Proceedings of the 1999 IEEE
International Conference on Robotics & Automation
Detroit, Michigan • May 1999

Interactive Task Training of a Mobile Robot through Human Gesture Recognition

Paul E. Rybski, Richard M. Voyles
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
{rybski,voyles}@cs.umn.edu

Abstract

This paper describes a demonstration-based programming system in which a mobile robot observes the actions of a human performing a multi-step task. From these observations, the robot determines which of its pre-learned capabilities are required to replicate the task and in what sequence they must be ordered. The focus of this paper is on the Hidden Markov Model method used to learn and classify the actions as "gestures". A preliminary system demonstration is also described in which the robot observes the human performing a block distribution task. During the demonstration, the robot actively follows the demonstrator to maintain its vantage point and to infer spatial relationships.

1 Introduction

Gesture-based programming [10], or programming by demonstration, is a powerful tool which can be used to impart abstract knowledge about a task to a robotic system in an extremely short amount of time. In this method of training, a task expert, such as a human (or another robot), does the actions necessary to complete a task, or gestures in such a way as to impart symbolic knowledge about the task.

The primary benefit of this method is that the trainer does not have to provide the robot with an exact model of all of the actions necessary to accomplish its goal. All the trainer needs do is present the parameters of these actions to the robot. Such parameters may include what kinds of objects to affect by the action, where the robot should be oriented while executing the action, and so forth.

In this paper, a programming by demonstration system is described. This system, implemented on a small mobile robot, makes use of the robot's vision system to analyze and classify particular actions that the human performs. These actions are classified using a Hidden Markov Model (HMM) [8] representation. HMMs are used because they are capable of identifying a particular decision model out of a seemingly random set of observable symbols. This approach is extremely powerful because it allows one to determine the underlying intention behind an action which may normally appear ambiguous.

2 Related Work

A large amount of research has been devoted to the gesture-based or demonstration-based programming paradigm. Voyles and Khosla [11] has developed a system of gesture-based programming based on a multi-agent model of "encapsulated expertise." Robot to robot action recognition and cooperation has been successfully accomplished using a stereo vision system by Kuniyoshi [5]. Bakker and Kuniyoshi [2] extended this to "learning by imitation" in which a robot learns its behaviors by observing the behaviors of another robot. Pook and Ballard [7] have generated a working system which learns the parameters for teleoperated manipulations. Zhu [16] made use of HMMs for the recognition and avoidance of obstacles for dynamic path planning of a mobile robot. Lee and Xu [6] built a gesture-based programming system in which HMMs were used to classify sign-language gestures from a Cyberglove. HMMs have also been used by Yang et al. [14] for learning the mapping from position trajectory in Cartesian and joint space as well as learning a velocity trajectory in Cartesian space from the teleoperated control of a manipulator. Another example of how HMMs can be used to classify continuous gestures and is in written symbol recognition [15].

HMMs have been used quite successfully in purely vision-based gesture recognition systems as well. Starner and Pentland have developed a system for rec-

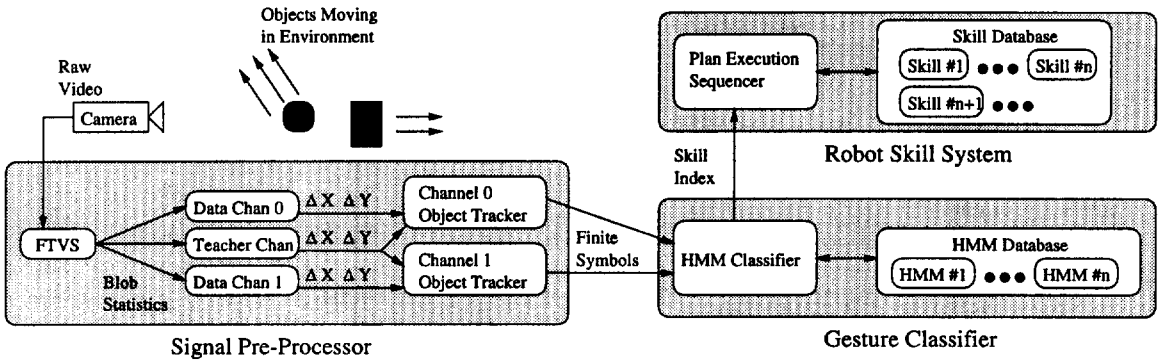

Figure 1: Architecture of the Demonstration-Based Programming System

ognizing American Sign Language (ASL) symbols [9]. Wilson and Bobick developed a system for classifying human motions using a variation on the standard HMM framework [12]. Brand et al recognize *Tai Chi* motions with another variant on the standard HMM framework [3].

