

Multimodal cognitive interface for robot navigation

Mohammed Elmogy · Christopher Habel ·
Jianwei Zhang

Received: 12 June 2010 / Accepted: 15 December 2010
© Marta Olivetti Belardinelli and Springer-Verlag 2010

Abstract To build effective interactions between humans and robots, they should have common ground of understanding that creates realistic expectations and forms the basis communications. An emerging approach to doing this is to create cognitive models of human reasoning and behavior selection. We have developed a robot navigation system that uses both spatial language and graphical representation to describe route-based navigation tasks for a mobile robot. Our proposed route instruction language (RIL) is intended as a semi-formal language for instructing the robot to execute a route in an indoor environment. We implemented an instruction interpreter to process the route description and generate its equivalent symbolic and topological map representations. A topological map is generated to describe relationships among features of the environment in a more abstract form without any absolute reference system to treat the ambiguity which can occur when the robot cannot recognize the current landmark. The symbolic and topological map representations are supplied to other system components as an initial path estimation to guide the robot while it plans its navigation task. We

conducted some experiments to evaluate the routes which are written by using the RIL instructions.

keywords Human–robot interaction · Multimodal cognitive interface · Route description · Symbol grounding

Introduction

Route instructions are considered as one of the more important natural language interfaces between humans and mobile robots for applying an effective human–robot interaction (HRI). They are an interesting combination of robotics, artificial intelligence, cognitive psychology, and natural language processing (MacMahon 2005; MacMahon et al. 2006). Route instructions are easily evaluated, despite the complexity of integrating modules doing linguistic modeling, abstract spatial reasoning, and moving a robot through a world. They frequently involve two kinds of expressions that connect to spatial structure (Eschenbach et al. 2000): a verb of motion (such as go, turn, enter, and throw) and a directional adverb or a directional prepositional phrase (such as into the zoo, through the park, back, and straight on).

Therefore, route instructions specify spatial information about the environment of the route and temporal information about the actions (movements, turns) to be performed (Tschander et al. 2003). Human route instructions are usually conveyed either verbally in spoken discourse or written texts, or by graphical means, i.e. by illustrating the route on a map, or drawing sketch-maps (Habel 2003). Whereas verbal route instructions focus on the actions to be performed and take the spatial environment as the frame for these actions, maps and other pictorial representations foreground the spatial environment without possessing

M. Elmogy (✉)
Information Systems Department, Faculty of Computers
and Information sciences, Mansoura University,
Mansoura, Egypt
e-mail: melmogy@mans.edu.eg

C. Habel
WSV, Department of Informatics, University of Hamburg,
Hamburg, Germany
e-mail: habel@informatik.uni-hamburg.de

J. Zhang
TAMS, Department of Informatics, University of Hamburg,
Hamburg, Germany
e-mail: zhang@informatik.uni-hamburg.de

adequate means for representing the actions. The major deficit of maps as means for route instruction is that they do not focus on the sequence of actions to be performed but on the spatial environment. A third possibility is to combine these two kinds of external representations leading to multimodal route instructions. Multimodal route instructions combine natural language route descriptions and visualizations of the route to follow, such that the strengths of both means for communication route knowledge are brought together.

Generally, verbal route instructions represent knowledge about spatial actions and spatial layouts (Tschander et al. 2003). A route instruction set is useful if it reliably guides followers to the intended destination. Verbal route instructions are explanations given by a director, intended to guide a mobile agent, the follower, toward a specific spatial destination. When following route instructions, the follower must parse and interpret the text, model the instruction's actions and descriptions, and enact the instructions in the world, by performing these actions and recognizing the descriptions. Typically, a follower cannot simply execute instructions without inference, since the necessary actions are not completely specified. Instructions often provide just a skeletal plan of action. A follower can resolve the ambiguities and omissions by using knowledge of language, an understanding of spatial actions and relations, and a model of the environment. The core measure of a set of instructions for a route is simple—did the follower end up at the intended destination?

Consequently, good route instructions should contain adequate information about the two aspects (Habel 2003). First, the aspect concerns navigation actions, in particular locomotion actions and perception actions, which are performed by the robot to reach its destination. The second is the spatial environment in which the intended locomotion of the robot will take place. The instructor's primary task is to choose a good combination of communicational means to transfer the relevant information concerning both aspects to the robot.

Related work

In the last three decades, there has been considerable research on spatial language and spatial reasoning. This motivates the research interest of using spatial language for interacting with artificial navigational agents. Many researchers (Litvintseva et al. 1997; Tschander et al. 2003; Tellex and Roy 2006; Skubic et al. 2004; MacMahon et al. 2006) have proposed frameworks using natural language commands in simulated or real-world environments to guide their artificial agents during navigation. In this

section, some implementations of natural language interfaces for mobile robots and simulated agents will be discussed.

Tschander et al. (2003) proposed the idea of a cognitive-oriented Geometric Agent (GA), which simulates instructed navigation in a virtual planar environment. This geometric agent can navigate on routes in its virtual planer environment according to natural language instructions presented in advance. GA is implemented to study the interaction between the spatial information given in route instructions and the spatial information gained from perception. In their approach, conceptual route instruction language (CRIL) is used to represent the meaning of natural language route instructions. CRIL expressions are constructed from a basic inventory of descriptive operators. CRIL and formal reasoning based on CRIL expressions can be used to test contrasting proposals for the semantics of spatial expressions regarding their consequences for the performance of an instructed navigator.

MacMahon (2005) introduced the MARCO system for understanding and executing natural language route instructions in 3D large-scale virtual indoor environments. MARCO is an agent that follows free-form, natural language route instructions by representing and executing a sequence of compound action specifications that model which actions to take under which conditions. Tellex and Roy (2006) implemented spatial routines to control the robot in a simulator. They defined a lexicon of words in terms of spatial routines and used that lexicon to build a speech-controlled robot in a simulator. Levit and Roy (2007) have developed components of an automated system that understands and follows navigational instructions. The system has prior knowledge of the geometry and landmarks of specific maps. This knowledge is exploited to infer complex paths through maps based on natural language descriptions. Their approach is based on an analysis of verbal commands in terms of elementary semantic units that are composed to generate a probability distribution over possible spatial paths in a map.

On the other hand, there are considerable research efforts in developing various command sets for mobile robots and robotic wheelchairs (Schulz et al. 2006; Tellex and Roy 2007; Pires and Nunes 2002; Simpson and Levine 1997). The mobile robot community has created systems that can understand natural language commands. Many research efforts focus on using spatial language to control the robot's position and behavior, or to enable it to answer questions about what it senses. In Lauria et al. (2001), Kyriacou et al. (2002), Bugmann et al. (2004), Instruction-Based Learning (IBL) is built to train mobile robots using natural language instruction to describe a navigate task. In this project, a robot is instructed on how to travel from one place to another in a miniature town. IBL uses

unconstrained language in a real-world robotic application that learns prior to execution. In IBL, the user's verbal instructions are converted into new internal program code that represents new procedures. Such procedures become part of a procedure pool that robots reuse to learn increasingly complex procedures. Hence, the robot should be capable of executing increasingly complex tasks.

