Template-Based Recognition of Pose and Motion Gestures On a Mobile Robot

Stefan Waldherr Sebastian Thrun Roseli Romero Dimitris Margaritis

Computer Science Department
Carnegie Mellon University
Pittsburgh, PA

Abstract

For mobile robots to assist people in everyday life, they must be easy to instruct. This paper describes a gesture-based interface for human robot interaction, which enables people to instruct robots through easy-to-perform arm gestures. Such gestures might be static *pose gestures*, which involve only a specific configuration of the person's arm, or they might be dynamic *motion gestures* (such as waving). Gestures are recognized in real-time at approximate frame rate, using a hybrid approach that integrates neural networks and template matching. A fast, color-based tracking algorithm enables the robot to track and follow a person reliably through office environments with drastically changing lighting conditions. Results are reported in the context of an interactive clean-up task, where a person guides the robot to specific locations that need to be cleaned, and the robot picks up trash which it then delivers to the nearest trash-bin.

Introduction

The field of robotics is currently undergoing a change. While in the past, robots where predominately used in factories for purposes such as manufacturing and transportation, a new generation of "service robots" has recently begun to emerge. Service robots cooperate with people and assist them in their everyday tasks. A landmark service robot is Helpmate Robotics's Helpmate robot, which has already been deployed at numerous hospitals worldwide (King & Weiman 1990). In the near future, similar robots are expected to appear in various branches of entertainment, recreation, health-care, nursing, etc., and we expect them to interact closely with people.

This upcoming generation of service robots opens up new research opportunities. While the issue of *mobile robot navigation* has been researched quite extensively (see e.g., (Kortenkamp, Bonassi, & Murphy 1998; Borenstein, Everett, & Feng 1996)), considerably little attention has been paid to issues of *human-robot interaction*. However, many service robots will be operated by non-expert users, who might not even be capable of operating a computer keyboard. It is therefore essential that these robots be equipped with "natural" human robot interfaces that facilitate the interaction of robots and people.

The need for more effective human robot interfaces has been recognized. For example, in his M.Sc. thesis, Torrance developed a natural language interface for teaching mobile robots names of places in an indoor environment (Torrance 1994). Due to the lack of a speech recognition system, his interface still required the user to operate a keyboard; however, the natural language component made instructing the robot significantly easier. More recently, Asoh and colleagues (Asoh *et al.* 1997) developed an interface that integrates a speech recognition system into a phrase-based natural language interface. They successfully instructed their "office-conversant" robot to navigate to office doors and other significant places in their environment, using verbal commands. Other researchers have proposed vision-based interfaces that allow people to instruct mobile robots via arm gestures. For example, Kortenkamp and colleagues (Kortenkamp, Huber, & Bonassi 1996) recently developed a gesture-based interface, which is capable of recognizing arm poses such as pointing towards a location on the ground. In a similar effort, Kahn and colleagues (Kahn *et al.* 1996) developed a gesture-based interface which has been demonstrated to reliably recognize static arm poses (pose gestures) such as pointing. This interface was successfully integrated into Firby's reactive plan-execution system RAP (Firby *et al.* 1995), where it enabled people to instruct a robot to pick up free-standing objects. Both of these approaches, however, recognize only static pose gestures. They cannot recognize gestures that are defined through specific temporal patterns of arm movements, such as waving. Motion gestures, which are commonly used for communication among people, provide additional freedom in the design of gestures. In addition, they reduce the chances of accidentally classifying arm poses as gestures that were not intended as such. Thus, they appear better suited for human robot interaction than static pose gestures.

This paper presents a vision-based human robot interface that has been designed to instruct a mobile robot through both pose and motion gestures. An adaptive dual-color tracking algorithm enables the robot to find, track, and follow a person around at speeds of up to one foot per second. This tracking algorithm can quickly adapt to different lighting conditions. Gestures are recognized by a real-time template-matching algorithm. This algorithm works in two phases: one that recognizes static arm poses, and one that recognizes gestures.

