Case Study: Robotic Person Following Using Fuzzy Control and Image Segmentation

M. Tarokh* and P. Ferrari
Intelligent Machines and Systems Laboratory
Department of Computer Science
San Diego State University
San Diego, CA 92182-7720
e-mail: tarokh@sdsu.edu

Received 26 April 2002; accepted 28 May 2003

The paper proposes a scheme for a mobile robot to follow a person in reasonably complex dynamic environments. The scheme consists of an image processing technique for person identification, and a fuzzy controller for following the person. The identification process utilizes image color, shape characteristics, and a region growing technique. The fuzzy logic controller receives image data such as the mass, center of mass, and their rates of change to produce steering and speed commands for the robot. A behavior control strategy is also incorporated to cope with several situations, such as hazard, that are not accounted for in the fuzzy controller. Descriptions of the software/hardware as well as experimental setup and results are provided to demonstrate the feasibility of the scheme.

1. INTRODUCTION

Mobile robots can assist a person in performing a variety of tasks, and can be used in many terrestrial and space applications. These include, for example, carrying a person's tools, delivering parts, and helping a doctor in his hospital rounds by carrying medical instruments and storing patients' data and charts. A rover assistant can aid an astronaut in his extravehicular activities by carrying tools and rock samples, supplying backup communications, providing life support equipment, acquiring and analyzing geological samples, and maintaining position data relative to the main spacecraft. One of the major requirements of such robotic assistants is the ability to track and follow a moving person through a nonpredetermined and unstructured environment. The robotic person following consists of two main tasks—person recognition and segmentation from the surrounding environment, and motion control to follow the person using the recognition results.

Research in person recognition has taken a number of forms. There has been a series of experiments using stationary cameras to track the motion of a

*To whom all correspondence should be addressed.

person.[1–3] Other approaches detect moving objects by subtracting two frames while the camera is stationary during the time these frames are captured.[4,5] Extensions of the stationary person-tracking approach include color retrieval and detection which attempt to recognize a person in a scene based on multi-modal signature of the person,[6] and the tracking and vision systems to follow the motion of a person's face based on skin color recognition.[7] Both of these approaches have had successes, but remain a stationary camera solution whereas in robotic person following the camera is attached to the robot and moves with it.

A number of robotic systems have been developed for the purpose of giving guided tours. Their focus has been mainly on navigation in crowded environments,[8] and developing successful people–robot interaction.[9] However, in the case of tour giving, it is the robot that leads the person, whereas in person following the opposite scenario takes place, and is a much more difficult problem.

A color-based tracking system capable of tracking color blobs in real time is implemented on a mobile robot,[10] but requires the person to wear a shirt of specified color and does not consider shape. An approach to recognition of a moving person by a camera mounted on a robot is provided in ref. 11 which also uses color recognition. These approaches are only effective in environments that do not contain objects whose color is similar to that of the person to be tracked.

There has also been considerable work in the area of autonomous robot navigation using fuzzy logic, neural networks, and genetic algorithms. In particular numerous fuzzy-logic base approaches have been developed (e.g., see ref. 12 for a review). Fuzzy logic has been applied to the wall following and obstacle avoidance problems.[13] Omni directional cameras, although expensive, are useful in sensing motion in every direction.[14] Such cameras allow creation of panoramic images of the environment, which can be used for navigation and control of a mobile robot. Research reported in ref. 15 uses vision to guide a mobile robot by comparing images to a database of images that are created during an initialization tour of the environment. Regardless of the approach, navigation and tracking using maps require that the environment be known prior to application, which limits flexibility and is not a valid approach to person following.

The main focus of the present paper is the development of a fuzzy robot control system to achieve person following using a simple real time image processing. Although the problem of person identification in a dynamic scene has been studied and there exists several sophisticated techniques for this task, however, most of them require specialized vision processing electronics.[16] In this paper, we present a simple real-time method for segmenting a person from the scene, and identifying and extracting relevant quantities for the use by fuzzy person following control.

2. PERSON SEGMENTATION

Consider a mobile robot equipped with a camera that is required to follow a person in relatively flat terrain such as those inside buildings or paved outdoor locations. We use both color and shape of the person's clothes (e.g., shirt or jacket) for identification. We apply an iterative thresholding method[17,18] to automatically select threshold values and to remove all objects in the scene that have colors (or gray level values in the case of a monochrome image) different from the person's shirt color. Since it is possible to have objects in the environment that have colors similar to the person's clothes, we segment the thresholded image into regions using standard region growing by planting seeds and aggregating the pixels (see, e.g., ref. 18). We then use several shape measures to select the region that has shape characteristics closest to the shape of the person. These measures are defined later in this section. The resulting processed image at each sample time is a binary image $f(x,y)$ whose value is either L, or 0 depending whether the pixel at location (x,y) belongs to the person, or is a background pixel, where L is the color (or graylevel) value. The x coordinate is across the field of view and the y coordinate is along the direction of the robot motion.

For image recognition, we use several image characteristics as follows. The mass of the image is defined as

$$M = \sum_{x=0}^{p} \sum_{y=0}^{q} f(x,y), \tag{1}$$

where M is the total image mass, p is the number of rows, and q is the number of columns in the image. Determining the mass of an image is important for person following because it gives a mearure of how close the camera is to the person being followed. A large mass is indicative of a person that is close to the camera mounted on the robot, whereas a small mass implies that the person is far away.

