Proc. 11th Int. Conf. on Intelligent Autonomous Systems,
pp. 89-98, Ottawa, Canada, Aug./Sep. 2010.

Development of a Person Following Robot and Its Experimental Evaluation

Jun MIURA [a,1], Junji SATAKE [a] Masaya CHIBA [a], Yuki ISHIKAWA [a], Kenta KITAJIMA [a], and Hiroaki MASUZAWA [a],

[a] *Department of Information and Computer Sciences, Toyohashi University of Technology*

Abstract. This paper describes a robot system which can follow a specific person while avoiding obstacles and other people walking around. This system consists of various functional modules, which are a stereo-based robust person detection and tracking, a laser range finder-based map generation, and an on-line randomized motion planning, implemented on a self-contained wheeled mobile robot. All these modules are implemented using RT-middleware, which enhances software development, maintenance, and reusability. The implemented system is tested in a reasonably complex environment with several walking people at a time. The experimental results show the feasibility of the system.

Keywords. Person following robot, Person detection and tracking, Path planning, RT-middleware

1. Introduction

Personal service robot is one of the promising areas to which robotics technologies can be applied. As we are facing the "aging society", the need for robots which can help people in their everyday life is increasing. For keeping the *Quality of Life* of elderly people, for example, making them feel easy to going out is one of the important issues. A robot that can move with people like a partner and carrying items will be a useful device for this purpose. We have been developing such a *partner robot*, and one of the important functions of which is following a specific person among obstacles and other people.

The person following task entails several functions: person detection and tracking, obstacle detection and mapping, motion planning, and robot control. Moreover, these functions should be reliable enough and should run within a limited time period with a coordinated way especially when the robot operates in dynamic environments with many people. It is, therefore, still a challenging work to realize an actual mobile robot that can follow a specific person in real environments.

Several person following robots have been developed [15,3,9]. These systems use vision and/or laser range finders to detect persons and to make free space maps. They do not consider occlusions between person. We have developed a stereo-based person following robot which can cope with occlusions among people [14]. This system, however, does not recognize static obstacles nor consider other persons in motion planning.

[1]Corresponding Author: Jun Miura, Toyohashi, Aichi 441-8580, Japan; http://www.aisl.ics.tut.ac.jp/.

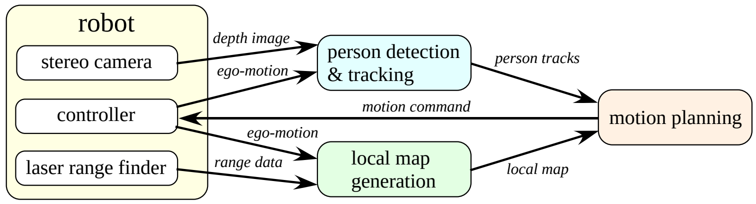


Figure 1. Configuration of the system.

This paper describes a person following mobile robot system and its experimental evaluation. Fig. 1 shows the configuration of the system. We deal with two kinds of objects in the environment: persons detected by stereo vision and static obstacles detected by a laser range finder (LRF). The functions of the three main modules are as follows:

- *person detection and tracking* module detects persons using stereo and tracks them using Kalman filtering to cope with occasional occlusions among people.
- *local map generation* module constructs and maintains an occupancy grid map, centered at the current robot position, using the data from the LRF. It performs a cell-wise Bayesian update of occupancy probabilities [12] assuming that the odometry error can be ignored for a relatively short robot movement.
- *motion planning* module calculates a safe robot motion which follows a specified target person and avoids others, using a randomized kinodynamic motion planner.

To develop and maintain the module-based software system, we use *RT-middleware* [1] [2] environment where each software module is realized as an *RT component*. Multiple RT components can run and communicate with on multiple computers distributed over a local area network.

Performance evaluation in real environments is important in developing dependable systems. We therefore tested the system in a reasonably complex environment with several walking people at a time and analyzed problems to arise.