Figure 1: AMELIA, the robot used in our research, is a RWI B21 robot equipped with a color camera mounted on a pan-tilt unit, 24 sonar sensors, and a 180° SICK laser range finder.

The algorithm can recognize both pose and motion gestures.

This approach has been integrated into our existing robot navigation and control software (Thrun *et al.* 1998; Burgard *et al.* 1998), where it enables human operators

- to provide direct motion commands (e.g., stopping),
- to guide the robot to places which it can memorize
- to indicate the location of objects (e.g., trash on the floor)
- and to initiate clean-up tasks, where the robot searches for trash, picks it up, and delivers it to the nearest trash-bin.

In a pilot study, we have successfully instructed our robot to pick up trash scattered in an office building, and to deposit it into a trash-bin. This task was motivated by the "clean-up an office" task, which was designed for the AAAI-94 mobile robot competition (Simmons 1995). Our scenario differs from the competition task in that a human interacts with the robot and initiates the clean-up task, which is then performed autonomously by the robot.

In our experiments, we found the interface to be reliable and relatively easy to use. While this is only an example application designed to test the utility of motion gestures in human robot interaction, we believe that our interface is applicable to a larger range of upcoming service robots.

Visual Tracking and Servoing

The lowest-level component of our approach is a color-based tracking algorithm, which enables the robot to find, track, and follow people in real-time. Visual tracking of people has been studied extensively (Darrel, Moghaddam, & Pentland 1996; Crowley 1997; Wren *et al.* 1997). Many existing approaches assume that the camera is mounted at a fixed location. Such approaches typically rely on a static background, so that human motion can be detected through image differencing. Some approaches (e.g., (Yang & Waibel 1995)) can track people if the camera is mounted on a pan-tilt unit, which can impose mild changes in illumination. Recognizing gestures with a robot-mounted camera is more difficult due to the occasional occurrence of drastic changes in background and lighting conditions that are caused by robot motion. This problem has previously been addressed by (Wong, Kortenkamp, & Speich 1995; Huber & Kortenkamp 1995), who successfully devised algorithms for tracking people visually similar to the one proposed here.

Since our algorithm for finding people is a specialization of our tracking algorithm, let us first describe the tracking algorithm. This algorithm tracks people based on a combination of two colors, namely face color and body color (i.e., shirt color). It iterates four steps:

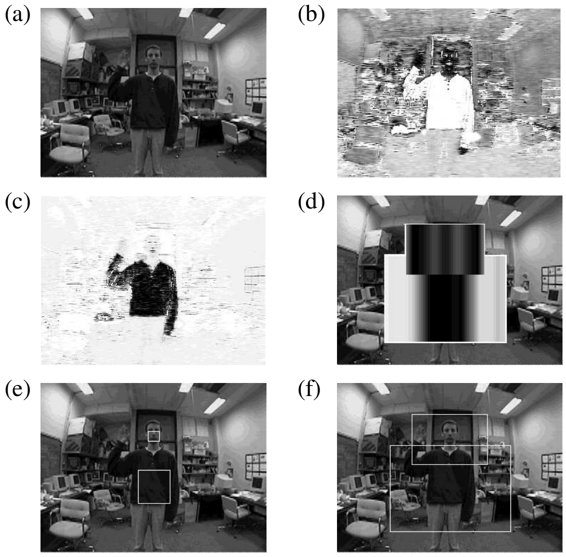

Figure 2: Tracking a person: (a) Raw camera image, (b) face-color filtered image, and (c) body-color filtered image. The darker a pixel in the filtered images, the smaller the Mahalanobis distance to the mean color. (d) Projection of the filtered image onto the horizontal axis (within a search window). (e) Face and body center, as used for tracking and adaptation of the filters. (f) Search window, in which the person is expected to be found in the next image.