The center of the mass (X_c, Y_c) of the image is defined as

$$X_c = \frac{1}{M}\sum_{x=0}^{p}\sum_{y=0}^{q} yf(x,y),$$
$$Y_c = \frac{1}{M}\sum_{x=0}^{p}\sum_{y=0}^{q} xf(x,y). \tag{2}$$

The center of mass is also of importance for person tracking because it provides the coordinates of the point to be tracked by the robot. In addition to the mass and its center, we are also interested in finding how these quantities change from the image taken at one instant of time t, to the image taken at the next time $(t+T)$ where T is the sample time. Suppose that the mass and its center at these two times are (M_1, X_{c1}, Y_{c1}) and (M_2, X_{c2}, Y_{c2}), respectively. Then the rates of change are

$$\dot{M} \equiv \frac{dM}{dt} \approx \frac{1}{T}(M_2 - M_1),$$
$$\dot{X}_c \equiv \frac{dX_c}{dt} \approx \frac{1}{T}(X_{c2} - X_{c1}), \tag{3}$$
$$\dot{Y}_c \equiv \frac{dY_c}{dt} \approx \frac{1}{T}(Y_{c2} - Y_{c1}),$$

where T is small. The mass rate $\dot{M}$ measures the change in the distance D_{rp} between the robot and the person. A positive $\dot{M}$ is indicative of the robot getting closer to the person whereas a negative $\dot{M}$ implies that the person is moving away from the robot. The magnitude of $\dot{X}_c$ represents how quickly the person is moving across the camera's field of view. A positive $\dot{X}_c$ indicates that the person moves from left to right, whereas a negative value implies motion in the other direction. The quantity $\dot{Y}_c$ represents how quickly the person is moving away or towards the robot (camera). When $\dot{Y}_c$ is positive, the person is moving away from the robot, whereas a negative $\dot{Y}_c$ indicates that the robot is getting closer to the person.

It is to be noted that $\dot{Y}_c$ provides similar and complementary information to that of $\dot{M}$. However, the mass M is more sensitive than y-center of the mass Y_c to variations in D_{rp}. This is due to the fact that mass is a measure of a two dimensional quantity (i.e., area), whereas Y_c is a measure of a one-dimensional quantity. This implies that the variation of M with D_{rp} is essentially quadratic whereas the variation of Y_c with this distance is in essence linear. Although the sensitivity of M to distance is desirable, however, this sensitivity causes noise when the derivative of mass, $\dot{M}$, is formed. As a result, we use M as a measure of robot–person distance D_{rp}, and $\dot{Y}_c$ as a measure of the rate of change of this distance.

Since there may be several regions in the image that have a color similar to the person's clothes, thresholding of the image is followed by region growing, as mentioned before. The identified regions are then evaluated using shape measures to determine if the region represents the person. These measures must be independent of the mass (area) of the image since the mass changes with the distance of the robot to the person. The three measures that satisfy this requirement are compactness, circularity, and eccentricity. We define the compactness K as

$$K = \frac{16M}{n^2}, \tag{4}$$

where M is mass as before, and n is the number of pixels on the boundary of the region. A square has the highest compactness with $K=1$, and a highly noncompact region has $K=0$. We define the circularity C as

$$C = \left(1 + \frac{1}{nd_a}\sum_{i=1}^{n} |d_i - d_a|\right)^{-1}, \tag{5}$$

where d_i is the distance of the mass center to a pixel on the boundary of the region, n is the number of the pixels on the boundary, and $d_a = (1/n)\Sigma_{i=1}^{n} d_i$ is the average distance of the center to the boundary. A perfectly circular region has $C=1$, and a highly noncircular region has $C=0$. Finally, we define eccentricity as

$$E = 1 - \frac{d_{\min}}{d_{\max}}, \tag{6}$$

where $d_{\max}$ and $d_{\min}$ are, respectively, the maximum and minimum distances of the center of the region to its boundary. A highly eccentric region, where $d_{\min} \ll d_{\max}$, has $E=1$, and the minimum value of E is zero. Note that other mass-independent shape measures can also be used instead of or in addition to the above three measures.

The process of determining the shape measures of the person is as follows. Before the start of person following, several images of the person in typical poses are taken with plain background, i.e., no objects

in the background. For these typical poses, each shape measure is found and is then averaged to get the reference shape measure values K_r, C_r, and E_r. During person following, the regions R_j, $j = 1,2,...,m$, detected through the iterative thresholding method described before, are grown and their shape measures K_j, C_j, and E_j are evaluated using (4)–(6), where m is the number of regions. In order to determine which of these regions corresponds to the person, we evaluate a closeness measure defined as

$$\sigma = \tfrac{1}{3}(e^{-a_k\delta_k} + e^{-a_c\delta_c} + e^{-a_e\delta_e}), \qquad (7)$$

where $\delta_k = |K_r - K_j| / K_r$, $\delta_c = |C_r - C_j| / C_r$ and $\delta_e = |E_r - E_j| / E_r$ are the relative deviations from the reference shape parameters, and a_k, a_c, and a_e are user specified coefficients that determine the relative weighting of each shape measure to the closeness factor. Note that the particular closeness function chosen in (7) produces 1 if all shape measures of the region are the same as the reference value, and approaches zero if the region shape measures are completely different. The region that has the largest value of σ is selected, and if this value is close to 1, the selected region is assumed to represent the person. The exponential function provides high sensitivity to mismatch between the region and the reference shape. If all the regions have small values of σ, then none are chosen and another image is taken and analyzed.

