Synthesis of Robot's Behaviors from few Examples

Louis Hugues Alexis Drogoul

Université Pierre et Marie Curie
MIRIAD - LIP6
PARIS - FRANCE
{*louis.hugues, alexis.drogoul*}*@lip6.fr*

Abstract

This paper addresses the problem of the acquisition of robot's behaviors for real environments. It insists on the interest of learning behaviors during robot's interaction with the environment under the control of a human tutor. The paper presents a learning model conceived to synthesize behaviors from a set of few examples and relying on a distributed representation of perception/action relations. The model is experienced on a real robot to learn a slalom task without giving any a priori information about the task or any element of the environment. The model exhibits properties that are well adapted to the interactive learning of concrete behaviors.

1 Introduction

The integration of autonomous mobile robots in real environments requires to conceive robot behaviors able to deal with uncompleted, un-precise and uncertain robots perceptions as well as partially unpredictable actions. Robots will have to be autonomous in un-modeled dynamic environments while behaving in ways useful to human users. The question is therefore *how can robots acquire those behaviors ?* Some behaviors can be explicitly programmed, but this requires an *explicit* description of the tasks and a model of the environment. Some behaviors can be learned using teleological methods such as reinforcement learning [11], or genetic algorithms [6]. This requires again, to define explicitly the behaviors by the intermediary of a reward or fitness function and to use a trial-and error scheme impossible to achieve in most environments.

From the human user point of view, a good way to define a behavior is to interact directly with the robot in the destination environment. A set of methods which could be grouped under the name of *Empirical Learning* are gaining interest in the literature. Those methods are Learning by demonstrations or from examples [1, 7, 12] , Imitation [9, 8, 3, 2, 4] or more classically Supervised Learning[14, 13]. Those works concentrate on the acquisition of actions sequence but they rely on predefined and simplified perceptions such as simple fixed shapes, fixed color or *ad-hoc* perceptual behaviors. Contrarily, our approach is to define a representation of a behavior being able to capture perceptions/relations without requiring *a priori* knowledge to model the perceptual space. We are particularly interested in finding a minimal representation of a behavior relying on very simple features. In the following sections we present the representation and the learning algorithm used to synthesize behaviors. The experimentations consisting in learning a slalom task are then described.

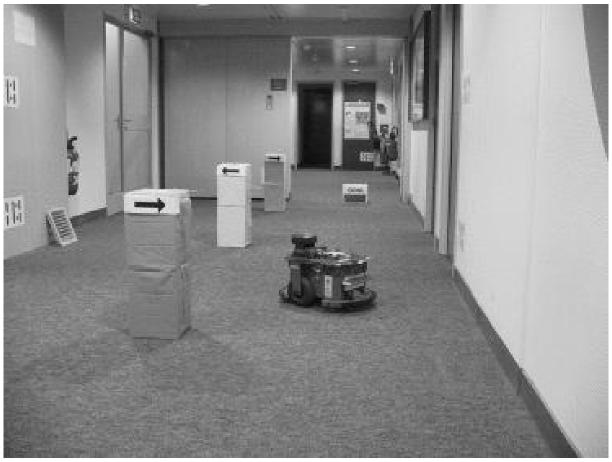

Figure 1: *The robot in the slalom task environment*

2 A Model for Behavior Learning

The framework of the model is a supervised learning scheme where the human tutor records several examples of a target behavior (ie: a docking manoeuvre, a complex succession of operations in a game or for person assistance ...) by remote controlling the mobile robot. The tutor shows several variants of the behavior from different postures. The purpose of the learning mechanism is therefore to produce a synthetic, robust and reusable behavior from those heterogeneous parts. If the tutor is not satisfied of the result he/she can add some new examples or remove some others. To achieve this the model relies upon three key points:

1. A collection of minimal perceptual features.
2. A learning procedure based on the sampling of the examples.
3. An ensemble method for autonomous behavior realization.

