Defining a human-like problem solving strategy for a robot

Mary Felkin*, Yves Kodratoff**

*CADIA
mary@ru.is
**LRI
yvko@free.fr

1 Abstract

We developed a system the purpose of which is obtaining a robot able to emulate the strategies used by a human facing a problem-solving task. We have been able to solve this problem within a very particular psychological setting, in which the human behaviour can be interpreted as ‘observables’ of his/her problem-solving strategy. Our solution encompasses the one of yet another problem, namely, how to close up a loop starting with the behaviour of several humans, its analysis and interpretation in terms of human observables, the definition of what was are the strategies used by the humans (including inefficient ones), the interpretation of the human observables in terms of movements of the robot, the definition of what is a “robot strategy" in terms of human strategies. The loop is closed with a programming language enabling us to program these robot strategies, making them observables in the same way as the human strategies are observables at the beginning of the loop. This paper is devoted to the detailed explanation of one of the above steps, that is, how we have been able to define in an objective way what we call a robot strategy. We shall see that our solution merges two different factors. The one aims at avoiding very ‘inhuman’ behaviours and is based on the mean of the behaviour of the set of humans we observed. The other provides ‘humanity’ to the robot by allowing it to deviate from this mean by n times the standard deviation observed, paralleling to deviations from the mean of the human it is supposed to emulate. Completely new human-like behaviours are also easy to program.

2 Introduction and Motivations

The goal of this paper is to explain how, in a particular setting, we have been able to define a a problem-solving strategy by only using statistical concepts, thus this ‘strategy’ can be fitted inside the robot controller. We started with the observation of humans who were themselves put in a situation of problem-solving, and we built such an interpretation of their behaviour that it was possible to transfer this interpretation in a program controlling the robot. Before explaining what is a robot’s human-like strategy, we have to explain as briefly as possible, the various stages by which we abstracted the human behaviour.

In a sequence of psychological experiments, blindfolded human volunteers explored a maze in search of a 'treasure' and, while doing so, expressed their search strategy by sequences of perception-actions pairs, which were recorded. Perception here was limited to touch, which could be observed on the videos. Actions where limited to moving in the maze and catching the treasure, which could be also be observed. The volunteers in the mazes had several different goals which they combined through some thought process akin to multi-criteria optimisation to mentally construct and evaluate their behaviours. On top of their given goal, finding the treasure, their overall strategies included the goals of not getting lost, of not exploring the same place twice, of not bumping into obstacles, etc. We performed a detailed analysis, including a digitalisation, of the videos showing the behaviour of 10 of these volunteers, called G1_1, G1_2, G1_3, G3_1, G3_2, G4_1, G4_2, G4_3, G7_1 and G7_2 in the following. We thus could 'run' an exact replicate of their behaviour in our system and look at this replicate. Obviously, our final goal was not to obtain such a replicate but to analyse it in order to try understanding what could have been the underlying strategy of the volunteer.

The gap between human strategies and perception-action pairs is too wide to be bridged in a single learning step. We followed cognitive science architectural models of the human cognitive processes to gradually increase the complexity of what was being learnt, from our raw data made of 'observables', i. e., perception-action pairs, to the primitives, which are meaningful sequences of observables, and onwards to tactics (composed of one or several primitives) and strategies (sets of tactics).

3 Learning by imitation

There have been a large amount of work done in the field of robot learning by imitation, a relatively new (about twenty years old) field of research, see for example Billard et Siegwart (2004), Dillmann (2004) and Schaal et al. (2003), which takes inspiration from a wide range of disciplines, including psychology, biology, neurobiology, etc. Alissandrakis et al. (2002), Billard et Hayes (1999), Demiris et Hayes (2001) and Calinon et Billard (2007). An example among others of the necessary multidisciplinarity is Alissandrakis et al. (2006) who propose a mathematical solution to the correspondence problem, which originally comes from animal psychology : they formalise the correspondences by giving mapping matrices to link agents with different morphologies. Other research papers present work which is less biomimetic, for example Calinon et al. (2007) who present an architecture for extracting the relevant features of a given task and then generalise the acquired knowledge to other contexts. They demonstrated the effectiveness of their architecture by implementing it on a humanoid robot learning to reproduce the gestures of a human teacher. There is a major difference between these works and our own. Many robots are good at learning to reproduce human gestures but they make no attempt at learning the underlying human strategies. We concentrate on the problem-solving behaviour, thus what we are *not* doing is an attempt at mimicking the human surface behaviours, such as smiling, speaking etc. Our main source of information is the position of the humans and the position of their various body parts (here their body, their head, their hands and their feet). We call this information the 'observables'. Similarly, we do not attempt to estimate the performance of these search strategies, we are only interested in the method of analysing and transferring them.