The above method of distinguishing the region corresponding to the person to be followed from other detected regions in the image is simple and yet quite effective. There are several reasons for this effectiveness. One is that the robot is controlled reasonably close to the person being followed and in the direction of person's motion, as will be seen in the next section. This allows only a few objects in the camera's view, making the person identification reasonably easy. Furthermore, the simplicity of image processing tasks allows fast computation, making it possible to achieve relatively high sample rates. This in turn allows identification of the person based on several images taken during one control sample time, rather than just a single image.

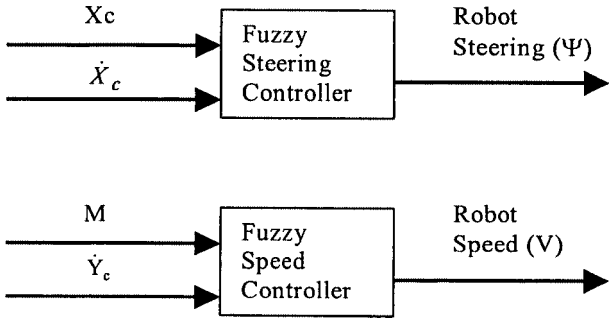

Figure 1. Block diagram of the two main controllers.

3. FUZZY PERSON-FOLLOWING CONTROL

In this section, we utilize the image information, namely image mass, its center, and their derivatives, to control the motion of the robot for person following. We also, discuss certain behavior control aspects to deal with hazards and unsafe motions. It will be assumed that the camera is mounted on the robot and is fixed with respect to the robot.

There are essentially two independent high level controllers, one for robot steering and the other for robot speed, as shown in Figure 1. The steering controller uses the image center of mass in the x-direction which is where the person is relative to the field of view at the time of image acquisition. It also receives the change in the center of mass in the x-direction which is the speed of the person motion across the field of view. These two inputs determine the robot steering (rotation) Ψ, in order to position the camera so that the person is close to the center of the field of view. The purpose of the speed controller is to adjust

Table I. Fuzzy set definitions for inputs and outputs of steering and speed controllers.

Fuzzy variable	Set 1	Set 2	Set 3	Set 4	Set 5
$\tilde{X}_c$	*FarLeft*	*Left*	*Center*	*Right*	*FarRight*
$\tilde{\dot{X}}_c$	*RapidLeft*	*Left*	*NoChange*	*Right*	*RapidRight*
$\tilde{M}$	*VerySmall*	*Small*	*Medium*	*Large*	*VeryLarge*
$\tilde{\dot{Y}}_c$	*RapidClosingIn*	*ClosingIn*	*Steady*	*MovingAway*	*RapidMovingAway*
$\tilde{\Psi}$	*BigTurnLeft*	*TurnLeft*	*NoChange*	*TurnRight*	*BigTurnRight*
$\tilde{V}$	*VerySlow*	*Slow*	*Nominal*	*Fast*	*VeryFast*

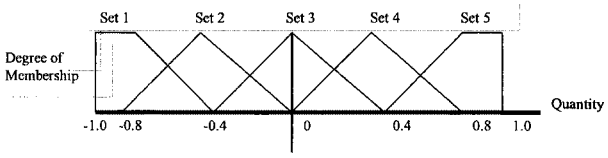

Figure 2. Membership functions of the fuzzy sets.

the robot speed V according to the speed of the person so as to maintain a reasonably constant distance between the robot and the person. This controller uses the image mass M which provides a good measure of the distance of the person to robot, as discussed in the previous section. However, for the rate of distance change, we use $\dot{Y}_c$ since $\dot{M}$ is quite noisy due to the sensitivity of M, as explained before. Note that fuzzy control allows the use of different inputs for the quantity and the rate, i.e., M and $\dot{Y}_c$ in the case of speed control.

The input and output quantities to the controllers are fuzzified each using five overlapping fuzzy sets. The fuzzy variables, distinguished by a tilde, and their associated fuzzy sets are defined in Table I. The normalized membership functions for $\tilde{X}_c$, $\tilde{\dot{X}}_c$, $\tilde{\dot{Y}}_c$, and $\tilde{\Psi}$ are given in Figure 2 and their range is from –1 to 1. The normalized membership functions for $\tilde{M}$ and $\tilde{V}$ are similar to those of Figure 2, but their range is from 0 to +1, as these two quantities are always positive.

The general fuzzy rule for the steering controller is of the form

if $\tilde{X}_c$ *is "Location" and* $\tilde{\dot{X}}_c$ *is "LocationChange," then* $\tilde{\Psi}$ *is "TurnAction,"*

where *Location, LocationChange* and *TurnAction* take on, respectively, one of the fuzzy sets associated with $\tilde{X}_c$, $\tilde{\dot{X}}_c$ and $\tilde{\Psi}$ given in Table I. The rule matrix is devised to relate the inputs to the outputs (actions). The rule matrix for the steering controller is given in Table II. The amount and direction of steering, given in Table II for various combinations of $\tilde{X}_c$ and $\tilde{\dot{X}}_c$, can be explained in a straightforward manner. For example, when the person is to the right of the robot ($\tilde{X}_c$ is *Right*) and is moving to the right ($\tilde{\dot{X}}_c$ is *Right*), then a large steering to the right ($\tilde{\Psi}$ is *BigTurnRight*) is needed to follow the person.