Step 1: Color Filtering. Two Gaussian color filters are applied to each pixel in the image. Each filter is of the form

$$c_i = \begin{pmatrix} e^{(X_i - \hat{X}_{\text{face}})^T \, \Sigma_{\text{face}}^{-1} \, (X_i - \hat{X}_{\text{face}})} \\ e^{(X_i - \hat{X}_{\text{body}})^T \, \Sigma_{\text{body}}^{-1} \, (X_i - \hat{X}_{\text{body}})} \end{pmatrix} \qquad (1)$$

where X_i is the color vector of the i-th image pixel, $\hat{X}_{\text{face}}$ and Σ_{face} are the mean and covariance matrix of a face color model, and $\hat{X}_{\text{body}}$ and Σ_{body} are the mean and covariance matrix of a body (shirt) color model. The result of this operation are two filtered images, example of which are shown in Figures 2b&c. These images are then smoothed locally using a pseudo-Gaussian kernel with width 5, in order to reduce the effects of noise.

Step 2: Alignment. Next, the filtered image pair is searched for co-occurrences of vertically aligned face and body color. This step rests on the assumption that a person's face is above his/her shirt in the camera image. First, the image is mapped into a horizontal vector, where each value corresponds to the combined face- and body-color integrated vertically. Figure 2d illustrates the results of this alignment step. The gray-level in the two center regions indicate graphically the horizontal density of face and body color. The darker a region, the better the match. Both responses are then multiplied, to determine the estimated horizontal coordinates of the person. Finally, the filtered image regions are searched vertically for the largest occurrence of the respective color, to determine the vertical coordinates of face and body. Figure 2e shows the results of this search. We found this scheme to be highly reliable, even for people that moved hastily in

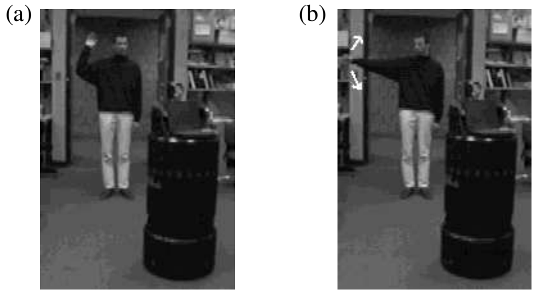

Figure 3: Example gestures: (a) stop gesture and (b) follow gesture. While the stop gesture is a pose gesture, the follow gesture involves motion, as indicated by the arrows.

front of the robot.

Step 3: Servoing. If the robot is in visual servoing mode (meaning that it is following a person), it issues a motion command that makes the robot turn and move towards this person. The command is passed on to a collision avoidance method (Fox, Burgard, & Thrun 1997) that sets the actual velocity and motion direction of the robot in response to proximity sensor data.

Step 4: Adaptation. Finally, the means and covariances $\hat{X}_{\text{face}}, \Sigma_{\text{face}}, \hat{X}_{\text{body}}, \Sigma_{\text{body}}$ are adapted, to compensate changes in illumination. The robot computes new means and covariances from small rectangular regions around the center of the face and the body (shown in Figure 2e). Let $\hat{X}^*_{\text{face}}, \Sigma^*_{\text{face}}, \hat{X}^*_{\text{body}}, \Sigma^*_{\text{body}}$ denote these new values, obtained from the most recent image. The means and covariances are updated according to the following rule, which is a temporal estimator with exponential delay:

$$\begin{aligned} \hat{X}_{\text{face}} &\longleftarrow \alpha \hat{X}^*_{\text{face}} + (1-\alpha)\hat{X}_{\text{face}} \\ \sigma_{\text{face}} &\longleftarrow \alpha \sigma^*_{\text{face}} + (1-\alpha)\sigma_{\text{face}} \\ \hat{X}_{\text{body}} &\longleftarrow \alpha \hat{X}^*_{\text{body}} + (1-\alpha)\hat{X}_{\text{body}} \\ \sigma_{\text{body}} &\longleftarrow \alpha \sigma^*_{\text{body}} + (1-\alpha)\sigma_{\text{body}} \end{aligned} \tag{2}$$

Here α is a decay factor, which we set to 0.1 in all our experiments.

