A Mobile Robot That Recognizes People

Carol Wong[*], David Kortenkamp[†] and Mark Speich[‡]

[*]Lockheed Engineering and Sciences Company
2400 NASA Road 1, C-33
Houston, TX 77058
wong@mickey.jsc.nasa.gov

[†]Metrica, Inc.
Robotics and Automation Group
Houston, TX 77058
korten@mickey.jsc.nasa.gov

[‡]University of Texas – Austin
Department of Electrical and Computer Engineering
Austin, TX 78712
mspeich@mickey.jsc.nasa.gov

Abstract

In order for mobile robots to interact effectively with people they will have to recognize faces. In this paper we describe a robot system that finds people, approaches them and then recognizes them. The system uses a variety of techniques: color vision is used to find people; vision and sonar sensors are used to approach them; a template-based pattern recognition algorithm is used to isolate the face; and a neural network is used to recognize the face. All of these processes are controlled using an intelligent robot architecture that sequences and monitors the robot's actions. We present the results of many experimental runs using an actual mobile robot finding and recognizing up to six different people.

1 Introduction

A long range goal of the Robotics Architecture Laboratory at NASA Johnson Space Center is to develop technologies that will allow for effective human-robot teams in dynamically changing environments. For example, a human might be working at a repair site and ask a robot to fetch tools or spare parts from another repair site or to deliver something to a person working at another site. For robots to cooperate with people in such a manner will require that they have many skills. This paper describes the development of one of the more important skills that a robot should have as part of a human-robot team—the ability to find and recognize faces.

Face recognition has received a great deal of attention in the literature and there are many experimental methods [1, 2, 3, 4]. Face recognition on a mobile robot, however, poses several additional challenges. First, speed is important because the computational resources onboard a robot are limited and transmission of images offboard the robot is time-consuming. Second, robustness is necessary because neither the camera nor the subject are in fixed positions. Third, the face recognition problem begins not with an image of the face, but with the problem of first finding a person, second finding the face and then recognizing the face. Finally, since the face recognition process is not a stand-alone application it must fail in such a way that the robot can take additional actions to identify the person.

We have broken the problem of finding and recognizing people into four distinct sub-tasks:

1. **Locating a person.** The robot needs to find people in a large, open environment. In order to do this we require that people wear a solid color shirt for which the robot can search.

2. **Approaching that person.** When the robot sees a color that indicates a person, it must approach that person so that it can isolate their face.

1995 IEEE International Conference on Tools with Artificial Intelligence

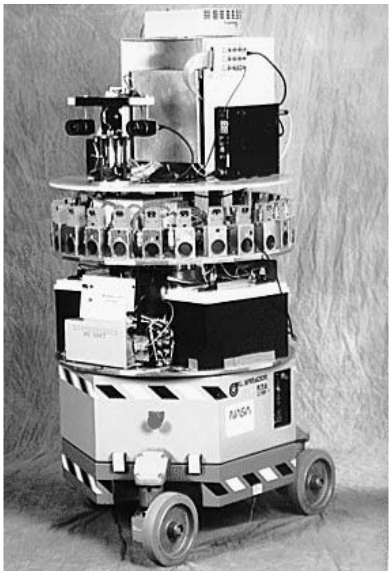

Figure 1: A mobile robot with a color vision system.

3. **Locating their face** The robot uses the colored shirt and a face-matching template to crop a 32 by 32 pixel image that contains only the person's face.

4. **Recognizing their face.** The cropped 32 by 32 pixel image is fed into a three-layer neural network that has been trained to recognize up to six different faces.

Each of these sub-tasks is described in Sections 2, 3, 4, and 5 of this paper respectively. All of them are implemented on an actual mobile robot that performs a "find-and-recognize" task within our laboratory. This robot is described in the next subsection of this paper. All robot control is done within the context of an intelligent control architecture that we are developing. This software control architecture is described in subsection 1.2 of this paper.

