Pixel-based Behavior Learning

Louis Hugues [1] and **Alexis Drogoul** [2]

Abstract. In this paper we address the problem of learning behaviors for autonomous mobile robots. We particularly focus on methods which enable a human user to train a robot in its real destination environment without giving an a-priori model. Using complex visual input typical of real situations in office environments we show that very simple visual features can be used to represent the perception/action relation specific to a given behavior. From this point we propose a learning model relying on a statistical collection of two-pixels features for representing a behavior. We then present the experiments made on a real robot and propose extensions of the model for active-perception and behavior selection.

1 INTRODUCTION

Robots of our near future will be situated in the real world [3, 15] and most will be in relation with humans [6]. They will have to behave in ways useful to human users while being autonomous in unmodeled dynamic environments. How can robots acquire those behaviors? This question interests robotics but also any fields where computers have to perceive and act in the real world. Some of the robots' behaviors can be explicitly programmed, but this requires an explicit description of the tasks and a model of the environment were invariants can be distinguished (invariants such as distance to wall , position of an object, etc...). Some behaviors can be learned using teleological methods such as reinforcement learning [16], or genetic algorithms [10]. 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. The problem of the explicit definition of the behavior is displaced but still has to be faced by an expert designer.

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. Several methods have been proposed in this direction : Learning by demonstrations or from examples, Memory-based learning [18, 1, 11] , Imitation [13, 12, 2, 7] or Supervised Learning [19]. Those approaches focus on the learning of complex action sequences but they rely on simple predefined and constrained percepts - most use well known shapes, centroids of simple color objects and minimal environments.

In our thinking, the perception and its intrinsic complexity should have a structural impact onto a behavior learning model. Perception is not a pattern recognition subproblem which can be studied separately of the problem of action learning in an environment. Perceptual features should have the properties of:

1. Compliance with realistic environments.
2. Robustness to noise induced by sensors or environmental conditions.
3. Robust to occlusions occurring frequently when a person, an object or even self occludes the visual field.
4. Postures distinction to discriminates between robots viewpoints , (ie: approaching an object from left or from center left, looking at a very close object...).
5. Tractability so as to obtain features at a low computational cost along real-time processing.
6. Support of the perception/relation established along a behavior.
7. Extendibility to allow behavior adaptation to environment changes.

[1] LIP6 - Université Pierre et Marie Curie, Paris, France, email: louis.hugues@lip6.fr
[2] LIP6 - Université Pierre et Marie Curie, Paris, France, email: alexis.drogoul@lip6.fr

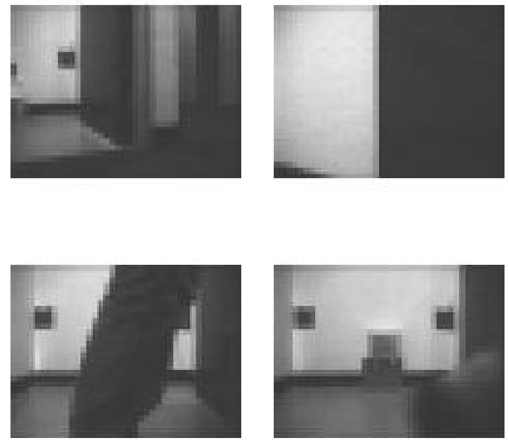


Figure 1. Typical robot vision in a structured environment - The images are obtained from the robot monoscopic color camera - they can be occluded for instance by persons passing near the robot.

Since the work of D. Marr [17], the Computer Vision field has provided a lot of general methods, ranging from recognition [4], image classification and retrieval by Content [20, 5] to three dimensional vision [9]. However those methods are not conceived in the perspective of behavior learning and not do not fit the above properties.

1.1 Selectivity of n-pixels features

Surprisingly very simple visual features made of 2, 3 or 4 pixels can be used to discriminates complex robot visual perceptions. This can be shown experimentally with the following $Select(p)$ test which measures the discriminating power of a feature made of $p = 1, 2, 3, ...$ distinct pixels - pixel defined by a tuple $(x, y, color)$.