Naked eye video analysis of the volunteers' behaviour showed some goal-directed tactics. These were listed for the purpose of studying whether and how they could eventually be implemented as heuristics for a robot controller. Here are a few of them : - "Keep a main direction". - "Avoid backtracking, except in a dead end". - "Avoid following the same path twice". - etc. On the other hand, the high-level heuristics discovered by the psychologists Tijus et al. (2007) and Iemmi (2005) are intelligent and could be useful guidelines for hand-coded robotic search behaviours. But they have not been learnt by a computer, they are the result of naked eye analysis of the videos of the psychological experiments. So for a different problem (or different settings and/or experimental conditions) the experiments would have to be done all over again in order to discover new behaviours, appropriate for the new problem. Automatically extracting the same information from the raw databases is another can of worms. After adequate preprocessing, perception/action pairs (i. e., what we call the observables) could be, and in some cases were, automatically defined while others were hand-crafted. For example, an observable relative to the right hand, called "isRightHandAtBodySide", has a value of one if the right hand is held by the body side, a value of two if the right hand is stretched out exploring the empty space, and a value of zero if neither of the two previous conditions hold. The correlation matrix, as shown in [], reveals that the observables are highly redundant but we made no effort to reduce this redundancy. Even those that are trivially redundant (such as 'walking' and 'following an obstacle') describe different phases of a human startegy and cannot be reduced to one single observable. The standard hypothesis in Machine Learning is that the descriptors are mutually independent. As we shall see in the following, achieving this independency would prevent us to built the intermediary descriptors (called the primitives) with which the tactics are described. This is why the raw databases were not designed to be exploited by any kind of inductive algorithm, they were designed to contain much information in a compact format.

4 From observables unto tactics and strategies

Automatically extracting from a database the heuristics, or strategies, used by humans in a problem-solving situation takes more than a good preprocessing and then running the database through the appropriate data mining algorithm. To go from the database of observables to heuristics, we had to define a middle ground. As we already said, the raw data contained in the database, called the observables, were grouped into higher level primitives. The main difference between observables and primitives is that observables are observable at every time step, what happened during the previous time step(s) notwithstanding, while primitives are combinations of observables. All movement descriptors, which require a comparison between at least two consecutive time steps, can only be primitives. Each tactic is defined by a sequence of observables and primitives and, in turn, a sequence of tactics defines a strategy.

4.1 Observables

First we record basic facts such as the position of the person in the maze at a given time step, the position of his/her hands, etc. These facts are called observables. Our "Maze" program, used as a recording tool for such a purpose, creates one database per run of a person in a maze. It records in the database, each quarter of a second, fifty observables which describe the action and the situation of a person in a maze.In these databases of observables, each row corresponds to a time step (four per second) and each column corresponds to an observable. Observables also correlate the location of the person in the maze with the location of objects and proximities are recorded. As an example, see below in Table 1, the 10 first observables, over the 50 we actually used. The choice of these observables, among the thousands of other possible ones, is by itself a matter of discussion : some are intuitively ‘obvious’ as are the coordinates of the individual. The criterion of choice is however more complex : some have added, other deleted on the ground that we could find observables that were expressing the same objective fact, but that were better suited to be part of a primitive. The whole process looks complex and it is long indeed, but it does not require detailed explanations. Similarly, choosing to define ‘near to’ as being ‘less than 20’ is quite intuitive when considering the maze. We must however admit that these choices are done by observation, as the ones done in Tijus et al. (2007) and the whole of our work does not completely eliminate subjectivity, it significantly pushes it backwards, however.

