A Neural-Network Based Approach for Recognition of Pose and Motion Gestures On a Mobile Robot

Stefan Waldherr, Sebastian Thrun and Roseli Romero

Computer Science Department
Carnegie Mellon University
Pittsburgh - PA - USA

Abstract

Since a variety of recent changes in both robotic hardware and software suggests that service robots will soon become possible, to find "natural" ways of communication between human and robots is of fundamental importance for the robotic field. 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, *that is, they involve motion (such as waving). Gestures are recognized in real-time at approximate frame rate, using neural networks. 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.*

1 Introduction

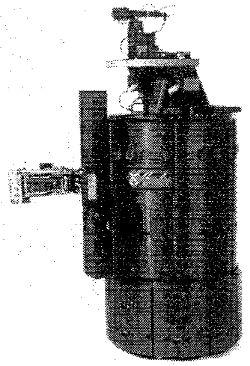
The field of robotics is currently undergoing a change. While in the past, robots where predominately used in factories for purposes such as manufacturing and transportation, a new generation of "service robots" has recently begun to emerge. Service robots cooperate with people, and assist them in their everyday tasks. A landmark service robot is Helpmate Robotics's Helpmate robot, which has already been deployed at numerous hospitals worldwide [8]. Helpmate, however, does not interact with people other than by avoiding them. In the near future, similar robots are expected to appear in various branches of entertainment, recreation, healthcare, nursing, and others, and to interact directly 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., [2]), considerably little attention has been paid to issues of *human-robot interaction.* The need for more effective human-robot interfaces has been recognized. For example, Torrance developed a natural language interface for teaching mobile robots names of places in an indoor environment [16]. 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. Recently, Asoh and colleagues [1] 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 [9] 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 his colleagues [7] 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 [5], where it enabled people to instruct a robot to pick up free-standing objects. Both of these approaches, however, recognize only static pose gestures.

Our approach extends this work to motion gestures, that is, gestures that are defined through specific temporal patterns of arm movements, such as

0-8186-8629-4/98 $10.00 © 1998 IEEE

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.



waving. Motion gestures, which are 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 an adaptive dual-color tracking algorithm which enables the robot to track and, if required, follow a person around at speeds of up to one foot per second while avoiding collisions with obstacles. This tracking algorithm quickly adapts to different lighting conditions. Gestures are recognized by a real-time neural-network based algorithm. This algorithm works in two phases: one that recognizes static arm poses, and one that recognizes gestures (pose and motion). In the first phase, the algorithm predicts the angles of the two arm segments relative to the person's body from image. In the second phase, the results of the first phase are temporally matched to previously recorded examples of gestures, using the Viterbi algorithm [12]. The gesture angles matcher can recognize both pose and motion gestures. The result is a stream of probability distribution over the set of all gestures, which is then thresholded and passed on to the robot's high-level controller.

This approach has been integrated into our existing robot navigation and control software, where it enables human operator to provide direct motion commands (e.g., stopping), to guide the robot to places which it can memorize, to point to 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.

2 Visual Tracking and Servoing

The lowest-level component of our approach is a vision-based tracking algorithm that enables the robot to track and follow people in real-time. Visual tracking of people has been studied extensively over the past few years [4, 3]. The vast majority of existing approaches assumes 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 more advanced approaches (e.g., [18]) can track people even if the camera is mounted on a pan-tilt unit. However, even in these cases, the illumination is usually fairly uniform. In addition, processing power is limited on our robot (200Mhz Pentium PC), which imposes an additional burden on the software design.

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. For example, in [18], a system was proposed which tracks human faces based on their color. This approach was reported to track people reliably with a camera mounted on a pan-tilt unit. When testing this approach on a mobile robot, however, changes in lighting conditions often made it impossible to follow a person through a building.

We therefore extended the color-based approach in a fairly straightforward way. Our approach tracks humans based on a combination of two colors, namely face color and body color (i.e., shirt color). Both colors are assumed to be arranged vertically in the image. The resulting algorithm iterates four steps:

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.