The selectivity test is performed on a set S of $n = 1200$ images similar to fig. 1 and obtained along robot wandering sessions.
For p-pixels select(p) is computed as follow:

1. Pick randomly an image of S.
2. In the image pick p pixels randomly.
3. Count the number r of images of S containing the complete feature (with the p pixels at same position with same color).
4. Report $Select(p)$ as the percentage of successful identification with p pixels..

$$Select(p) = \left[1 - \frac{r-1}{n-1}\right] \times 100 \quad (1)$$

The results are averaged over 100 trials to obtain a better statistical significance. The plot of $E[Select(p)]$ in figure 2 shows that, *on an average*, 4 pixels are sufficient to discriminate between several visual perceptions.

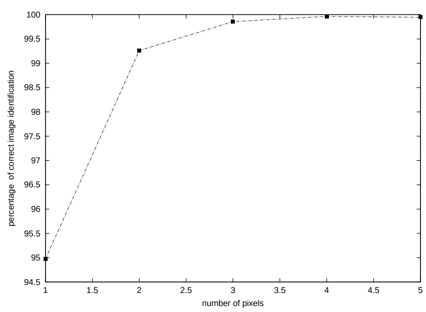


Figure 2. $E[Select(p)]$: averaged percentage of successful image identification. A simple p-pixels feature (picked randomly) is used to identify an image in a set of 1200 images collected by the mobile robot.

2 A MODEL FOR BEHAVIOR LEARNING

We can use the p-pixel features to construct a model for robot learning by examples. In this supervised learning scheme, 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 behavior from those heterogeneous parts.

2.1 Input examples

The examples are sequences made of $\{X_t, Y_t\}$ couples where X_t is the perceived video input image at time t and Y_t the corresponding action vector. In practice X is a 40x30 pixels image and Y is a 2D vector made of right and left wheels velocities.

$$Y = \begin{bmatrix} left\ velocity \\ right\ velocity \end{bmatrix} \quad (2)$$

The concatenation of all the examples gives the set F of frames which are used as input of the learning algorithm.

$$F = \{\{X_1, Y_1\}, \{X_2, Y_2\}, \ldots \{X_n, Y_n\}\} \quad (3)$$

2.2 Collection of Two-Pixels Features

A behavior is modeled by a redundant collection of two-pixels features (see eq. 4) which encodes the perception/action relation specific to the behavior. In this collection each feature is associated to an action vector $\overline{y}$ giving a statistical representation of the action to perform when the feature is detected. The choice of two-pixels rather than three or more is a compromise between a sufficient discriminating power and the possibility to assimilate similar images. The collection is redundant, it contains more features than the number of example frames, thus several features can be detected in a given image. This is the averaged contribution of several detected features which is used to determine the robot actions. The collection has the following form:

$$B = \{\{pixel1_1, pixel2_1, \overline{y}_1\}, \ldots, \{pixel1_m, pixel2_m, \overline{y}_m\}\} \quad (4)$$

Where $pixel$ denotes a $(x, y, color)$ t-tuple and $\overline{y}$ denotes an action vector. This representation is robust to noise, occlusions, and minor environment changes due to its distributed and redundant nature, it also permits to fuse several examples into a single structure because it remains at the pixel level.

2.3 Pixel-Based Learning Algorithm

The learning algorithm used to build B from the examples is derived from $select(p)$ and is made of three steps. The step 1 is used to evaluate the examples complexity and deduce the number of features needed. The step 2 samples the examples frames to obtain the features. The step 3 associated a statistical representation of the action to each feature.

1. Count np the number of distinct tuple $(x, y, color)$ in F and set $m = \alpha \times np$ (in experiments $\alpha = 0.2$).
2. For each of m features:
   (a) Pick a frame randomly in F.
   (b) In this frame pick randomly $pixel1$ 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, carpet... and we sample so as the resulting density of the present pixel classes in the collection B is uniform. The algorithm can be rearranged to obtain a time-complexity in $O(n)$ where n is the number of frames.

