Robotica (2009) volume 27, pp. 715–727. © 2008 Cambridge University Press
doi:10.1017/S0263574708005092 Printed in the United Kingdom

Multi-agent system for people detection and tracking using stereo vision in mobile robots

R. Muñoz-Salinas†,*, E. Aguirre‡, M. García-Silvente‡, A. Ayesh$ and M. Góngora$

†*Department of Computing and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain.*
‡*Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain.*
$*Intelligent Mobile Robots and Creative Computing Research Group, Computer Engineering Division - School of Computing, De Montfort University, Leicester, UK.*

(Received in Final Form: August 21, 2008. First published online: September 30, 2008)

SUMMARY
People detection and tracking are essential capabilities in order to achieve a natural human–robot interaction. A great portion of the research in that area has been focused on monocular techniques. However, the use of stereo vision for these purposes concentrates a great interest nowadays. This paper presents a multi-agent system that implements a basic set of perceptual-motor skills providing mobile robots with primitive interaction capabilities. The skills designed use stereo and ultrasound information to enable mobile robots to (i) detect an interested user who desires to interact with the robot, (ii) keep track of the user while they move in the environment without confusing them with other people, and (iii) follow the user along the environment avoiding obstacles in the way. The system presented has been evaluated in several real-life experiments achieving good results and real-time performance on modest computers.

KEYWORDS: Stereo vision; Mobile robots; Service robots; Human–robot interaction.

1. Introduction

There is a considerable potential for using robots in domestic applications such as personal robots,[16,36] robotic pets, (21) tour guides robots,[10,45] and robotic assistants for disabled or elderly people,[30,40] among others. However, in order to make robots present in everyday life, it is necessary to achieve a natural and intuitive human–robot interaction (HRI),[7,18,43] *i.e.*, both robots and humans must be able to communicate with little prior knowledge about each other, and the communication should be natural. Classical communication devices such as mouses, keyboards, or touch screens, have been commonly used for HRI tasks. Nevertheless, it seems appropriate to complement these interaction devices with communication mechanisms more similar to those employed by humans, like speech, hand gestures, facial expressions, postural gestures, *etc*. A prerequisite for achieving the desired level of communication to provid robots with the abilities of people detection and tracking.

*Corresponding author. E-mail: rmsalinas@uco.es

The people detection and tracking topics have attracted the interest of researches in many areas. The problem has been tackled by the use of several type of sensors such as laser,[42] vision,[23,29,41,44,46] or a combination of both.[4,20] A great part of the effort has been focused on monocular vision and laser sensors. However, stereo vision is nowadays a technology very interesting for these purposes due to the advantages that it brings. First, all the methods designed for tracking in monocular images can be applied, but with much richer per-pixel information (color or luminance plus depth). Although the range information provided by stereo vision is less accurate than that provided by laser sensors, it can be effectively employed to calculate the three-dimensional position and body configuration of a user at a lower cost. Stereo vision can be employed for keeping track of people in cluttered environments reducing the confusions with background elements. Second, disparity information is relatively invariable to illumination changes. Therefore, systems employing stereo vision are expected to be more robust in real scenarios where sudden illumination changes often occur.

Several approaches to the people detection and tracking problems have been proposed using stereo vision. Among the first works in that area, we find the one of Cipolla and Yamamoto.[12] They present a stereo-based method for tracking objects using the *visualized locus* method. In ref. [15], Eklundh *et al.* study the uses of active vision concluding that the integration of multiple cues is necessary in order to accomplish real-life tracking tasks. In ref. [14], Darrel *et al.* present an interactive display system capable of detecting and tracking several people. People detection is based on integrating information provided by the following three modules: a skin detector, a face detector, and the disparity map provided by a stereo camera. Perseus[33] constitutes a remarkable example of using stereo vision for interacting with human users. Perseus is able to detect and track a single user and is able to determine where does he/she points to. The Perseus architecture is also employed by Franklin *et al.*[19] to develop a robotic waiter controlled by simple hand gestures. Haritaoglu *et al.* present in ref. [25] a complete system that integrates gray-level and stereo information for tracking people. Among the features that the system exhibit

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

is its remarkable ability to locate different human parts: head, hands, feet, *etc*.

The common aspect of these works is that tracking is performed in the camera image. However, when using stereo vision sensors, a ground-based representation of stereo data (named *plan-view map*) brings several advantages. First, an orthogonal projection allows to reduce the great amount of information provided by stereo systems but preserving meaningful spatial information that is useful for tracking purposes. Second, people do not tend to be overlapped in the floor plane as much as they are in the original camera images. Therefore, the tracking process is more reliable in the projection and it avoided the need of using a segmentation method that determines to which person belong each pixel of the camera image. Third, plan-view maps allow to easily integrate information provided by several stereo sensors into a single representation model.

The work of Harville *et al.*[26] constitutes an excellent example of using plan-view maps for people detection and tracking in video surveillance tasks. In first step, a background model of the environment is created and employed to detect entering objects. Then, they combine the information provided by occupancy and height maps in order to detect and track people using adaptive templates in combination with Kalman filters. While the cells of an occupancy map register the amount of three-dimensional points projected in them, cells of height maps register the maximum height of the points projected. Later, in ref. [26], they show the ability of height maps to learn human body poses. Zhao *et al.* present in ref. [48] a system for tracking multiple people in plan-view maps using multiple stereo cameras. They formulate the multi-object tracking task as a mixture model estimation problem solved by the Expectation Maximisation method. Hayashi *et al.*[27] present a people detection and tracking system especially designed for video surveillance. The environment is modeled using an occupancy map and people detection is performed in it based on a simple heuristic: a person is a peak in the map whose height is in a normal range. Later, in ref. [28], the same authors propose the use of a variable sized window to represent people according to their distance to the camera. Thus, when people are far from the camera, the window is enlarged to account for stereo errors. In ref. [11], the authors propose the use of stereo plan-view maps with sound plan-view maps. Sound and stereo information are combined to detect the speaker in a conference room. An important point worth noticing of all these works is that color information is not employed and that cameras are placed in over-head slanting positions.

The works previously indicated have shown the benefits of plan-view maps for people detection and tracking using static cameras. Nevertheless, the use of plan-view maps has not been appropriately explored in the field of autonomous robots. When using static cameras, background models are employed to distinguish background objects from foreground. Then, people are easily detected as foreground objects. However, tracking using a mobile platform makes very difficult to maintain a background model of an unknown environment because of the robot movement. Then, more sophisticated mechanism for detecting people are required. As per our knowledge, only Beymer and Knolige[5] have employed plan-view maps for people tracking in the robotics field. They employed a stereo camera placed at the height of people's knees. People are detected and tracked by their legs using an occupancy map. In their work, the authors do not consider any other source of information than the occupancy map created from the people legs. Their approach is very limited in the sources of information employed. Thus, the proposed system is likely to confuse the person being tracked with surrounding people or even obstacles. Unfortunately, the authors do not perform a quantitative evaluation of the method in reproducible conditions.

1.1. Proposed contribution

This work presents a complete multi-agent architecture for people detection and tracking intended for mobile devices that is based on using plan-view maps. The main novelty of the work is the detection and tracking techniques employed. Detection is carried out using a probabilistic combination of stereo information and a face detector. We employ a moving camera placed at aproximately 1 m height so that people's faces are visible. Then, we employ two different plan-view maps: *occupancy maps* allowing the detection of human-shaped objects and *height maps* allowing to discard non human objects by examining their height. Both pieces of information are combined in a *likelihood map* indicating regions with high likelihood of being occupied by people. Finally, face detectors are employed in order to verify if the regions of high likelihood in the map correspond to people. The combination of these three pieces of information is a novel approach that allows our system to reliably detect people without requiring a background model of the environment.