1	time	the current time step
2	ang	the current body orientation
3	headAngle	the current head orientation
4	X	the current body X coordinate
5	Y	the current body Y coordinate
6	isNearAwall	number of walls within a dist. = 20
7	isNearAstraightTable	number of straight tables within a dist. = 20
8	isNearAsidewayTable	number of sideway tables within a dist. = 20
9	isNearAstraightTableCorner	number of straight table corners within a dist. = 20
10	isNearAsidewayTableCorner	number of sideway table corners within a dist. = 20

Table 1 : The first ten observables (among 50)

3.2 Primitives and tactics

From the databases of observables we construct databases of primitive behaviours, called “primitives" for short, such as “Exploring a table top with one hand". These correlate observables over several consecutive time steps (movements can only be seen through such comparisons, so the observables can be considered static descriptors while the primitives are dynamic). We only look for some, tactic-related, movements and not for any possible movement. For example, if the person in the maze scratches his/her nose, this movement is not recorded.

Primitives are combined to describe four tactics : the goal-related treasure hunting tactic, called the “search tactic", the tactic used by the volunteer to cope with the fact that he/her has to move around blindfolded, called the “moving tactic", the tactic causing the behaviour of the volunteer encountering an obstacle, which has a mixed purpose of treasure hunting and spatial orientation, called the “obstacle following tactic" and the personal safety tactic called the “obstacle detection tactic". When we refer either to a human volunteer or to a robot, we will call an ‘individual’ the person/robot performing the actions.

The obstacle detection tactic and the obstacle following tactic could be described by a single attribute (by ‘attribute’ we mean either an observable or a primitive, in this case it is a primitive) while the other two tactics, the search tactic and the obstacle exploration tactic, needed several attributes to be described. A complete description can be found in Felkin (2008). Below, in table 2, is a list of some of the primitives we used. They are also highly redundant.

For instance, being ‘near’ something obviously implies being ‘near’. In some cases, however, it is important to be able to explicitly express that the individual is simply ‘near’ whatever it is near to.

1	Searching	19	Near a cupboard
2	Exploring object	20	Near a radiator
3	Bending	21	Near a wall
4	Sweeping	22	BC (Body near a corner)
5	Walking	23	LHC (Left hand near a corner)
6	Moving	24	RHC (Right hand near a corner)
7	Going straight	25	BU (Body near a unique object)
8	Curving	26	LHU (Left hand near a unique object)
9	Turning	27	RHU (Right hand near a unique object)
10	Staying near obstacle	28	Stopped near obstacle
11	Going off from obstacle	29	Moving in empty space
12	Arriving	30	Moving hands
13	Obst to obst	31	Unique hands
14	Following obstacle	32	Turning body
15	Obstacle detection	33	Obstacle detection in empty space
16	Bump	34	Obstacle detection in contact
17	Near	35	Unique All
18	Near a table	36	Class

Table 2 : The list of the 36 primitives we used

For instance, the 6th primitive ‘Moving’ tells us if the individual (human or robot) is or not moving : it indicates body or hand movement between time t and time t-1. When the individual is not moving, it describes a stopped individual, and it can take several values depending where the individual stopped : we chose to make the difference between stopped in empty space, stopped near a table, stopped near a cupboard, stopped near a radiator and stopped near a wall. Thus, most of these primitives are multivalued. In practice, we also made the choice to increase the number of primitives so as to only ‘observe’ binary primitives. The behaviour curves given below make use of such binary primitives.

Note that the primitives are context-dependent and so can be grouped according to the context in which they start to happen and the context in which they result :

Empty space	->	Empty space
Empty space	->	Obstacle
Obstacle	->	Same obstacle
Obstacle	->	New obstacle
Obstacle	->	Empty space

Primitives can also be grouped according to their duration. The most basic ones are the result of a comparison between the value of some variable(s) at the current time step and the value of the same variable(s) at the time step immediately preceding it, such as :

4.2 Strategies

Finally we define strategies as combinations of primitives either staying in an “on" state for a given time length or alternating between “on" and “off" states with a given frequency. This implies in-depth analysis and it will be described in the next section.