Finding a person. To find a person and acquire an initial color model, the robot scans the image for face color only, ignoring its body color filter. Once a color blurb larger than a specific threshold is found, the robot acquires its initial body color model based on a region below the face. Thus, the robot can track people with arbitrary shirt colors, as long as they are sufficiently coherent.

Our tracking algorithm, which can be viewed of a two-color extension of basic color-based tracking algorithms such as those described in (Yang & Waibel 1995; Swain 1991), was found to work reliably when tracking people and following them around through buildings with rapidly changing lighting conditions. The tracking routine is executed at a rate of 20 Hertz on a 200 Mhz Pentium PC. This speed is achieved by focusing the computation (Steps 1 and 2) on a small window around the location where the person was previously seen. Thus far, we tested the tracker only for various types of single-color shirts with long sleeves; experiments for multi-colored shirts were not yet conducted.

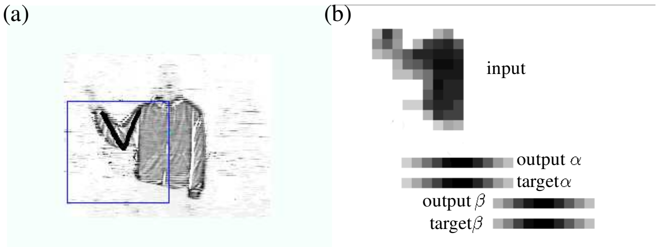

Figure 4: Neural network pose analysis: (a) Camera image, with the two arm angles as estimated by the neural network superimposed. The box indicates the regions which is used as network input. (b) The input to the neural network, a down-sampled, color-filtered image of size 10 by 10, and the outputs and targets of the networks for the two angle.

Template-Based Recognition of Pose and Motion Gestures

A primary goal of this work has been to devise a vision-based interface that is capable of recognizing both pose and motion gestures. Pose gestures involve a static configuration of a person's arm, such as the "stop" gesture shown in Figure 3a, whereas motion gestures are defined through specific motion patterns of an arm, such as the "follow me" gesture shown in Figure 3b.

Our approach employs two phases, one for recognizing poses from a single image (pose analysis), and one for recognizing sequences of poses from a stream of images (temporal template matching).

Pose Analysis

In the first phase, a probability distribution over all poses is computed from a camera image. Our approach integrates two alternative methods for image interpretation: a neural-network based method and a graphical template matcher (called: pose template matcher). Both approaches operate on a color-filtered sub-region of the image which contains the person's right side, as determined by the tracking module. They output a probability distribution over arm poses.

The *neural network-based approach* predicts the angles of the two arm segments relative to the person's body from the image segment. The input to the network is a down-sampled image segment, and the output are the angles, encoded using multi-unit Gaussian representations (Pomerleau 1993). In our implementation, we used 60 output units, 30 for each of the two arm angles. The network was trained with Backpropagation, using a database of 758 hand-labeled training images. After training, the average error was $4.91°$ for the angle of the upper arm segment and $5.54°$ for the angle of the lower arm segment. These numbers were obtained using an independent set of 252 testing images. Figure 4a shows an example image. Superimposed here are the two angle estimates, as generated by the neural network. Figure 4b shows the input, output, and target values for the network. The input is a down-sampled, color-filtered image of size 10 by 10. The output is Gauss-encoded. The nearness of the outputs (first and third row) and the targets (second and forth row) suggests that in this example, the network predicts the angle with high accuracy.

Figure 5: Examples of pose templates. The excitatory region is shown in black and the inhibitory in gray. White regions are not considered in the matching process.