1.1 Hardware infrastructure

Our experimental platform is a Cybermotion K2A base with a ring of 24 sonar sensors (see Figure 1). Onboard the robot are two 68030 processors that control all non-vision related functions. On top of the robot is a color CCD camera with a 6mm lens, which gives a horizontal field of view of 58 degrees and is mounted on a pan/tilt head. A multi-axis motion controller provides azimuth and elevation control of the pan/tilt head. The color camera output video signal is in Y/C (or S-Video) format. The Y/C video signal is converted into RGB component signals and digitized by an on-board Datacube Digicolor board. Color segmentation and tracking is performed by an on-board Datacube MaxVideo 200 board. A 68040 MPU board is used as the host processor for the robot's vision system and performs the cropping and neural network computations. All vision processing and motion control is located on-board the robot.

1.2 Software infrastructure

The integration of the vision system and the mobile robot takes place within the context of an intelligent architecture we are developing for control of robots. The architecture is composed of a set of skills and methods for turning on and turning off subsets of these skills to achieve different tasks.

Skills are defined as a closed loop of software that seeks to achieve or maintain a state (either internal or external). For the task of finding and recognizing a person, the robot has skills that allow it to move around a room, to search for and track colors, to approach people and to detect and recognize faces. Subsets of these skills may be active concurrently to achieve a desired robot behavior. A system called the *skill manager* [5] is responsible for coordinating skills that are active. Section 6 gives details about our robot's skills.

The responsibility for deciding which skills should be active and when they should be activated in order to achieve a task rests with a sequencer called the Reactive Action Packages (RAP) system [6]. The RAP system takes into consideration the robot's task, its current state and the state of the world and then selects a set of skills that should run to accomplish the task.

2 Locating a person

Locating a person in a dynamic environment is a complicated task. To simplify this task, we require that the person to be recognized wear a single solid color shirt. The robot uses the color to locate and approach a person, but not to recognize them. Locating a person is split into two components: 1) A visual search from a fixed location; and 2) Systematically moving the robot to new locations within the room and then doing (1). First we will describe how the vision system uses color to locate a person within its field of view and then we will describe how we search a room for a person.

2.1 Color detection

To find a person, we use a real-time color recognition technique that detects a color by comparing every pixel in an image frame with the target color at frame rate [7]. The normalized color component (NCC) values of each pixel in an image frame are computed and are compared to a pair of target NCC values, r_o and g_o. The NCC values, $\hat{r}$ and $\hat{g}$ are defined by approximating the chromaticity coordinates as follows:

$$\hat{r} = RH,$$

$$\hat{g} = GH,$$

$$H = Q\left(\frac{1}{R+G+B}\right)$$

R, G, and B are the red, green, and blue color components, respectively. $Q(.)$ is a linear quantizer that maps a floating point value $\left(\frac{1}{R+G+B}\right)$ to a n-bit value.

The NCC values of each pixel in an image frame are compared to r_o and g_o using the following criteria:

$$p_{i,j} = \begin{cases} 1, & \text{if } r_o(1-\delta_r) < \hat{r}_{i,j} < r_o(1+\delta_r) \text{ and} \\ & g_o(1-\delta_g) < \hat{g}_{i,j} < g_o(1+\delta_g); \\ 0, & \text{otherwise.} \end{cases}$$

$p_{i,j}$ is the pixel value at location (i,j) of the resultant binary image. δ_r and δ_g are chosen such that $r_o(1-\delta_r)$, $r_o(1+\delta_r)$, $g_o(1-\delta_g)$, and $g_o(1+\delta_g)$ together define a rectangular NCC region corresponding to a group of colors that are visually indistinguishable from the color defined by r_o, g_o. The color-matching results of every pixel in an image frame are represented by a binary image with 1 indicating a positive match and 0 indicating otherwise. The binary image is then convolved with a local averaging filter to eliminate any isolated pixel areas. Positively matched pixels form a blob in the resultant image. The centroid (x,y) of the blob provides the location of the person.

2.2 Searching

The procedure described above is packaged into a robot skill (called SEARCH-COLOR) that sweeps a 180 degree area with the camera head, performing the color matching process while sweeping. If a person is located, the vision system keeps the person within the camera field of view by coordinating the pan/tilt movement of the camera head with the centroid location of the detected blob in real-time. This skill also informs the RAP system of its success or failure in finding a person. If the search was successful, then the RAP system will instruct the robot to begin approaching the person (described in the next section). Otherwise, the RAP system deactivates the SEARCH-COLOR skill, activates a skill (TURN-RELATIVE) to turn the robot 180 degrees and then reactivates the SEARCH-COLOR skill to perform another sweep.

