High Level Imitation Learning in a Treasure Hunting Task

Mary Felkin
CADIA
Reykjavik University
mary@ru.is

Yves Kodratoff
LRI
Université Paris Sud
yvko@free.fr

Abstract

High level learning by imitation consists of learning to emulate high level human cognitive processes. We have been able to solve this problem within a specific setting, searching for a treasure in a maze. The human behaviour can be recorded as time series of observable variables, representing the portion of the human's problem-solving strategy which is apparent to an outside observer. These can then be subjected to several layers of constructive generalisation, until the strategy is abstracted from the original settings and can thus be applied to new ones.

1 Introduction

Current agent architectures are often built on the model of human cognitive processes such as they are described by cognitive scientists. The lower level corresponds to reactive processes, such as reflexes, only dealing with the immediate situation; the middle level corresponds to deliberative reasoning and as such can also deal with the past and the future; the top level corresponds, among other mental tasks, to even more long-term strategical decision making which leads to goal generation (see for example [Sloman, 2001]).

Learning can occur at any or all levels of the control architecture of an artificial agent but 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 the observable perception-action pairs which constituted our raw data, to meaningful sequences of them which constituted basic actions, and onwards to tactics and strategies. The success of our attempt contributes to the validation of these cognitive science models.

Inducing the plans of a human placed in a problem solving context is also a plan recognition task. The human makes mental plans about how to solve the problem and from the time series of his actions these plans become partially apparent. In this paper the "mental plans" are called the tactics.

We were not interested in the performance of such or such human strategy, we were interested in how they could be learnt. But high level learning by imitation has two major effects. As the basis of our system is the average human behaviour, "inhuman" mistakes such as going round in a circle or going backwards and forwards along the same path are avoided. As the intrisic variability of human behaviour is reproduced, but bounded by a function of the standard deviation of the humans' behaviours, the resulting robotic strategy turned out to be an efficient sweeper.

For the final step, from tactics to strategies, we found that a robotic strategy could be given a mathematical formulation. Comparing possible robotic strategies to human strategies enabled us to define a similarity measure of "human likeness" for the robotic strategies. Such definitions, even in a limited context, are a step towards building an engineering blueprint of human intelligence.

2 Related works

A large amount of work has been done in the field of robot learning by imitation, a relatively new (about twenty years old) field of research, see for example [Billard and Siegwart, 2004], [Dillmann, 2004] and [Schaal *et al.*, 2003]. This field takes inspiration from a wide range of disciplines, including psychology, biology, neurobiology, etc. [Alissandrakis *et al.*, 2002], [Billard and Hayes, 1999], [Demiris and Hayes, 2001] and [Calinon and 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. A formal definition of plan recognition can be found in [Kraus, 1991].

3 Experimental settings

In a sequence of psychological experiments, blindfolded human volunteers explored a maze[1] in search of a 'treasure'

[1]The mazes were not virtual, they were built with rows of tables and sometimes cupboards in a large room.

and, while doing so, expressed their search strategy[2] by sequences of perception-actions pairs, which were recorded. Perception here was limited to touch, which could be observed on the videos. Actions were limited to moving in the maze, touching objects and picking up the treasure, these could also be observed. The psychologist [Iemmi, 2005] and the mixed team [Tijus *et al.*, 2007] showed that 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 most often used 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 a close[3] 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 understand the underlying strategy of the volunteer and to become able to reproduce it in new contexts.

4 From observables to control variables

Automatically extracting from a database the 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 [Felkin, 2008]. To go from the database of observables to heuristics, we had to define a middle ground.

4.1 The observables

The raw data contained in the database, called the observables, are indicated in fig.1. They are the basic facts such as the position of the person in the maze at a given time step, the position of his/her hands, etc. 50 observables were recorded every quarter of a second in what we call the "log files". Each maze also has a static description indicating the position of the obstacles, of the treasure, etc.

Fig.1 models the human's cognitive processes as a very simplified version of the HCog-Aff (Human Cognitive Affects, [Sloman, 2001]) model, and superimposes our definitions. Our use of the HCog-Aff model is explained in details in [Felkin, 2008].