Our approach also employs a template-based algorithm for analyzing poses, which compares poses to a set of pre-recorded *pose templates*. More specifically, the color-filtered image is correlated with a set of 16 pre-recorded templates of arm poses. Each pose template consist of three regions: an *excitatory region*, which specifies where the arm is to be expected for a specific pose, an *inhibitory region*, where the arm should *not* be for this pose, and a *don't-know region*, which is not considered when computing the correlation. Figure 5 shows examples of pose templates. Here excitatory regions are shown in black, whereas inhibitory regions are shown in gray. The templates are constructed from labeled examples of human arm poses (one per template), where the excitatory region is extracted from the largest coherent region in the filtered image segment, and a straightforward geometric routine is employed to determine a nearby inhibitory region. In a experiment involving 122 example images, we recorded that in 85.8% of all cases, the correct pose was identified; in 9.68% a nearby pose was recognized, and in only 4.52% the pose template matcher produced an answer that was clearly wrong. While the choice of 16 as the number of templates is somewhat ad hoc, we found that it covered the space of possible arm poses in sufficient density for the tasks at hand.

Both methods, the neural network-based method and the pose template matcher, generate multi-dimensional *feature vectors*, one for each image. The integration of two different methods for pose analysis was originally driven by the goal of understanding the different strengths and weaknesses of the approaches. While neural networks generate information at much higher accuracy (floating-point accuracy vs. 1-out-of-16), in preliminary evaluations we observed that they are less robust to occlusions of body parts. These results, however, are preliminary, and a systematic experimental comparison is currently underway.

Temporal Template Matching

In the second phase, a temporal template matcher compares the temporal stream of feature vectors with a set of pre-recorded prototypes (gesture templates) of individual gestures.

Figure 6 shows examples of gesture templates, for the gestures "stop" (Figure 6a) and "follow" (Figure 6b). Each of these templates is a sequence of prototype feature vectors, where time is arranged vertically. The size of the white boxes indicates the magnitude of the corresponding numerical value. Gesture templates are composed of a sequence of feature vectors, constructed from a small number (e.g., 5) of training examples. For graphical clarity, only feature vectors with 16 components obtained with the pose template matchers are shown here. As can be seen in this figure, the stop gesture is a pose gesture (hence all feature vectors look alike), whereas the follow gesture involves motion. Both types of gestures—pose gestures and motion gestures—are encoded through such gesture templates.

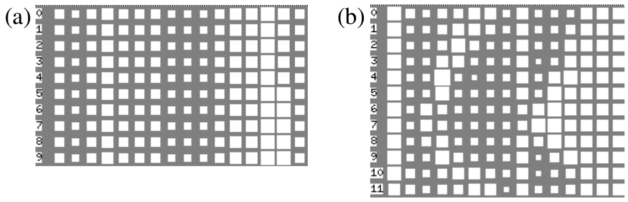

Figure 6: Examples of gesture templates. Gesture templates are sequences of prototype feature vectors. Shown here are gesture templates for (a) stop gesture (does not involve motion), (b) follow gesture (involves motion, as indicated by the change over time).

The temporal template matcher continuously analyzes the stream of incoming feature vectors for the presence of gestures. It does this by matching the gesture template to the most recent n feature vectors, for varying numbers of n (in our implementation, $n = 40, 50, \ldots, 80$); notice that the gesture templates are much shorter than n. To compensate differences in the exact timing when performing gestures, our approach uses the Viterbi algorithm (Rabiner & Juang 1986) for time alignment. The Viterbi alignment employs dynamic programming to find the best temporal alignment between the feature vector sequence and the gesture template, thereby compensating for variations in the exact timing of a gesture. Figure 7 shows an actual sequence, during which a person performs the follow gesture. The reader should notice that this example exhibits a similar pattern as shown in Figure 6b. The arrow marks the point in time at which the follow gesture is complete and recognized as such by the Viterbi algorithm.

Learning

Both sets of templates, the pose templates and the gesture templates, are learned from examples, just like the neural network. To teach the robot a new gesture, the person presents itself in front of the robot and executes this gesture several times, in pre-specified time intervals. Our approach then segments these examples into pieces of equal length, and uses the average feature vector in each time segment as a prototype segment. We found the current training scheme to be robust to variations of various sorts, as long as the person exercises reasonable care when training the robot. In most cases, a single training gesture suffices to build a reliable template.