If this second sweep is also unsuccessful, then RAPs instructs the robot to move to another position in the room. This is accomplished by activating three obstacle avoidance skills: VFH-MAP, VFH-FREE-DIR, and VFH-MOVE, which implement the VFH obstacle avoidance technique [8]. While moving, the robot is also looking for people and will stop if it sees one. When the robot attains its new position (all search positions are fixed) it again performs two visual sweeps. The robot continues this process until it finds a person.

3 Approaching a person

Once the robot has located a person, it has to approach them so that their face can be cropped. Approaching a person is a two step process. First, visual information is used to approach within 2 meters of the person. Then robot's sonar sensors can be used to approach to 1.5 meters from the person.

To approach a person visually, the RAP system activates a TRACK-COLOR skill. This visual skill moves the camera head to keep the tracked color in the center of the field of view. This skill also feeds the heading and distance of the tracked color to the obstacle avoidance and robot movement skills, which allows the robot to pursue the color while avoiding obstacles. Distance to the tracked color is roughly determined by measuring the size of the color (in pixels) in the image. The robot will continue to pursue the color until it estimates that it is within 2 meters of the color.

When the robot is within two meters of the color, the RAP system deactivates the obstacle avoidance skills and activates a special skill that will approach the person. This skill is called SONAR-APPROACH and it moves the robot in the direction of the target color while continually checking the forward sonar sensors until they read 1.5 meters. The camera head is still tracking the color and feeding updated headings to the SONAR-APPROACH skill, so the robot can still pursue the person should they move. Once the robot is at 1.5 meters from the person the SONAR-APPROACH skill will attempt to maintain that distance by moving forward or backward as the person moves. The skill also signals RAPs that it has approached the person so RAPs can begin to activate the skills for locating and recognizing a face.

4 Locating a face

Locating human faces is a difficult first step in automatic face recognition. Yet the task of locating a face is often avoided by either segmenting the image manually

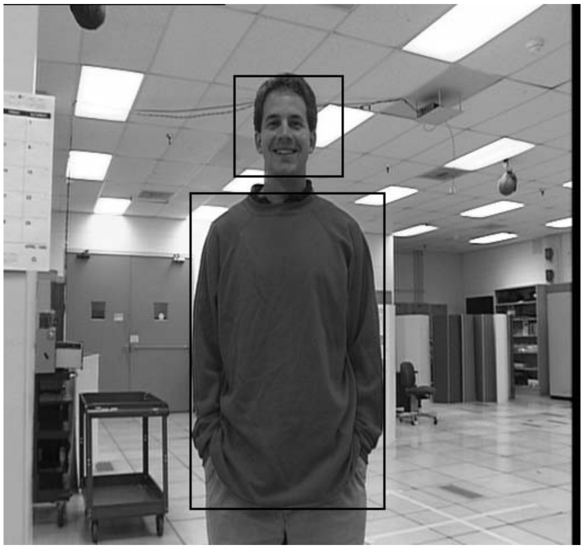

Figure 2: The face subimage (upper rectangle) is based on the rectangular boundary of the colored shirt (lower rectangle).

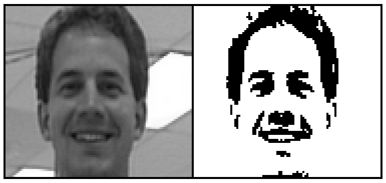

Figure 3: The greyscale subimage and the corresponding binary subimage.

or by using a known uniform background. For a mobile robot that is required to work in a dynamic environment, automatic face detection is essential to the success of the face recognition process. With the approximate location of the face computed based on the location of the colored shirt, a simple binary template matching technique, suitable for real-time computation, can effectively locate the frontal view of a human face.

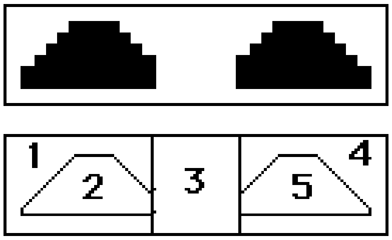

Figure 4: A binary template that models human eyes (top). The template is divided into five regions (bottom).