There are many others working on gesture recognition with HMMs, but these contributed most to our inspiration for this paper. Our HMM is described in sections 3 and 4.2.

3 Hidden Markov Models

Hidden Markov models are used to model the underlying processes of a system whose inner workings cannot be completely observed. This is a useful method for determining the underlying processes behind the individual components of a gesture. A fundamental assumption that can be made about human gestures is that simply observing a gesture is not sufficient to extract the underlying structure behind it [15]. The same gesture may appear to a naive system to be different when repeated by different people under different circumstances. A HMM attempts to classify the underlying structure of the gesture and correlate it with the actual observed input.

Gestures or speech (or any other kind of signal continuous signal) can be discretized and represented as single-dimensional strings of observation symbols, $O = (o_1, o_2, \ldots, o_n)$. An algorithm called the Forward-Backward algorithm [8] is used to determine the likelihood that a given HMM, λ, produced a string of observed symbols. In order to adjust the parameters of an HMM to recognize a particular class of observation symbols, an algorithm known as Baum-Welch is used [8].

4 Demonstration-Based Programming

The demonstration-based programming system described in this paper consists of three sections, as shown in Figure 1. The first part is a signal preprocessor, which filters the raw sensor data into a form that is usable by the rest of the system. The second part is the gesture classifier which uses a HMM representation of gestures to recognize those made by the teaching human. The third part is the robotic skill system which contains all of the sensory-motor skills necessary for the robot to interact with its environment.

This system is implemented on an RWI Pioneer 1 [1] mobile robot outfitted with a Newton Labs FastTrack Color Vision System (FTVS) [13]. All software is written in C++ using the Saphira 6.1f API [4] running under Linux. The vision system performs color segmentation on the image, given user-defined parameters. The FTVS has three separate data channels which it can use to track different colors. Regions in the image which correspond to these colors are analyzed and statistics about largest single blob in the image are computed at 60Hz. One channel is defined explicitly for the teacher color. The other two channels are defined as "data" channels which colors of objects that the robot can manipulate are stored in.

4.1 Signal Pre-Processor

Several steps are necessary to compress the raw visual information into a form that the robot can understand and immediately act upon. The FTVS returns statistics about each of the three colored blobs that it tracks. These statistics include center of mass, area and perimeter of a bounding box surrounding the blob of color. These statistics are passed into data modules which discretize the sensor data at 10Hz. This first-

pass discretization consists of reporting ΔX and ΔY of the center of mass for each channel from one time step to another.

Two object tracking modules analyze the values of ΔX and ΔY coming from the previous step and determines whether the motions of the blobs in the image frame correspond to gesture segments. Gestures are represented as sequences of symbols which describe the motion of an object through time and space. If an object is seen to move, this movement is classified as either a horizontal or a vertical displacement. If the relative positions of the blobs in the teacher channel and a data channel changes, this movement is classified as an increase or decrease in relative proximity of those two channels. In either case, these symbols are generated and concatenated into a single-dimensional stream and are passed into the gesture classifier.

4.2 Gesture Classifier

Each gesture that the robot must recognize is represented as a unique HMM. When a new gesture is to be learned by the robot, a new HMM representation must be created and trained on sample gesture data. The human teacher provides a data set of sample gestures that is used by the Baum-Welch algorithm to train the new HMM. Once trained, this HMM is loaded into a database and is ready for use.

When the human performs a gesture for the robot, the strings of symbols, $O = (o_1, o_2, \ldots, o_n)$, generated by both the object tracking modules are fed into the HMM classifier. In order to classify this gesture, the value of $P(O|\lambda_i)$ must be generated for each HMM in the database. The Forward-Backward algorithm is applied to calculate the likelihood for each HMM. Once all the values of $P(O|\lambda_i)$ have been calculated, a confidence measure (similar to Lee & Xu [6]) is calculated for each λ_i. This confidence measure is defined as:

$$C_i = \sum_{k \neq i} \frac{\log(P(O|\lambda_k))}{\log(P(O|\lambda_i))}.$$

Finally, a HMM is only chosen if $C_j > \sum_{k \neq j} C_k$. Under this restriction, the "best" HMM in this winner-take-all strategy must produce a value of $P(O|\lambda_i)$ that is significantly better than all of the others. If no such HMM can generate this confidence, the system reports that it cannot classify the gesture as it is too ambiguous.