Integration and Results

The gesture-based approach has been integrated into our previously developed mobile robot navigation system, thereby building a robot that can be instructed by natural means (Thrun *et al.* 1998; Burgard *et al.* 1998). In a nutshell, our navigation methods enable robots to navigate safely while acquiring maps of unknown environments. A fast motion planner allows robots to move from point to point or, alternatively, to explore unknown terrain. Collisions with obsta-

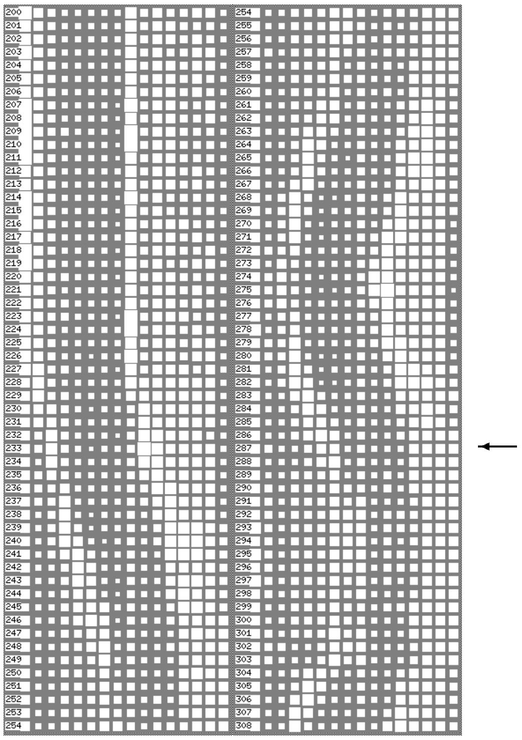

Figure 7: Example of a feature vector stream that contains a *follow* gesture. The arrow indicates the point in time when *follow* is recognized.

cles are avoided by a fast, reactive routine, which sets the velocity and travel direction of the robot according to periodic measurements of the robot's various proximity sensors (laser, sonar, infrared, tactile). While the robot is in motion, a concurrent localization method continuously tracks the robot's position, by comparing sensor readings with the learned map of the environment. Permanent blockages lead to modifications of the map, and thus to new motion plans.

We tested the effectiveness of the gesture-based interface in the context of a clean-up task that involved human user interaction and mobile manipulation. The specific choice of the task was motivated by past AAAI mobile robot competitions, in which robots were asked to find and remove all sorts of objects (trash, tennis balls, etc.).

- **Follow**: If the robot is shown a follow gesture, it follows a person around. The follow gesture involves waving the arm, as indicated in Figure 3a.
- **Stop:** If the robot is shown a stop gesture, it immediately stops. The stop gesture is a pose gesture, as shown in Figure 3b.
- **Pointing:** If the person points towards an object on the ground, the robot starts searching for an object within its visual field. If it succeeds, it moves towards the object, picks it up, and then returns to the nearest trash-bin. The location of trash-bins is known to the robot. In our implementation, the pointing gesture is actually a motion gesture, which involves moving the arm from the body towards the pointing direction and back. We found this version of the pointing gesture to lead to fewer false-positives.
- **Abort:** If the person shows an abortion gesture, the robot moves back to its initial position and waits for a new person.

	recognition result					
	follow	stop	point-1	point-2	abort	no gesture
follow	59	-	-	-	-	-
stop	-	39	1	-	-	-
point-1	-	-	44	-	-	6
point-2	-	-	-	42	-	-
abort	1	2	-	-	16	1
no gesture	-	-	-	-	-	50

Table 1: Recognition results. Each row shows the recognition result for examples of a specific gesture (including 50 tests where the human subject did not perform any gesture).