2.1 Input examples

The examples are sequences made of $\{X_t, Y_t\}$ couples where X_t is the video input image perceived by the robot at time t and Y_t is the control order coming from the tutor. The concatenation of all the examples gives the set F of frames, this set is the input of the learning algorithm.

$$F = \{\{X_1, Y_1\}, \{X_2, Y_2\}, ..., \{X_n, Y_n\}\} \qquad (1)$$

In practice X_t is a 40x30 pixels image and Y_t is a 2D action vector representing the two (differential) wheels velocities.

$$Y_t = \begin{bmatrix} left\ wheel\ velocity \\ right\ wheel\ velocity \end{bmatrix} \qquad (2)$$

2.2 Collection of Pixel-Based Features

A behavior B is modeled by a set of simple features which encodes the perception/action relation specific to the behavior. Those features are obtained by sampling the examples at the pixel level. The behavior representation has the following form:

$$\begin{aligned} B = & \\ & \{\{pixel1_1, pixel2_1, \overline{y}_1\}, \\ & \{pixel1_2, pixel2_2, \overline{y}_2\}, \\ & ..., \\ & \{pixel1_m, pixel2_m, \overline{y}_m\}\} \end{aligned} \qquad (3)$$

In each of those m features $pixel1$ and $pixel2$ denote two precise pixels in the image and $\overline{y}$ denotes an action vector. A pixel is defined by a triplet $\{xpos, ypos, class\}$, the class information is described in paragraph 2.6 and corresponds somehow to the pixel's color. The action vector $\overline{y}$ associated to each feature is the action to perform when the feature is detected.

The collection of features is *redundant* because several features can be detected in a given input image (as it is the case for the image in figure 2). This is the *averaged contribution of several detected features* which is used to determine the robot actions. This representation is robust to noise, occlusions, and minor environment changes due to its distributed and redundant nature. It permits to fuse several examples into a single structure because it remains at the pixel level and do not rely on macro-features.

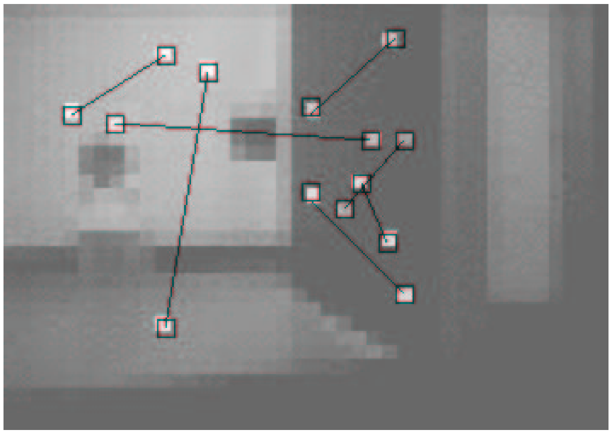

Figure 2: *Autonomous realization of behavior B. Features detected in current input image X are represented by linked pixels.*

As it can be seen in figure 2 a simple couple of pixels impose constraint onto the robot's position. The choice of two pixels has proven to be adapted in real experimentation, features made of three or more pixels impose too much constraint on the robot's posture and can rarely be matched , contrarily a feature made of a single pixel is not discriminative enough.

2.3 Learning Algorithm

The learning algorithm used to build B from the examples is a sampling procedure which receives F for input and outputs the collection B. It is made of three steps. The step 1 is used to evaluate the examples complexity and deduce the number of features needed based on the number of distinct pixels occurrences. The step 2 samples the examples by picking randomly features in the frames. The step 3 associates an action to each feature.

1. Count np the number of *distinct* pixels $\{xpos, ypos, class\}$, in F and set the number of features: $m = \alpha \times np$ (in experiments we used $\alpha = 0.2$).
2. For each of the m features pick a frame randomly in F, in this frame pick randomly two pixels: $pixel1$ and $pixel2$.
3. For each feature f, compute $\overline{y}_f$, the arithmetic mean of Y over the frames containing f.

In Step 2 we do not want to over-sample the less informative surfaces like large walls or carpet. Thus we constraint the sampling so as the density of the classes in the resulting collection B is uniform. The Steps 3 corresponds to the estimation of $\overline{y} = E[Y|pixel1, pixel2, F]$. The algorithm has a time complexity in $O(n)$ where n is the number of frames.

2.4 Autonomous realization of a behavior

To control the robot in real-time the behavior has to produce cyclically an effector vector Y from the input video image X. This is obtained by averaging the contributions of detected features.