The purpose of the speed controller is to keep the distance between the robot and the person relatively constant. The general fuzzy rule for the speed controller is of the form

if $\tilde{M}$ *is "Magnitude" and* $\tilde{\dot{Y}}_c$ *is "LocationChange," then* $\tilde{V}$ *is "SpeedAction,"*

where *Magnitude, LocationChange* and *SpeedAction* take on, respectively, one of the fuzzy sets associated with the fuzzy variables $\tilde{M}$, $\tilde{\dot{Y}}_c$, and $\tilde{V}$ given in Table I. The rule matrix for the speed controller is given in Table III. For example, when the mass is small (i.e., large distance) and the person is moving away rapidly, then the robot must move very fast to reduce the distance. On the other extreme, when the mass is large (i.e., small distance) and the robot is rapidly closing in, then the speed must be reduced substantially. This action corresponds to the fuzzy set *VerySlow* in the rule matrix.

The defuzzification is performed using the center of gravity method, e.g., the crisp value of the speed (or steering), is obtained from

$$V = \frac{\Sigma_i c_i A(\mu_i)}{\Sigma_i A(\mu_i)}, \tag{8}$$

where the summation is performed over all fuzzy sets whose rules are fired, c_i is the center of the membership function, and $A(\mu_i)$ denotes the area under the clipped membership function of the fired fuzzy set.

Table II. Rule matrix for steering control.

		$\tilde{X}_c$				
		FarLeft	*Left*	*Center*	*Right*	*FarRight*
$\tilde{\dot{X}}_c$	*RapidLeft*		*BigTurnLeft*	*BigTurnLeft*	*TurnLeft*	*NoChange*
	Left		*BigTurnLeft*	*TurnLeft*	*NoChange*	*TurnRight*
	NoChange	*BigTurnLeft*	*TurnLeft*	*NoChange*	*TurnRight*	*BigTurnRight*
	Right	*TurnLeft*	*NoChange*	*TurnRight*	*BigTurnRight*	
	RapidRight	*NoChange*	*TurnRight*	*BigTurnRight*	*BigTurnRight*	

Table III. Rule matrix for speed control.

		$\tilde{M}$				
		VerySmall	*Small*	*Medium*	*Large*	*VeryLarge*
$\tilde{\dot{Y}}_c$	*RapidMovingAway*		*VeryFast*	*VeryFast*	*Fast*	*Nominal*
	MovingAway		*VeryFast*	*Fast*	*Nominal*	*Slow*
	Steady	*VeryFast*	*Fast*	*Nominal*	*Slow*	*VerySlow*
	ClosingIn	*Fast*	*Nominal*	*Slow*	*VerySlow*	
	RapidClosingIn	*Nominal*	*Slow*	*VerySlow*	*VerySlow*	

For a symmetric triangular membership function with a maximum height of 1, and a clipped height of h_i, simple geometry gives $A(\mu_i) = w_i h_i (1 - h_i/2)$.

The crisp values of the rover steering and speed are in fact the set points for the corresponding lower level controllers. The details of the implementation are provided in Section 4.

3.1 Behavior Control

In addition to following the person, the robot must avoid collision with walls and other objects, and furthermore must not come very close to the person. The latter implies that when the person stops, the robot must also come to a halt at a reasonable distance from the person. In addition, the robot must produce an appropriate action if the person is not seen by the camera, or is not detected by the image processing system. To account for these various conditions, we employ a behavior control system. There are four inputs to the behavior controller, namely, image mass M, image y-center of the mass Y_c, average distance of robot to object D_{ro} as measured by sonar sensors or other proximity sensors, and a binary signal indicating if the image of the person is found. These behaviors are described as follows.

Explore/Alarm: If the image processing software is unable to detect the person after several attempts, it is concluded that the person is not in the camera's view and the behavior "explore" is activated. In explore behavior the robot stops, and the camera is rotated and zoomed until the person is in the camera's view. This requires the availability of a pan-tilt-zoom camera. If such a camera is not available, as is the case with our experimental setup, then the robot must be rotated on the spot using steering. If for any reason the system is not able to identify the person after several sample times, then it sounds an alarm, signaling to the person that he should come in the camera's view.

Hazard: This behavior is activated if certain conditions exist on M, Y_c and D_{ro}. These conditions can be, for example, that one of the three quantities (M, Y_c, D) is outside its safe range (i.e., M larger than a threshold, or Y_c and D_{ro} smaller than some limits). Alternatively, this behavior can be activated if two of the three quantities show unsafe values. The hazard behavior will result in the immediate stopping of the robot. If the problem is the robot getting close to the person as detected by the vision system (M and Y_c), then new images are taken at regular intervals until the person moves again and M and Y_c are within their safe values. On the other hand, if the hazard is due to collision detection by the sonar sensors, then a collision avoidance maneuvering takes place. This maneuvering, can be, for example, turning away from the obstacle and then turning towards the person. Alternatively any other available collision avoidance techniques can be incorporated.