A subjective impression about the current search activity of the person in the maze was recorded at each time step (it is our 50^{th} observable). As the videos ran in slow motion, this value was set to one of three possible values: "Systematic search", "Random search", "No search". We call this observable the "class" variable and it's use is justified in section 5 (it can be dispensed with but using it reduces run-time complexity).

[2]Our use of the expression "search strategy" here does not imply the volunteers were searching according to an explicit plan. Randomly searching through the maze is also a "search strategy", and so is "not searching at all".

[3]Our replicates are only "close" because the recordings are noisy. We believe this noise contributes to the robustness of the ensuing generalisations but we did not test this hypothesis.

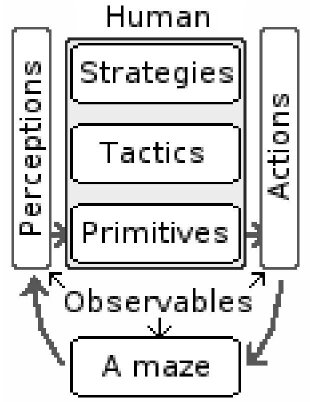

Figure 1: From observables to strategies

4.2 The primitives

The first 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, and sometimes of observables and static maze descriptors. All movement descriptors, which require a comparison between at least two consecutive time steps, can only be primitives.

Another important difference is that primitives exclusively make use of the information available to the human. We are no longer modelling a maze and a person moving in it as seen from the outside, we are modelling the human's perceptions and basic actions. So all the (X, Y) coordinates information is removed and replaced by descriptors such as "the person is near a table". What is more, this type of information is only recorded when the human knows it (in this case, once the person has touched the table).

We often encountered questions such as "when a person is walking near an object, which is the angle interval between the edge of the object and the direction of mouvement which would best describe whether this person is following the object or not?". We solved this question, and many others, with the decision tree inducer C4.5[4] [Quinlan, 1993] [Quinlan, 1996]. With 49 observables, and no fixed number of time steps limit, we could generate a huge number of possible combinations.

Some "temporary class" descriptions were sometimes recorded on subsets of the logs. In our example, we recorded a "temporary class" which has three possible values for the obstacle following behaviour: "strong", "weak" and "none". Then we let C4.5 build the corresponding primitive descriptions from the observables.

We settled our choice on 36 primitives, some of which were entirely hand-crafted while others were automatically or partially automatically constructed.

[4]We tried a few classification algorithms which met the following criteria: generating explicit models, being readily available and being fast. C4.5 turned out to be the best predictor of our class attribute among them.

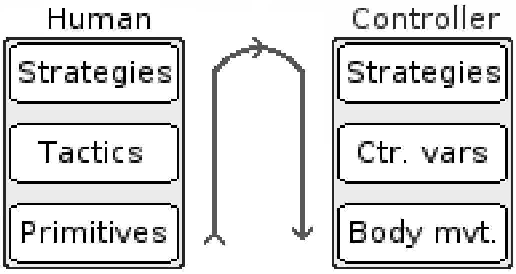

Figure 2: From primitives to body mouvements

4.3 The tactics

Each tactic is defined by a combination of observables and primitives. We defined 4 tactics:

- the goal-related treasure hunting tactic, called the "search tactic",
- the tactic used by the volunteer to cope with the fact that he or she 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"
- the personal safety tactic called the "obstacle detection tactic".

In [Felkin, 2008] we explain in details why and how this particular decomposition was chosen. The combined effect of enacting one of each 4 types of tactics is a strategy, a formal definition of a strategy is given in the next section.

4.4 The control variables

Fig.2 shows the path we followed in this work: First a bottom-up generalisation, in several steps, which started with the log file recording the movements of the human and was achieved with the help of machine learning algorithms (as seen above). Then the top-down implementation of the induced strategies into control variables ("Ctr. vars" in fig.2) and robotic body mouvements ("Body mvt").