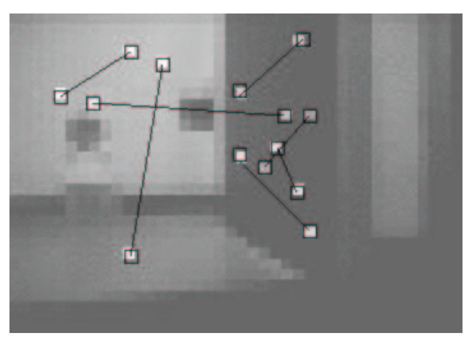


Figure 3. Autonomous realization of behavior B. Each feature detected in current input image X is represented by two linked pixels.

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 input image X, determine the subset $A(X)$ of features of B found in X.

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

$$\overline{Y} = \frac{1}{|A(X)|} \sum_{f \in A(X)} \overline{y}_f. \tag{5}$$

3. set $Y \leftarrow \overline{Y}$.

This approach benefits of advantages which are common to Ensemble Methods [8] 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.The autonomous behavior is reactive and does not need explicit reference to time.

2.5 Utility measure

Beside the action response $\overline{Y}$, the model can provide a simple measure of the pertinence of the current visual perceptions for a given behavior B. 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 6). $U_B(X)$ 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.

$$U_B(X) = |A_B(X)| \tag{6}$$

This utility measure can be used for *active perception* and *behavior selection*.

2.6 Active perception

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 the utility measure $U_B(X)$ is below a given threshold and it can actively search for a better posture before continuing to move. This can be done by rotating the robot itself or the camera and looking where the $U_B(X)$ measure reaches a local maximum. This approaches associates to the learned material a build-in scan/search schema. The plot of $U_B(X)$ in figure 5 has been obtained while the camera was quickly scanning the environment by doing a panoramic movement (see fig 4). $U_B(X)$ is maximal for the views which can be used by the behavior.

2.7 Behavior selection

A real application needs to combine several behaviors and it must be able to select one among several by considering the current context, the goals to be reached and the robot's internal state. The $U(X)$ provides a useful information to determine which behaviors could be activated in a given context and compare their chance of success. As shown in figure 6, for two distinct behaviors $B1$ and $B2$ respectively learned in environments $E1$ and $E2$, the measures $U_{B1}(X)$ and $U_{B2}(X)$ obtained during wandering in environment $E1$ indicates that $B1$ is more appropriated than $B2$. An action selection scheme can use this situated information to determine the right behavior.

Figure 4. A behavior B has been learned, first consisting 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 5

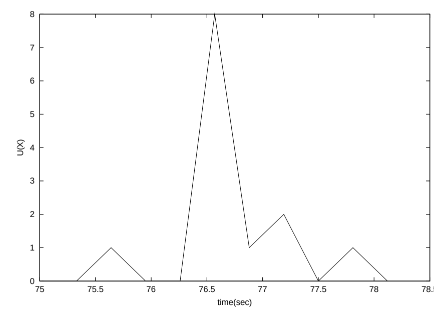


Figure 5. $U_B(X)$ measured while the camera is performing a camera panoramic. Corresponds to the successive views of figure 4

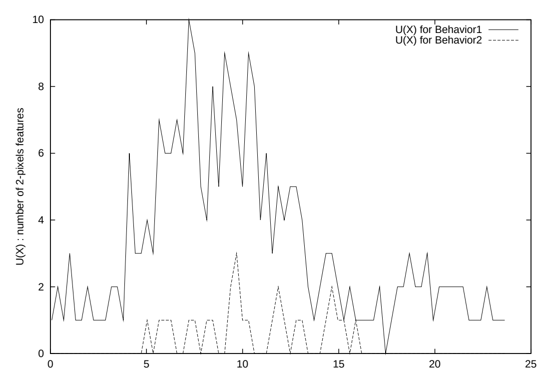


Figure 6. Comparison of the U(X) measures for two behaviors $B1$ and $B2$, when the robot wanders in the environment where $B1$ has been learned.Clearly U(X) reports than $B1$ can be realized with more success than $B2$. A maximum at $t = 9s$ can be exploited by an action selection algorithm to launch $B1$.

3 EXPERIMENTS

3.1 Experimental platform

For the experimentation we use a mobile robot Pioneer 2dx (fig. 10) equipped with a monoscopic color camera and an Pentium 200MHz on-board computer. During recording of examples the robot is tele-operated 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. The images are reduced to a resolution of 40x30 and normalized. The two motored wheels are controlled separately with a precise value ranging from -600.0 to 600.0 mm/s.