An important aspect to take into account when designing human–robot interfaces is the fact that the user being tracked might establish close interactions with other people. In that situation, tracking based exclusively on position information is not appropriate since the person being tracked might change his trajectory (to avoid collisions) or even stop walking to start a chat. Then, the prediction about his trajectory becomes unreliable and the person being tracked might be confused with another. Color is a powerful cue that can be employed to distinguish the person being tracked from others. However, it has not been employed by other authors that have employed plan-view maps. Our tracking approach combines depth and color information in order to avoid confusing the identity of the people being tracked with others in his surroundings. The tracking method proposed in this work uses a Kalman filter to keep track of the person direction and an adaptive color model of his clothes. Our approach fuses these pieces of information (position and color) dynamically in the following manner: when the uncertainty of person location is high and when color information becomes more important in deciding where the person is located. Nevertheless, when the person is properly located, both position and color are employed for tracking.

The proposed visual approach is integrated into a multi-agent system for the task of people detection and tracking. We have designed a system divided in three levels based on

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

the concept of building complex behaviors (named skills) based on the combination of simpler behaviors.[3,8] The skills designed combine visual and range information in order to enable mobile robots to (i) detect an interested user who desires to interact with the robot; (ii) keep track of the user while they move in the environment without confusing them with other people; and (iii) follow the user along the environment avoiding obstacles in the way. The system proposed has been designed using a multi-agent philosophy[13,32,34,38,39] in order to allow further expansions.

The remainder of the paper is structured as follows. Section 2 gives a description of the hardware employed and an overview of the system. Sections 3, 4 and 5 explain the middle and higher levels of the system respectively. Section 6 shows the results of the proposed system in several real-life experiments. Finally, Section 7 draws some conclusions.



Fig. 1. Robots for which the system is operative.

2. Multi-agent System Overview

2.1. Hardware description

The system presented has been developed to run both on the Nomad 200 and Peoplebot robots. Both of them are equipped with a ring of 16 sonar sensors that are employed to measure the distance of the surrounding obstacles. Although both robots have on-board computers, we have opted to run the system on a laptop computer in order to have standard computational power for both platforms. The laptop employed is a Pentium IV running at 3.2 GHz that communicates with the computer of the robot via Ethernet using TCP/IP. The computer of the robot is only used to run a daemon that informs about the data from the sensors and manages the movement commands to the motors. All the agents of our system run on the laptop computer under Linux OS.

In order to perform the visual acquisitions, a pan-tilt unit (PTU) (model *PTU-D46-17* of *Direct Perceptions*) and a *Bumblebee* stereo camera (of *Point Grey Research*) have been used. The PTU is able to move 139° in the horizontal axis (both to the left and right side), 47° down-side, and 31° up-side. This device allows the movement of the camera independently from the movement of the robot. The stereo camera connects to the firewire port of the laptop computer and is able to send images of resolution 320 × 240 at 15 fps. The camera does not perform the stereo processing itself, it is done in the laptop computer via software. Figure 1 shows the appearance of the robots for which the system is currently operative.

2.2. Architecture overview

The multi-agent system designed is shown in Fig. 2 where the rounded rectangles represent agents. An *agent* can be seen as an independent software process aimed to get or keep a goal implicit in its own design, and that has the capability to communicate with other agents. Unlike classical three-levels deliberative-reactive architectures,[2,22] our approach is based on a functional design to solve the proposed problem and is conceived to be expandable with further levels if required. Following, a brief description of each level is provided.

The *Hardware Managers* level is comprised of agents that act as wrappers for the particular hardware of the system in order to abstract its particularities and to enable an easy portability. The *RobotPlatform* agent wraps the set of minimum services that the robots employed provide. They are (i) setting the translational and rotational speed of the robot, (ii) getting the current position of the robot using an odometric system, and (iii) returning the values of the sonar sensors. Similarly, the *Ptu* and *StereoCamera* agents are abstractions of the PTU and stereo camera respectively.

The *Behavior* level contains a set of agents that implement simple behaviors. A behavior can be seen as the capability of performing an action based on the current perceptions.[3,8] Depending on the type of behavior, the action can be a physical action like moving the PTU or an internal action or decision. The level named *Skills* is comprised of a set of agents that develop complex behaviors based on the concurrent or sequential execution of the simple behaviors in the middle level. Finally, the agent called *Facilitator* (shown in the right side of the architecture) is a *yellow pages* agent. Agents that desire to interact in our system must register themselves prior to offer their services.

3. Behaviors

The *Behaviors* level is comprised of a set of agents that implement behaviors which in the upper level are combined together to create more complex ones (named *skills*). This level is composed by a total of the following five behaviors (see Fig. 2):

(a) *Inspect Around*: This behavior moves the PTU slowly around an indicated direction. It is employed when the system is actively looking for something in the surroundings. In our case, it will be used to look for a person to interact with.
(b) *Fixate Point*: This is a general-purpose behavior that leads the PTU toward a three-dimensional point in the space so that this point becomes the center of the following images captured by the camera. The behavior is not directly called to perform the action but it responds to events.

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

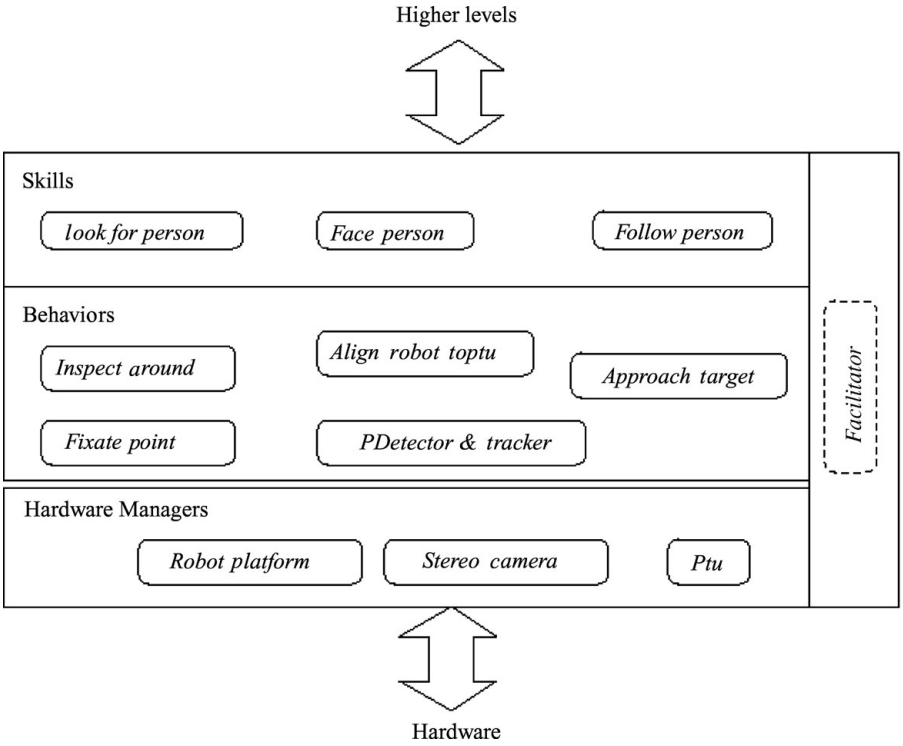


Fig. 2. Multi-agent system proposed.

(c) *Align Robot To Ptu*: This behavior aligns the robot's heading direction with the PTU's pointing direction. The goal is to direct the robot toward the direction that the visual system is looking at. The behavior is implemented as a rule-based fuzzy system.
(d) *PDetector & Tracker*: This is the behavior in charge of detecting and tracking users using the stereo information provided by the camera. A detailed explanation of the agent and the approach employed for people detection and tracking is given in Section 4.
(e) *Approach Target*: This behavior moves the robot from its current location to an indicated one (relative to its current position) avoiding the obstacles in the path. The *virtual forces fields* (vff)[6] method is employed to navigate safely. The vff technique models the desired position to go as an attractive force while the objects around the robot acts as repulsive forces. In our case, the repulsive forces are calculated employing the sonar readings and the attractive force is represented by the person position.