Normal: If the person is detected and no hazard is present, this behavior is activated. In the normal behavior, the robot will perform the person following, i.e., images are taken, mass and its center and their derivative are found, these quantities are fuzzified, and the rules are applied to finally determine the next steering and speed set points.

4. HARDWARE AND SOFTWARE ENVIRONMENTS

In order to test the applicability of the ideas discussed above and evaluate the performance of the system, we have implemented the person-following scheme on a small mobile robot. Figure 3 shows the block diagram of the person-following system which consists of two main parts, i.e., the robot and the computer (PC) each contain various components of the person-following system. The PC is stationary and is located separately from the robot. The components of the system are camera, imaging board, transmitters and receivers, person-following software, and low level controllers.

The robot is a Pioneer 2DX mobile robot from

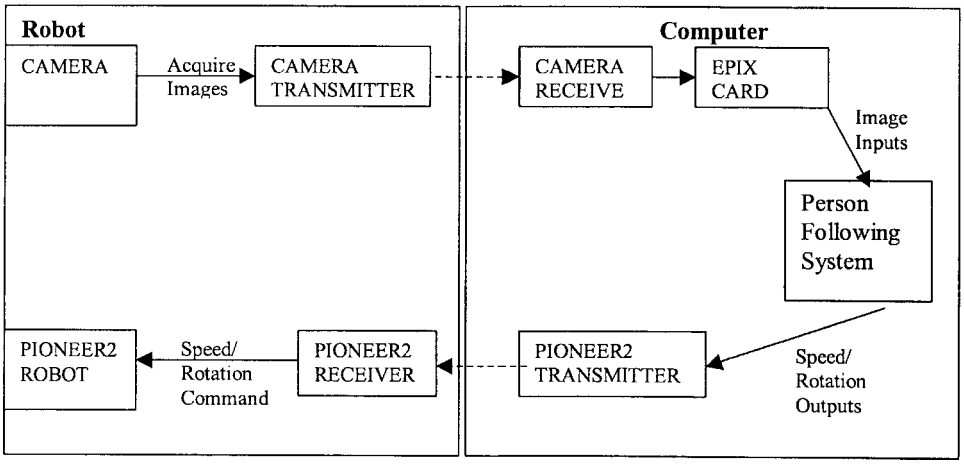

Figure 3. Block diagram of the overall person following system.

Active Media and is 21.5 cm high, 38 cm wide, and 45 cm long. The Pioneer is a two-wheel drive with separate motors for each wheel. It also has a third caster free wheel with no drive. It is controlled from the PC via a radio modem. There is one modem onboard the robot that is hardwired to the motor controller board of the robot. The other radio modem is located next to the PC and is connected via a serial cable to a serial port on the PC. This permits two-way communications between the robot and the computer. For this project the majority of the transmissions are steering and speed commands from the computer to the robot, and a set of sonar readings from the robot to the computer. The robot is equipped with an array of eight sonar sensors whose range can be adjusted from 10 cm to 5 m. The sonar sensors are used in our project to detect objects, walls, etc. for collision avoidance.

The Pioneer is equipped with a software program called Saphira that contains C++ library functions for handling communications between the PC and the robot. It also contains a two-level control architecture, a higher level for receiving set points and a lower level for the interaction of a micro-controller with the drives motors. In the person-following project, we communicate with the higher level component.

The vision system, GFP-5005 from Polaris Industries, was separately purchased and integrated into the person-following system. It consists of a color CCD camera with a built-in transmitter and a separate receiver unit. We have mounted the camera directly on the Pioneer robot, as shown in Figure 4. The vision system is capable of using either NTSC or PAL video format. The vision transmitter and receiver operate on a frequency of 2.4 MHz, and have a range of about 100 m for line of sight operation and about 30 m through walls and doors.

An imaging board from Epix Inc. was also separately purchased and installed on the PC for interfacing between the camera and the PC. It is capable of acquiring live or snapped images in a number of formats, and can be configured for a variable number of frame buffers depending on the image size and resolution. All frame buffers are accessible in RAM at any time using a C++ software called PIXIPL. This software consists of a number of library functions for enhancing, filtering, analysis, and file manipulations. The person-following software utilizes some of the functions in this library.

The fuzzy person follower is written in Microsoft Visual C++ to facilitate communication and interfacing with image processing library (PIXIPL) and robot control software (Saphira), both of which are written in C++. In addition, Microsoft Visual C++ with its Foundation Class makes creation of the visual environment with dialog boxes and graphic display

Figure 4. The robot and the mounted camera.

possible. This is an important tool for displaying the images after acquisition and processing. An additional advantage of C++ is its rapid processing speed. This is critical for motion control applications since it ensures that the cycle time and processing speed will be the same for every iteration of the program, and that the program will not interrupt the control application in order to handle data clean up, storage, or automatic garbage collection. The fuzzy person follower presents the user with graphic interface, allowing the user to enter inputs or select inputs from a menu. It will then coordinate the process of image acquisition and analysis. The results of image analysis are then used by the fuzzy controllers, which in turn determine the set points for the low level steering and speed controllers. These values are sent to the Saphira robot controller for execution on the motors.