The robot controller programming language which we implemented gives the actions which should be performed for all possible situations. (See the next section for a definition of "situation variables" and "control variables"). When we reached this stage we knew how to express the strategies in terms of tactics and the tactics in terms of primitives and observables. So we sorted our task-relevant actions (see below for how these were determined) according to body parts. The legs are considered a single body part. The lower levels of the controller does not need to know whether the robot should move forwards because it is part of its search tactic or because it is part of its obstacle following tactic. This was also a way of eliminating the conflicts which would certainly have arised, in a probabilistic context, if we had attempted to implement the tactics independently one from the other.

4.5 The body mouvements

This part translates control variable values, such as "follow the table by moving alongside it and sliding your nearest hand along the edge of it to guide yourself" into the corresponding body mouvements. The feedbak from the environment is again expressed strictly according to what the person in the maze would know. For obstacle following[5] the person needs to know whether his/her hand is touching the flat top of an object, the edge of it, the side of it, whether his/her arm is extended,etc.

5 Definitions

5.1 General definition of a strategy

A situation variable is a descriptor of perceptions of the environment external to the controller[6]. Each of the M situation variables has a finite and known number of possible values.

A control variable is a descriptor of robot action. Each of the N control variables has a finite and known number of possible values.

Formally, a robotic strategy is:

- A finite set of external situation states, E. Each situation state of E is expressed by a vector of M situation variables values: (e_1, ..., e_M).
- A finite set of internal action states, I. Each action state of I is expressed by a vector of N control variables values: (i_1, ..., i_N).
- An action transition matrix mapping all possible situation states to all possible action states. The values contained in this matrix are the probabilities of the robot enacting the behaviour described by an action state given a situation. We call it $\Lambda_A = a_{ij}$.
- An action duration mean transition matrix mapping all possible situation states to all possible action states. The values contained in this matrix are the means, should the robot enact the behaviour described by an action state, of the duration of all control variables of this behaviour. We call it $\Lambda_{AD} = a_{d-ij}$.
- An action duration standard deviation transition matrix mapping all possible situation states to all possible action states. The values contained in this matrix are the standard deviations, should the robot enact the behaviour described by an action state, of the duration of all control variables of this behaviour. We call it $\Lambda_{ASD} = a_{sd-ij}$.

Whenever the situation state of the robot changes, the robot goes into a certain action state chosen randomly according to the probabilities of Λ_A. It draws durations, in independent draws, for all the control variables values according to the Gaussian probability distributions defined by Λ_{AD} and Λ_{ASD} and starts a count down to implement these durations.

[5]We use this example all along because the obstacle following tactic was the simplest to describe.

[6]Not necessarily "external to the robot", the input from a sensor describing the state of the internal battery would be a situation variable value.

When the situation state of the robot does not change but one of the control variable values reaches the end of its randomly assigned number of time steps, the robot goes into another action state chosen randomly, according to the probabilities of Λ_A, among all action states which have the same values for all the other control variables and a different value for the control variable which is due for a change. It draws a duration for this new control variable value according to the probability distributions defined by Λ_{AD} and Λ_{ASD}.

So switches from one action state to another can be triggered both externally, by a change of situation state, and internally, by the reaching the end of some control variable value life span. When the log file shows a such a switch happening independently of these two conditions, it corresponds to a change of strategy. Changes of strategy are defined by a subset of situation states, either of which triggers the change, by a consecutive sequence of situation states belonging to this subset, or by a time limit assigned to each consecutive strategy.

When a strategy is reproduced from a given log file, the probabilities of transitions which never occurred can be set to zero and the corresponding values of Λ_{AD} and Λ_{ASD} left undefined.

This definition of a strategy resembles a Hidden Markov Model (HMM). Our Λ_A corresponds to the confusion matrix of a HMM. But we have no internal state transition matrix, it is replaced by two external matrices, grounding every internal transition probability into the external context and time. This also removes the need for the vector of initial internal state probabilities. Another difference is that the count down mechanism makes the internal state of the system dependant not only upon the previous internal state, nor upon any fixed number of previous states, but upon a variable number of previous states.