There are two data channels feeding their observed symbol streams into the gesture classifier. All of the gestures in the database are channel-independent, so it is possible that a HMM could be chosen for one string of observed symbols and another HMM could be chosen for the other string. In this case, the HMM with the highest value of C_j is chosen and that gesture (in the appropriate channel) is selected. The system is currently not able to classify two gestures occurring simultaneously.

The gestures that the robot is programmed to recognize are the following:

- Move Towards Object
- Move Away from Object
- Drop Object
- Grab Object

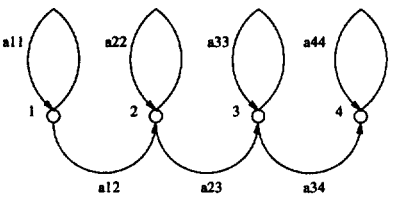

Figure 2: Bakis (left-right) HMM, with labeled states and state transitions.

In order to determine the proper topology of the HMM (how many states are necessary) to appropriately classify gestures, several different topologies of HMM were tried. For the gestures that were to be analyzed, the best number of states (in terms of expressiveness and generality) was found to be four. Thus, for the implementation of this system, each HMM had four states of which each state could generate six observable symbols. The topology of the HMMs was also kept reasonably simple. For the recognition of the gestures, a simple Bakis (left-right) Hidden Markov Model is used, as shown in Figure 2. As the model changes state, moving from left to right, the states to the left become inaccessible. This particular structure of HMM is used extensively in the speech-recognition community [8] because of its ability to classify time-dependent sequences of symbols. Speech data is assumed to be time-dependent and non-cyclic and this assumption works well for gesture classification. The following additional restriction is placed on the state transitions for this model: $a_{ij} = 0$ if $j < i$ or $j > (i + 1)$. This means that each state in the Bakis models used for this data can only transition to itself or to the state immediately to the right.

4.3 Robot Skill System

The final part of the demonstration-based programming system is the database of skills that the robot is

programmed with. A skill is a sensor-motor primitive that allows the robot to interact with its environment. Without this basic level of competence, the robot is unable to do any useful work.

All of the gestures that the robot knows how to recognize in its HMM database have a corresponding skill associated with them. When a robot recognizes a gesture, it determines the skill that corresponds to that gesture and records the index of that skill as well as the Cartesian coordinates of where it was when it saw that gesture in its plan execution sequencer. When the robot has learned the task (i.e. the human has stopped demonstrating), the robot executes each action stored in its plan execution sequencer in the order that it saw them.

All of the known gestures have a corresponding skill associated with them. However, not all skills have a corresponding gesture. The mapping between the robot's gestures and some of its skills is shown in Table 1. The skills that do not have an associated gesture are generally used as part of the training process or are used for assisting the robot as it moves about the environment on its own. The complete list of skills is:

- **Approach:** This skill uses a closed-loop control routine to move about in its immediate location and search for an object of an appropriate color. Once one is found, the robot visually servos near to the object but not close enough to disturb it.
- **Retreat:** This skill moves the robot away from an object so that it will not disturb or push it inadvertently when it travels to a different location.
- **Grasp:** This skill allows the robot to maneuver itself close enough to the object and then use a open-loop control routine to grasp the object as it moved out of range of the camera (which happened when the object was within 6 inches of the robot).
- **Release:** This skill is simply the inverse of the grab object skill.
- **Follow:** This skill is used exclusively when the robot is being trained by the task expert. When the robot comes within a meter of the teacher, it stops and waits until the teacher gestures or moves again.
- **Travel:** This skill moves the robot from one location (stored in Cartesian coordinates) to another location.

Gesture	Skill
Move Towards Object	Approach
Move Away from Object	Retreat
Drop Object	Grasp
Grab Object	Release

Table 1: The mapping from gesture to skills

5 Experiments

5.1 Gesture Recognition

Each HMM in the gesture classification database is trained with 25 sample gestures of a particular type. To test the classification system, 100 additional test samples of each kind of gesture are obtained. Each sample is fed into the HMM classifier, and the values for $P(O|\lambda_i)$, and C_i are computed for each. The results are shown in Table 2. In the second column, a value of less than 100% in the $P(O|\lambda_i)$ column indicates that the system could potentially mis-classify gestures if the classification was accomplished using likelihood calculations alone. The values in the third column represent the percentage rate of how many times the system was confident of its classification. A low value here would mean that the system finds that particular gesture too ambiguous and would elect not to classify it all instead of risking a misclassification. No mis-classifications occurred for this initial experiment.