5. EXPERIMENTAL RESULTS

The tests performed consisted of a person who would tour a laboratory, exit the laboratory, and walk down the length of a hallway in a serpentine path. The person would then turn left around a corner into another hallway. The total distance traveled in each experiment was about 30 m. The robot was expected to follow the person while maintaining a distance of about 2 m behind the person without hitting walls or other objects. If the person were to come to a complete stop, the robot was expected to zero in on the person's position and come to a stop at about 1.5 m from the person. The robot was also expected to avoid hitting the tables and other objects in the laboratory, and the walls in the hallway.

The images produced are 8-bit graylevel with the image size of 752 by 480 pixels. The origin of the image is placed at the center, thus X_c and Y_c range (−376 to +376) and (−240 to +240), respectively, where the lower limit corresponds to the normalized −1, and the upper limit corresponds to normalized +1. The ranges of $\dot{X}_c$ and $\dot{Y}_c$ are (−500 to +500) and (−250 to +250), respectively, which correspond to the normalized range of (−1 to +1). The mass ranges from (0 to 50 000) pixels, corresponding to normalized values of 0 to +1. Finally, the sampling time for the experiments was chosen as $T = 0.3$ s. In fact using a 700 MHz PC, it was possible to perform all image processing and fuzzy and behavior control calculations in about 0.15 s. However, for our experiments twice this time enabled the robot to follow accurately the person who walked at a speed of about 0.8 m/s.

In the experiments, the person wore a black shirt, as shown in Figure 5, but any other plain colored clothes would be equally acceptable. Figure 5 shows several images taken by the robot's camera during several experiments in different locations along the tour route and involving different desired robot-to-person distances. Figure 5(a) is taken at the start of the tour in the laboratory where the person is identified from the somewhat similar shape and color of the back of a chair. In Figure 5(b) the person being followed is segmented from the other persons and the announcement board that are dark colored. Similarly Figures 5(c) and 5(d) show other persons in dark clothes, but the system has been able to distinguish them from the person being followed using the shape characteristics. Finally, Figure 5(e) shows a scene where the system failed to identify the person due to the difficult situation of occlusion and overlapping. In this case the robot continued the motion in the same direction and with same speed as the previous sample time. However, as severe occlusion persisted for several sample times, the robot stopped and sounded an alarm, signaling to the person that he should come in the camera's view. Finally, the system was also used for person following with different people, and Figure 5(f) shows another person starting his tour in the laboratory.

A typical sequence of eight consecutive images after color and shape recognition and negatization are presented in Figure 6 which show typical motions of the person. Note that the shape parameters have enabled the system to discard several regions in the image that do not satisfy the shape characteristics of the person.

The non-normalized parameters for the images in Figure 6, namely mass, its center, and the change of the center are given in Table IV. For illustration, typical fuzzy values for images in Figures 6(a) and 6(b) are provided in Table V where Set 1, Set 2, etc. are defined in Table I. The last two rows of Table V are the resulting clipped fuzzy sets for steering and speed where the given values represent the heights of the appropriate clipped membership functions. Finally, Table VI shows the crisp values of the steering and speed commands for the images of Figures 6(a)–6(h). By examining Figure 6 and Table VI, we see that the person starts out to the right of the camera's view [Figure 6(a)], but then moves to slightly the left [Figure 6(b)]. This corresponds to a steering command of 0.2 degree, which is very small to keep the person close to the center of camera's view. Then as the person moves to the right again [Figures 6(c)–6(f)], the robot steers by −2.5 degrees and again by −2.6

Figure 5. Images taken by the robot's camera during several experiments in different locations along the tour route.

degrees, where the negative values represent steering to the right. Finally, the person starts heading off to the left [Figures 6(g) and 6(h)], and the robot starts steering to the left by 7.7 degrees, as one would expect. The speed of the person ($\dot{Y}_c$) in the Figures 6(a) and 6(b) is the highest (see Table IV), therefore the speed of the robot is also the largest. As the speed of the person ($\dot{Y}_c$) begins to decrease, so does the speed of the robot until in Figures 6(g) and 6(h) the mass of the person has decreased considerably, causing the robot to speed up in order to catch up with the person.

Note that the results provided above represent a small typical sample of about 120 images covering a traversed distance of about 30 m during one of the experiments involving the laboratory and hallway tour. We performed some ten different experiments each involving above 30 m of traversal, some in reasonable cluttered environments such as those shown in Figure 5. In these experiments, the person recognition algorithm performed quite well, despite existence of several objects and persons in the scene that had similar color to the person's clothes. It was observed that during the whole trials, the robot followed the person very well and maintained a safe distance, despite several occasions where the person suddenly stopped or made relatively quick turns. Only in one of these experiments was the vision system unable to identify the person, causing the robot to stop and send a sound signal asking the person to come into the camera's view.