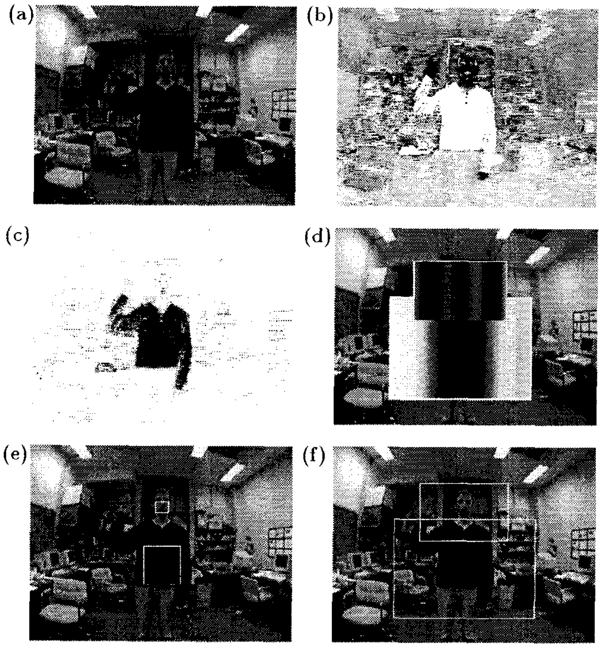


Figure 2: Tracking a person: (a) Raw camera image, (b) face-color filtered image, and (c) body-color filtered image. (d) projection of the filtered image onto the horizontal axis (within a search window). (e) Face and body center, as used for tracking and adaptation. (f) Search window, in which the person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

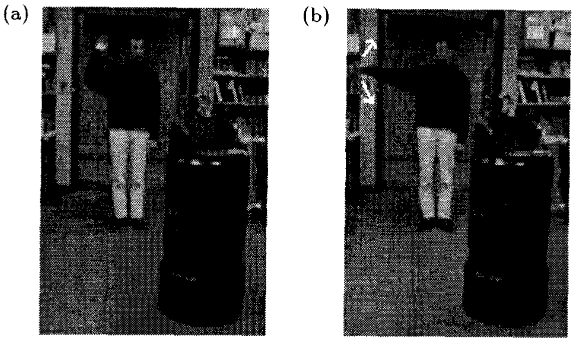


Figure 3: Example gestures: (a) stop gesture and (b) follow gesture. While the stop gesture is a pose gesture, the follow gesture involves motion, as indicated by the arrows.

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 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 [6] 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, that are obtained from the most recent image only. The means and covariances are updated according to the following rule, which is a temporal estimator with exponential delay:

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

Here α is a learning rate, which we set to 0.1 in all our experiments.

Finding a person. To find the 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.

This straightforward extension of the basic color-based tracking approach 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.

3 Recognition of Pose and Motion Gestures

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

The approach proposed here employs two phases, one for recognizing poses from a single image (pose

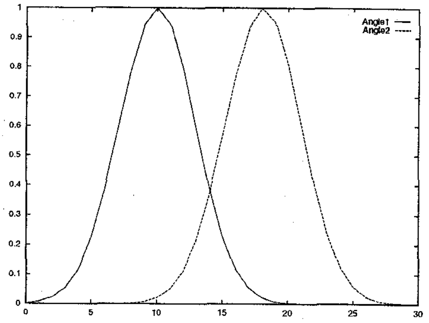


Figure 4: Gaussian output encoding for a pair of angles

analysis), and one for recognizing sequences of poses from a stream of images (temporal angles matching).

Pose analysis. In the first stage a probability distribution over all poses is computed from a camera image. A neural-network based method is used for image interpretation. The approach operates on a color-filtered sub-region of the image which contains the person's right side, as determined by the tracking module.

The neural network-based approach predicts the angles of the two arm segments relative to the person's right side from the image. The input to the network is a downsampled image segment constituted by a vector of 100 components, and the output corresponds to the angles of the arm segments, encoded using multi-unit Gaussian representations, just like in to Pomerleau's ALVINN [11]. The output encoding uses multiple units to encode a single scalar value, by generating Gaussian-like activations over the array of output units. Like Pomerleau, we found that those representations gave the best results among several ones that we tried during the course of this research.