The color information associated to each pixel is not directly the red,green, blue information obtained from the video input. Instead to reduce noise the color combines the hue,saturation and value (HSV) of the pixel and a class describing the pixel's immediate neighborhood. The hue, saturation and value of the pixel are obtained from a classical HSV transform and are respectiveley coded with 4, 3 and 1 bits. The class of a pixel is obtained by comparing its 5x5 pixels neighborhood to each of a set of 30 fixed patterns (figure 7). The 30 patterns have been obtained separately by training a Kohonen Self Organizing Feature Map (SOMF) [14] on a set of images coming from various places in our office. Finally the color of a pixel is an expression equivalent to: $color = hue \wedge saturation \wedge value \wedge class$. This coding has proven to provide in practice stable information in various indoor environments.

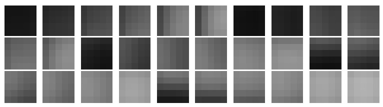


Figure 7. The 30 fixed patterns (5x5 pixels each) used to associate a class to pixel.

3.2 Experiment without active perception

The first behavior experimented is chosen to mimic a docking maneuver: entering in a small place and approaching a device symbolized by a color box. To learn a behavior we record 7 examples movies each with a duration of a few tenth of seconds. The number of features generated for such behavior is approximately 20.000. A first observation is that the learning phase is very fast, typically 20 seconds on a pentium II 300MHz wich is much faster than a neural networks training phase on similar data. It be can asked whether the representation used is able to encode the examples, or say differently if it is able to learn the regression function $Y = f(X)$ compatible with the given data. The figure 8 shows the value of the left wheel effector recorded along an example and the response of the behavior to the same succession of images. The response fits the example despite that the behavior also encodes 7 other examples. This is confirmed if we compare the statistical distribution of actions in the examples to the statistical distribution of actions in the corresponding behavior (supported by $\overline{y}$ values).

What are the model's performances while realizing the behavior in real situations ? For a behavior consisting in negotiating the entrance in a small room and then approach a device and slow down when situated near the device, the robot performed well in 45 percents of the cases. The cases where it failed where mainly due to its confrontation to unknown scenes. However this can be sensibly improved with an active perception mechanism described below.

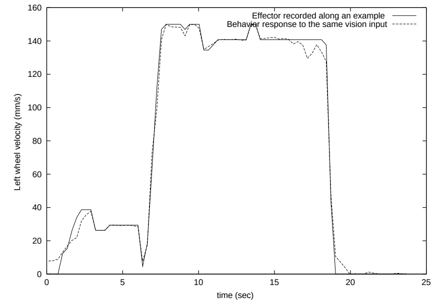
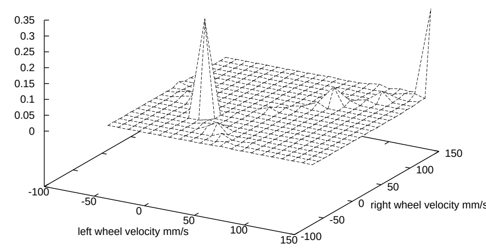
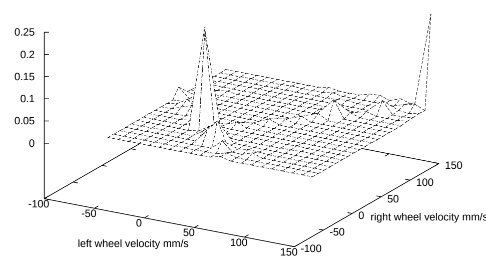


Figure 8. Comparison of actions recorded along an example to the response of a behavior to the same visual input (only left wheel velocity is shown)



a)



b)

Figure 9. 2-D histograms representing the statistical distribution of actions in a set of examples (a). It can be compared to the statistical distribution of 2d vector $\overline{y}$ in the generated behavior(b).