When we were determining the kinds of strategies people used in the mazes, we decided from these observations that a strategy would be composed of four tactics, as below :

Search tactic
Moving tactic
Obstacle following tactic
Obstacle detection tactic

5 The analysis

This step aims to caracterise the variation of the various descriptors during a run, thus obtaining a description of each run a set of simple statistical values.

5.1 Binarisation

For this purpose, we binarised all our primitives and drew the corresponding curves over time. An example of such a curve is given below in figure 2. Binarisation was necessary because the curves corresponding to multi-valued primitives were very hard to read and interpret. From now on, when we speak of primitives we will always mean the binarised version of them. These curves showed us, for each of the ten runs, what the humans were doing time step by time step in terms of primitives.

5.2 Average values and dividing the runs in four parts

This lead to another problem : The difference in time length between the different runs made comparison awkward. We noticed that the human individuals tend to change their strategy over the time. In order to take this fact into account, we had to divide the recording of their behaviour in parts. As a matter of simplifying the problem, we decided two divide each run in four equal parts and to average the values of the runs quarter by quarter. This gave us other curves, shown in appendix 1, which display the average, for all runs, of the values of the primitives taken one by one. Figures 5.3, 5.9 and 5.10 are examples of such. These curves are repeated with the average over all runs and the standard variation added.

5.3 Average durations

More information was extracted from the curves describing the values of the primitives over time : the average duration of consecutive series of positive values, the average duration of consecutive series of negative values, and their respective standard variations.

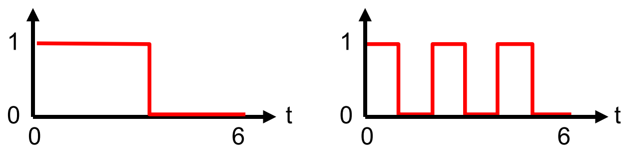

FIG. 1 – *Different behaviours with the same average value*

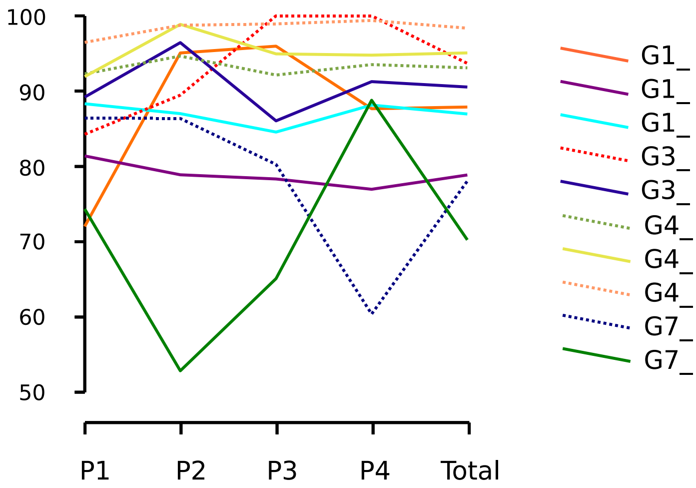

FIG. 2 – *The average values of the "Search" variable for each quarter and for the total runs*

5.4 Example

If figure 1 corresponded to the values of the "Walking" primitive, the figure on the left could be taken from G1_2 who sometimes alternated between rather fast walking, apparently at random, with rather long stops. The figure on the right could be taken from G7_1, who explored tables by walking slowly along them, sometimes so slowly that for one or more consecutive time steps (the duration of which is a quarter of a second) she appeared to be standing still. The average value of the variable represented here is 0.5 for both figures, but they correspond to different behaviours. This can be expressed by the average duration of consecutive series of positive and of negative values. In the figure on the left of figure 5.2, both averages would be 3.0 while in the figure on the right both averages would be 1.0.

The P1, ..., P4 values of figure 2 correspond to these partitions into quarters. A quarter corresponds to a different number of time steps from one run to another. This division is arbitrary and a different value could be chosen. Due to the relative shortness of the run we observed on the volonteers, more than four would complicate the situation to little avail. We observed that less than four was not in fact sufficient to detect strategy changes since strategies change within a run. This value is thus specific to our experiment and should be reevaluated for experiments done in a different setting.