1. From the current input image X, determine the subset $A_B(X)$ of features of B found in X.

2. Compute $\overline{Y}$ as the average of $\overline{y}$ over the elements of $A_B(X)$:

$$\overline{Y} = \frac{1}{|A_B(X)|} \sum_{f \in A_B(X)} \overline{y}_f. \qquad (4)$$

3. use $\overline{Y}$ for current wheels velocities.

This approach benefits of advantages which are common to Ensemble Methods [5] particularly the statistical determination of the solution. It is adapted to the frequent cases where the robot perceives its environment partially, occluded or even changed. A feed-forward neural network (ie: a multi layer perceptron trained by backpropagation) would take into account all the input images even in unseen situation , contrarily the collection of features reacts to already seen pixels and neglects other data.

2.5 Active perception

The model can provide a measure of the pertinence of the current visual perceptions for a given behavior B and this information can be used in real-time to improve the autonomous realization of the behavior. A *utility measure* $U_B(X)$ is defined by the number of features of B found in perception X (equation 5). $U_B(X)$ is low if the robot is in front of a totally unknown scene, contrarily $U_B(X)$ is high in front of a scene belonging to the learning examples. If the robot perceives a partial or shift image with respect to the examples, $U_B(X)$ has a medium value.

$$U_B(X) = |A_B(X)| \qquad (5)$$

Along the realization of a behavior a robot can be in situations where its perceptions are difficult to exploit. This happens for instance if someone passes near the robot or if it is in front of an unknown scene. In those situations the robot can determine that $U_B(X)$ is below a given threshold and it can actively search for a better posture before continuing to move. In the experiments this is done by rotating the camera and looking where the $U_B(X)$ measure reaches a local maximum. The learned material is therefore exploited by adding a a build-in scan/search schema. To exemplify this view, the plot of $U_B(X)$ in figure 4 has been obtained while the camera was quickly scanning the environment by doing a panoramic movement (see fig 3). $U_B(X)$ is maximal for the views which can be used by the behavior.

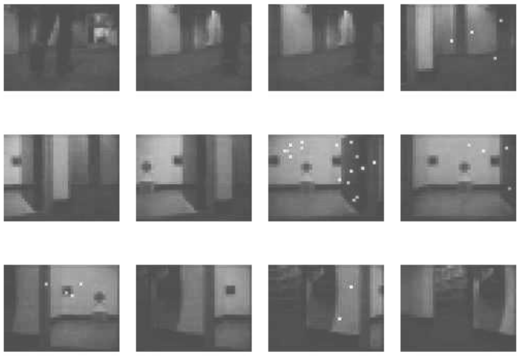

Figure 3: *A behavior B has been learned: entering a small room. This figure shows a camera panoramic movement of a corridor and room entrance. The light pixels correspond to detected features of B.Their are more features detected in front of the room entrance,indicating when B can be applied. $U_B(X)$ is reported in figure 4*

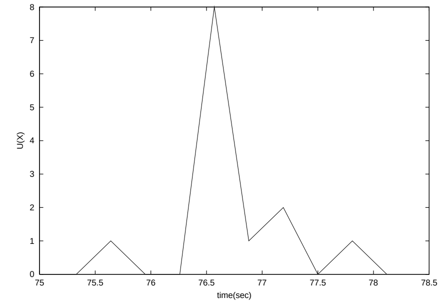

Figure 4: *$U_B(X)$ measured while the camera is performing a camera panoramic. Corresponds to the successive views of figure 3*

2.6 Class of a pixel

To remove noise and illuminancy impact we do not use directly the red, green, blue information of the video input. Instead we associate to a pixel a class which combines the *color* of the pixel and a *pattern* describing its immediate neighborhood. The color is an eight bits integer obtained by a HSV transform where Hue, saturation, value are respectively coded with 4, 3 and 1 bits. The pattern for a pixel is obtained by comparing its 5x5 pixels neighborhood to each of a set of 30 fixed patterns (figure 5). The 30 patterns have been obtained separately by training a Kohonen Self Organizing Feature Map (SOMF) [10] on a set of images coming from various places in our office. Finally the class of a pixel combine the two informations in a single expression equivalent to: $class = pattern \wedge color$. This coding has proven to provide stable information in several indoor environments. It exploits two complementary aspects of the image which are the color and local shape.