Skubic et al. (2004) implemented robot spatial relationships combined with a multimodal robot interface that provides the context for the human-robot dialog. They showed how linguistic spatial descriptions and other spatial information can be extracted from an evidence grid map and how this information can be used in a natural human-robot dialog. Pradel and Hoppenot (2006) proposed a method for symbolic trajectory description in unknown indoor environments. The chosen form uses a panoramic description called fresco. The method uses distance measurements from a 2D laser range finder, digitizes the robot's visibility area, eliminates superfluous data, and reorients their presentation. The landmarks are then extracted and organized into the fresco, which is validated by means of neighborhood rules.

Kollar et al. (2010) presented a system that follows natural language directions by extracting a sequence of spatial description clauses from the linguistic input and then infers the most probable path through the environment given only information about the environmental geometry and detected visible objects. They used a probabilistic graphical model that factors into three key components. The first component grounds landmark phrases in the perceptual frame of the robot by exploiting co-occurrence statistics from a database of tagged images. Second, a spatial reasoning component judges how well spatial relations describe a path. Finally, verb phrases are modeled according to the amount of change in orientation in the path.

In this paper, we have developed a navigation system for a mobile robot to enable computer-language-naive users to instruct their mobile robots by using a multimodal cognitive interface. Our main goal is to let the robot execute navigational tasks reasonably with some degree of autonomy to adapt to the user's route description. This requires both the adaptivity to unknown environments and the ability to generate plans. Our approach is presented to qualify the mobile robot to walk autonomously in miniature city or indoor environments. We proposed a spatial language to describe route-based navigation tasks for a mobile robot. This language is implemented to present an intuitive interface that will enable users to easily and naturally describe a route to a mobile robot in indoor and miniature city environments. An instruction interpreter is implemented to analyze the user's route to generate its equivalent symbolic and topological map representations,

which are used as an initial path estimation for the humanoid robot. A topological map is generated to describe relationships among features of the environment in a more abstract representation without any absolute reference system. It is used to handle uncertainty conditions when the robot cannot recognize the current landmark.

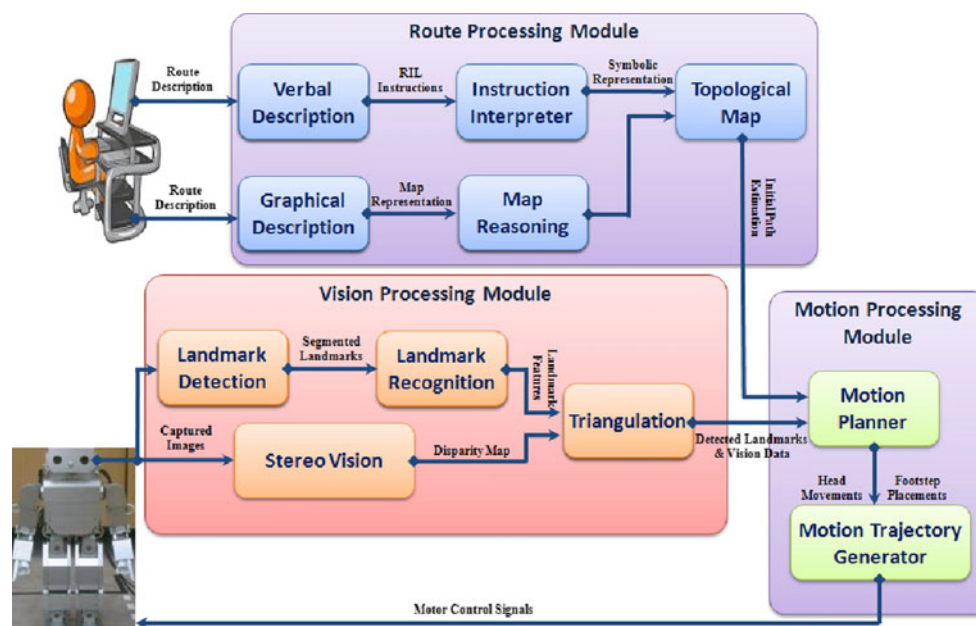
The robot navigation system

We have developed a robot navigation system to enable novice users to instruct the mobile robots in indoor environments by using verbal and graphical route instructions. The user describes the route to the robot verbally or graphically by using a Graphical User Interface (GUI), and the robot has no prior spatial knowledge of the environment's layout. The route description includes the start point, target point, actions, spatial relationships, and landmarks; whereas landmark locations are built in real-time from stereo vision. Our system combines vision-based sensing with a motion planner to allow the humanoid robot to navigate toward a desired goal position while avoiding obstacles.

Figure 1 shows the main building blocks of our system and the output from each stage. The robot navigation is based on the route described by the user to generate an initial path estimation which is supplied to the motion planner. The humanoid robot begins from the start point and moves along that path to collect information and recognize the landmarks by using its stereo vision. Based on the new findings and the processed route, the path is then re-planned to adjust the robot's position during navigation. Therefore, the robot re-plans its path and footstep placements autonomously based on the detected landmarks and the initial path estimation which is resulted from route description processing. The system is composed of three main modules: route processing, vision processing, and motion processing modules.

The route processing module receives and processes the route description to generate symbolic and topological route representations. We present a spatial language to describe route-based navigation tasks for a mobile robot. In addition, we also present a graphical representation to provide the user with a simple interface to sketch the route for the robot. The instructions of this spatial language and graphical representation are implemented to provide an intuitive interface with which the computer-language-naive user can easily and naturally describe a navigation task to a mobile robot in any indoor environment. In our system, the instructions of the processed route are analyzed to generate a symbolic representation of the navigation task via the instruction interpreter. The resulting symbolic representation is used to

Fig. 1 The architecture of the humanoid robot navigation system



generate a topological map of the route to supply the robot with the information about the route's environment and the relationships between the landmarks. It is also supplied to the robot motion planning stage as the initial path estimation of the route description to ground the action and landmark symbols with their equivalent physical procedures and objects, respectively.

The vision processing module processes the captured images from the humanoid robot's cameras to detect, recognize, and localize landmarks during navigation (Elmogy and Zhang 2008). The outputs of the last two modules are supplied to the motion processing module to calculate and execute the shortest humanoid robot footstep placements (Elmogy et al. 2009). In this paper, the route processing module is discussed in detail.

Multimodal route instructions

In our system, the route description can be presented in two different ways. It can be represented in written form by using our proposed semi-formal language or by using a simple graphical interface to indicate the path between the start and end points for the robot. These two methods of route descriptions will be discussed in the ensuing subsections.

Verbal route description

In our system, we present a spatial language—called Route Instruction Language (RIL)—to describe route-based navigation tasks for a mobile robot. This language is

implemented to present an intuitive interface that will enable novice users to easily and naturally describe a route to a mobile robot in indoor and miniature city environments. We proposed this language to avoid ambiguity and misunderstanding during route description. Therefore, a non-expert user can describe the route for the mobile robot by using simple and easy to understand instructions.

The RIL is developed to describe the route between the start and end points to a mobile robot. It is intended as a semi-formal language for instructing robots, to be used via a structured graphical user interface. RIL provides elementary instruction statements which are processed to supply the robot with a sequence of motion actions. During navigation, this sequence of actions is processed by the motion planner to determine the footstep placements which will be effected by the humanoid robot to execute the route. Each statement in the RIL constitutes a spatial instruction which relates verbally coded motion concepts to one or more landmarks by use of a suitable spatial relationship. The commands of the RIL and their syntaxes are shown in Table 1. Each instruction of the RIL specifies motion verbs, directions, destinations, and landmarks.