5.5 Notations

Finally, the siginificant parameters seem to us to be of two kinds. One is the mean and the standard deviation of the set of behaviours for each primitive feature. This describes a 'reasonable' behaviour, by which we mean it avoids absurdities a human would not like to perform (for example going round and round in a circle). For each quarter P1, ..., P4 and for each subject, the other significant feature describing the subject's strategy is his/her deviation from the mean. It seemed to us convenient to describe this deviation in terms of mean deviation. The table below exemplifies the way we represented the variation of the variables in each qurter, nad for all volunteers. It show no comment when the subject stays near the mean, and a number of pluses or minuses saying how many times the mean deviation they deviate from the mean in the interval Pi. In order to avoid a large number of minuses, the comment 'none' means that this specific action is not performed at all by the subject. When no interval is indicated, this means that the behaviour is observed within all intervals. When an interval is omitted without comment, this means that this interval is in the mean. Notice that we very seldom observed a deviation from the mean of three times the standard deviation, and and this was the maximum one ever observed on the human volunteers, except when a behaviour was totally absent.

6 Run by run strategy analysis

6.1 Notations

Finally, the siginificant parameters seem to us to be of two kinds. One is the mean and the standard deviation of the set of behaviours for each primitive feature. This describes a 'reasonable' behaviour, by which we mean it avoids absurdities a human would not like to perform (for example going round and round in a circle). For each quarter P1, ..., P4 and for each subject, the other significant feature describing the subject's strategy is his/her deviation from the mean. It seemed to us convenient to describe this deviation in terms of mean deviation. The table below exemplifies the way we represented the variation of the variables in each qurter, nad for all volunteers. It show no comment when the subject stays near the mean, and a number of pluses or minuses saying how many times the mean deviation they deviate from the mean in the interval Pi. In order to avoid a large number of minuses, the comment 'none' means that this specific action is not performed at all by the subject. When no interval is indicated, this means that the behaviour is observed within all intervals. When an interval is omitted without comment, this means that this interval is in the mean. Notice that we very seldom observed a deviation from the mean of three times the standard deviation, and and this was the maximum one ever observed on the human volunteers, except when a behaviour was totally absent.

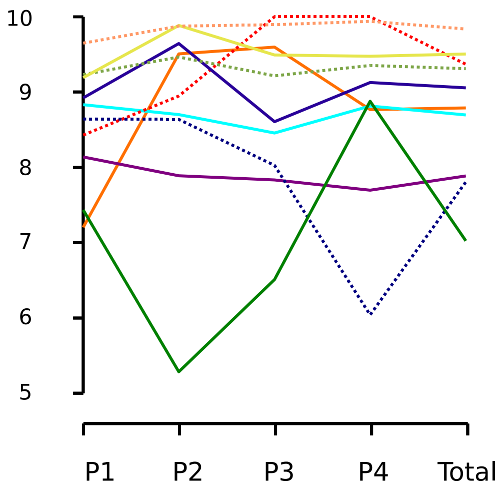

FIG. 3 – *The average value of the “Searching" binary variables for the ten databases, the runs colour code is the same as in figure 2*

	Obs det	Obs empt space	Obs contact	Moving	Search
G1_1	P1-, P2- -, P3- -, P4- -	none	none	P1-	P1- -, P2+, P3+, P4+
G1_2		none	P1+, P2+, P3+		-
G1_3		none			
G3_1	+	-		P1-, P3+, P4+	P3+, P4+
G3_2	P2-, P3+	P1-, P2-, P4-		P2+	
G4_1			P3-		
G4_2			P4+		+
G4_3	P1+, P2+, P3++	P3+	P1+, P2++, P3+	+	+
G7_1	P2-	P1++, P2+, P4++	-	P3-, P4–	P3-, P4- -
G7_2	P1- -, P2-, P3+	P1-, P3 ++, P4+	none	P2- -, P3-	P1-, P2- - -, P3-

Table 3 : The “Obstacle detection", “Obstacle detection in empty space", “Obstacle detection in contact", “Moving" and “Searching" behaviours for the ten volunteers

In G1_1 “Obs det" we observed that G1_1 detects the obstacles less than the mean in interval P1, this by an amount approximately equal to the standard deviation. This observation is noted by P1- in the table.