Our two external matrices Λ_{AD} and Λ_{ASD} models the variability over time intrinsic to human behaviour. In every day speech, a "robotic behaviour" has come to mean an inhumanly rigid and repeating behaviour. Our goal being to learn human problem-solving strategies our model needed this flexibility over time. Moreover, we believe that this flexibility is one of the advantages humans have over robots, an advantage which contributes to making humans more efficient in real-life situations. Λ_{AD} and Λ_{ASD} also have "the opposite effect" as they enable the robot to "remember what it is doing" and so preclude erratic behaviour.

Fig.3 illustrates the importance of considering the durations. If the binary variable represents "Moving forwards", the top part of fig.3 would be a slow walk, the discretisation in time steps and integer distance units filling the log with alternating values "1" and "0". The lower part would be a fast walk followed by a pause. The average value of this variable is obviously the same top and bottom, one half, so these two very different behaviours can only be distinguished by the average durations of consecutive series of values.

5.2 Simplifications and necessity of using Machine Learning algorithms

Brute force mimicking of a human strategy according to this definition of a strategy would be trivial (the matrices can be

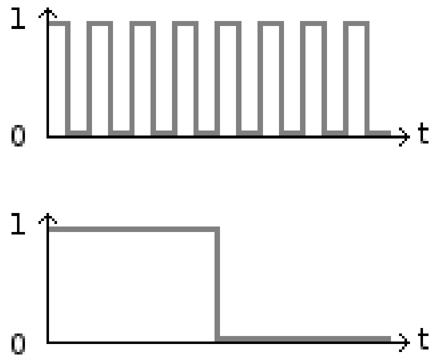

Figure 3: Same average, different behaviour

filled by counting the relevant occurrences in the log file corresponding to the strategy of this human) but it would also be intractable in any but the most basic settings and it would require very large log files. Luckily strong simplifications are possible without deterring from human-like behaviour. In a complex situation these simplifications cannot be handcrafted, they can only be achieved through the use of machine learning algorithms.

Our definition has an inherent simplicity in that it only takes into account the influences of situation variables upon action variables. The influences of action variables upon situation variables are not part of our definition of a strategy. The influences of action variables upon situation variables, for example the probability of reaching a table after having walked straight ahead during 10 seconds, describe the maze and not the strategy.

One simplification consists of reducing the length of the vectors of E. This means clustering all possible situations and identifying for every cluster the most significant descriptors. This solves the problem of defining what should be done if the robot encounters a situation the human never encountered. In our settings we managed to reduce the length of the vectors of E to 1 and to reduce all our situation variables to binary variables. We were unable to make our situation descriptors mutually exclusive, but we were able to order them according to their influence, which is the next best thing.

Another simplification is to eliminate all task-achievement-irrelevant behaviours (even if the psychologists find them meaningful). Attribute construction and attribute selection algorithms are used to build and select useful descriptors. A class attribute can be defined and its values set to correspond to different types of strategy, even if differentiating among these particular types of strategies is irrelevant for the intended robotic controller. This can be done upon a subset of the log but it has to be done if this simplification is to be used. Otherwise the attribute construction and selection algorithms have no way of "knowing" whether we are interest in goal-directed actions or in the actions indicating, for example, whether the person in the maze is enjoying him/herself.

For implementation purposes it is obvious that the lower tail of the Gaussian distributions controlling the duration of an action should be trimmed. It would be meaningless for the robot to attempt to do something during -3 time steps, so the lower tail is trimmed at zero or above. The higher tails are also trimmed, the reason for this is explained in the next section.

5.3 Definition of human-like strategies

We define a human-like robot strategy as a strategy which component behaviours are enacted not more than three standard deviations away from the average number of time steps during which the humans enact them, or not enacted at all. (The "not enacted at all" provision is necessary to include settings where not every possible behaviour can be enacted, for example fingering a dangling set of keys cannot be enacted if there are no dangling set of keys in a given maze). The number of standard deviations required to be within limits is set to three according to our empirical observation that humans stay within this range, for all behaviours.