4. People Detection and Tracking

The abilities of detecting and tracking people are fundamental in robotic systems that desire to achieve a natural HRI. The agent *PDetector & Tracker* is in charge of implementing these abilities. They are achieved by combining stereo vision and color using a plan-view map representation. The steps involved in the people detection and tracking tasks are outlined in Fig. 3. First, a stereo image is obtained from the *StereoCamera* agent. Second, a plan-view map representation of the stereo information is created. Third, the map is analyzed to detect possible objects that, according to their height and dimensions, might belong to human beings. They are named *person candidates*. These three steps are common for the people detection and tracking tasks. Then, people detection is performed by looking if any of the person candidates detected shows a face in the camera image. On the other hand, tracking is treated as an assignment problem, *i.e.*, deciding which one of the person candidates detected is the person being tracked. Following, these steps are explained in detail.

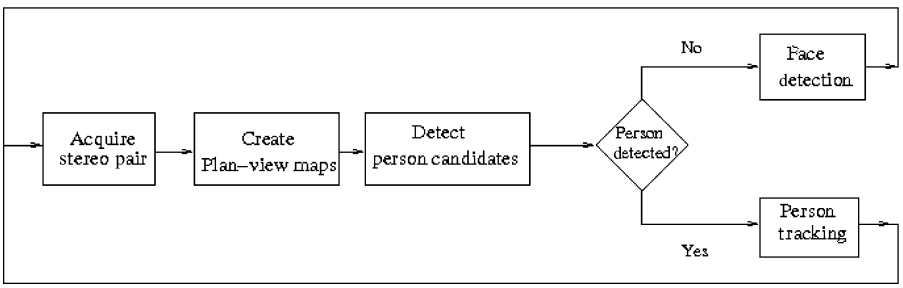


Fig. 3. People detection and tracking process.

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

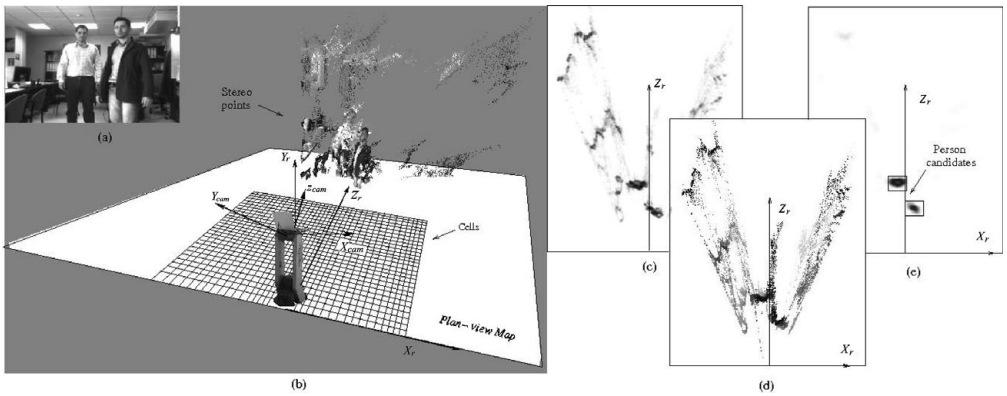


Fig. 4. (a) Image of the right camera captured with the stereo system. (b) Three-dimensional reconstruction of the scene showing the reference systems employed. (c) Occupancy map. (d) Height map. (e) Likelihood map.

4.1. Plan-view map creation

StereoCamera agent provides the last image captured and its corresponding disparity map to the agent that request it. Using standard stereo equations,[9] the three-dimensional positions of the matched points (pixels with disparity) can be obtained. Let us denote the three-dimensional coordinates of the points matched by the stereo vision systems as $P_{cam} = \{p^i_{cam} = (X^i_{cam}, Y^i_{cam}, Z^i_{cam}) \mid i = 1, \ldots, M\}$. The three-dimensional positions p^i_{cam} are referred to the stereo camera reference system. However, it is preferable for our purposes to translate the position of the points matched to a *"robot"* reference system. It is placed at the center of the robot, at ground level, and in the robot's heading direction Z_r. Knowing the height of the camera in relation to the floor plane, we dynamically calculate the linear transformation matrix that translates the points p^i_{cam} into $p^i_r = (X^i_r, Y^i_r, Z^i_r)$.

Figure 4(a) shows an example of a scene captured with our stereo camera (the image corresponds to the right camera). Figure 4(b) shows the three-dimensional reconstruction of the scene captured using the points detected by the stereo camera. The *"robot"* and camera reference systems have been superimposed in Fig. 4(b).

As it can be noticed, the number of matched points is very high. In order to perform a reduction of the amount of information obtained, preserving structural information, stereo data is orthogonally projected in plan-view maps. A plan-view map divides a region of the environment into a set of cells of fixed size δ (see Fig. 4(b)). The cell (x^i, y^i) in which a three-dimensional point p^i_r is projected can be calculated as indicated in Eq. 1. The selection of δ must be made taking into account that while a high value helps to decrease the computational effort and the memory used, a low value not only increases the precision but also the computational requirements. In this work, we have opted for setting $\delta = 3$ cm which is an adequate trade-off between both requirements according to our experimentation,

$$x^i = \left(X^i_r / \delta\right) ; \; y^i = \left(Z^i_r / \delta\right). \tag{1}$$

The set of points that are projected on each cell is defined as

$$P_{(x,y)} = \left\{ i \mid x^i = x \wedge y^i = y \wedge Y^i_r \in [h_{min}, h_{max}] \right\},$$

where $[h_{min}, h_{max}]$ is a height range that has two purposes. First, the upper limit h_{max} avoids using points from the ceiling or from objects hanging from it (*i.e.,* lamps). Second, the lower limit h_{min} excludes from the process low points that are not relevant (floor points) and thus helps to reduce the computing time. The height range $[h_{min}, h_{max}]$ should be such that, at least, an important part of the body of the person to detect should fit in it (including his/her head). The rest of the points whose projection is outside the limits of the plan-view map are not considered.

Two different plan-view maps are employed in this work. They are named occupancy map $\mathcal{O}$ and height map $\mathcal{H}$. The occupancy map registers in each cell $\mathcal{O}_{(x,y)}$ the amount of points that are projected in it. It is calculated as

$$\mathcal{O}_{(x,y)} = \sum_{j \in P_{(x,y)}} \frac{\left(Z^j_{cam}\right)^2}{f^2}. \tag{2}$$

The idea is that each detected point increments the cell in which it is projected by a value proportional to the surface that it occupies in the real scene.[26] Points closer to the camera correspond to small surfaces and vice versa. If the same increment is employed for every cell, the same object would have a lower sum of the areas the farther it is located from the camera. This scale in the increment value will compensate the difference in size of the objects observed according to their distance to the camera.

Cells of the height map, $\mathcal{H}_{(x,y)}$, register the height of the highest point projected in them. The height map is calculated as

$$\mathcal{H}_{(x,y)} = \begin{cases} max\left(Y^j_w \mid j \in P_{(x,y)}\right) & \text{if } P_{(x,y)} \neq \emptyset \\ h_{min} & \text{if } P_{(x,y)} = \emptyset. \end{cases} \tag{3}$$

Figures 4(c) and (d) show the occupancy and height maps of the scene in Fig. 4(a). In Fig. 4(c) the darker a pixel is the higher is its occupancy. In Fig. 4(d) the darker a pixel is the higher is its height.