As noticed above, humans tend follow strategies that avoid behaving more than three standard deviations from the mean. When we set up a programming language as briefly described in the next session, we tried to enlarge this ‘definition by observation’. We very soon observed

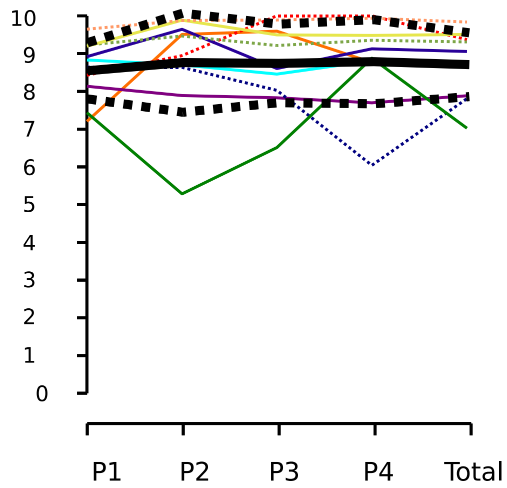

FIG. 4 – *The average value of the “Searching" binary variables for the ten databases, with the global average and standard deviation*

that the programs that do not follow this rule will behave in ‘typically non human way’, that is repeating the same action loop, being caught for ever in a strategy etc. This is why we now define a human-like robot strategy as a control system that is able to observe a set of human, define a mean and a standard deviation in the human behaviour and, finally that generates a control respecting the human law of “either you do nothing or, if you act, then never go over the limit of three times the standard deviation".

As another important consequence, these results enable us to render the descriptions of the behaviours somewhat more objective. The ‘feeling’ a psychologist may have in front of a given behaviour can now be commented without resorting to a psychological explanation. For instance, two of our volunteers (psychological explanation : they have been visibly tired of searching to no avail) suddenly stopped their searching behaviour and started to dance in the middle of the room. Whatever were their motivations for dancing may be a point but it is also very interesting to notice that their searching suddenly dropped down to ‘none’, that their movement in empty space became very high. We also observed that this seemingly absurd behaviour, a so typically human one, happens to have been a success strategy. In the programming language we will describe in the following, it would be quite easy to program the robot for a thorough search, followed by a drastic change of strategy after some time of failure.

7 Programming the strategies

Programming a langage in which strategies similar to the human ones can be used to control a robot asks for some decisions that are not obvious to take. At first, we simply programmed a way to reproduce the various primitives and to include them into tactics ans strategies. For instance, in order to deal with movement in empty space, we introduced the “Empty Space" tactic control, which can take four possible values : Not_Walk, Walk_Straight, Walk_Curve, Walk_Turn.

It was also necessary to introduce six control variables. When acting autonomously, the robot's actions are controlled through six main variables, which combine to reproduce all types of recorded actions. These variables are :

EXPLORE_OBJECT
EXPLORE_GROUND
EMPTY_SPACE
GO_OFF
FOLLOW_OBJECT
OBJECT_DETECTION

EXPLORE_OB has five possible values :

None :	The hands stay by the body side or searches the empty space in the obstacle detection behaviour (explained below)
1HnE :	One hand not efficient. One hand follows the near side of the object, only covering a small percentage of its surface as the robot goes along. The other one is a the body side or exploring the empty space
2HnE :	Two hands not efficient. Both hands act as described above. They are following one another along the near edge of the obstacle the robot is following
1HE :	One hand efficient. One hand sweeps the whole surface or nearly the whole surface as the robot goes along
2HE :	Two hands, at least one acting efficiently

Here is, for example, a detailed description of EXPLORE_OBJECT, the object exploration behaviour.

Exploring an object means in this context feeling its surface with one or two hands. As the robot goes along an obstacle, the hand on the side towards the obstacle can either follow the edge of this obstacle, sweep the obstacle, or stay by the body side. Sweeping is considered to be the efficient exploration, while following the side of the obstacle is not an efficient way to check whether the treasure is on the table.

Figure 5 should be read from left to right. The robot was moving from right to left.