The position commands are used to indicate the current position of the robot during navigation. These instructions are primarily used to identify the start and end positions of the robot. They can also be used during the robot route description to describe relevant confirmations of the robot's current position with respect to one or more landmarks. These instructions are represented in RIL by using three different commands: \$START(), \$STOP(), and \$BE().

Table 1 The command set of the route instruction language (RIL)

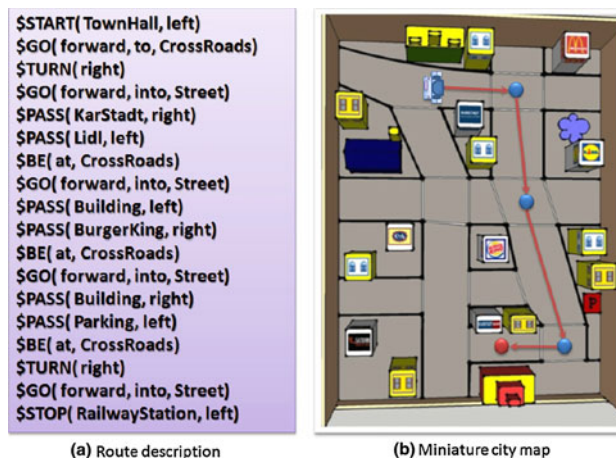
Instruction	Syntax
\$START()	\$START([Pre1 Direction], Landmark1, [Pre2], [Landmark2])
\$STOP()	\$STOP(Pre1 Direction, Landmark1, [Pre2], [Landmark2])
\$BE()	\$BE(Pre1 Direction, Landmark1, [Pre2], [Landmark2])
\$GO()	\$GO([Count], [Pre1 Direction], [Landmark1], [Pre2], [Landmark2])
\$CROSS()	\$CROSS([Pre1], Landmark1, [Pre2], [Landmark2])
\$PASS()	\$PASS([Pre1], Landmark1, [Direction], [Pre2], [Landmark2])
\$FOLLOW()	\$FOLLOW([Landmark1], Pre, Landmark2)
\$ROTATE()	\$ROTATE(Direction, Pre, Landmark)
\$TURN()	\$TURN([Count], [Pre1], Direction, [Pre2], [Landmark])

The Locomotion commands are used to instruct the robot to move in the spatial environment in a specific direction or to follow a certain path. In other words, these instructions give the robot the order to move to a particular region or to go in a particular direction with respect to one or more landmarks. In RIL, the locomotion commands are introduced by using four basic instructions: \$GO(), \$CROSS(), \$PASS(), and \$FOLLOW().

The last category is the change of orientation commands, which are used to rotate around a landmark or turn in a certain direction. These commands are used to change the direction of the robot by turning or rotating to a specific direction. \$TURN() and \$ROTATE() commands are used in RIL to represent the changes in orientation of the robot's current position during navigation.

Table 1 shows the syntaxes of the RIL instructions. The instruction's syntax consists of a command word and an arbitrary number of arguments. The command word indicates the action which will be taken by the mobile robot and is represented in the imperative form of the verb, e.g., GO, TURN, BE, etc. Each argument is a place holder for a specific group of words such as prepositions, directions, the number of turns, and landmarks. To add more flexibility to the command syntax, multiple kinds of command syntaxes have been defined. Mandatory arguments are typed without any brackets, whereas optional arguments are placed between rectangular brackets “[]”. The pipe symbol “|” indicates an OR operator.

For example, the parameters of the \$GO() command can be represented as follows. The “Count” presents the number of turns in a specific direction. For example, \$GO(2, **RIGHT**) means that the robot will take the second

**Fig. 2** “Town hall—railway station” route description by using RIL

right turn. “Pre” represents the formal counterpart to a preposition or an adverb which will be used in the spatial statement, such as to, between, and at. “Direction” specifies the direction of the turn or a landmark, i.e. left, right, forward, and backward. Finally, “Landmark” represents the name of a pre-defined landmark in the knowledge database.

To prevent ambiguity and misunderstanding, RIL uses an extrinsic reference frame to refer to all landmarks and actions in route description, which is based on the robot's viewing perspective. It also prevents transformation between different reference types while the route is processed. Figure 2 shows an example of a route description from the town hall to the railway station in our miniature city using RIL.

In this route, the robot is instructed to begin at a starting point with the town hall to its left. First, the robot has to move to the crossroads, and then it should turn right. Afterward, the robot is instructed to walk straight on, to pass the Karstadt store on its right, to pass the Lidl supermarket on its left, and to move to the next crossroads. Then it is to walk straight on, pass a building to the left, and the Burger King restaurant to the right, and go on until reaching the next crossroads. The next instructions include the robot's crossing of the street, going straight on, passing a building to the right, passing a parking place to the left, and then turning right at the next crossroads. Finally, the robot has to keep walking down the street until it is standing to the left of the railway station, which is determined as its destination.

Graphical route description

We proposed another way to describe the route for the robot by using a simple graphical representation. The main aim of this interface is to provide the user with a simple

tool that attempts to discretize the route environment as a graph-like representation of places connected by paths. It represents the environment as a graph where nodes represent places and edges represent connections between places.

On the other hand, the graphical route representation helps to facilitate reasoning of the route, mainly due to the compactness of the representation. It represents the route in a symbolic nature that allows higher-level reasoning (such as order, connectivity, and regions), which can enhance the spatial knowledge of the route environment.

In our implementation, we proposed three main actions of the robot: move, notify, and stop. The move action is used to draw the estimated path for the mobile robot. It is presented as solid straight lines. Robot turns are indicated as straight lines in different directions. The notify action presents the current robot's position with respect to one or more landmarks. It also indicates the position of landmarks during robot motion in a specific direction. It is represented as dashed line which connects landmarks with the robot's path. Finally, the stop action is used to indicate the target point of the robot with respect to one or more landmarks.

In the graphical representation, the robot's position is shown at the bottom of the map as an origin of the route. The user begins to draw the map by starting from the robot's position until he reaches the target point. At the beginning, the user localizes the current robot's position by drawing one or more landmarks connected to the robot's image in a specific direction. Then, he starts to draw the robot's path by using motion action and identifies the path by choosing some landmarks. Finally, he uses the stop action to finish the robot's route and indicates the target point.

For map reasoning, during the map drawing, the user's actions are analyzed and converted to a symbolic representation. The initial robot's position is considered as the start point of the route with respect to one or more landmarks. The stop position is considered as the robot's target. The move actions are translated into motion and turn actions.