4.2. Detection of person candidates

A person candidate is considered to be a scene element that according to its shape might belong to a human being. Detection of person candidates is done by creating a

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

likelihood map $\mathcal{L}$, combining the occupancy and height maps. Each cell $\mathcal{L}_{(x,y)}$ indicates the likelihood of the corresponding environment region to be occupied by a person. The idea underling the creation of the likelihood map is that, when using plan-view maps, people project in an area proportional to their real dimensions. Usually, a person placed at the plan-view map position (x, y) with his/her arms partially extended fits in a rectangular region $\mathcal{R}_{(x,y)}$ whose size, ζ_R, varies from 0.4 m to 0.6 m. For that purpose, we define a pair of measures ($O_{\mathcal{R}(x,y)}$ and $H_{\mathcal{R}(x,y)}$) that are used for the calculation of the likelihood map. The first one, $O_{\mathcal{R}(x,y)}$, provides information about the number of points occupied in the region $\mathcal{R}_{(x,y)}$. It is calculated as

$$O_{\mathcal{R}(x,y)} = \sum_{i \in \mathcal{R}(x,y)} \mathcal{O}_{(x^i,y^i)}. \quad (4)$$

In Eq. 4, $\mathcal{R}_{(x,y)} = \{i \mid max(|x^i - x|, |y^i - y|) < \frac{\zeta_M}{2}\}$ represents the set of cells from a squared region of size ζ_M centred at (x, y). The parameter ζ_M, represents the dimension of a person's projection, ζ_R, expressed in cells (in our case $\zeta_R = 0.6$ m).

The second measure, $H_{\mathcal{R}(x,y)}$, represents the maximum height of the points projected in $\mathcal{R}_{(x,y)}$ and is calculated as

$$H_{\mathcal{R}(x,y)} = max(\mathcal{H}_{(x^i,y^i)}) \forall i \in \mathcal{R}(x, y). \quad (5)$$

When a person is located at the map position (x, y), the occupancy of the region, $O_{\mathcal{R}(x,y)}$, can be expected to have values in a certain range. Similarly, the maximum height of that region, $H_{\mathcal{R}(x,y)}$, can be expected to be normally distributed along the mean people height. Under these considerations, we define the likelihood of a cell to be the center of a person using the following Gaussian distribution:

$$\mathcal{L}_{(x,y)} = \frac{exp\left(-\left(\frac{(O_{\mathcal{R}(x,y)}-\mu_o)^2}{2\sigma_o^2} + \frac{(H_{\mathcal{R}(x,y)}-\mu_h)^2}{2\sigma_h^2}\right)\right)}{2\pi\sigma_o\sigma_h}. \quad (6)$$

In Eq. 6, the parameters μ_o and σ_o, represent the expected mean and standard deviation of $O_{\mathcal{R}(x,y)}$ when a person is in the region $\mathcal{R}(x, y)$. Similarly, the parameters μ_h and σ_h, represent the expected mean and standard deviation for the height of a standing person. The values μ_h and σ_h are selected in this work to detect and track adult people. We found that $\mu_h = 1.5$ m and $\sigma_h = 0.5$ m provide good results. However, the values for the parameters μ_o and σ_o depend on the stereo sensor resolution and must be appropriately selected so that partial occlusion of the targets can be handled. In our case, using a resolution of 320×240 pixels, we found experimentally that the values $\mu_o = 3500$ and $\sigma_o = 2000$ provide the best results. Equation 6 is applied to every cell of the plan-view maps in order to create the likelihood map $\mathcal{L}$. Figure 4(e) shows the likelihood map created from the occupancy and height maps in Figs. 4(c) and (d).

Person candidates are determined as peaks in $\mathcal{L}$ employing a two-steps iterative process. First, the cell $\mathcal{L}_{(x,y)}$ with maximum likelihood is selected as a new person candidate. This cell is assumed to be the candidate's location. Second, all the cells of the likelihood map in the region $\mathcal{R}_{(x,y)}$ (size occupied by a person) are set to zero in order to avoid including the person again as a candidate in the next iteration. The two steps are repeated until the cell with maximum likelihood is below a threshold θ_{min}. Figure 4(e) shows the two person candidates detected in the likelihood map.

4.3. Person detection

Person candidates detected in the previous phase correspond to objects whose shape and height are similar to humans. However, they can be caused by structural elements of the environment, *e.g.*, a coat in a hanger. To avoid the confusion, a face detector is applied on the camera image for people detection. Face detection is a process that can be time consuming if applied on the entire image. Thus, it is only applied on regions of the camera image where the head of candidates are expected to be (*i.e.*, the top part of the candidate). If a face is detected, the system considers the candidate as a person. This reduction of the search region where to apply the face detector which brings two main advantages. First, it reduces the computational time as smaller regions are analyzed. Second, it reduces the number of false positives as stated in ref. [35] at the cost of increasing the false negatives. The face detector employed is based on the approach proposed by Viola and Jones[47] which was later improved by Lienhart.[37] We have employed the detector implemented in the OpenCV library[31] that is trained to detect frontal human faces and works on gray-level images. We must indicate that face detection is a step required only in the person detection phase. Once the person is detected, it can be tracked without the necessity of detecting his face. The requirement of detecting the user's face not only helps to reduce the possibility of false positives but also helps to detect some degree of interest of the user in the interaction.

When all the person candidates have been examined, the behavior emits an event indicating the positions of the people detected. Then, it is the responsibility of the agent that activated this to decide what to do with that information.

4.4. Person tracking

When one or more people are detected in the scene, this agent can be commanded to keep a visual track of one of them. As previously indicated, relying exclusively on position information for tracking is not appropriate when the user is in the presence of other people. The interacting user might establish close interactions with other people in the surroundings, thus causing confusions to the tracking system. In this work, both position and color information are combined to enhance tracking. When the person is first detected, a color model of his clothes is created. Let us denote the color model of the person being tracked by $\hat{p}$.

4.4.1. Color modeling. A color model $\hat{q}$ is a histogram created with the color of the points that project in the person candidate region $\mathcal{R}_{(x,y)}$. The HSV space[17] has been selected to represent color information in this work because its robustness to illumination changes. The histogram $\hat{q}$ is comprised by $n_h n_s$ bins for the hue and saturation. However, as chromatic information is not reliable when the value or saturation components are too small, pixels on these situations are not used to describe the chromaticity. Because

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

these "color-free" pixels might have important information, the histogram is also populated with n_v bins to capture its luminance information. The resulting histogram is composed by $m = n_h n_s + n_v$ bins.

Let $\{x_j^*\}_{j=1,\ldots,n}$ be the pixels employed to create the color model. We define a function $b : \Re^2 \to \{1, \ldots, m\}$ which associates to the pixel x_j^* the index $b(x_j^*)$ of the histogram bin corresponding to the color of that pixel. The color density distribution for each bin $\hat{q}(u)$ of x^* is calculated as

$$\hat{q}(u) = K \sum_{j=1}^{n} \kappa\big[b(x_j^*) - u\big]. \tag{7}$$

The function κ represents the Kronecker delta function and K is a normalization constant calculated by imposing the condition

$$\sum_{u=1}^{m} \hat{q}(u) = 1.$$

4.4.2. Kalman filtering. A Kalman filter is employed in this work to track the person position. The state transition model in our system is given by the equation

$$\mathbf{x}_t = \mathbf{F}\mathbf{x}_{t-1} + N(0, \mathbf{Q}_t),$$

where

$$\mathbf{F} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

represents the transition matrix. The state vector,

$$\mathbf{x}_t = (x_t, y_t, \dot{x}_t, \dot{y}_t)^T$$

contains the person's location and velocity in the plan-view map, and the process noise is given by

$$\mathbf{Q}_t = I_{4x4} q^2,$$

where I is the $4x4$ identity matrix and q^2 has been experimentally estimated as 0.01.

At time t, an observation (or measurement) $\mathbf{z}_t$ of the true state $\mathbf{x}_t$ is made according to

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + N(0, \mathbf{R}_t).$$

The observation vector,

$$\mathbf{z}_t = (x_{obs}, y_{obs})^T$$

represents the observed position of the person being tracked in the plan-view map. The observation is provided by the data association step explained below. The observation model in this work is

$$\mathbf{H} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

The measurement noise,

$$\mathbf{R}_t = I_{2x2} r^2$$

represents the error in the estimation given by the stereo information. In our system, it has been experimentally determined that $r^2 = 0.01$ provides good results.