When the robot is approximately 1.5 meters away from the person, the vision system grabs an image frame and crops the portion of the image above the shirt (Figure 2). A wider subimage is initially cropped from the grabbed image to ensure that the entire face is included. A 96 by 90 subimage, located 18 pixels above the center of the upper edge of the rectangular boundary of the detected color shirt, is initially cropped from the grabbed image. The precise location of the face relative to the subimage is determined using a binary template matching technique. The final size of the face subimage to be cropped is 53 by 90 pixels if the robot to person distance is 1.5 meter. The sizes and the vertical location of the subimages to be cropped are scaled linearly according to the exact robot to person distance provided by the sonar sensors. The cropped color face subimage is first transformed to a grey scale subimage. The grey scale subimage is then converted to a binary subimage by comparing each pixel to a threshold. (Figure 3). In this experiment, a threshold of 70 is used.

A binary template that models human eyes is formed based on real face images of different persons. The template is divided into five regions. Figure 4 shows the 34 by 9 binary template used in this experiment. The binary template is compared, pixel by pixel for each of the five regions, to all possible 34 by 9 pixel blocks within the upper portion of the binary subimage. A pixel block is considered as a possible match if the number of matching pixels in each of the five regions is greater than a threshold. A different threshold is chosen for each of the five regions based on the total number of pixels in each region of the template. The pixel block that has the maximum total number of matching pixels is considered as the best match. If no possible match is found, the vision system reports the failure in locating a face to the *skill manager*.

The relative location of that pixel block in the subimage is used to compute the location of the face. The face subimage is then extracted from the grey scale subimage accordingly. Each extracted face subimage is shrunk to a fixed size, 32 by 32 pixels, before it is processed for recognition.

The face locating process described above has been tested on nine different persons, three women and six men. Each test begins with the robot tracking and approaching the person in a large room. This ensures that variations associated with the movements of the robot and the person, and the background environment are included in these tests. Experimental results show that the vision system is able to accurately locate the faces of eight out of the nine persons in over 90 percent of the tests. However, the vision system can only correctly locate the face of one man in the test group in approximately 20 percent of the tests. The performance of the face locating process can be improved for any particular person by adjusting the threshold used in the color to grey scale image conversion, or by adjusting the shape and the size of the binary template to be a more accurate representation of the eyes of that person. Using a bank of different templates instead of a single fixed template, that requires increasing processing time, should be able to improve the success rate of face detection.

The approach described above has been packaged as a skill called GET-FACE that can be activated by the RAP system at the appropriate time (i.e., when the robot has approached within 1.5 meters of a person). After a successful GET-FACE the RAP system will immediately activate the skill for recognizing a face, described in the next section of this paper.

5 Recognizing a face

We use a neural network to recognize faces. The earliest connectionist work on face storage and recognition was by Kohonen and his colleagues [9]. Their experiments used two layer linear networks. They showed that the network would classify images at novel orientations properly via linear interpolation. Midorikawa [10] used a three-layer back-propagation network to classify face patterns. He found that the network was robust against a large amount of noise, and that the success of the network depended on the initial weights rather than on the number of hidden units. The recent development of powerful connectionist learning algorithms such as back propagation has made it possible to program computation in networks by example rather than algorithm [11, 12]. This is especially useful when no algorithm is known. The face recognition technique described here is based on Cottrell et al work on categorization of faces using unsupervised feature extraction [11]. An image compression network is used to automatically extract image features for pattern recognition. Extracted features are used as input to a two layer network trained to distinguish faces and to attach an identity to the face image. The back-propagation neural network is implemented based on the parallel distributed processing models of McClelland and Rumelhart [13].

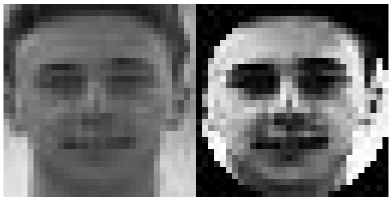

Figure 5: Left: The original face image. Right: The face image after preprocessing.