We define a similarity measure of "human likeness" as the average distance between the average number of time steps a robot enacted a behaviour and the average number of time steps the observed humans enacted this behaviour. Behaviours which are never enacted are not taken into account. Our similarity measure becomes less and less meaningful when the robotic environment become more and more different from the human one.

We define a higher level imitation learning control system as a control system that is able to observe a set of human behaviours, find out the mean and standard deviation of the number of occurrences of these human behaviours and generate a control respecting the human law of "either you do nothing or, if you act, then never go either over or under the limit of three times the standard deviation".

Justification: as already mentioned, humans tend follow strategies which composing behaviours avoid being more than three standard deviations away from the average of these behaviours. When we set up a programming language, we tested what would happen when this "definition by observation" was not implemented. We very soon observed that the programs that do not follow this rule will behave in typically inhuman ways, repeating the same actions in a loop, being caught for ever in a particular behaviour etc.

Restriction: The settings have to be comparable. If our robot, after having observed humans in mazes mostly made of tables and sometimes of cupboards, was sent out to explore a maze mostly made of radiators and only containing one small table, it would make no sense for it to spend as much average time as the humans exploring a table.

6 Curiosity and an example of strategy

Curiosity is a natural drive towards exploration and a natural incentive for learning. It has been shown to be stronger in wild animals. [Skinner, 1922] wrote that a flighty animal such as the pronghorn antelope will approach a person lying on the ground waving a red flag. It can be found in their genomes: In an experiment with chickens, [Murphy, 1977] found that chicks from a flighty genetic line were more likely to panic when a novel ball was placed in their pen, but they were also more attracted to a novel food than birds from a calm line. It is not a coincidence that curiosity is genetically linked with the propensity to take flight [Murphy, 1977]. Curiosity can lead to dangerous behaviour. If among some hypothetical swarm of Mars robot some are more curious than others, they would also have to be programmed to explore new and interesting texture carefully and to communicate their results to the other robots. So if one is curious about a landslide it could either escape it or, at least, prevent the others from also getting caught in it. A robot landing on an unexplored planet should not ignore objects it was not programmed to encounter, or it my pass by some interesting new discovery (or ignore something dangerous). Rodney Brooks, among others, argues that a fleet of cheap autonomous robots would be a more efficient Martian explorer than a few more sophisticated but remotely controlled robots [Brooks and Flynn, 1989]. In a population, curiosity leads to a wide variety of behaviours and so increases the likelihood of discovering efficient strategies. The behaviours of the volunteers in the mazes which were due, according to the psychologists [Tijus *et al.*, 2007] to curiosity were varied and often unique. This diversity, and the diversity of the objects which were the subjects of the volunteers' curiosity, lead to the generalisation of these objects under the denomination "unique objects" because there was only one of each kind in the mazes. They included a dangling set of keys, a video camera lying on a table, etc.

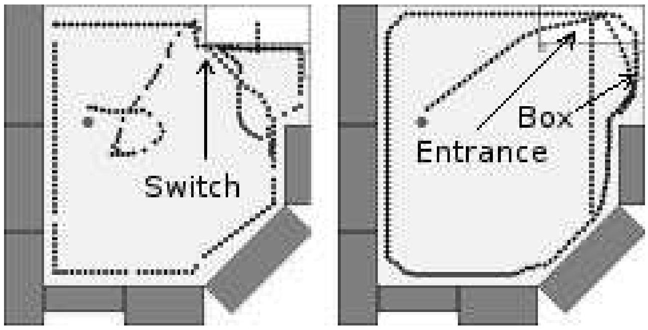

Figure 4: Left: actual G7_2 run. Right: simulated G7_2 run. The unique objects, in white, are the entrance, the light switch and a box on the wall.