Instruction interpreter

The instruction interpreter is used to discriminate, identify, and categorize the motion actions of the processed route description. It converts the verbal description of the route produced by using RIL into a sequence of actions that the robot must take in order to successfully follow paths anticipated by the instruction giver. It combines definitions from the lexicon according to the parse structure of the instruction, creating a symbolic script that describes the navigation process. The generated symbolic representation

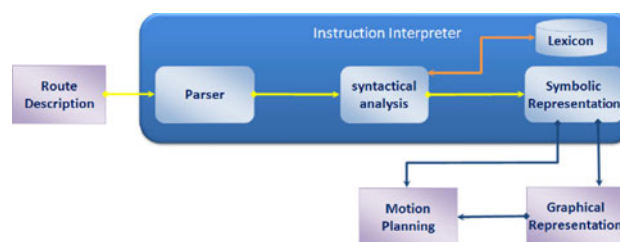


Fig. 3 The structure of the instruction interpreter

is used to create a topological map for the route environment. It is also supplied to the motion planner as an initial path estimation of the navigation task to help in generating the footstep placements for the humanoid robot.

The instruction interpreter contains a simple parser, a lexicon, a syntactic analysis, and a symbolic generator (see Fig. 3). The parser is supplied by the route description text to split it into a sequence of words. The resulting list is entered at the syntactical analysis stage to identify the structure of instructions by consulting in the lexicon to obtain the type and features of the resulting words. Finally, the result will be supplied to the symbolic representation stage to generate an equivalent symbolic script of the route.

The parser analyzes the raw route description text supplied by the user and prepares it for the syntactic analysis stage. It parses the route descriptions that are presented in both RIL and controlled natural language forms. The parser separates the text into individual instructions. Each instruction is split into a sequence of words using space and punctuation characters as delimiters. The resulting list is entered at the syntactical analysis stage to identify the structure of instructions.

The syntactical analysis stage is provided by a list of the resulting words from the parser to identify the structure of the instructions. It compares the structure of the processed instruction with a list of all kinds of instruction syntaxes that are understandable by the robot. Each word is looked up in the lexicon to obtain its type and features. The syntactical analysis is robust to unexpected input. If it encounters a constituent that cannot be modeled, it will ignore it while modeling the remainder of the instruction. Likewise, if it cannot analyze one instruction from a set of route descriptions, it will analyze the others. Additionally, the syntactical analysis is used to divide the route description into segments. Each segment begins with a motion action (i.e. \$GO() instruction) and ends before the next one—except for the starting and ending statements. All statements in the segment are processed in series and executed as a single sub-route in the robot navigation task.

The lexicon is a module of linguistic knowledge that maps words onto structures representing their meaning. It combines syntactic and semantic information about the words such that the syntactic structure can support the

derivation of the meaning of the instructions. Thus, the task to construct the meaning of a route instruction presupposes a coherent and consistent system of entries in the spatial lexicon. The lexicon is used to find the type of the parsed words and also combines these words with their definitions. The available types of words in the lexicon are command verbs, directions, prepositions, numbers of turns, and landmarks. Each verb entry in the lexicon consists of an action verb and an associated script composed from the set of its primitives and depends on the specified arguments passed to its instruction. It is defined as a script of primitive operations that run on data extracted from the analyzed instruction. In other words, each entry in the lexicon consists of a symbol and an associated script composed from the set of primitives. Depending on the processed instruction, the lexicon picks the suitable subscript for the processed word. Some spatial routines in the lexicon return a subscript as their result.

After analyzing the route instructions syntactically and connecting each resulting verb with its motion procedure, the symbolic representation of the route is generated. This symbolic script is based on CRIL representation that was developed by Tschander et al. (2003). The resulting symbolic script consists of three basic components: motion actions, spatial relationships, and landmarks. The motion actions are classified into the following four essential actions:

!BE_AT() Action: It presents the position of the robot during navigation. It identifies the start, current, and end positions of the robot during navigation.

!GO() Action: It indicates the motion actions which should be taken by the mobile robot.

!VIEW() Action: It is used to notice a landmark in a certain direction or region during navigation.

!CH_ORIENT() Action: It is used to indicate a change in the current orientation of the mobile robot motion during navigation based on a specific direction or landmark.

The spatial relationships are classified into two types. First, relations represent a location with respect to a landmark. Second, relations specify a direction with respect to one or two landmarks. The spatial or directional relationships can be divided into four classes. The first class is the goal relationships which specify the end of the path. The second one is the source relationships which give the start of the path, the third class is the course relationships which characterize the intermediate course of the path, and the final class is the shape relationships which identify the shape of the path.

Landmarks in our miniature city are classified into definite and indefinite landmarks depending on their features. Definite landmarks have unique characteristics which single them out from among the other landmarks in the

```

1-!BE_AT(x,p1);;LOC(p1,LEFT(LM1,rsys1));;LM1( TownHall,Shape)
2-!GO(x,d1);;TO(d1,FORWARD(LM2,rsys2));;LM2( CrossRoads,Color)
2-!CH_ORIENT(x,d2);;TO(d2,RIGHT(rsys3));;-----
3-!GO(x,d3);;LOC(d3,FORWARD(rsys4));;-----
3-!VIEW(x,p2);;LOC(p2,RIGHT(LM3,rsys5));;LM3( KarStadt,Symbol)
3-!VIEW(x,p3);;LOC(p3,LEFT(LM4,rsys6));;LM4( Lidl,Symbol)
3-!BE_AT(x,p4);;LOC(p4,AT(LM5,rsys7));;LM5( CrossRoads,Color)
4-!GO(x,d4);;LOC(d4,FORWARD(rsys8));;-----
4-!VIEW(x,p5);;LOC(p5,LEFT(LM6,rsys9));;LM6( Building,Texture)
4-!VIEW(x,p6);;LOC(p6,RIGHT(LM7,rsys10));;LM7( BurgerKing,Symbol)
4-!BE_AT(x,p7);;LOC(p7,AT(LM8,rsys11));;LM8( CrossRoads,Color)
5-!GO(x,d5);;LOC(d5,FORWARD(rsys12));;-----
5-!VIEW(x,p8);;LOC(p8,RIGHT(LM9,rsys13));;LM9( Building,Texture)
5-!VIEW(x,p9);;LOC(p9,LEFT(LM10,rsys14));;LM10( Parking,Color)
5-!BE_AT(x,p10);;LOC(p10,AT(LM11,rsys15));;LM11( CrossRoads,Color)
5-!CH_ORIENT(x,d6);;TO(d6,RIGHT(rsys16));;-----
6-!GO(x,d7);;LOC(d7,FORWARD(rsys17));;-----
6-!BE_AT(x,p11);;LOC(p11,LEFT(LM12,rsys18));;LM12( RailwayStation,Shape)

```

Fig. 4 The resulting symbolic representation of the “town hall—railway station” route description

miniature city, such as the Burger king restaurant, the Saturn store, and the town hall. On the other hand, indefinite landmarks have a number of properties that are not unique such as buildings, crossroads, and streets.

Figure 4 shows the resulting symbolic representation of the route described in Fig. 2. It displays the three types of information extracted from the route description: actions, spatial relationships, and landmarks. Where “rsys” refers to a spatial reference system that has to be anchored relative to the conceptual representation of the preceding segments of the route instruction. The “x”, “LM”, “w”, “p”, “t”, and “r” symbols refer to the robot, landmark, path, position, track, and region, respectively.