The network was trained using Backpropagation algorithm [13], considering a database of 1708 hand-labeled training images, that is, for each image relative to the person's right side, we have computed two angles corresponding of the two arm segments through a graphic interface, in order to build the training data set. We used 60 output units, 30 for each one of the two arm angles. As a simple example, Figure 4 shows two different angles: Angle1 = 124.3 degrees and Angle2 = 224.5 degrees codified by using the Gaussian Output representation given by:

$$y_i = e^{-50*d_i^2}$$

where $d_i = \frac{angle}{360} - \frac{i}{29}$ for $i = 0, 1, \ldots, 29$ and $angle \in R$. Figure 5(b) 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.

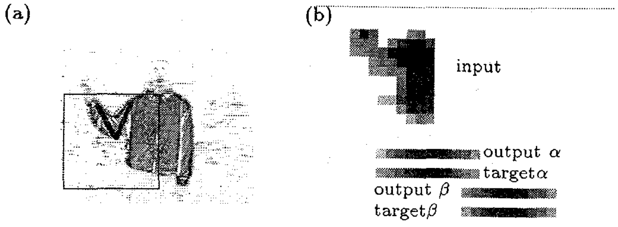


Figure 5: Neural network pose analysis: (a) Camera image, with the two arm angles as estimated by the neural network superimposed. The box indicates the region which is used as network input. (b) The input to the neural network and the outputs and targets of the networks for the two angle.

Topology	Average error Upper arm segment	 Lower arm segment
100-200-60	4.85	5.24
100-100-60	4.73	5.45
100- 50-60	4.96	5.56

Table 1: Average error obtained by neural network in the testing data

After the training phase was concluded, since the neural network output is a vector with 60 components in which the 30 first components correspond to the upper segment of the arm whereas the 30 last components correspond to the lower segment of the arm, it was necessary to convert the network's output activation levels into two angles (in degrees) for interpretation of the results. For this, the gaussian of width specified during training that best fits the output activation levels was determined as for the first 30 output units as for the last 30 output units. In our case, the technique used was the Discrete Search technique considering the LMS given by:

$$E(a) = \Sigma_i\ [y_i - f(x_i)]^2$$

where $f(x_i) = e^{-\frac{(x_i - a)^2}{2592.}}$; $x_i = i/29. * 360.0$ and y_i is the i^{th} component of the output of the neural network. The value of a corresponding the minimum value of $E(a)$ has been considered as the angle corresponding to those output activation levelss, that is, the angle a^* provided by the network is determined by the value corresponding to the best fit gaussian's peak along the output given by:

$$a^{*} = arg_{a}min\{\Sigma_i \; [y_i - e^{-\frac{(x_i-a)^2}{2592.}}]\}$$

The network's average error for the angle of the upper arm segment, and for the angle of the lower arm segment, for an independent set of 569 testing images, is shown in Table 1 considering three different topologies and a learning rate equals to 0.025. Other rate learning has been considered but as it was expected as much one requires precision in the learning of training set as less precision is obtained in the testing data. So, we decided to choose this value as the value of learning rate. The tests in real time were performed considering 100 neurons in the hidden layer, that is, the topology 100-100-60. Figure 5(a) shows an example image. Superimposed here are the two angle estimates, as generated by the neural network.

The neural network-based method generates two-dimensional *feature vectors*, one per image. The neural network generates two angles that are compared to two angles (desired angles) of each image in training set. The result is a vector of 2 components, where each value corresponds to the distance between the desired angle and angle provided by network. These feature vectors form the basis of the temporal angle matching.

Temporal Angle Matching. In the second phase, a temporal angle matcher compares the temporal stream of feature vectors with a set of pre-recorded prototypes of individual gestures. Each of these templates is a sequence of prototype feature vectors, where time is arranged vertically. Gesture templates are composed of a sequence of feature vectors, constructed from a small number of observations.