Finally, let us also denote by $(\hat{x}_t, \hat{y}_t)$ the position predicted by the filter for time t and by $(p_{x,t}^2, p_{y,t}^2)$ the uncertainty associated to that prediction (given by the predicted estimate covariance matrix). These variables are employed in the data association step to determine the most likely person location in the next time step as explained below. For more information about the Kalman filter, the interested reader is referred to ref. [24].

4.4.3. Data association step. In the tracking phase, whenever a new stereo image is obtained from the *StereoCamera* agent, the plan-view maps are created and person candidates are located as explained in the previous sections. The data association step determines which of the person candidates detected correspond to the person being tracked. For that purpose, let us denote by

$$O = \{o^i = (x^i, y^i, \hat{q}^i) \mid i = 1, \ldots, N\}$$

the plan-view map locations (x^i, y^i) of the N person candidates detected and their corresponding color models $\hat{q}^i$.

The data association step is solved using a maximum likelihood estimate (MLE) approach

$$o^b = \arg\max_{o^i} S(o^i),$$

i.e., the person candidate o^b that obtains the highest value for the likelihood function $S(o^i)$ is considered to be the valid track. However, in order to avoid confusing the person being tracked with other people in case of occlusion, the association is considered valid only if the highest likelihood exceeds a certain threshold θ_v (0.5 in our experiments).

The likelihood function is defined as the following Gaussian distribution:

$$S(o^i) = \frac{e^{-\frac{1}{2}\left(\frac{(x_o^i - \hat{x}_t)^2}{p_{x,t}^2} + \frac{(y_o^i - \hat{y}_t)^2}{p_{y,t}^2} + \frac{B(\hat{q}^i, \hat{p})^2}{\sigma_c^2}\right)}}{(2\pi)^{\frac{3}{2}} p_{x,t}\, p_{y,t}\, \sigma_c} \tag{8}$$

that combines position and color information. On one hand, the distance between the person candidate location (x^i, y^i) and the predicted position of the person $(\hat{x}_t, \hat{y}_t)$ are calculated using as deviation the uncertainty associated to the prediction $(p_{x,t}^2, p_{y,t}^2)$. Thus, the more uncertainty is associated to the person position, the more relevant color information becomes. On the other hand, the distance of the color models are estimated using the Bhattacharyya distance $B(\hat{a}, \hat{b})$,[1] that is calculated as

$$B(\hat{a}, \hat{b}) = \sqrt{1 - \sum_{u=1}^{m} \sqrt{a(u)b(u)}}. \tag{9}$$

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

It is a normalized distance so that $B(\hat{a}, \hat{b}) \in [0, 1]$. The parameter σ_c has been experimentally determined as 0.3.

As can be noticed, Eq. 8 allows to dynamically weigh the relative importance of color and position information in the data association step. When the uncertainty about position grows, color becomes more relevant in Eq. 8 and vice versa.

4.4.4. Model update. When the most likely person candidate has been determined, both the Kalman filter and the color model are updated. The color model of the person being tracked $\hat{p}$ is updated using the color histogram of the best candidate $\hat{q}^b$ as

$$\hat{p}_t(u) = (1 - \lambda_c)\hat{p}_{t-1}(u) + \lambda_c \hat{q}^b(u), \ \forall u = 1, \ldots, m. \quad (10)$$

The parameter $\lambda_c \in [0, 1]$ weighs the contribution of the observed color model to the updated one. In this work we have set this parameter to 0.1. The resulting color histogram $\hat{p}_t(u)$ is then normalized dividing each cell by the sum of all of them.

Finally, this agent emits a *PersonTracked* event to inform about the results of the tracking. If the person has been successfully detected, the message contains the position of the person and a prediction of its future position. However, if the person is not tracked, the *PersonTracked* event is also emitted but indicating the predicted position for the person and the uncertainty associated with the prediction. Therefore, even if the person is lost momentarily, the system can continue working with the predictions until the person is relocated. Nevertheless, if the person remains unseen for too long and the uncertainty about his/her position becomes very high, the agent considers that the person is lost and emits an event indicating it. Afterward, the agent stops until new orders are received.

5. Skills

The level named *Skills* is composed by a set of agents that develop particular *skills* based on the concurrent or sequential execution of the previously explained behaviors. The three skills implemented are shown in Fig. 2.

(a) *Look For Person*: The goal of this skill is to look for people in the surroundings of the robot. The skill is achieved through the concurrent execution of the behaviors *Inspect Around* and *PDetector & Tracker*. The former moves the PTU to scan the environment. The later examines the stereo images in order to detect people entering in the scene. The skill can be configured so that only people nearer a certain distance d_l is considered. Thus, only people approaching the robot is considered and other people passing by might be ignored.

(b) *Face Person*: This skill is employed once a person has been detected. Its aim is to keep track of a desired person while he/she walks around the robot at near distances. The skill is accomplished by the concurrent activation of the behaviors *PDetector & Tracker*, *Fixate Point,* and *Align Robot To Ptu*. The first one is employed to keep visual track of a desired person. The second one is employed to continuously point toward the direction of the desired person, avoiding him/her to go away of the field of vision. The third behavior makes the robot to spin around itself in order to align the robot and PTU axis. The resulting skill turns the robot in the same direction of the person being tracked.

(c) *Follow Person*: This skill allows the robot to follow the person being tracked up to a certain distance d_{fp} avoiding the obstacles in the path. The skill is achieved by the concurrent execution of the following behaviors: *PDetector & Tracker*, *Fixate Point,* and *Approach Target*. The first behavior is in charge of keeping track of the desired person. The second behavior is configured so that after each tracking step, the PTU is directed toward the person position. Finally, the third behavior is employed to allow the robot to follow the person. After each tracking step, *Approach Target* is commanded to move toward a position d_{fp} meters from the person. As this process is continuously repeated, the target position for the *Approach Target* agent is continuously updated.

6. Experimental Results

A total of five sets of experiments have been performed in order to test the proposed system. The first three sets test the ability of the behavior *PDetector & Tracker*. The first one tests its ability to detect person candidates at different distances. The second one evaluates the performance of the behavior *PDetector & Tracker* to detect people at different distances, *i.e.*, detecting faces associated to person candidates. The third set of experiments has been performed in order to evaluate the process of detecting a person using the *Look For Person* skill, *i.e.*, evaluate whether the integration of perception and motion is appropriate in order to achieve the goal established for the *Look For Person* skill. Finally, the fourth and fifth set of experiments evaluate the *Face Person* and *Follow Person* skills respectively.

6.1. Detection of person candidates

The aim of the experiments is to evaluate the performance of the *PDetector & Tracker* behavior in detecting person candidates. As detecting person candidates constitutes the first step for both people detection and tracking, the success of this behavior sets the maximum performance expected for the others.

To perform the experiment, a person was instructed to walk naturally for approximately 30 s in the field of view of the camera keeping a fixed distance while a color-with-depth video sequence was being recorded. The video sequence was recorded at 15 fps and during which both the camera and the robot were still. In addition, the person was instructed not only to avoid looking at the camera all the time but also to appear in lateral and backward positions. The experiment was repeated for the same person at different distances and for a total of four different people. The total recording time was 8 min summing 7365 frames.

The videos were processed off-line by the *PDetector & Tracker* behavior. Whenever the behavior detected correctly the presence of the person as a person candidate, it was considered a success. Otherwise, it was considered as a failure. The success rate of the behavior in correctly detecting

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

Table I. Success of the *PDetector & Tracker* behavior in detecting person candidates.

d (cm)	Frames analyzed	Success
100	1043	95.3%
150	1121	94.1%
200	1035	94.2%
250	1050	85.3%
300	1037	83.2%
350	1058	80.1%
400	1021	68.4%

a person candidate, in relation to its distance to the camera, is shown in Table I.

As it could be expected, there is a degradation behavior performance as the person is located farther from the camera. However, the detection of a person candidate can be done with an acceptable degree of success up to distances of 350 cm.