Topological map

After creating the symbolic representation of the route, the robot requires an adequate representation of the route environment. This representation should be abstract enough to facilitate higher-level reasoning tasks like strategic planning or situation assessment, and still be detailed enough to allow the robot to perform lower-level tasks like path planning/navigation or self-localization. The topological map is created to describe relationships among features of the environment in a more abstract representation without any absolute reference system. Our implementation of the topological map represents the robot’s workspace in a qualitative description (Elmogly et al. 2008). It presents a graph-like description of the route where nodes correspond to significant, easy-to-distinguish landmarks, and arrows correspond to actions or action sequences which will be executed by the mobile robot.

The generated topological representation of the route is mainly used to treat the ambiguity which can be occur in situations where a landmark can be described not only

by its properties, like color and shape, but also by its relations to other landmarks. By considering relations, we may be able to resolve cases where the known properties of the landmark are not sufficient to distinguish it from other similar landmarks, or the robot cannot recognize the current landmark. Therefore, the topological map is used to comprehend the route instructions by building up a mental representation of the processed route. This representation contains spatial information about the route, the sequence of actions to be performed, and the relations between landmarks. In the navigation phase, the robot has to match the internal representation (i.e. topological map) against the perceived scenes.

Symbol grounding

In the motion planner, the symbol grounding stage is implemented to incorporate the high-level cognitive processes with their corresponding sensorimotor processes. The high-level cognitive processes are presented in our system as a processed route description to perform abstract reasoning and generate plans for robot actions. They use symbols to denote both landmarks and robot actions. Otherwise, the sensorimotor processes observe the physical world and execute actions in the route environment. They operate from the cameras' data that originate from observing these landmarks. If the overall system is to perform its tasks successfully, it needs to make sure that these processes are successfully connected to indicate the same physical objects.

To solve the symbol grounding problem, a methodology is needed to resolve situations where the sensors detect several landmarks that are consistent with the symbolic description of a desired landmark. In order for an autonomous system to function robustly when faced with such ambiguous situations, it needs to reason and act in a way that allows it to distinguish between the perceived objects and determine their correct correspondents. The plan involves finding out relevant information about the landmarks until the correct landmark is identified. We used the perceptual anchoring via conceptual spaces to connect the symbolic cognitive system (Σ) with its corresponding sensorimotor perceptual system (Π). Figure 5 shows the anchoring process of the "BurgerKing" landmark that connects its symbolic representation with its perceptual region in the captured image. The conceptual space is a metric space whose dimensions, called qualities (Gärdenfors 2000), are related with the quantities processed by the robot sensors. Points in a conceptual space, called knoxels (Chella et al. 2004), represent the epistemologically primitive elements at the considered level of analysis. For logo landmarks (such as BurgerKing), the conceptual space represents three different

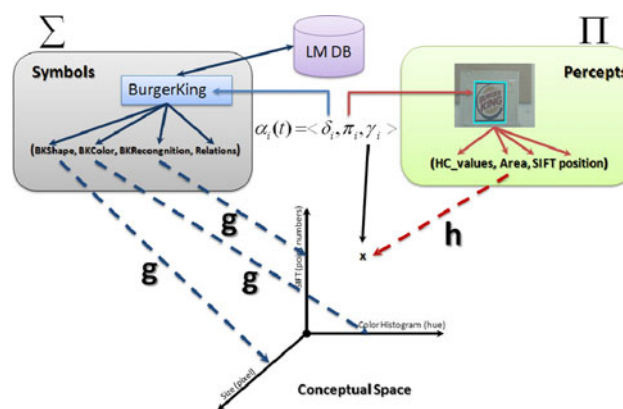


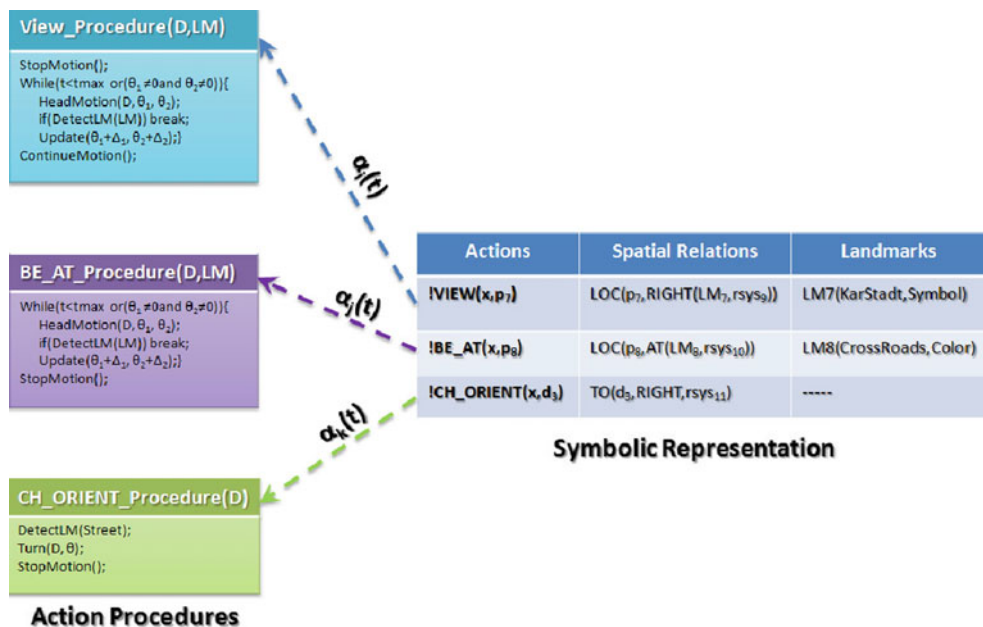
Fig. 5 The anchoring process for a landmark between its symbolic and perceptual data

quantities: the correlation values to the hue component of the stored color histograms, landmark shape and size range, and the number of the matched SIFT points.

On the one hand, the symbolic system manipulates individual symbols for each landmark to denote its physical object. The predicate grounding function (g) associates each individual symbol with a set of symbolic predicates that assert properties of the corresponding landmark. It associates unary predicates in Σ to areas in the conceptual space. In other words, the g function gives semantics to symbolic predicates in terms of observable quantities in the conceptual space. On the other hand, the perceptual system generates percepts from the observation of physical landmarks that are represented as regions in the captured images. The sensor model function (h) associates each percept with its observed values of a set of measurable attributes. It transforms a measurement vector from the sensor system into a set of knoxels in the conceptual space.

Therefore, the correspondence between symbols and percepts is reified in a data structure called anchor ($\alpha(t)$) that contains pointers to the corresponding symbols (σ_i) and percepts (π_i). In addition to these pointers, it has a pointer to an estimate of the current values of some attributes of the landmark which it refers to. This pointer is called the signature and denoted by γ_i which indicates its corresponding knoxels in the conceptual space. An anchor can be considered as a model of a physical object that reflects the persistence of the object, and which can be shared across different subsystems of the agent. Once an anchor has been created, it should be continuously updated to account for changes in the landmark's attributes and handle the connections of this landmark with its neighbor landmarks in the route description. This connection is made depending on the relationships that are retrieved from the processed route to handle the uncertainty during robot navigation.

Fig. 6 The symbolic and dynamic procedures of some robot locomotion actions



We extended the anchoring process to handle not only the landmarks but also robot actions. The anchoring process is used to connect the symbolic representation of the robot locomotion actions to their corresponding dynamic procedures which are controlled by both path and footstep planners. Figure 6 shows symbolic representations of some robot actions and their corresponding dynamic procedures.