To illustrate an example of the amount of variation between gestures of the same type, Figure 3 shows the likelihood values from testing 100 different instances of a Grab Object gesture. The four connect line graphs represent all four of the HMMs stored in the system. The top-most set of points (denoted by '+' symbols) represents the likelihood returned from the HMM trained to recognize the Grab Object gesture. The other three set of points represent the likelihood returns from the other gestures. According to Table 2, every value for $P(O|\lambda_i)$ correctly classifies the data, even though the log of the probability values returned from the Forward-Backward algorithm fluctuates between -50 and -100.

The calculation of the confidence values over the same set of gestures and HMMs is shown in Figure 4. As in the previous figure, the data returned from the HMM that was trained on the Grab Object (once again delimited by a '+') has a much higher value than the three other HMMs. However, there are gestures

Gesture	$P(O\|\lambda_i)$	C_i
Grab Object	100%	94%
Drop Object	100%	92%
Move Towards Object	100%	97%
Move Away from Object	100%	99%

Table 2: Success rate for classification of gestures

in this sequence which the system is not very confident about, $C_j \leq \sum_{k \neq j} C_k$, and thus the percentage correct classification for the confidence factor is only 94%.

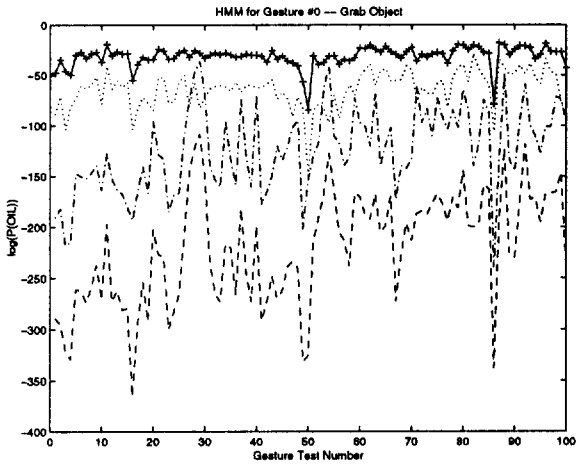

Figure 3: Calculation of $P(O|\lambda_i)$ for all four HMMs, using the gesture data from Grab Object

5.2 Initial System Demonstration

Having proven that the recognition system was reasonably robust, the whole system was put through a preliminary test. A pile of boxes of two different colors was assembled and placed in the center of a room. The task for the robot was to sort the boxes into to separate piles by observing a human do it first, as seen in Figure 5. The robot successfully learned each of the gestures and was then able to complete the entire sorting task by continuously applying the sequence of gestures that it had originally learned from the human. Even though the conditions for the test were overly simplified and the test itself was somewhat contrived, the initial results were encouraging.

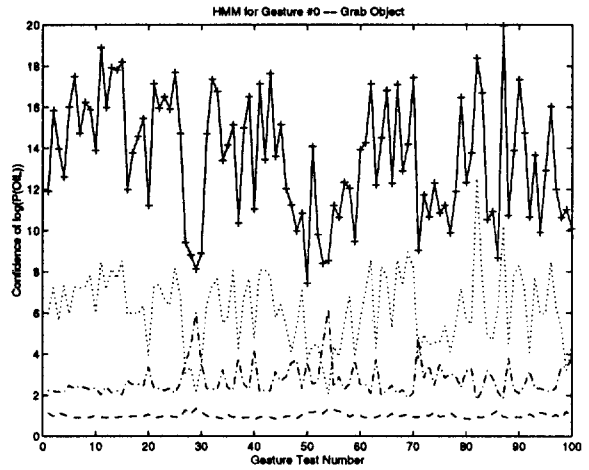

Figure 4: Calculation of C_i for all four HMMs, using the gesture data from Grab Object

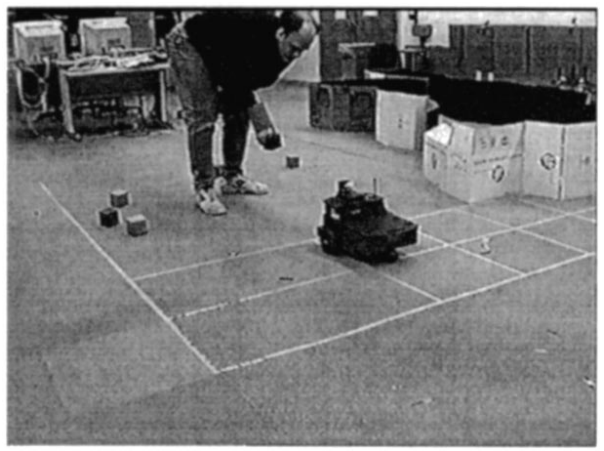

Figure 5: Training the robot to sort blocks

6 Summary and Conclusions

A demonstration-based programming system was developed which allows a human to train a robot on a task by performing a series of actions or gestures. By demonstrating the actions for the robot, the human can let the robot extract relevant parameters for the task (such as the Cartesian position where the action should take place). The robot follows the human around the environment and tries to be as unobtrusive as possible so as to let the human complete its task. The robot provides feedback to the human when it fails to recognize a gesture so that the human can know to re-demonstrate the task.