The time sequence in figure 5 does not show consecutive time steps. The robot takes more than one time step to move along one table. It takes about 12 time steps (3 seconds) to walk the length of an average table while exploring its surface.

We needed a flexible way to combine our six control variables, hard coding all the combinations we might possibly need to implement a strategy was not possible. So we implemented a very simple and problem-dependent programming language.

Here is an example of a program simulating a crisscrossing behaviour in empty space

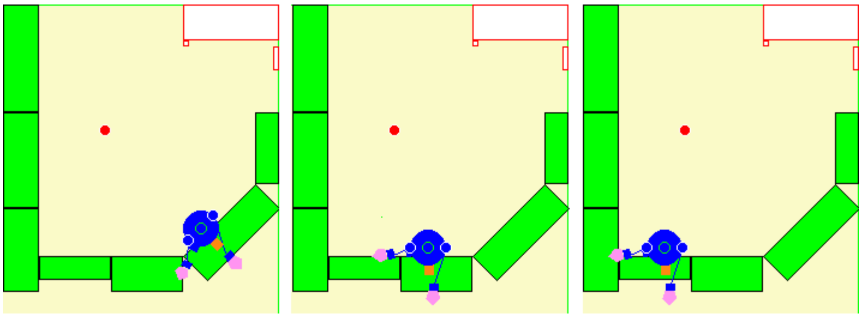

FIG. 5 – *A time sequence of screenshots of a simulated robot enacting EXPLORE_OB = 2HE and FOLLOW_OB = Serious*

LABEL1 :
Go away from any obstacles and walk straight DURING (10) GO_OFF = 2, EXPLORE_GR = 2, EMPTY_SPACE = 2
Walk straight until you encounter an obstacle UNTIL (OB) EXPLORE_GR = 2, EMPTY_SPACE = 2
Loop IF (TRUE) GOTO LABEL1

This very basic example describes an open space crisscrossing behaviour : the robot goes away from any obstacle it encounters in a randomly chosen open direction.

We produced also complex programs that can reproduce a human behaviour. In order to ease the psychologists' programming efforts, we built an automatic program generator that analyses human behaviour in the same way as we did, and produces the parameters of program that, instead of reproducing the way the human acted, reproduces his/her strategies. This enables the psychologist to obtain an infinity of different traces, all pertaining to the same type of behaviour. The 'robot' actions appear as a numerised video entirely similar to the ones we build from the 'real life' videos provided to us by the psychologists.

8 Conclusions

8.1 human/robot problem solving

The primary purpose of this work is showing that humans when they are in position of solving a problem make use of strategies that can be analysed and transferred to a robot.

In order to accomplish this purpose, we had to define and to reduce the scope of what is a "human problem solving". In our case, we chose to analyse the behaviour of humans placed in a maze in which they had to find a "treasure". Since the mazes were relatively small, the subjects were blindfolded. This obviously is a really particular case of problem solving and we cannot claim to have found a general solution to the simulation of the problem solving human behaviour. It was also frequently the case that several individuals were asked to cooperate in solving the problem. Due to time limitations we considered the sole problem of an individual behaviour. It follows that the problem we really addressed is the one of solving the behaviour of blinded single individuals trying to discover a treasure within a maze. Even though our solution is not directly applicable to every problems, we shall try to show now that it provides the steps necessary to solve any other particular case by observing humans solving the problem themselves.

Obviously, it could be possible to reproduce the exact behaviour of the observed humans and we do have several “robotic simulations" which are the exact behaviours of a human being. These traces are very useful in order to compare what a “real human" does with what a “simulated human" does. Inversely, these exact reproductions are useless to robotics since the humans are always observed in a particular setting and the slightest change in this setting would make the trace useless. Similarly, they are not very useful to the psychologist since they are nothing more than a digitalised version of the video we started from.