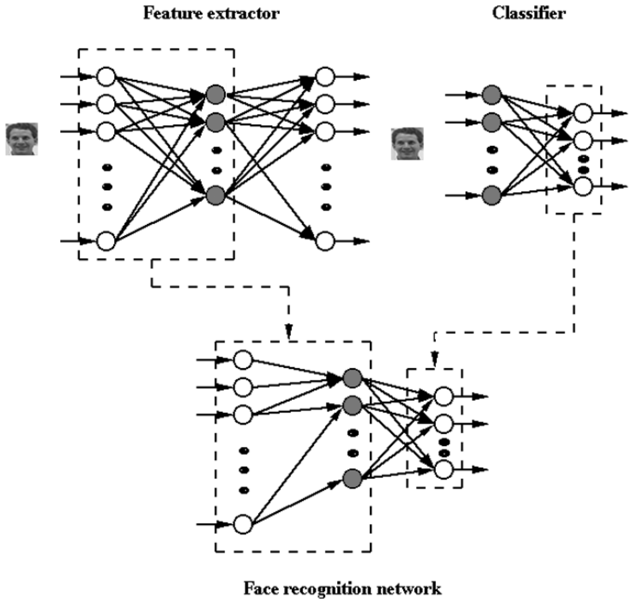

Figure 6: The architecture of the human face recognition network.

5.1 Image preprocessing

Since human heads are generally oval in shape, an elliptic mask is applied to each image such that pixels outside of the ellipse are excluded from further processing. This eliminates most of the background pixels in the image and reduces number of pixels per image to be processed from 1024 to 793 pixels. The masked face image is then normalized using histogram equalization to reduce the lighting effect on the image (figure 5). Further, the image data are converted from 255 grey scale values linearly to floating point values of the range 0 to 1 before being input to the neural network. The same preprocessing procedure is applied to all of the training and testing images.

identity	target output pattern
Linda	1 0 0 0 0 0
Eric	0 1 0 0 0 0
Dave	0 0 1 0 0 0
Carol	0 0 0 1 0 0
Jodi	0 0 0 0 1 0
Mark	0 0 0 0 0 1

Table 1: Example target output patterns of the classifier network

5.2 Network architecture

The architecture of the face recognition network (Figure 6) consists of two parts: (1) a feature extractor which computes a feature map extracted from the raw image; (2) a classifier working on the previously computed feature map. Each person is considered as a separate class. The output of the classifier gives the class, or identity, of the presented image. Such a system is able to separate classes which are not linearly separable.

The feature extractor is a three-layer backpropagation network. This network is trained to reproduce the input image on its output layer. Both the input and the output layers each has 793 units (size of a face image) and the hidden layer has 80 units. The input and output layers are much larger than the hidden layer of the network. This size differential indicates a compact internal representation of the image information at the hidden unit level. The classifier is a two-layer network. The input layer has 80 units corresponding to the 80 hidden units of the feature extractor network. The output layers has 6 units, each representing a different face identification. The classifier network is trained to match the compressed hidden unit representations of the training images with their identities. An example of target output patterns corresponding to each identity or class is shown in table 1.

The face recognition network (Figure 6) is constructed by combining the first two layers of the feature extraction network and the output layer of the classifier network. Therefore, the face recognition network is a three-layer network with a 793 X 80 X 6 architecture. The input layer has 793 units corresponding to 793 pixels of an input image. The 80 hidden units are the internal representation of that image. The 6 units in the output layer each represents the identity of one of the 6 subjects.

5.3 Training

A database of 105 face images of 8 different persons has been created with images taken by the robot's vision system using the person and face locating techniques described in the previous sections (see Figure 7). Different training data sets are generated from images in the database. Each set of training data consists of 36 images, 6 images per person. The initial weight values of the feature extraction networks are generated randomly. The weight changes are accumulated over all input images presented within an epoch, and the weights are changed only at the end of the epoch. The training of the feature extraction network is completed when the sum of square errors of all images is less than 8. The performance of the feature extraction network can also be evaluated by comparing the input image and the reproduced image visually. The hidden units of each training image generated by the trained feature extraction network form a data set of features corresponding to the training images. This feature data set is used for training the classifier network. The initial weight values of the classifier network are also generated randomly. The network is then trained until it can accurately label all of the image representations in the training set and the total sum of errors is less than 0.003. The activation level, in the range of 0 to 1, of each output unit computed by the network represents the confidence level of the corresponding