Experimental results

To evaluate the routes that are written by using the RIL instructions, we conducted two different experiments. The first experiment is carried out to test the suitability of the RIL for the novice users. The second experiment is conducted to analyze the resulting RIL routes and compare them with their equivalent verbal routes. It also tests the usability of both RIL instructions and the graphical user interface of our system.

RIL for novice users

We carried out an experiment to test the suitability of the proposed route instruction language as an interaction tool between a computer-language-naive user and a mobile robot. Eighteen participants took part in the experiment (mean age = 29.5, SD = 3.35). None of the participants had any background knowledge on route instructions and robotics.

The experiment was conducted in single sessions. The participants were first supplied with instructions about the

purpose of the experiment, a general idea about RIL, and the estimated time of the experiment. Then, we gave them a description of the RIL syntax, a map of the miniature city, and an example of a suitable route description. Figure 7 shows the example route which represents the route from the railway station to the town hall in the miniature city. Afterward, we asked them to describe a route between the railway station and the McDonald’s restaurant as depicted in the last figure. Finally, we gave them a questionnaire to know their impression of the usage of the RIL. There were no time limitations in the experiment. The entire session took approximately 15–20 min.

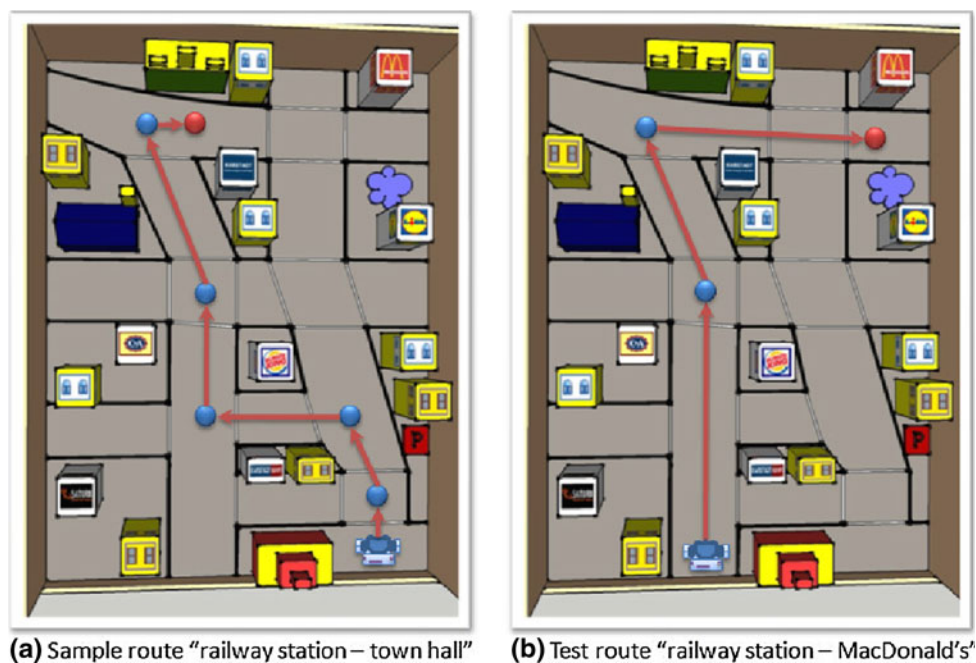
After analyzing the tested route and the answered questions for each participant, we found the following:

- 89% of the participants described the route correctly, but the rest were confused about how to use some commands and parameters.
- 83% of the participants stated that the RIL is simple and easy to learn, but the rest of them preferred to use a controlled natural language without any specific syntax for the instructions.
- 78% of the participants agreed that it is better to provide the commands of RIL with many optional parameters than to restrict them to a single syntax.

Route description analysis

We conducted a second experiment to analyze the route descriptions written by RIL using a GUI. This analysis had two main purposes. The first aim is to dissect the resulting

Fig. 7 The sample and tested routes of the first experiment



verbal and RIL routes produced by the participants. The second purpose is to analyze the RIL instructions to know the common instructions used by the users.

Fifteen participants (7 male, 8 female) took part in this study, ranging in age from 23 to 35 (mean age = 31.07, SD = 3.49). The participants had no previous experience with natural language processing and robotics. They were tested individually. Each participant was asked to describe 5 different routes in the miniature city. Figure 8 illustrates the routes used in the study.

At the beginning, the participants were asked to describe these routes verbally. After finishing the verbal description of the five routes, we explained the command syntax of the RIL to the participants and how to instruct the robot to travel from one place to another in the miniature city. Then, they were asked to describe these routes again by using the RIL via a GUI. After finishing all route descriptions, the users were asked to answer a simple questionnaire to get their impression of describing the routes by using the GUI.

We analyzed the resulting 75 verbal route descriptions (5 routes/user). We found that 98% of the resulting descriptions can be presented by using RIL. For the RIL routes, we had 830 instructions resulting from the participants' routes. Table 2 shows the average and standard deviation of the time, number of instructions per route, and correct percentage of instructions for the resulting route descriptions. Only 47 complete routes succeeded in reaching the goal and did not have any wrong instructions, whereas the other 28 routes contained one or more wrong instructions which could not be analyzed completely to produce the equivalent symbolic and topological map

representations of the route. The wrong routes resulted from misunderstanding of one or more parameters in the RIL instructions. Seventeen wrong routes, which contained one or more wrong instructions, were executed by ignoring the wrong parameters and using some default parameters. On the other hand, the other 11 routes were not executed at all because they had two or more consequence wrong instructions.

Figure 9 shows the frequency of the three instruction categories of RIL in the resulting route descriptions. It is obvious that the locomotion instructions are the most used RIL category. On the other hand, Fig. 10 illustrates the frequency of the RIL instructions in the resulting participants' routes. It is observed that the "Go" instruction is the most used command and the "FOLLOW" and "ROTATE" instructions are the less used commands in the resulting route descriptions. Consequently, the "GO" instruction clearly dominates among all analyzed instructions and the "forward" type that includes instructions such as "GO ahead", "GO down", "GO along" etc., is the most frequent type among other "GO" instructions as shown in Fig. 11.