2. Person detection and tracking

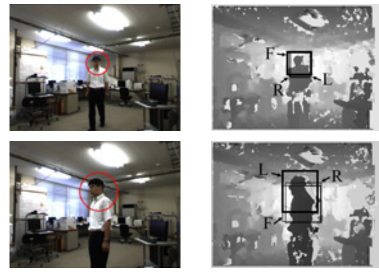
2.1. Related works on visual person detection and tracking

Beymer and Konolige [2] developed a stereo-based person detection based on background static obstacles subtraction. Howard et al. [7] proposed a visual person detection method which first converts a depth map into a polar-perspective map on the ground and then extracts regions with largely-accumulated pixels. Occlusions are not handled there. Ess et al. [5] proposed to integrate various cues such as appearance-based object detection, depth estimation, visual odometry, and ground plane detection using a graphical model for pedestrian detection. Although their method exhibits a good performance for complicated scenes, it is still costly to be used for controlling a real robot. Many local feature-based human detectors (e.g., HOG-based detector [4]) have also been shown

[2] RT-middleware is a specification on a component model and infrastructure services applicable to the domain of robotics software development, authorized by OMG (Object Management Group).

Left Front Right

Figure 2. Depth templates.



(a) Input images (b) Depth images

Figure 3. Person detection results.

(a) positive samples (b) negative samples

Figure 4. Training samples for the SVM-based verifier.

to be useful, but they do not provide reliable range information, which is most useful information in controlling a mobile robot.

2.2. Depth template-based person detection

Our stereo-based person detection and tracking method [14] uses *depth templates*, which are the templates for human upper bodies in depth images (see Fig. 2). We made the templates from the depth images where a person was at about 2 *m* away from the camera. We currently have three templates corresponding to three body directions, and use them simultaneously to find the one with the highest matching value. From the camera position on the robot, we can constrain the relationship between the person position in the image and the size and the depth value of the template. This greatly reduces the search space of the depth templates in person detection. When the robot is continuously tracking a person, even more reduction is possible by position prediction.

Fig. 3 shows examples of person detection using the depth templates. Three rectangles in each depth image are detection results with the three templates and the one with bold lines shows the template with the highest evaluation value. In spite of the change of body direction, a person can be tracked stably.

2.3. SVM-based false rejection

A simple template-based approach is effective in reducing the computational cost but may produce many false detections for objects with similar silhouettes to person. We thus apply an SVM-based classifier using image intensity to candidate regions detected by depth information in order to reject false positives.

Each image region detected by the templates is resized to 20×20 pixels and a set of pixel values is directly used as the input to the SVMs. We use 438 positive and 146 negative images for the SVM learning. Fig. 4 shows examples of positive and negative sample images for training. For about 1500 test images, the learned SVM produced no false positives against negative cases and six percent false negatives against positive cases.

2.4. EKF-based tracking

We adopt the Extended Kalman Filter (EKF) for robust data association and occlusion handling. The state vector includes the position and the velocity in the horizontal axes (X and Y) and the height (Z) of a person. The vector is represented in the robot local coordinates and a coordinate transformation is performed from the previous to the current robot pose every time in the prediction step, using the robot's odometry information. Data association is carried out by the gating method based on the Mahalanobis distance. Color information of the clothing is also used for identifying the target person to follow.

2.5. Performance evaluation of person detection and tracking module

We first evaluated the performance by fixing the robot for 2000 frames. In the frames, 42 persons in total appears and the maximum number of persons in a frame is four. The success tracking rate was 93.8%. We then evaluated using another set of 2000 frames taken during the person following by the robot. In the frames, 26 persons in total appeared and the maximum number of persons was three. The success tracking rate was 96.0%.

3. Motion planning

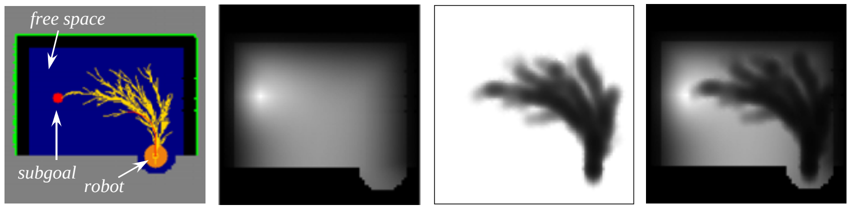
Motion planning is especially important in realizing an actual person following robot in dynamic environments. The robot needs to choose a reasonably efficient safe motion within a limited time period. It is also necessary to consider kinodynamic constraints of the robot [8,10]. In this paper, we propose a variant of randomized kinodynamic motion planner based on a predefined motion set [8], which additionally utilizes a bias for generating goal-directed safe motions.

Inputs to the planner are static obstacles, composed of cells with probabilities higher than a threshold (currently, 0.7) in the local map, and dynamic ones (i.e., persons) with their position and velocity estimates. The current position of the target person is set as the local destination for planning. The planner is invoked every time person and obstacle information is updated.