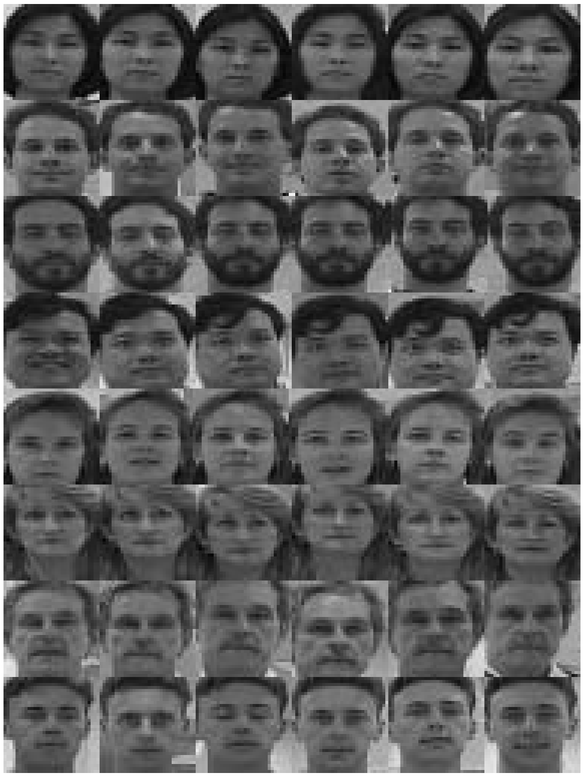

Figure 7: Part of the face database containing images taken autonomously by the mobile robot.

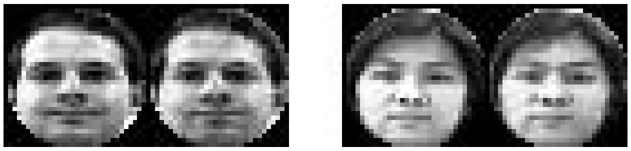

Figure 8: Two examples of input (left) and output (right) images of the feature extractor

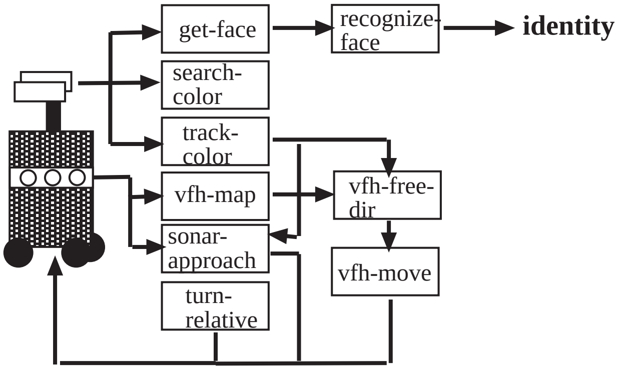

Figure 9: The skill network for our robot.

identification. An output unit is considered to be ON if its activation level is greater than 0.70. Each of the output units corresponds to a different identity. The weights of the trained feature extraction and classifier networks are used for building the face recognition network.

5.4 Results

The feature extraction and the face recognition networks have been tested separately. Figure 8 shows an example of the feature extraction network results. Test results show that the feature extraction network is able to reproduce all of the training images with only minor distortions. This indicates that the activation levels of the hidden units provide an accurate internal representation of the input training image.

The face recognition network formed by combining the feature extraction and classifier networks has been tested with both trained and untrained images in the database. The network can accurately classify all the trained images. When tested with untrained images, the network can correctly identify approximately 84 percent of the images. Similar results are obtained from networks trained with different training data sets.

6 Experimental runs with the robot

We have on-board our robot the following skills (see Figure 9):

- SEARCH-COLOR, which sweeps the camera head looking for a specific color.
- TRACK-COLOR, which moves the camera head to keep the color within the field of view of the camera.
- VFH-MAP, VFH-FREE-DIR, and VFH-MOVE, which move the robot while avoiding obstacles.
- TURN-RELATIVE, which turns the robot to a certain angle.
- SONAR-APPROACH, which approaches a person to within 1.5 meters and maintains that distance.
- GET-FACE, which crops a 32X32 pixel window that contains the face.
- RECOGNIZE-FACE, which feeds the 32X32 pixel window into a neural network that has been trained on up to six different faces.