After analyzing the answered questions of the given questionnaire, we found the following:

- 88% (SD = 0.13) of the participants stated that the GUI is simple and easy to use.
- 93% (SD = 0.11) of the participants declared that the RIL is simple and easy to learn.
- 90% (SD = 0.21) of the participants did not face any ambiguity or misunderstanding during route description.
- 78% (SD = 0.39) of the participants agreed that it is better to provide the commands of RIL with many

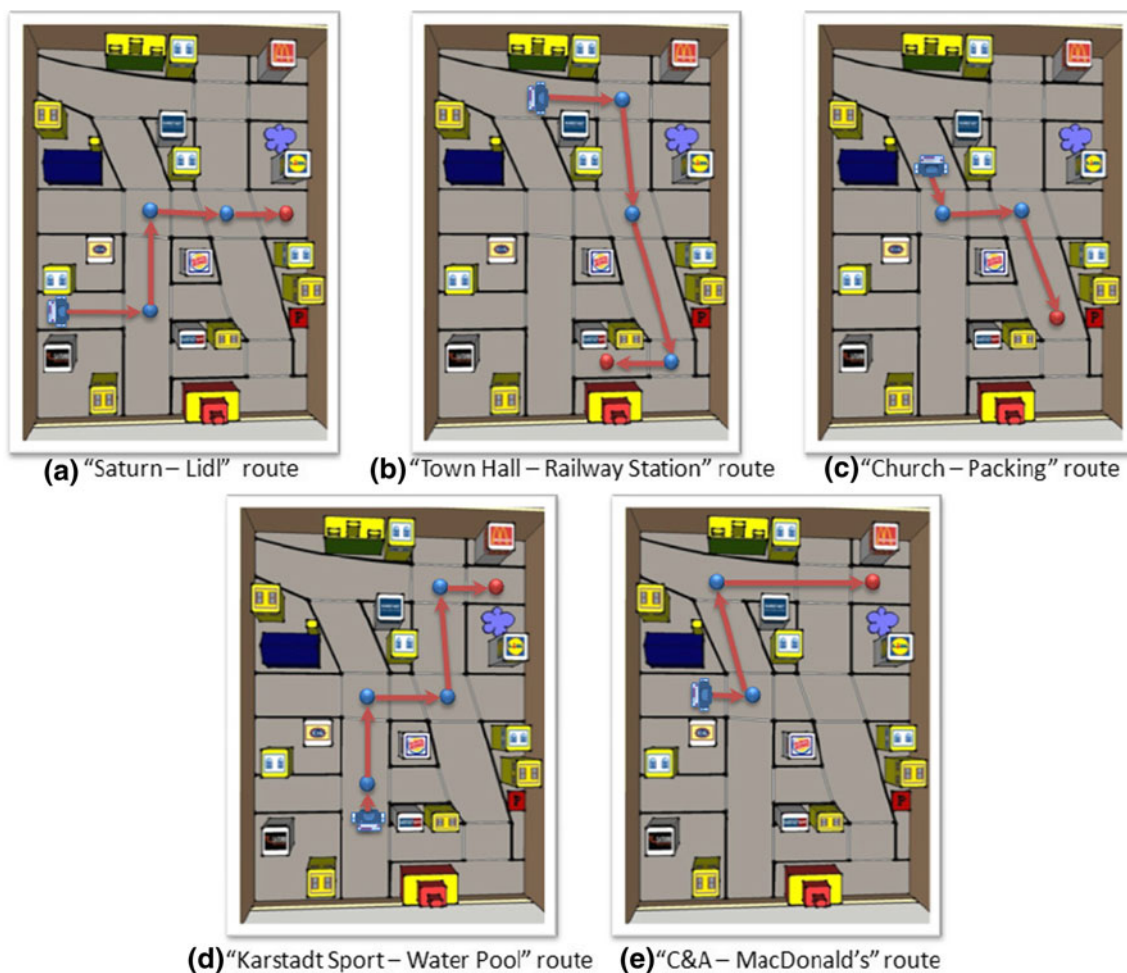


Fig. 8 The five routes used in the experiment

Table 2 Statistical analysis of the resulting route descriptions

Route	Time (SD)	Instructions (SD)	% Correct (SD)
1	7.53 (2.95)	11.07 (2.25)	92% (0.13)
2	6.73 (2.76)	11.93 (1.79)	92% (0.14)
3	4.00 (1.13)	08.87 (1.77)	95% (0.09)
4	5.40 (1.24)	13.20 (2.21)	96% (0.07)
5	4.07 (1.10)	10.27 (1.87)	94% (0.10)

optional parameters than to restrict them to a single syntax.

At the end, the results of the experiments confirmed that RIL is simple to learn and it is well suited to describe the route in indoor environments. The GUI facilitates the route description and lets the novice user describe the routes easily without ambiguity and misunderstanding. On the other hand, we have found that most of the commands that participants chose can be classified or decomposed into these categories, and by considering only such commands, we can replicate the paths with reasonable accuracy.

Conclusion

In this paper, we proposed a multimodal interface that can be used easily by inexperienced users to describe navigation tasks for mobile robots. The routes can be described to the robot verbally or graphically. Our proposed route instruction language is intended as a semi-formal language for instructing the robot to execute a route in indoor environments. We conducted some experiments to evaluate the routes which are written by using the RIL instructions. The results of the experiments confirmed that RIL is simple to learn and it is well suited to describe the route in indoor environments. The GUI facilitates the route description and lets the novice user describe the routes easily without ambiguities and misunderstandings. On the other hand, we have found that most of the commands that participants choose can be classified or decomposed into the RIL categories, and by considering only such commands, we can replicate the paths with reasonable accuracy.

For route analysis, we implemented an instruction interpreter to process the route description and generate its

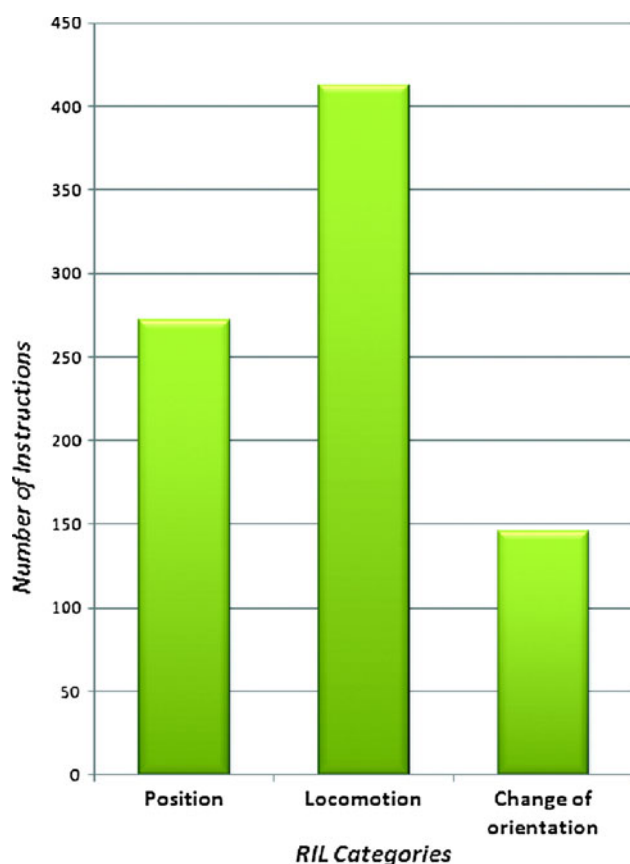


Fig. 9 Occurrence statistics of RIL categories in the resulting routes

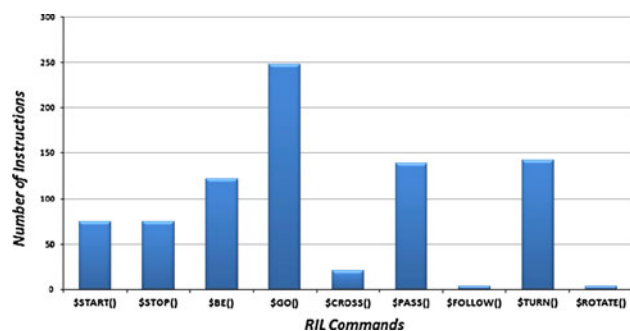


Fig. 10 Occurrence statistics of RIL commands in the resulting routes

equivalent symbolic and topological map representations. A topological map is generated to describe relationships among features of the environment in a more abstract form without any absolute reference system. It is mainly used to treat the ambiguity which can occur when the robot cannot recognize the current landmark. It handles these situations by considering the relations between the landmarks. Therefore, the topological map is used to comprehend the route instructions by building up a mental representation of the processed route. These high-level cognitive representations—i.e., symbolic and topological map representations—are supplied

