the navigation system and the phrases recognized by the speech system.

Since the speech system is independent of the physical robot, it can be interfaced with a number of robots. One of the extensions of this project is to install a natural language interface for some of the other robots being built in the AI lab and eventually to use the same natural language interface with more than one robot at a time.

## Acknowledgments

We thank Leon Sterling and Liz Sonnenberg for the support of the Computer Science Department for this project, Andrew Howard for letting us use J.Edgar and for his help and advice throughout, Elise Dettman, Meladel Mistica and John Moore for their enthusiasm and dedication, and all the people in the AI Vision Lab for their help.

## References

Colleen Crangle and Patrick Suppes (1994). *Language and learning for robots.* CSLI lecture notes 41. Stanford: CSLI.

Andrew Howard and Les Kitchen (1997a). *Vision-Based navigation Using Natural Landmarks,* FSR'97 International Conference on Field and Service Robotics. Canberra, Australia.

Andrew Howard and Les Kitchen (1997b). *Fast Visual mapping for Mobile Robot Navigation,* ICIPS'97 IEEE International Conference on Intelligent Processing Systems, Beijing.