6.2. Detection of people

The purpose of the experiments is to evaluate the performance of the *PDetector & Tracker* behavior in detecting faces in person candidates. A total of four people were instructed to walk keeping a fixed distance to the camera, but this time always looking at it. Sequences of 15 s were recorded for the same person at different distances. The 4356 frames acquired were analyzed off-line and the results obtained are shown in Table II.

As it can be noticed there is a sharp cut in the success between the distances 200 cm and 250 cm. This is because the method employed for detecting faces[47] requires the faces to have a minimum size in the image. In our system, the face detector is configured to detect faces that fit in a squared window of at least 15 pixels. Faces smaller than this size are not to be detected. Although this limit can be changed, the smaller it is, the higher computational effort is required to analyze the images. Analyzing the results of the experiment in Table II, it can be noticed that at distances farther than 200 cm, the size of the people's faces in the images begin to be too small to be detected. However, a distance of 200 cm seems to be a reasonable limit for detecting people that desire to interact with the robot.

6.3. Experiments for Look For Person skill

The two previous sets of experiments evaluate the behaviors used by the *Look For Person* skill. However, this third experiment is designed to evaluate the whole skill. It evaluates if the robot is able to successfully detect a person who wants to start an interaction with it and for how long. For that purpose, the robot is commanded to start the execution of the *Look For Person* skill. Then, a person approaches the robot and tries to be detected by it. The time that elapsed between the person approaches the robot nearer than 2 m and he/she is detected as a person, is measured. An excessive amount of time used by the robot to detect a user could make the user feel frustrated thus avoiding future interactions.

Table II. Success of the *PDetector & Tracker* behavior in detecting people's faces.

d (cm)	Frames analyzed	Success
100	838	95.2%
150	842	83.4%
200	843	70.8%
250	944	14.7%
300	889	0.0%

Table III. Times employed by the *Look For Person* skill in detecting potential users.

Person	Avrg. time (ms)	Min. time (ms)	Max. time (ms)
1	515	139	1623
2	677	134	1682
3	429	112	1886
4	512	143	1782

The experiment was repeated 10 times for the same person starting from different positions (always in a cone of 120° around the heading direction of the robot) and for a total of four different people. Table III shows the results of the experiment. Each row shows the results for the ten tries of each person. The first column names the person. The second column indicates the average time that was required by the robot to detect the person in the ten tries (expressed in milliseconds). Finally, the third and fourth column indicate the minimum and maximum times respectively (expressed in milliseconds) employed for the ten tries of each person.

As it can be noticed, the average time necessary to be detected by the proposed skill is around half a second. We consider that this is an appropriate performance for the skill and that potential users should not feel frustrated having to wait for too long to be detected.

6.4. Experiments for Face Person skill

This fourth set of experiments aims to evaluate the performance of the skill *Face Person* to keep track of a person moving in the presence of other people. As explained in Section 5, the behaviors *PDetector & Tracker*, *Fixate Point,* and *Align Robot To Ptu* are run concurrently to develop that skill. This experiment does not only evaluates the performance of each one of the three behaviors but also the coordination between them.

The experiments consist of a person (named $P0$) approaching the robot, being detected using the *Look For Person* skill. Once it has been detected, the skill *Face Person* starts and the person begins to move following the trajectory described in Fig. 5. In the same environment there are two other people ($P1$ and $P2$) that can distract the robot from tracking $P0$. The trajectory is such that $P0$ has to pass in front of $P1$ and behind $P2$, *i.e.*, $P0$ is occluded for a period of time. The length of the path was approximately of 6.5 m.

The experiment was repeated five times for four different people. An experiment was considered successful if the robot was able to keep track of $P0$ while he/she was moving along the specified trajectory without confusing him/her with $P1$ or $P2$. The experiment was considered to fail if the robot

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

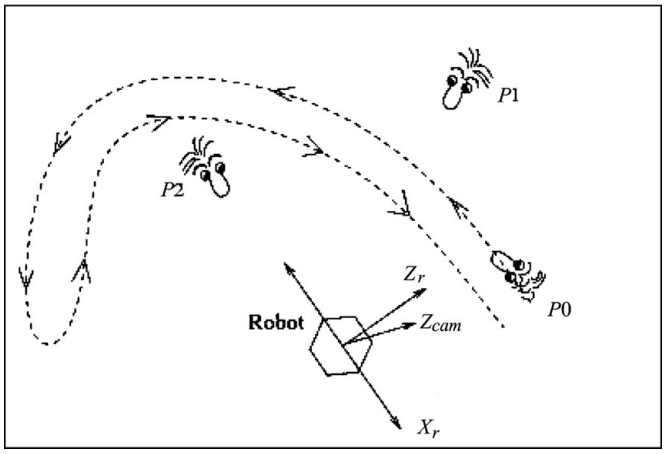


Fig. 5. Person trajectory for testing the *Face Person* skill.

confused the person $P0$ with another or lost track of him/her for more than two consecutive seconds.

Figure 6 shows images taken from the camera of the robot while performing the experiment. Initially, the person $P0$ looks at the camera to be detected by the robot. As $P0$ moves along the specified trajectory, the other people ($P1$ and $P2$) appear in the images. As it can be seen, the system is able to keep track of the person even while moving behind $P2$ (see images 4–5 and 8–9 of Fig. 6).

Table IV summarizes the results of the experiments. Each row indicates the results of the five times the experiment was repeated for each person. The first column indicates the person. The second column indicates the number of times that the experiment was successful (out of the maximum value of five). The third column indicates the average time employed by the person to complete the experiments (only for successful experiments). The fourth column indicates the total average number of frames analyzed in each experiment by the *Track Person* behavior. The last column indicates the percentage of the frame analyzed in which the person was successfully tracked.

Table IV. Success of the *Face Person* skill.

Person	No. of success	Time (s)	No. of frames	Track
1	4	25	131	83.0%
2	4	29	151	80.3%
3	4	26	136	94.8%
4	5	22	113	92.9%

As it can be noticed, the experiment failed one time for all the people except for the last one. The cause of the failure in the first two cases was because $P0$ and $P1$ wore clothes with very similar colors and as they came too near, it became confusing for the robot. The reason of the failure in the third case was because the person being tracked remained for more than two consecutive seconds behind person $P1$ so the robot lost track of him/her.

The results obtained show that the proposed skill is able to coordinate the three basic behaviors appropriately to keep track of the person $P0$ without confusing him/her with the rest of the people for majority of the times that the experiment was repeated. In addition, we observed that the system might confuse the person being tracked with another person with similar colored clothes when both interact at very close distances. This is because the observation model proposed in Eq. 8 is not able to distinguish them in that situation. In future works we will try to find a solution to this problem with the use of additional information to describe each person.

6.5. Experiments for Follow Person skill

This set of experiments are designed to evaluate the success of the *Follow Person* skill in moving the robot while following a person. As in the previous case, the experiments do not only test the operation of the individual behaviors employed but also test the performance of their combination. As explained

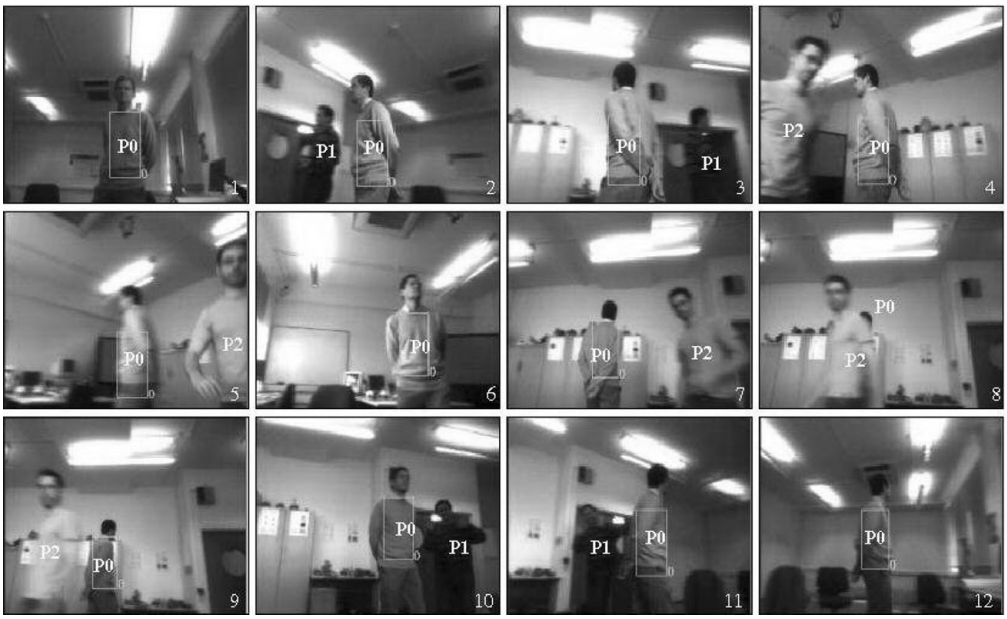


Fig. 6. Images captured from the camera of the robot while testing the *Face Person* skill.