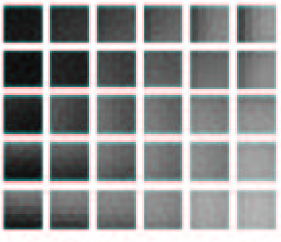

Figure 5: The 30 fixed patterns (5x5 pixels each) used to associate a class to pixel.

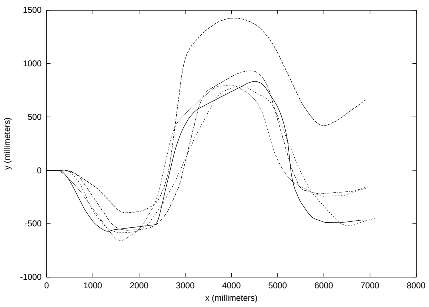

***Figure 6:** Trajectories of the robot for the five examples of the learning set. The figure is a plot of the odometer values, as usual it is subject to drift and is not used by the leaning model - The trajectories correspond to the five movie in figure 7.*

3 Experiments

For the experimentation we have used a mobile robot Pioneer 2dx equipped with a monoscopic color camera and an Pentium 200MHz on-board computer. Only video image and wheel control are used. During recording of examples the robot is teleoperated with a joystick via radio-ethernet. The video images are acquired at a rate of 3 images per second for the recording of examples as well as for autonomous behavior realization. Prior to training and realization phases the images are preprocessed as follow : pixelisation to reduce to a 40x30 image, normalization of each of the red, green, blue planes to limit illumination variations. Each of the two motored wheels are controlled by a precise value ranging from -600.0 to 600.0 mm/s.

We experiment the model to learn a slalom task the environment of which is shown in the figure 1. The robot has to slalom between three colored stakes and then approach a goal box. The robot does not react particularly to a specific color, color of the stakes and those other objects are equivalent to him. Most of the time it does not even see the stakes because it is turning around. To learn the task we recorded 5 examples movies (fig. 7). Each example is a possible variation of the slalom, their trajectories can be seen in the figure 6. The learning phase is fast and takes 65 seconds on a pentium II 300MHz. This can be compared to the much longer training phase of a feed-forward neural network which would have required much more examples anyway. For the evaluation all trials start from the start position in front of the first stake, a trial is considered successful if the robot follows correctly the path, do no touch the stakes and reaches the goal with a 20 cm error. The robot is considered to have achieved 2/3 of the task if it passes correctly the first two stakes. The results are recapitulated in figure 1 and correspond to the average over 30 successive trials. The robot has succeeded completely 16 times over 30 and has succeeded 2/3 of the task 25 times over 30.

	percentage of success
2/3 of the task	83%
complete task	53%

***Table 1:** Experimentation results over 30 trials.*

The behavior is reactive and the robot can be kidnaped and displace in the task environment, it then continues the task from this new point. If we place a chair to occlude partially its view the robot can continue its job, providing that we do not hide too much of its vision. The failures of the robot occur nearly always in the same place where the robot has only a grey wall in front of him, in this case few information can be used to differentiate perceptions.

4 Discussion

The results presented in the preceding paragraph show that it is possible to quickly learn a behavior starting with a few examples and minimal features, the result is a behavior which is anchored in a real environment and corresponds to the tutor intentions. The kind of behaviors learned can be helpful in a wide range of applications, the proposed model provides a way to ground components of a larger robotic application. What has been demonstrated is a practical feasabily. However the behavior still needs to be improved and this improvement can be envisaged by taking benefit of the following properties.

Manageability of the representation. The model provides a certain level of *intelligibility*. Each cell can be considered as a simple rule saying: *if pixel*($xpos_1, ypos_1, class_1$) $\wedge$ *pixel*($xpos_2, ypos_2, class_2$) *then perform action* $\tilde{y}$. A behavior is thus a large set of rules of this kind, it can therefore be interpreted in various ways to find, for instance, objects that influence the robot or situations that cause behavior failure. The simplicity of the representation enables off-line processing such as: fusion of several behaviors, compression or filtering of behaviors. We are currently experiencing the fusion of several behaviors learned in different places but all corresponding to the same abstract action (ie: dock to a power station). We therefore try to obtain genericity starting from the concrete situations and not by a generalization mechanism.