3.1. Randomized kinodynamic motion planning

The planner is given a predetermined set of possible motions, each of which is specified by a pair of the translational and the rotational velocity and satisfies the kinematic constraints. To consider dynamic constraints, we perform motion planning in a state space describing the robot pose and its derivative. By considering the limited acceleration of the wheels, possible transitions (i.e., motions) from a state are enumerated as follows:

1. Convert the velocity pair of the state into a velocity pair of both wheels.
2. Calculate the ranges of the right and the left wheel velocity considering the acceleration limitation.
3. Choose a set of motions whose wheel velocities are within the above ranges.



(a) map and tree expansion (b) goal-directed bias (c) density bias (d) combined bias

Figure 5. Biases and tree generation.

We perform a weighted randomized planning, by expanding a tree with the current state being the root. The possible set of feasible motions is calculated for each node *on-line* and the weight of each motion is given by the bias at the predicted end position of the motion. We repeat the tree expansion until the time reaches the limit. Once the tree construction ends, we select the most promising path and use only the first motion in the path as the robot command to be executed in the next cycle.

3.2. Collision check by state prediction in time-space

A motion can be a candidate if it does not cause collision with obstacles. Collision with dynamic obstacles (i.e., persons) is checked in time-space [11] on a fixed set of *time slices*. Using the cycle time (currently, 500 ms) and the number of slices (currently, 5), we predict in each time slice both the state of the robot to be achieved by a selected motion and the possible spatial range of each dynamic obstacle in order to see if any collisions occur. At present, a simple constant speed motion model is used for motion prediction of other persons.

3.3. Biases for controlling randomized tree expansion

Randomized motion planning algorithms often give biases to nodes (i.e., reachable states) in the tree being expanded so that the tree grows in a favorable way. A density bias is, for example, often used for expanding the tree towards less densely-sampled regions [8]. It is also important to guide the expansion to globally better directions. Rodriguez et al. [13] proposed to combine a PRM-based global planner with a kinodynamic local planner. We take a similar approach but set a potential field over the entire free space, instead of selecting only one most promising global motion, for guiding local planning.

We need to provide a goal-directed bias in the tree expansion so that the resultant motion is efficient (near to the shortest path) *and* safe (keep some distance from obstacles). We therefore set a potential over the free space on which such a motion will gain a large potential increase. Since this potential calculation should be done on-line, we propose to adopt a level set method-based potential calculation [6]. This first sets a velocity field in which the velocity of a position becomes higher as the distance from obstacles to the position becomes larger, and then propagates a wavefront using the velocity field.

We use above-mentioned two kinds of biases: the goal-directed bias and the density bias. Fig. 5 shows an example of the biases and the tree expansion. For a free space map (blue region in Fig. 5(a)), the goal-directed bias (see Fig. 5(b)) and the density bias (see

Fig. 5(c)) are generated and their multiplication is used as the final bias (see Fig. 5(b)) to guide the tree expansion (yellow lines in Fig. 5(a)).

3.4. Path selection

Each path from the root to a leaf represents a feasible robot path. We would like to select a longer and faster path. So we evaluate each leaf node using its bias value (i.e., the potential value at that node position) and the averaged speed from the root to that node.

Suppose there are K leaf nodes. For each leaf node k, let $\Delta B(k)$ be the increase of the bias value from the root to node k and $V(k)$ is the average velocity of the path to node k, respectively. These two values are nondimensionalized by using the means and the variances of the corresponding values of every leaf node. The best leaf node k^* (and therefore the best path which leads to k^*) is determined by:

$$k^* = \arg\max_k \{w \cdot \Delta B(k) + (1 - w) \cdot V(k)\}, \tag{1}$$

where w $(0 \leq w \leq 1)$ is a weight determined empirically. Once the best path is selected, the very first motion of the path is selected and sent to the robot controller.

3.5. Reuse of the previous path

Every time the environment information (i.e., obstacle map and persons) is updated using the latest sensor data, the whole path planning process, which is composed of the bias calculation, the tree expansion, and the path selection, is performed from scratch. The only exception is the reuse of the selected path in the previous cycle. Since this path is likely to be effective in the current cycle, we examine the path from the root to the leaf and the partial path to the node which is safe (i.e., collision-free) and the farthest from the root is used as a part of newly-expanded tree.