3.3 Experiment with active perception

With the active perception improvement it is now possible to reproduce a complex task such as a slalom. In the slalom task (fig. 10) the robot has to slalom between three stakes and then approach a goal box.To learn the task we recorded 5 examples each example being a possible variation of the slalom. 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

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

Table 1. Slalom Task results over 30 trials with active perception

Figure 10. The slalom task

two stakes. The results are recapitulated in table 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.

4 CONCLUSION

In this paper we have proposed a novel approach for the learning of robot's behaviors relying on the use of minimal pixels features. Our model is conceived so as to capture the perception/action relation which supports a behavior demonstrated by a tutor in a real environment. Because it is fast, requires few examples and need no a-priori information it is well suited to our central objective which is the on-line training of robots by non human users. The representation used for the behaviors allow various operations which permit to adapt the behavior to the environment, it can be extended or shrink-ed, filtered or fused and is sufficiently minimal to fit a lot of algorithms. The intelligibility of the representation allows to analyze a-posteriori the behavior data. This last property is particularly useful for providing an understanding of robots' actions to the human users. Finally in our thinking the pixel-based approach is not limited to robot learning but could be transposed for problems involving 2-D perceptual devices. Our future work is oriented toward online interactive adaptation of the behaviors after a learning phase.

REFERENCES

[1] G. Atkeson and S. Schall, 'Robot learning from demonstration', in *Proceedings of the 1997 international Conference on Machine Learning (ICML'97)*, ed., J.D.H Fisher, (1997).

[2] P. Bakker and Y. Kuniyoshi, 'Robot see, robot do : An overview of robot imitation', in *Aisb 96 Workshop on learning in Robots and Animals*, (1996).

[3] R.A. Brooks, 'Intelligence without reason', Technical Report 1293, MIT AI Lab, (1991).

[4] R. Brunelli and T. Poggio, 'Face recognition: Features versus templates', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**(10), 1042–1052, (October 1993).

[5] O. Chapelle, P. Haffner, and V. N. Vapnik, 'Support vector machines for histogram-based image classification', *IEEE Transactions on Neural Networks*, **10**(5), (September 1999).

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

[7] J. Demiris and M. Mataric, 'Perceptuo-motor primitives in imitation', in *Autonomous Agents'98 Workshop on Agents In Interaction - Acquiring Competence Through Imitation*, (1998).

[8] T.G. Dietterich, 'Ensemble methods in machine learning', in *First International Workshop on Multi Classifier Systems*, eds., J. Kittker and F. Roli. NewYork: Springer Verlag, (2000).

[9] O. Faugeras, *Three-Dimensional Computer Vision, a Geometric Viewpoint*, 1993.

[10] D. Floreano and F. Mondada, '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).

[11] H. Friedrich and R. Dillman, 'Robot programming based on single demonstration and use intention', in *Third European Workshop on Learning Robots - ECML'95*, (1995).

[12] P. Gaussier, S. Moga, J.P Banquet, and M. Quoy, 'From perception-action loops to imitation processes: A bottom-up approach of learning by imitation', *Applied Artificial Intelligence*, **1**(7), (1997).

[13] G. Hayes and J. Demiris, 'A robot controller using learning by imitation', in *Proceedings of the 2nd International Symposium on Intelligent Robotic System*, (1994).

[14] T. Kohonen, *Self-Organizing Maps*, Springer-Verlag, New York, 1997.

[15] P. Maes, 'Situated agents can have goals', *Journal of Robotics and Autonomous Systems*, **6**, 49–70, (1990).

[16] S. Mahadevan and J. Connell, 'Automatic programming of behavior-based robots using reinforcement learning', *Artificial Intelligence*, **55**, 311–365, (1992).

[17] D. Marr, *Vision : a Computational Investigation into Human Representation and Processing of Visual Information*, W.H. Freeman and Company , NY 26-61, 1982.

[18] A. Moore, C. Atkeson, and S. Schaal, 'Memory-based learning for control', Technical Report CMU-RI-TR-95-18, CMU Robotics Institute, (1995).

[19] D.A Pomerlau, 'Knowledge-based training of artificial neural networks for autonomous robot driving', in *Robot Learning*, eds., J.Connell and S. Mahadevan, (1993).

[20] M.J. Swain and D.H Ballard, 'Color indexing', *International Journal of Computer Vision*, **7**(1), (1991).