Interactive Learning. A first scheme for interactive learning is the possibility to add or remove example and judge immediately how it affects the behavior. It is possible here due sufficiently fast learning and the possibility to encode different examples. The representation of the perception/action relation also permits to conceive online mechanisms to adapt and tune the behavior on-line under the control of a tutor, particularly in case of contradictory examples. It is possible to identify the cells which cause a particular action and reward cells positively or negatively according to some tutor's request. Under the tutor control it is possible to over-sample or sub-sample some parts of the environment.

Noise, occlusions. The model is well suited to resist to partial occlusions of the visual field, which often happen in a dynamic environment. Due to collective structure, a behavior can still decides the right action even if the environment features are partially visible (if someone passes in front of the robot for instance). The model can also deal with noise: the incorrect perception of a pixel doesn't affect the whole behavior and is compensated by other cells.

5 Conclusion and future work

We have presented a model which can be considered as a distributed stochastic representation of a behavior from a Machine Learning point of view, or as a multi-agents representation of a behavior from a Multi-Agents point of view. This model exhibits properties suited to interactive behavior learning under the conduct of a human. Those properties are: a minimal and extendable model, no need of a priori information, a fast learning speed and a realistic use in noisy and occluded environment. Those properties are difficult to found in classical learning methods such as feed forward neural networks. They cannot be found either in methods relying on high-level descriptions of the environment. The learning phase described above can be though of as a *bootstrap phase* where a proto-behavior is learned. In our on-going work we are now working on the possibility to adapt the proto-behavior on-line under the control of the human tutor.

To conclude we believe that teaching behaviors using schemes such as examples based learning, imitation, learning from demonstrations, all from an empirical point of view, constitute one of the most realistic approach to human use of agents and robots. We think that those approaches will benefit to take into account the intrinsic nature of the perceptions (occluded, partial, noisy, uncertain ...) using a distributed model as it is the case in biological systems, and not to rely onto an abstract idealized vision of a world made of few well-identified objects. This should have consequences onto the learning mechanisms themselves as well as on the structure of behaviors.

References

[1] ATKESON, G., AND SCHALL, S. Robot learning from demonstration. In *Proceedings of the 1997 international Conference on Machine Learning (ICML'97)* (1997), J. Fisher, Ed.

[2] DAUTENHAHN, K. Getting to know each other – artificial social intelligence for autonomous robots. *Robotics and Autonomous Systems 16* (1995).

[3] DEMIRIS, J., AND HAYES, G. Imitative learning mechanisms in robots and humans. In *Proceedings . of the 5th European Workshop on Learning Robot* (1996), V. Klingspor, Ed.

[4] DEMIRIS, J., AND MATARIC, M. Perceptuo-motor primitives in imitation. In *Autonomous Agents'98 Workshop on Agents In Interaction - Acquiring Competence Through Imitation* (1998).

[5] DIETTERICH, T. Ensemble methods in machine learning. In *First International Workshop on Multi Classifier Systems* (2000), J. Kittker and F. Roli, Eds., NewYork: Springer Verlag.

[6] FLOREANO, D., AND MONDADA, F. Automatic creation of an autonomous agent: Genetic evolution of a neural-network driven robot. In *Proceedings of the Conference on Simulation of Adaptive Behavior* (1994).

[7] FRIEDRICH, H., AND DILLMAN, R. Robot programming based on single demonstration and use intention. In *Third European Workshop on Learning Robots - ECML'95* (1995).

[8] GAUSSIER, P., MOGA, S., BANQUET, J., AND QUOY, M. From perception-action loops to imitation processes: A bottom-up approach of learning by imitation. *Applied Artificial Intelligence 1*, 7 (1997).

[9] HAYES, G., AND DEMIRIS, J. A robot controller using learning by imitation. In *Proceedings of the 2nd International Symposium on Intelligent Robotic System* (1994).

[10] KOHONEN, T. *Self-Organizing Maps*. Springer-Verlag, New York, 1997.

[11] MAHADEVAN, S., AND CONNELL, J. Automatic programming of behavior-based robots using reinforcement learning. *Artificial Intelligence 55* (1992), 311–365.

[12] NICOLESCU, M., AND MATARIC, M. Learning and interacting in human-robot domains. *IEEE Transactions on Systems, Man, Cybernetics 31*, 5 (2001).