The psychological experiment [Iemmi, 2005] this work extends established that the volunteers built mental maps of their surroundings by asking the volunteers to draw a map of the maze they had visited, and though none of them had ever seen it with his/her eyes the psychologists got some rather accurate maps. The people in the maze were in the situation of a robot with limited sensor capabilities, and in this uncertain world, unique objects acted like tactile beacons. G7_2 could have had an underlying strategy very different from that of the robot, and the apparent matching could be purely coincidental. The plan she drew, though, indicates that she had fully grasped the circular nature of the G7 maze. She also indicated where the door was. G7_2 had been following the walls or the rows of tables against the walls when she encountered the door frame and its light switch for the second time. G7_2 realised she had gone all round the G7 maze. She completely modified her search behaviour and started exploring

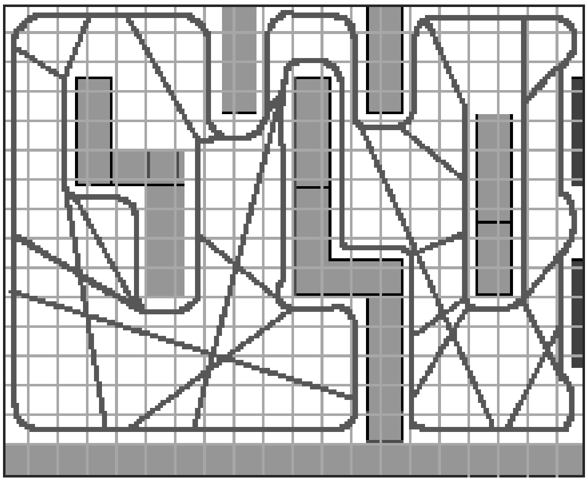

Figure 5: 10 minutes run in the G4 maze

the empty space. We can see in fig.4 (left) that her strategies before and after the second contact with a unique object are very different.

Our robot learnt to emulate these strategies, and the change of strategies depending upon the second encounter with a unique object. A (very) simplified rewritting of these rules is:

- At first, go straight until you encounter something.
- Then, choose a random direction and follow the obstacles. Remember all unique objects you encounter.
- When you come for the second time upon a unique object, stop following the obstacles. Choose a random open direction and go off into the empty space.

On the run illustrated fig.4 (right), the robot was luckier than the human, as it found the treasure after having covered a shorter distance. With other initialisations of the random number generator, the robot's path can be longer, this version was chosen because it is easy to interpret it visually.

7 Results

We consider that the fact that some human problem-solving strategies are learnable is more important than the actual strategies being learnt here.

7.1 Reasons for the lack of comparisons between robot and human performance

The purpose of the volunteers was to find the treasure (or, in some cases, for one of them to find the treasure, after which they could all walk out). So their performance in terms of time to achievement and/or ground coverage depended too much upon the original location of the treasure for performance comparisons to be made. We only had the logs of ten human runs. Each human only went through a maze once. They did not all go through the same maze. Three humans definitely did not spend all their time in the maze searching for the treasure. So such comparisons would in any case have had low statistical significance.

The logs of "performant humans", who found the treasure quickly, do not generate better controllers. In fact in one case it was quite the opposite: G1_1 used a systematic right-handed blind man walk: she followed the wall on her right, thoroughly exploring all objects she encountered on her right and ignoring anything on her left. She quickly found the treasure because it was located on the ground, near a radiator which was fixed to the wall. The corresponding controller makes the robot go round in circles following the outside perimeter of the maze and never explores the interior so is definetely not a good sweeper. G1_1 only covered about a tenth of the maze before finding the treasure. In terms of time to achievement she was fast, in terms of ground coverage she was very slow.

On average the humans took 11'32" to find the treasure. Our controller induced from the combined logs of all humans reached between 78% and 94% of all reachable squares (ground and tables) at least once after 10 minutes, depending on maze size and complexity. But concluding that, given uniformly random treasure locations, the robot did better than the humans would be, to say the least, premature.

Given these facts and lack of human-related data we regretfully forwent comparison measurements.

7.2 Robot results

Our program can read the log file of any individual human volunteer and automatically translate it into an implemented robotic strategy. But the results we give here are the results of the program generated from all logs joined together.

Fig.5 records the track of such a run of our program in the G4 maze. The unreached squares, about 14%, are the ones which are part of the (white) ground and not touched by the track. We could also have unreached squares on the tables, though this is not the case here: the robot went at least once along a side (or both) of all table-made "maze walls" and explored them all (the track shows the position of the body of the robot, not the position of the hands).