Table 1 surveys experimental results for the recognition accuracy of the gesture-based interface. Each row corresponds to a number of experiments, in which a human subject presented a specific gesture. Because of the diversity of possible pointing gestures, this specific gesture was realized by two different gesture templates, one for pointing towards the floor (labeled "point-1" in Table 1) and one for pointing horizontally (labeled "point-2"). In some experiments, no gesture was shown, to test the robot's ability to detect gestures only if the person actually performed one.

As can be seen in Table 1, our approach recognizes gestures fairly accurately. In 211 experiments in which a human showed a gesture and an additional 50 experiments where the human did not show a gesture, the robot classified 95.8% of the examples correctly. With 95% confidence, the overall accuracy is in the range $[93.3\%; 98.3\%]$. Some gestures had a 100% recognition rate (follow and point-2), whereas the point-1 gesture was only recognized with 88% accuracy.

Figure 8 shows an example run, in which our robot AMELIA is instructed to pick up a piece of trash. Shown there is a map of the the robot's environment, constructed using an occupancy grid technique (Moravec 1988; Thrun *et al.* 1998), along with the actual path of the robot and the (known) location of a trash-bin. Initially, the robot waited in the corridor for a person. The person instructed the robot to follow him into the lab (using the follow gesture), where it first stopped the robot (using the stop gesture), then pointed at a piece of trash (a can). The robot picked up the can, and returned to the corridor where it deposited the trash in a bin. We found that operating the robot at low speed (0.7 feet per second) made it easier to instruct the robot. Tests at higher speeds (1.5 feet per second) made it difficult to position the robot reliably close to a trash can, even though our software can manage much higher speeds. In the future, we plan to extend the interface so that the person can select the speed, and in addition give direct commands to the robot (such as "rotate left" or "move 2 feet back").

Discussion

This paper described a gesture-based interface for human-robot interaction. A hybrid approach, consisting of an adap-

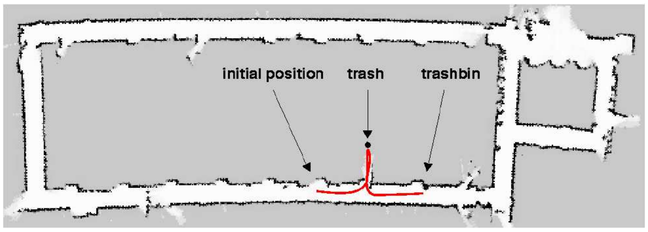

Figure 8: Map of the robot's operational range (80 by 25 meters) with trace of a specific example of a successful clean-up operation. The robot waited in the corridor, was then guided by a human into a lab, where it picked up a can and later deposited it into a trash-bin.

tive color-filter, two template matchers and an artificial neural network, were described for recognizing human arm gestures from streams of camera images. Our approach is capable of recognizing both static pose gestures and dynamic motion gestures. The paper demonstrated the usefulness of the interface in the context of a clean-up task, where a person cooperated with the robot in cleaning up trash.

We believe that finding "natural" ways of communication between humans and robots is of importance for the field of robotics, as a variety of recent changes in both robotic hardware and software suggests that service robots will soon become possible, and commercially viable. This research is intended to be a step towards this goal.

There are several open questions and limitations that warrant further research. For example, the tracking module is currently unable to deal with multi-colored shirts, or to follow people who do not face the robot. We believe, however, that the robustness can be increased by considering other cues, such as shape and texture, when tracking people. Secondly, our approach currently lacks a method for teaching robots new gestures. This is not really a limitation of the basic gesture-based interface, as it is a limitation of the robot's finite state machine that controls its operation. Future work will include providing the robot with the ability to learn new gestures, and to associate those with specific actions and/or locations. Together with our existing mapping and navigation software, this should lead to a robot that can be taught a collection of tasks in novel indoor environments. Thus, by providing the robot with the ability to learn new things, we seek to explore further the practical utility of gesture-based instruction in the context of mobile service robotics. Finally, we believe it is worthwhile to augment the interface by a speech-based interface, so that both gestures and speech can be combined when instructing a mobile robot.