[13] POMERLAU, D. Knowledge-based training of artificial neural networks for autonomous robot driving. In *Robot Learning* (1993), J.Connell and S. Mahadevan, Eds.

[14] POMERLAU, D. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation 15*, 88 (1997).

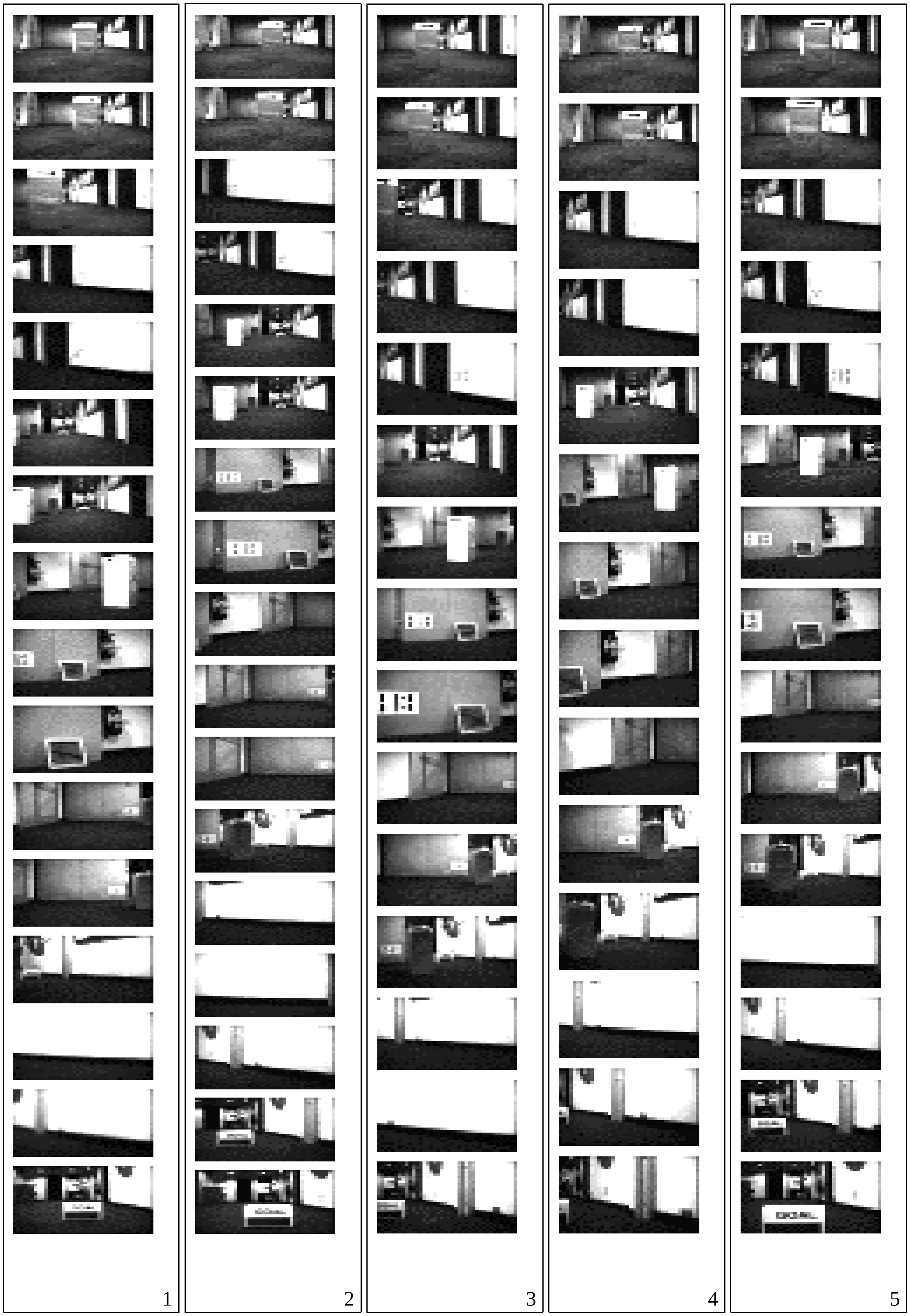

Figure 7: *The successive video frames of the five examples used to learn the slalom task (only 1 frame over 10 are shown)*