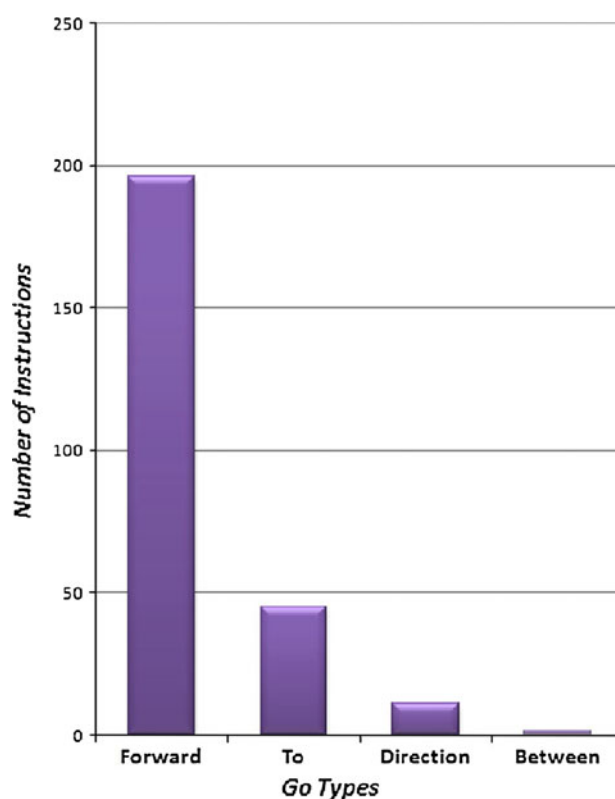


Fig. 11 Occurrence statistics of “GO” command types in the resulting routes

to other system components as an initial path estimation to guide the robot while it plans its navigation task. Finally, the symbolic representation is supplied to the motion planner to ground the landmark symbols in their equivalent physical objects by using perceptual anchoring.

Currently, we are developing a controlled spatial language that can be used to describe a navigation task for mobile robots or simulated artificial agents. The controlled spatial language contains simple natural language commands without special syntax. In future, we will try to test the performance of this language and compare it with the proposed RIL instructions.

References

- Bugmann G, Klein E, Lauria S, Kyriacou T (2004) Corpus-based robotics: a route instruction example. In: Proceedings of intelligent autonomous system, Amsterdam, pp 96–103
- Chella A, Coradeschi S, Frixione M, Saffiotti A (2004) Perceptual anchoring via conceptual spaces. In: Proceedings of the AAAI-04 workshop on anchoring symbols to sensor data. AAAI Press, Menlo Park
- Elmogy M, Zhang J (2008) Robust real-time landmark recognition for humanoid robot navigation. In: Proceedings of the 2008 IEEE international conference on robotics and biomimetics (ROBIO), Bangkok, pp 572–577, Dec 14–17

- Elmogy M, Habel C, Zhang J (2008) Robot topological map generation from formal route instructions. In: Proceedings of the 6th international cognitive robotics workshop at 18th European conference on artificial intelligence (ECAI), Patras. IOS Press, pp 60–67, July 21–22
- Elmogy M, Habel C, Zhang J (2009) Online motion planning for hoap-2 humanoid robot navigation. In: Proceedings of the 2009 IEEE international conference on intelligent robots and systems (IROS), St. Louis, pp 3531–3536
- Eschenbach C, Tschander L, Habel C, Kulik L (2000) Lexical specifications of paths, Lecture Notes in Computer Science, vol 1849/2000. Springer, Berlin, pp 127–144
- Gärdenfors P (2000) Conceptual spaces: the geometry of thought. MIT Press, Cambridge
- Habel C (2003) Incremental generation of multimodal route instructions. In: Natural language generation in spoken and written dialogue, AAAI spring symposium 2003, Palo Alto, pp 44–51
- Kollar T, Tellex S, Roy D, Roy N (2010) Toward understanding natural language directions. In: Proceedings of the 5th ACM/IEEE international conference on human–robot interaction (HRI'10), Osaka, pp 259–266
- Kyriacou T, Bugmann G, Lauria S (2002) Vision-based urban navigation procedures for verbally instructed robots. *Robot Auton Syst* 51:1326–1331
- Lauria S, Bugmann G, Kyriacou T, Bos J, Klein E (2001) Training personal robots using natural language instruction. *IEEE Intell Syst* 16:38–45
- Levit M, Roy D (2007) Interpretation of spatial language in a map navigation task. *IEEE Trans Syst Man Cybern B* 37(3):667–679
- Litvintseva L, Tanaka T, Yamafuji K, Ulyanov V (1997) Intelligence computing for direct human–robot communication using natural language and cognitive graphics. In: Proceedings of the IEEE international symposium on computational intelligence in robotics and automation (CIRA'97), pp 332–337
- MacMahon M (2005) Marco: a modular architecture for following route instructions. In: Proceedings of the AAAI workshop on modular construction of human-like intelligence, Pittsburgh, pp 48–55
- MacMahon M, Stankiewicz B, Kuipers B (2006) Walk the talk: connecting language, knowledge, and action in route instructions. In: Proceedings of the 21st national conference on artificial intelligence (AAAI-2006), Boston, pp 1475–1482
- Pires G, Nunes U (2002) A wheelchair steered through voice commands and assisted by a reactive fuzzy-logic controller. *J Intell Robot Syst* 34:301–314
- Pradel G, Hoppenot P (2006) Symbolic trajectory description in mobile robotics. *J Intell Robot Syst* 45(2):157–180
- Schulz R, Stockwell P, Wakabayashi M, Wiles J (2006) Towards a spatial language for mobile robots. In: Proceedings of the 6th international conference on the evolution of language, pp 291–298
- Simpson RC, Levine SP (1997) Adaptive shared control of a smart wheelchair operated by voice control. In: Proceedings of the 1997 IEEE/RSJ international conference on intelligent robots and systems (IROS '97), vol 2, pp 622–626
- Skubic M, Perzanowski D, Blisard S, Schultz A, Adams W, Bugajska M, Brock D (2004) Spatial language for human–robot dialogs. *IEEE Trans Syst Man Cybern C Appl Rev* 34(2):154–167
- Tellex S, Roy D (2006) Spatial routines for a simulated speech-controlled vehicle. In: Proceedings of the 1st ACM SIGCHI/SIGART conference on human–robot interaction, Salt Lake City, pp 156–163
- Tellex S, Roy D (2007) Grounding language in spatial routines. In: Proceedings of AAAI spring symposium on control mechanisms for spatial knowledge processing in cognitive/intelligent systems
- Tschander LB, Schmidtke H, Habel C, Eschenbach C, Kulik L (2003) A geometric agent following route instructions, Lecture Notes in Computer Science, vol 2685/2003. Springer, Berlin, pp 89–111