A set of simple gestures and corresponding actions was defined and implemented on a mobile robot. The gesture-recognition system was tested and found to

be reasonably robust in its classification of gestures. The whole system was put through a preliminary test, and the results and outlook for the system are very encouraging.

7 Future Work

The first extension will be to develop more gestures which involve more than just the visual system of the robot. There is a rich set of information that can be obtained from using the robot's other sensors such as its sonars and bumpers. By fusing this information with the visual gesture data, more powerful and descriptive gestures can be developed to describe more complicated tasks.

An interesting departure from strictly human to robot gesture recognition is that of robot to robot gesture/action recognition. If a single robot is programmed with a particular task and executes it, another robot could be programmed with that task simply by watching the first one. In teams of robots where there are many parallel tasks that must be done, two specific classes of robots could be used: specialists and floaters. The specialists would be programmed ahead of time to do a particular task, while the floaters would move about and assist the specialists as needed. The floaters would observe the specialists doing their tasks and then be able to assist them appropriately. Once the floaters were no longer needed, they would move off to find another specialist to assist. Future extensions of this work will take this scenario and others like it into account.

Acknowledgments

We would like to acknowledge the support of NSF under grant NSF/DUE-9351513. Additional support was provided by the Air Force Research Laboratory under contract number F30602-96-2-0240. Any opinions, findings, conclusions or recommendations expressed herein are those of the authors and do not reflect the views of the Air Force Research Laboratory, Carnegie Mellon University or the University of Minnesota.

References

[1] ActivMedia, Inc., Peterborough, NH. *Pioneer Operation Manual v2*, 1998.

[2] P. Bakker and Y. Kuniyoshi. Robot see, robot do: An overview of robot imitation. In *AISB Workshop on Learning in Robots and Animals*, Brighton, UK, April 1996.

[3] M. Brand, N. Oliver, and A. Pentland. Coupled hidden markov models for complex action recognition. Technical Report 407, Vision and Modeling Group, MIT Media lab, Cambridge, MA 02139, USA, November 1996.

[4] K. G. Konolige. *Saphira Software Manual.* SRI International, 1997.

[5] Y. Kuniyoshi. Vision-based behaviors for multi-robot cooperation. In *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, volume 2, pages 925–932, Munich, 1994.

[6] C. Lee and Y. Xu. Online, interactive learning of gestures for human/robot interfaces. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 2982–2987, Minneapolis, 1996.

[7] P.K. Pook and D.H. Ballard. Recognizing teleoperated manipulations. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 578–585, 1993.

[8] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[9] T. Starner and A. Pentland. Visual recognition of american sign language using hidden markov models. In *International Workshop on Automatic Face and Gesture Recognition (IWAFGR)*, Zurich, Switzerland, 1995.

[10] R. M. Voyles and P.K. Khosla. A multi-agent system for programming robotic agents by human demonstration. In *Proc. of AI and Manufacturing Research Planning Workshop*, volume 2, Albuquerque, 1998.

[11] R. M. Voyles and P.K. Khosla. Gesture-based programming: A preliminary demo. In *IEEE International Conference on Robotics and Automation*, Detroit, 1999.

[12] A. Wilson and A. Bobick. Learning visual behavior for gesture analysis. In *Proceedings of the IEEE Symposium on Computer Vision*, Coral Gables, Florida, 1995.

[13] A. Wright, R. Sargent, C. Witty, and J. Brown. *Cognachrome Vision System User's Guide.* Newton Research Labs, Redmond, WA, 1996.

[14] J. Yang, Xu Y, and C.S. Chen. Hidden markov model approach to skill learning and its application to telerobotics. *IEEE Transactions on Robotics and Automation*, 10(5):621–631, 1994.

[15] J. Yang, Y.Xu, and C.S. Chen. Human action learning via hidden markov model. *IEEE Trans. on Systems, Man, and Cybernetics*, 27(1):45–56, 1997.

[16] Q. Zhu. Hidden markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3):390–397, 1991.