By activating different combinations of the above skills, the robot can be made to perform any of the locating and recognizing sub-tasks. In experimental trials, we had the robot find, approach and recognize a variety of people, each many different times. Our database of images is relatively small due to the time-consuming nature of acquiring images by having the robot find and approach a person. However, because we are describing a complete robot system and not just a face recognition technique, it is essential that all of our testing be done using images that are acquired autonomously by the robot. For this reason we cannot use existing face databases for testing.

We have tested the robot system on six people, each about twelve times. Of those runs, the robot recognizes the person about 70 percent of the time. Note that the system as a whole has a higher failure rate than the neural network itself (Section 5.4), because the robot system can fail at other points along the recognition process.

Failures fall into three categories: 1) The robot does not successfully locate the face; 2) The robot thinks it has located a face, but has not; 3) The neural network fails to recognize a good face image. In the first case, the GET-FACE skill reports that it cannot locate a face and the RAP system runs that skill again. Failure to locate a face is often the result of the person moving or because the background contains the color of the shirt or a person's long hair obscuring the colored shirt. If the GET-FACE skill fails twice then the robot asks the person to move a little bit and it will try again. In the second case, the GET-FACE skill does not return an error, but it has, in fact, not correctly isolated the face. This can happen because the template-matching system has found something that looks like eyes in a different part of the image. When this happens, the RECOGNIZE-FACE skill reports that it cannot recognize the person. This is not the neural network's fault as the cropped image does not contain a complete face, however the failure is noticed by the RECOGNIZE-FACE skill. In the last case, a good face image is passed to the neural network, which simply doesn't recognize it. This often happens if someone is not looking at the camera or has a vastly different facial expression or orientation than in their training images. In the last two cases, the RAP system will invoke each skill again. If they fail a second time, the robot will ask the person to move and it will try again.

6.1 Conclusion

In order for robots to interact effectively with people they will have to be able to recognize faces. The process described in this paper is a first step in that direction. We use a variety of techniques to locate, approach, isolate and recognize people. All of these are implemented on an actual mobile robot. All of the robot's processes are controlled by an intelligent software architecture that sequences and monitors the robot's actions. In the future we hope to add additional human interaction skills, including gesture and pointing recognition and speech recognition. Then we hope to combine these skills in the context of our intelligent architecture to execute long running scenarios with human-robot teams.

References

[1] B. Moghaddam and A. Pentland, "Face recognition using view-based and modular eigenspaces," in *Automatic Systems for the Identification and Inspection of Humans, SPIE Vol. 2277*, 1994.

[2] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, , 1993.

[3] D. J. Beymer, "Face recognition under varying pose," Technical Report 1461, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1993.

[4] C.-L. Huang and C.-W. Chen, "Human facial feature extraction for face interpretation and recognition," *Pattern Recognition*, vol. 25, no. 12, , 1992.

[5] S. T. Yu, M. G. Slack, and D. P. Miller, "A streamlined software environment for situated skills," in *Proceedings of the AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space (CIRFFSS '94)*, 1994.

[6] R. J. Firby, "Task networks for controlling continuous processes," in *Proceedings of the Second International Conference on AI Planning Systems*, 1994.

[7] C. Wong, "A real-time color recognition technique," *Color Research and Application*, 1995.

[8] J. Borenstein and Y. Koren, "Histogramic inmotion mapping for mobile robot obstacle avoidance," *IEEE Journal of Robotics and Automation*, vol. 7, no. 4, , 1991.

[9] T. Kohonen, P. Lehitio, E. Oja, A. Kortekangas, and K. Makisara, "Demonstration of Pattern Processing Asymmetric Threshold Network," in *Proceedings International Conference on Cybernetics and Society*, 1977.

[10] H. Midorikawa, "The face pattern identification by back propagation learning procedure," *Neural Network*, vol. 1, , 1988.

[11] G. W. Cottrell, P. Munro, and D. Zipser, "Learning Internal Representations of Gray Scale Images: An Example of Extensional Programming," in *Proceedings Ninth Annual Cognitive Science Society Conference*, 1987.

[12] M. K. Fleming and G. W. Cottrell, "Categorization of Faces Using Unsupervised Feature Extraction," in *Proceedings IJCNN International Joint Conference on Neural Networks*, volume 2, 1990.

[13] J. L. McClelland and D. E. Rumelhart, *Explorations in Parallel Distributed Processing*, MIT Press, 1988.