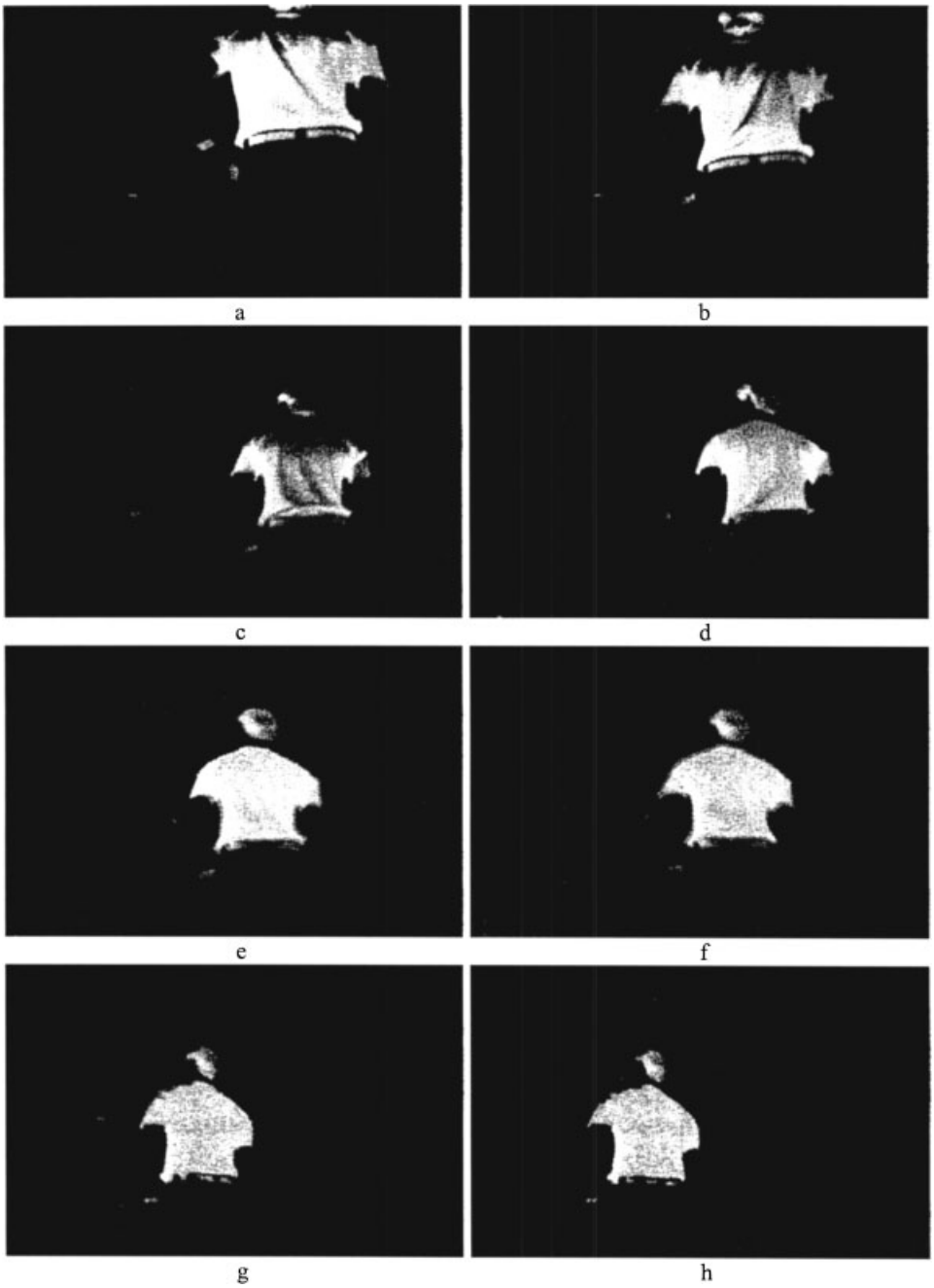

Figure 6. Sequence of images after color and shape recognition.

6. DISCUSSION AND CONCLUSIONS

We have presented a scheme for a person-following robot moving in a previously unknown environment. The scheme consists of a simple and fast person recognition using both color and shape measures, and a fuzzy logic controller for motion control. We have implemented the scheme on a mobile robot and performed various experiments in reasonably cluttered environments and have demonstrated the feasibility of the proposed scheme. We have also identified several important image characteristics for satisfactory person following control. The main benefit of the fuzzy logic person follower is its robustness to complex and uncertain situations. For example, there are

Table IV. The parameters of images (a)–(h) in Figure 6.

Image	X_c	Y_c	M	$\dot{X}_c$	$\dot{Y}_c$
6(a)	116	−108	26 620	−110	213
6(b)	94	−65	21 560		
6(c)	134	10	10 580	−32	−70
6(d)	124	−10	15 680		
6(e)	64	0	20 780	58	−16
6(f)	74	1	18 480		
6(g)	−26	30	14 900	−150	15
6(h)	−56	31	14 870		

instances when the person may become partially occluded, or other moving persons are in the scene. Despite this, the fuzzy person-following scheme functions satisfactorily provided there is enough motion in the scene for adequate detection. However, there are difficult situations where the vision system cannot identify the person, but these are infrequent.

The emphasis of the person segmentation presented here is on simplicity, speed of processing, and identifying quantities such as mass, its center coordinates, and their derivative which are very suitable for the use by the fuzzy controller. However, when the scene includes persons or objects that are very similar in color and shape to the person being followed such a simple method may not segment the person from the scene, and it is possible for the system to misidentify. In such cases, more sophisticated image processing will be required. However, such image processing methods entail the use of specialized vision processing hardware for real time person-following control, which substantially add to the cost and complexity of the system. On the other hand, the relative simplicity of the proposed method allows several image acquisitions and processing during each speed and steering control period. This makes the system less prone to misidentification, and results in better overall control.