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

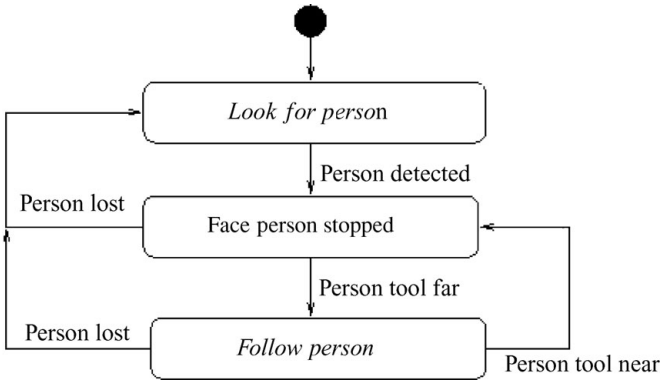


Fig. 7. Application scheme for testing the *Follow Person* skill.

in Section 5, this skill is achieved through the combination of the behaviors *Fixate Point*, *PDetector & Tracker,* and *Approach Target*. The simple application shown in Fig. 7 was build to test this skill. It was designed as an agent that was included in our system.

Initially, the robot starts waiting for a user to interact with him/her using the *Look For Person* skill. Once he/she is detected at distances nearer to 80 cm, the *Look For Person* skill is stopped and the skill *Face Person* is activated. *Face Person* is configured to work up to distances of $d_f = 1$ m, *i.e.*, when the person goes away at further distances, the skill stops. At further distances, the application employs the *Follow Person* skill to approach the person up to a distance of 80 cm. When this is achieved, the application activates the *Face Person* skill again. In case that the person is lost for more than two consecutive seconds by either the *Follow Person* or *Face Person* skills, the *Look For Person* skill is activated again.

The experiments carried out consisted of a person $P0$ approaching the robot to start an interaction. The user should make the robot follow him/her around our laboratory along a path of approximately 8 m. During the experiment, another person $P1$ was in the laboratory moving around $P0$ and even crossing between the robot and the person. This was to test if the robot got confused and followed $P1$ instead of $P0$. Person $P0$ was instructed to regain the attention of the robot and continue the experiment if it lost track of him/her for more than two consecutive seconds.

Table V. Success of the *Follow Person* skill.

Person	No. of success	Time (s)	No. of frames	Track	Restart
1	4	42	252	85.0%	1
2	4	59	360	74.2%	2
3	4	55	310	92.4%	2
4	4	67	387	86.7%	2

Figure 8 shows images taken from an external camera while the experiment was performed. The images show how the person to be followed ($P0$) starts approaching the robot to be detected (images 1–2 of Fig. 8). Then, $P0$ start moving toward the opposite side of the room and the robot follows him/her (images 3–7). As it can be seen, $P1$ moves near $P0$ and even crosses between the robot and $P0$. However, the robot is able to keep track of $P0$. When the robot has reached the end of the room, $P0$ comes back to the initial position while the robot continues following (images 8–12). Again, while $P0$ is being followed, $P1$ is moving as well, and even crossing between $P0$ and the robot.

The experiments were repeated four times for four different people. It was considered a success if $P0$ was able to lead the robot from one side of the lab to the other and then go back without being confused with $P1$. The results of the experiments can be seen in Table V. The first column indicates the person that performed the experiment. The second indicates the number of times that the experiment was successful. The third column indicates the average time employed to complete the experiment in cases that were successful. The fourth column indicates the average number of frames analyzed in each successful experiment and the fifth



Fig. 8. Images captured from an external camera while testing the *Follow Person* skill.

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

column indicates the percentage of these frames in which the person $P0$ was successfully tracked. Finally, the sixth column indicates the average number of times that the skill lost track of $P0$ for more than two consecutive seconds and the application had to restart the *Look For Person* skill. The results of these experiments show that the skill is able to successfully coordinate the three basic behaviors in order to follow human users.

A selection of videos that show some of the tests performed are publicly available at http://decsai.ugr.es/∼salinas/humanrobotvideos/.

7. Conclusions

The paper has presented a multi-agent system that provides a basic set of perceptual-motor skills useful for many mobile robotic applications that require to interact with human users. The skills designed use stereo visual information with ultrasound information to enable mobile robots to (i) detect an interested user who desires to interact with the robot, (ii) keep track of the user while he/she moves in the environment without confusing him/her with other people, and (iii) follow the user along the environment avoiding obstacles in the way. These skills constitute essential components for many mobile robotic applications that require to establish HRI.

User detection and tracking are performed using stereo vision and a plan-view map representation of the data. The plan-view map approach employed is an efficient representation mechanism for people detection and tracking in mobile platforms providing a great separability of the user's in case of partial occlusion. Since relying exclusively on position information for tracking is not appropriate when the target person interacts with others (he/she might be confused due to their proximity or due to their occlusions), our tracking system combines color information of the user's clothes with position estimation given by a Kalman filter.

The system proposed has been tested in numerous real-life experiments demonstrating the ability to detect people with a high success rate and to track them handling partial and total occlusion caused by other people in the surroundings of the robot. Besides, the system achieves real-time performance running in a single laptop computer. For that purpose, the agents of the system work at a configurable operation frequency allowing an efficient sharing of the computing resources.

Future works include the development of techniques for gesture recognition using stereo vision using the information provided by the stereo detector and tracker presented in this work.

Acknowledgments

This work has been partially supported by the Spanish MEC project TIN2007-66367 and Andalusian Regional Government project P06-TIC1670.