The temporal angle 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 ($n = 40, 50, \ldots, 80$); 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 [12] for time alignment, which employs dynamic programming to find the best temporal alignment between the feature vector sequence and the gesture template.

4 Integration and Results

The gesture-based approach has been integrated into our previously developed mobile robot navigation system, to build a robot that can be instructed by natural means [15]. In a nutshell, our navigation methods enable robots to navigate safely while acquiring maps of unknown environments.

gestures given		gestures recognized				
		stop	follow	point-2	point-1	no gesture
	241	40	62	51	45	41
stop	40	40	-	-	-	-
follow	59	-	59	-	-	-
point-2	50	-	-	50	-	-
point-1	42	-	-	-	42	-
no gesture	50	-	3	1	3	41

Table 2: Recognition results. Point-1 means gestures pointing towards to floor and Point-2 for pointing horizontally .

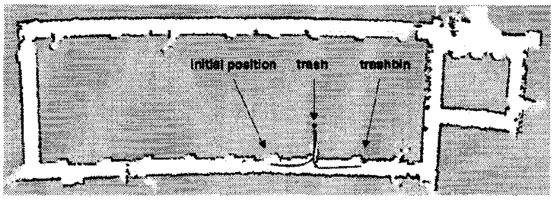


Figure 6: Map of the robot's operational range (80 by 25 meters) with trace of a specific example of a successful clean-up operation. The robot waited in the corridor, was then guided by a human into a lab, where it picked up a can and later deposited it into a trash-bin.

Table 2 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. In some experiments, no gesture was shown, to test the robot's ability to detect gestures only if the person actually performed one. Since false-positives (the robot recognizes a gesture that was not shown by the instructor) are generally worse than false-negatives (the robot fails to recognize a gesture), we tuned our thresholds such that the number of false-positives was small.

As can be seen in Table 2, our approach recognizes gestures fairly accurately. In 191 experiments where a human showed a gesture, and an additional 50 experiments where the human did not show a gesture, the robot classified 97.2% of the examples correctly. All errors were not of the type that the robot failed to recognize a gesture. There were no misclassifications among different gestures.

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.

Figure 6 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

[10, 15], 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.

5 Conclusion

This paper described a gesture-based interface for human-robot interaction. A hybrid approach, consisting of an adaptive color-filter and an artificial neural network, was 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.

There are several open questions that warrant further research. First, our approach has several limitations. 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. 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.

Acknowledgent 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

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

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

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

[4] Darrel, T., Moghaddam, B., and Pentland, A. Active face tracking and pose estimation in an interactive room. In *Proceedings of the IEEE Sixth International Conference on Computer Vision*, pages 67–72. 1996.

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

[6] Fox, D. and Burgard, W. and Thrun, S. The Dynamic Window Approach to Collision Avoidance. IEEE Robotics and Automation, vol. 4, number 1. 1997.

[7] Kahn, R., Swain, M., Prokopowicz, P., and Firby, R. Gesture recognition using the perseus architecture. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 734–741, San Francisco, CA. 1996.

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

[9] Kortenkamp, D., Huber, E., and Bonasso, P. Recognizing and interpreting gestures on a mobile robot. In *Proceedings of AAAI-96*, pages 915–921. AAAI Press/The MIT Press. 1996.

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

[11] Pormeleau, D. Neural Network Perception for Mobile Robot Guidance. Kluwer Academic Publishers, Boston, MA. 1993.

[12] Rabiner, L. and Juang, B. An Introduction to Hidden Markov Models. In *IEEE ASSP Magazine.* 1986.

[13] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. In Rumelhart, D. E. and McClelland, J. L., editors, *Parallel Distributed Processing. Vol. I + II.* MIT Press. 1986.

[15] Thrun, S. Learning maps for indoor mobile robot navigation. *Artificial Intelligence.* 1998.

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

[17] Wong, C., Kortenkamp, D., and Speich, M. A mobile robot that recognizes people. In *Proceedings of the IEEE International Conference on Tools with Artificial Intelligence.* 1995.

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