Table V. The result of fuzzification for the parameters given in the first two rows of Table IV. Also shown are the fuzzy steering and speed clipped sets.

Quantity	Set 1	Set 2	Set 3	Set 4	Set 5
X_c	0	0.	0.61	0.39	0
$\dot{X}_c$	0	0.44	0.56	0	0
M	0	0.62	38	0	0
$\dot{Y}_c$	0	0	0	0.3	0.7
Ψ	0	0.44	0.56	0.39	0
V	0	0	0	0.3	0.62

Table VI. Crisp values of steering and speed for the images in Figure 6 and data in Table IV.

Images	6(a) and 6(b)	6(c) and 6(d)	6(e) and 6(f)	6(g) and 6(h)
Steering (deg)	0.2	−2.5	−2.6	7.7
Speed (cm/s)	110	74	76	96

In the existing setup most image processing and control calculations are done on a PC located in a laboratory separate from the moving robot, as show in Figure 3. This requires wireless transmission of images and other signals between the robot and the PC. The sample time can be reduced if a lap top PC is placed directly on the robot, thus eliminating the need for signal transmission. This is currently being implemented.

The method described in the paper assumes that the robot moves on a relatively flat terrain as explained before. For person following over uneven terrain, articulate rovers with a camera having an actuated base are required to adjust the tilt and pan to account for the terrain topology. Articulate robots traversing bumpy or rocky terrain, such as those used in space applications,[19] are generally equipped with accelerometers or inertial navigation systems for sensing pitch and roll. Alternatively pitch and roll can be computed using the robot kinematics and sensed quantities such as robot linkage joint angles.[20] The pitch and roll measurements can be used to control the tilt and pan so as to adjust the camera orientation to account for changes in the terrain topology. This is currently under investigation.

REFERENCES

1. N. Jakobi, Evolving motion-tracking behavior for a panning camera head, in Simulation of Adaptive Behavior, 1998.
2. D. Koller, J. Weber, and J. Malik, Robust multiple car tracking with occlusion reasoning, Proc. 3rd European Conference on Computer Vision, Sweden, 1994, pp. 189–196.
3. J. Little and J. Boyd, Recognizing people by their gait: the shape of motion, J Comput Vision Res, Videre 1(2), Winter 1998.
4. Q. Cai, A. Mitchie, and J.K. Aggrarwal, Tracking human motion in an indoor environment, 2nd Int.

Conf. on Image Processing, October 1995.
5. C. Richards, C. Smith, and N. Papaikolopoulos, Detection and tracking of traffic objects in IVHS vision sensing modalities, Proc. 5th Annual Meeting of ITS America, 1995.
6. J. Matas, D. Koubaroulis, and J. Kittler, Color image retrieval and object recognition using the multimodal neighborhood signature, Proc. 6th European Conf. on Computer Vision, Dublin, Ireland, 2000.
7. R. Stiefelhagen, J. Yang, and A. Waibel, Tracking eyes and monitoring eye gaze, Proc. Workshop on Perceptual User Interfaces, 1997.
8. J. Schulte, C. Rosenberg, and S. Thrun, Spontaneous short-term interaction with mobile robots, Proceedings of IEEE Int. Conference on Robotics and Automation, Detroit, Michigan, 1999.
9. S. Thrun, When robots meet people: Research directions in mobile robots, IEEE Intell Syst, May–June 1998.
10. C. Schlegel, J. Illmann, H. Jaberg, M. Schuster, and R. Worz, Integrating vision based behaviors with an autonomous robot, J Comput Vision Res, Videre 1:(4) (2000), 32–60.
11. R. Tanawongsuwan, A. Stoytchev, and I. Essa, Robust tracking of people by a mobile robotic agent, College of Computing, Georgia Institute of Technology, February 1999.
12. A. Saffiotti, The uses of fuzzy logic in autonomous robot navigation: a catalogue raisonné, Technical Report 2.1, IRIDIA. Universite Libré de Bruxelles, Brussels, Belgium, November 1997.
13. R. Braunstingl, P. Sanz, and J. M. Ezkerra, Fuzzy logic wall following of a mobile robot based on the concept of general perception, Proc. 7th Int. Conf. on Advanced Robotics, Spain, 1995, pp. 367–376.
14. J. Gaspar, N. Winters, and J. Santos-Victor, Vision-based navigation and environmental representation with omni-directional camera, IEEE Trans Robot Automat 16:(6) (2000), pp. 722–730.
15. J. Weng and S. Chen, Vision-guided navigation using SHOSLIF, Neural Networks 1 (1998), pp. 1511–1529.
16. Pyramid Vision Technology, A division of Sarnoff Company, http://www.pyramidvision.com/products/examples.htm
17. J. R. Parker, Algorithms for image processing and computer vision, Wiley, New York, 1997.
18. R. Jain, R. Kasturi, and B. Schunk, Machine Vision, McGraw-Hill, 1995.
19. S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, R. Petras, and S. Laubach, The Rocky 7 Rover: A Mars sciencecraft prototype, Proc. IEEE Int. Conf. on Robotics and Automation, Albuquerque, NM, April 1997, pp. 2458–2464.
20. M. Tarokh, G. McDermott, S. Hayati, and J. Hung, Kinematic modeling of a high mobility Mars rover, Proc. IEEE Int. Conf. on Robotics and Automation, Detroit, MI, May 1999.