The settings for the following were the four mazes from which one or several logs had been drawn, and six extra (invented) mazes used for testing purposes.

- The performance was not better in the "known" mazes than in the invented mazes, showing that the strategies had really been abstracted from their original settings.
- All tables had been explored after at most 11 minutes (scale and speed corresponding to the real settings).
- On average, 83% of the tables had been explored after 3 minutes.
- Dividing the ground in squares 20 pixels across[7], which corresponds to the average "width" of a human as seen on the videos, between 78% and 94% of all reachable squares had been reached at least once after 10 minutes, the actual average values varying according to maze size and complexity.

[7]We were given the plans of the mazes by the psychologists, but these plans had no scale. When we inquired about it we were told that the corresponding rooms still existed and could be measured. We declined and used pixels for our distance unit.

- These percentages increase with run duration.

Many performance measurements can be applied to such settings. Strategies were also evaluated in other terms, including according to very empirical impressions.

Our goal was to show that the strategies of humans in a problem-solving situation are learnable, not to implement an efficient sweeper. We noticed that we produced proper models of inefficient behaviours as well as of efficient ones. For instance, the psychological bias "bottles are on tables and not on the ground" was learnt by our robot. We consider this a success even though it detracts from search efficiency when the treasure is on the ground.

8 Conclusion

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.

Obviously, it could be possible to reproduce the exact behaviour of the observed humans and we have the "robotic simulations" which do so, (noise excepted). 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.

This is why it is necessary to analyse the human behaviour and to generalise it to problem solving strategies that can applied in any setting, and to other problems as well, as long as these strategies are meaningful for the new problem.

Given a relevance indicator (the class attribute), attribute construction and selection can be performed with the use of inductive algorithms to go from observables to primitives and from primitives to tactics. Once the tactics are defined, the "shapes" the strategies can have become apparent. A strategy "shape" can then be automatically filled with relevant numerical values by several means, including automated ones.

We believe that with the use of learning algorithms the whole process could be automated even more. In the future, given an adequate mean to acquire a set of observables, a robot might be able to learn a human strategy with very little human intervention.

References

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

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

[Billard and Hayes, 1999] Billard and Hayes. Drama, a connectionist architecture for control and learning in autonomous robots. *Adaptive Behaviour*, 7:35–64, 1999.

[Billard and Siegwart, 2004] A. Billard and R. Siegwart. Robot learning from demonstration. *Robotics and Autonomous Systems*, 47:65–67, 2004.

[Brooks and Flynn, 1989] R. Brooks and A. M. Flynn. Fast, cheap and out of control: a robot invasion of the solar system. *Journal of the British Interplanetary Society*, 42:478–485, 1989.

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

[Calinon *et al.*, 2007] S. Calinon, F. Guenter, and A. Billard. 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, 2007.

[Demiris and Hayes, 2001] Demiris and Hayes. *Imitation in animals and artifacts*. 2001.

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

[Felkin, 2008] M. Felkin. Learning by observation and induction the strategies of humans placed in a problem-solving context. *PhD Thesis*, 2008.

[Iemmi, 2005] Valentina Iemmi. Les apports des nouvelles technologies a la psychologie clinique: Les robots comme compagnons therapeutiques. 2005.

[Kraus, 1991] Henri Kraus. A formal theory of plan recognition and its implementation. *Reasoning About Plans, by J.F. Allen, H.A. Kautz, R.N. Pelavin, and J.D. Tennenberg*, pages 69–126, 1991.

[Murphy, 1977] L. B. Murphy. Responses of domestic fowl to novel food and objects. *Applied Animal Ethology*, 3:335–349, 1977.

[Quinlan, 1993] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[Quinlan, 1996] J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

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

[Skinner, 1922] M. P. Skinner. The pronghorn. *Journal of Mammalogy*, 3:82–106, 1922.

[Sloman, 2001] A. Sloman. Varieties of affect and the cogaff architecture schema. *Proceedings Symposium on Emotion, Cognition, and Affective Computing AISB'01 Convention*, pages 39–48, 2001.

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