References

1. F. Aherne, N. Thacker and P. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetica* **32**, 1–7 (1997).
2. R. C. Arkin, "Path Planning for a Vision-Based Autonomous Robot," **In**: *Proceedings of the SPIE Conference on Mobile Robots*, Cambridge, MA (1986) pp. 240–249.
3. R. C. Arkin, *Behavior-Based Robotics* (MIT Press, Cambridge, MA, 1998).
4. M. Bennewitz, W. Burgard, G. Cielniak and S. Thrun, "Learning motion patterns of people for compliant motion," *Int. J. Robot. Res.* **24**, 31–48 (2005).
5. D. Beymer and K. Knolige, "Tracking people from a mobile platform," *Experimental Robotics VIII* **5** (2003) 234–244.
6. J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Trans. Syst. Man Cybernet*. **19**(5), 1179–1187 (1989).
7. C. Breazeal, "Social interactions in HRI: The robot view," *IEEE Trans. Syst. Man Cybernet., Part C* **34**, 181–186 (2004).
8. R. A. Brooks, "A robust layered control system for a mobile robot," *IEEE J. Robot. Autonom.*, RA-**2**, 14–23 (1986).
9. M. Z. Brown, D. Burschka and G. D. Hager, "Advances in computational stereo," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**, 993–1008 (2003).
10. W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thurn, "Experiences with an interactive museum tour-guide robot," *Artif. Intell.* **144**, 3–55 (1999).
11. N. Checka, K. Wilson, V. Rangarajan and T. Darrell, "A Probabilistic Framework for Multi-modal Multi-person Tracking," **In**: *Conference on Computer Vision and Pattern Recognition Workshop* (2003) pp. 100–107.
12. R. Cipolla and M. Yamamoto, "Stereoscopic tracking of bodies in motion," *Image Vis. Comput.* **8**, 85–90 (1990).
13. C. Colombo, A. Del Bimbo and A. Valli, "Visual capture and understanding of hand pointing actions in a 3-D environment," *IEEE Trans. Syst. Man Cybernet., Part B* **33**, 677–686 (2003).
14. T. Darrell, G. Gordon, M. Harville and J. Woodfill, "Integrated person tracking using stereo, color, and pattern detection," *Int. J. Comput. Vis.* **37**, 175–185 (2000).
15. J.-O. Eklundh, P. Nordlund and T. Uhlin, "Issues in Active Vision: Attention and Cue Integration/selection," **In**: *British Machine Vision Conference* (September 1996) pp. 1–12.
16. E. Falcone, R. Gockley, E. Porter and I. Nourbakhsh, "The personal rover project: The comprehensive design of a domestic personal robot," *Robot. Autonom. Syst.* **42**, 245–258 (2003).
17. J. D. Foley and A. van Dam, *Fundamentals of Interactive Computer Graphics* (Addison Wesley, Boston, MA, 1982).
18. T. Fong, I. Nourbakhsh and K. Dautenhahn, "A survey of socially interactive robots," *Robot. Autonom. Syst.* **42**, 143–166 (2003).
19. D. Franklin, R. E. Kahn, M. J. Swain and R. J. Firby, "Happy Patrons make Better Tippers: Creating a Robot Waiter using Perseus and the Animate Agent Architecture," **In**: *International Conference on Automatic Face and Gesture Recognition* (1996) pp. 14–16.
20. J. Fritsch, M. Kleinehagenbrock, S. Lang, T. Plötz, G. A. Fink and G. Sagerer, "Multi-modal anchoring for human-robot interaction," *Robot. Autonom. Syst.* **43**, 133–147 (2003).
21. M. Fujita and H. Kitano, "Development of an autonomous quadruped robot for robot entertainment," *Autonom. Robot.* **5**, 7–18 (1998).
22. E. Gat, Reliable Goal-Directed Reactive Control of Autonomous Mobile Robots *Ph.D. Thesis* (Virginia Polytechnic Institute, 1991).
23. S. S. Ghidary, Y. Nakata, T. Takamori and M. Hattori, "Human Detection and Localization at Indoor Environment by Home Robot," *IEEE International Conference on System Man, and Cybernetics*, **2** (2000) pp. 1360–1365.
24. M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice* (Prentice Hall, Englewood Cliff, NJ, 1993).
25. I. Haritaoglu, D. Harwood and L. S. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal. Mach. Intell.* **22**, 809–830 (2000).

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press

26. M. Harville, "Stereo person tracking with adaptive plan-view templates of height and occupancy statistics," *Image Vis. Comput.* **2**, 127–142 (2004).
27. K. Hayashi, M. Hashimoto, K. Sumi and K. Sasakawa, "Multiple-person Tracker with a Fixed Slanting Stereo Camera," **In**: *6th IEEE International Conference on Automatic Face and Gesture Recognition* (2004) pp. 681–686.
28. K. Hayashi, T. Hirai, K. Sumi and K. Sasakawa, "Multiple-person tracking using a plan-view map with error estimation," *Computer Vision – ACCV 2006*. Lectures Notes on Computer Sciences, **3851** (2006), pp. 359–368.
29. N. Hirai and H. Mizoguchi, "Visual Tracking of Human Back and Shoulder for Person Following Robot," **In**: *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, **1** (2003) pp. 527–532.
30. P. Hoppenot and E. Colle, "Localization and control of a rehabilitation mobile robot by close human-machine cooperation," *IEEE Trans. Neural Syst. Rehabil. Eng.* **9**, 181–190 (2001).
31. Intel. OpenCV: Open source Computer Vision library. *http://www.intel.com/research/mrl/opencv/*.
32. N. Jojic, B. Brumitt, B. Meyers, S. Harris and T. Huang, "Detection and Estimation of Pointing Gestures in Dense Disparity Maps," **In**: *4th IEEE International Conference Automatic Face and Gesture Recognition* (2000) pp. 468–475.
33. R. E. Kahn, M. J. Swain, P. N. Prokopowicz and R. J. Firby, "Gesture Recognition using the Perseus Architecture," **In**: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96)* (1996) pp. 734–741.
34. R. Kehl and L. Van Gool, "Real-time Pointing Gesture Recognition for an Immersive Environment," **In**: *6th IEEE International Conference on Automatic Face and Gesture Recognition* (2004) pp. 577–582.
35. H. Kruppa, M. Castrillon-Santana and B. Schiele, "Fast and Robust Face Finding via Local Context," *Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance* (2003).
36. S. Lauria, G. Bugmann, T. Kyriacou, J. Bos and A. Klein, "Training personal robots using natural language instruction," *IEEE Intell. Syst. Appl.* **16**, 38–45 (2001).
37. R. Lienhart and J. Maydt, "An Extended Set of Haar-like Features for Rapid Object Detection," *IEEE Conference on Image Processing* (2002) pp. 900–903.
38. P. Maes, T. Darrell, B. Blumberg and A. Pentland, "The alive system: Full-body interaction with autonomous agents," *IEEE Press in Computer Animation* (1995) pp. 11–18.
39. R. Mu noz-Salinas, E. Aguirre, M. García-Silvente and M. Gómez, "A multi-agent system architecture for mobile robot navigation based on fuzzy and visual behaviours," *Robotica* **23**, 689–699 (2005).
40. J. Pineau, M. Montemerlo, M. Pollack, N. Roy and S. Thrun, "Towards robotic assistants in nursing homes: Challenges and results," *Robot. Autonom. Syst.* **42**, 271–281 (2003).
41. H. Saito, K. Ishimura, M. Hattori and T. Takamori, "Multi-modal Human Robot Interaction for Map Generation," **In**: *41st SICE Annual Conference (SICE 2002)*, **5** (2002) pp. 2721–2724.
42. D. Schulz, W. Burgard, D. Fox and A. B. Cremers, "People tracking with mobile robots using sample-based joint probabilistic data association filters," *Int. J. Robot. Res.* **22**(2), 99–116 (2003).
43. K. Severinson-Eklundh, A. Green and H. Hüttenrauch, "Social and collaborative aspects of interaction with a service robot," *Robot. Autonom. Syst.* **42**, 223–234 (2003).
44. H. Sidenbladh, D. Kragic and H. I. Christensen, "A Person Following Behaviour for a Mobile Robot," **In**: *IEEE International Conference on Robotics and Automation*, **1** (1999) pp. 670–675.
45. R. Siegwart, K. O. Arras, S. Bouabdallah, D. Burnier, G. Froidevaux, X. Greppin, B. Jensen, A. Lorotte, L. Mayor and M. Meisser, "Robox at expo.02: A large-scale installation of personal robots," *Robot. Autonom. Syst.* **42**, 203–222 (2003).
46. L. Sigal, S. Sclaroff and V. Athitsos, "Skin color-based video segmentation under time-varying illumination," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 862–877 (2004).
47. P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," **In**: *IEEE Conference on Computer Vision and Pattern Recognition* (2001) pp. 511–518.
48. T. Zhao, M. Aggarwal, R. Kumar and H. Sawhney, "Real-time Wide Area Multi-camera Stereo Tracking," **In**: *Computer Vision and Pattern Recognition (CVPR 2005)* (2005) pp. 976–983.

https://doi.org/10.1017/S0263574708005092 Published online by Cambridge University Press