Acknowledgment

Partial financial support by Daimler-Benz Research, the Defense Advanced Research Projects Agency (DARPA) via the Airforce Missile System Command under contract number F04701-97-C-0022, and FAPESP (Brazil) under grant number 96/4515-8 is gratefully acknowledged.

References

Asoh, H.; Hayamizu, S.; Hara, I.; Motomura, Y.; Akaho, S.; and Matsui, T. 1997. Socially embedded learning of office-conversant robot jijo-2. In *Proceedings of IJCAI-97*. IJCAI, Inc.

Borenstein, J.; Everett, B.; and Feng, L. 1996. *Navigating Mobile Robots: Systems and Techniques*. Wellesley, MA: A. K. Peters, Ltd.

Burgard, W.; Cremers, A.; Fox, D.; Hähnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W.; and Thrun, S. 1998. The interactive museum tour-guide robot. In *Proceedings of AAAI-98*.

Crowley, J. 1997. Vision for man-machine interaction. *Robotics and Autonomous Systems* 19:347–358.

Darrel, T.; Moghaddam, B.; and Pentland, A. 1996. Active face tracking and pose estimation in an interactive room. In *Proceedings of ICCV-96*, 67–72.

Firby, R.; Kahn, R.; Prokopowicz, P.; and Swain, M. 1995. An architecture for active vision and action. In *Proceedings of IJCAI-95*, 72–79.

Fox, D.; Burgard, W.; and Thrun, S. 1997. The dynamic window approach to collision avoidance. *IEEE Robotics and Automation* 4(1).

Huber, E., and Kortenkamp, D. 1995. Using stereo vision to pursue moving agents with a mobile robot. In *Proceedings of IEEE ICRA-95*.

Kahn, R.; Swain, M.; Prokopowicz, P.; and Firby, R. 1996. Gesture recognition using the perseus architecture. In *Proceedings of the IEEE CVPR-96*, 734–741.

King, S., and Weiman, C. 1990. Helpmate autonomous mobile robot navigation system. In *Proceedings of the SPIE Conference on Mobile Robots*, 190–198. Volume 2352.

Kortenkamp, D.; Bonassi, R.; and Murphy, R., eds. 1998. *AI-based Mobile Robots: Case studies of successful robot systems*. Cambridge, MA: MIT Press.

Kortenkamp, D.; Huber, E.; and Bonassi, P. 1996. Recognizing and interpreting gestures on a mobile robot. In *Proceedings of AAAI-96*, 915–921.

Moravec, H. P. 1988. Sensor fusion in certainty grids for mobile robots. *AI Magazine* 61–74.

Pomerleau, D. 1993. *Neural Network Perception for Mobile Robot Guidance*. Boston, MA: Kluwer Academic Publishers.

Rabiner, L., and Juang, B. 1986. An introduction to hidden markov models. In *IEEE ASSP Magazine*.

Simmons, R. 1995. The 1994 AAAI robot competition and exhibition. *AI Magazine* 16(1).

Swain, M. 1991. Color indexing. *International Journal of Computer Vision* 7.

Thrun, S. et al. 1998. Map learning and high-speed navigation in RHINO. In Kortenkamp, D.; Bonassi, R.; and Murphy, R., eds., *AI-based Mobile Robots: Case studies of successful robot systems*. Cambridge, MA: MIT Press.

Torrance, M. C. 1994. Natural communication with robots. Master's thesis, MIT Department of EECS, Cambridge, MA.

Wong, C.; Kortenkamp, D.; and Speich, M. 1995. A mobile robot that recognizes people. In *Proceedings of ICTAI-95*.

Wren, C.; Azarbayejani, A.; Darrell, T.; and Pentland, A. 1997. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Learning* 19(7):780–785.

Yang, J., and Waibel, A. 1995. Tracking human faces in real-time. Technical Report CMU-CS-95-210, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.