8.2 The generalisations we performed

Generalization primarily took place when we defined the “list of primitives" with which we wanted to analyse the behaviours of the tested humans. As a matter of fact, these primitives are features that take observable values in each experimental situation. For instance the primitive “near a table" can take the values true or false. Whatever the experimental setting, we can observe if the subject (a human or a robot) is near or far from a table. In that sense, this primitive feature generalises the fact that the subject is placed near a table. A further generalisation took place when we decided to define the feature “near" which says if the subject is near or far from a list of potential obstacles. We cannot insure that this list of features, nor their relationships, are useful in all situations. For instance, in a really three dimensional setting, the features “above" and “under" should certainly have to be added ; in a street circulation setting, the features of the obstacles should be evaluated as a function of the danger they bring, etc. Nevertheless, within the limited universe of our experiments, we defined a set of features that seems to be quite satisfactory in rendering a human behaviour. We lacked time to be able to analyse in detail the relative importance of each of these primitive features. We succeeded at least in making possible this analysis since we can program a robot to behave as we want, while it is almost impossible to ask a human to go on acting natural while suppressing or exaggerating one of these features.

8.3 What is a strategy ?

The first strategy, we call a ‘core strategy’, is the one that avoids what a human would consider as being absurd. Staying for ever at the same place, going in small circles, doing again and again the same thing, looping trajectories etc. may happen to a human who is lost, but they will do their best to avoid the behaviours they consider as being absurd. Instead of proposing strategies dedicated to the solution of each of these problems, we observed that, in the mean, the 10 subject whom behaviour we digitalised, avoided these traps. As a consequence, we used the mean values rendering their behaviour. The result is that the robot which runs with these mean values does none of these mistakes. In a sense, the combination of all the values of the primitive features observed on humans provides a very simple solution to a very difficult problem.

Particular strategies. Some individuals deviate from this mean in two different ways. It is possible that some values of the primitive features are never activated. The subject is then infinitely below the mean value. It is also possible that some individual primitive values are far over the mean value. Nevertheless, as shown above, they at most reach three times the mean value. In that sense, all individuals never deviate too much from the means values because they obviously all follow a kind of core strategy to avoid absurd behaviours. What we will then call an individual strategy is defined by a set of deviations from the mean, in no way an attitude that could not be linked with the 'normal' social behaviour.

Références

Alissandrakis, A., C. L. Nehaniv, et K. Dautenhahn (2002). Learning to imitate corresponding actions across dissimilar embodiments. *IEEE Transactions on Systems, Man, and Cybernetics 32*, 482–496.

Alissandrakis, A., C. L. Nehaniv, et K. Dautenhahn (2006). Correspondence mapping induced state and action metrics for robotic imitation. *Special Issue on Robot Learning by Observation, Demonstration and Imitation*.

Billard et Hayes (1999). Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behaviour 7*, 35–64.

Billard, A. et R. Siegwart (2004). Robot learning from demonstration. *Robotics and Autonomous Systems 47*, 65–67.

Calinon, S. et A. Billard (2007). *Imitation and Social Learning in robots, Humans and Animals: Behavioural, Social and Communicative Dimensions*. Cambridge University Press.

Calinon, S., F. Guenter, et A. Billard (2007). On learning, representing and generalising a task in a humanoid robot. *IEEE Transactions on Systems, Man and Cybernetics 37 (2), Part B, Special Issue on Robot Learning by Observation, Demonstration and Imitation*.

Demiris et Hayes (2001). *Imitation in animals and artifacts*.

Dillmann, R. (2004). Teaching and learning of robot tasks via observation of human performance. *Robotics and Autonomous Systems 47*, 109–116.

Felkin, M. (2008). *Learning Human Strategies by Imitation and Induction*. Thèse de doctorat, Univ. Paris-Sud.

Iemmi, V. (2005). Les apports des nouvelles technologies a la psychologie clinique : Les robots comme compagnons therapeutiques.

Schaal, S., A. Ijspeert, et A. Billard (2003). Computational approaches to motor learning by imitation. *Philosophical Transactions of the Royal Society of London: series B, Biological Sciences 358*, 537–547.

Tijus, C., N. Bredeche, Y. Kodratoff, M. Felkin, C. Hartland, E. Zibetti, et V. Besson (2007). Human heuristics for a team of mobile robots. *Proceedings of the 5th IEEE International Conference on Research, Innovation and Vision for the Future (RIVF'07), Hanoi, Vietnam,*.