3.6. Best-first planner in simple situations

The kinodynamic motion planner inherently has a slight possibility of producing an inefficient motion due to its randomized nature. In a simple situation, like the one where the robot can see the target person in a wide space, an efficient motion can easily be generated. Since it is difficult to judge if the current situation is simple (for motion planning, of course) only from the generated map and person information, we always try to generate a simple motion using a heuristic motion planner. The planner runs with some time limit and if the planning is not successful, the kinodynamic motion planner is invoked.

The simple planner we use is a *best-first* version of the kinodynamic planner. It selects the best motion repeatedly until the node reaches the goal position or the allocated time elapses. Each node is evaluated by the following expression:

$$K_L \cdot L + K_\theta |\theta|,$$

where L is the Euclidean distance to the goal position and θ is the angle difference between the robot orientation and the direction of the robot; K_L and K_θ are constant weights. The smaller this expression is, the better the node is.

This best-first planner does not consume much time because it is very efficient and runs with a limited time.

4. Experiments

4.1. Hardware configuration

We use the robot (PeopleBot by MobileRobots Inc.) (see Fig. 6) for in-room experiments. It is equipped with a stereo camera (Bumblebee2 by Point Grey), a laser range finder (UHG-08LX by Hokuyo), and a Note PC (Core2Duo, 2.6GHz, 3GB memory). The PC is in charge of all necessary processing; the image processing and the motion planning part actually run in parallel using the two cores. We also use another robot system which has the same sensors on a different mobile platform, a computer-controllable electric wheelchair (by Kanto Auto Works Ltd.) (see Fig. 7).

4.2. Control algorithm and implementation

The robot repeats the following steps: (1) person detection and tracking (Sec. 2) and mapping, (2) motion planning (Sec. 3), and (3) motion execution. Since the cycle time is set to 500 ms and Step 1 takes about 100 ms, Step 2 can use about 400 ms. The number of nodes generated during the randomized sampling depends on the complexity of the environment but $1,400 \sim 1,500$ nodes are usually generated. Note that the cycle time is a parameter that can be adjusted to respective situations. All functional modules have been developed as RT-components, as described above.

4.3. Experimental result

4.3.1. In-room experiment

Fig. 6 shows a result of person following in a small room. The robot automatically follows the red person while avoiding static and dynamic obstacles. Rows indicate, from top to bottom, the views from the camera in the environment, those from the robot camera, and the map generation and the motion planning results, respectively. In the map, blue, green, and black regions indicate free spaces, static obstacles, and margins considering the robot size, respectively; the orange and the red circles in the maps indicate the robot and the target person.

In the second and the third column, another person passes between the robot and the target person. The triangles in the maps moving rightward indicate that passing person. The robot recognizes the situation correctly and plans a safe motion. In the first and the fourth column, the robot find a path without the randomized planning, because the target person is right ahead and no obstacles exist between the robot and the person.

4.3.2. Experiments in a more complex environment

We next chose the university cafeteria as a more complex environment for person following. We performed experiments more than ten times and the robot successfully follows a specified person while avoiding other persons and static obstacles. The average speed of the robot was about 0.3 m/s. The target person walked at a similar speed so that the robot caught him up, while other persons walked at a normal speed of about 1.0 m/s. Fig. 7 shows snapshots of an experimental run of about 40 m long.

Fig. 8 shows an example of the recognition and the planning result. From the stereo data (see Fig. 8(a)), the robot detected two persons, the target on the left and the other on

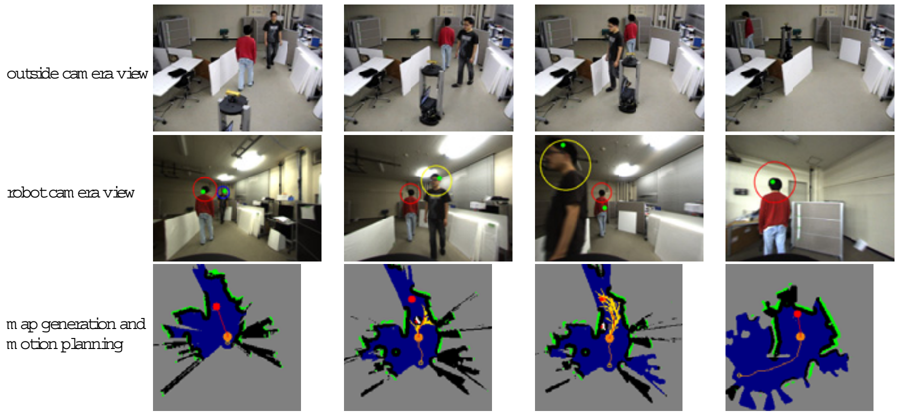
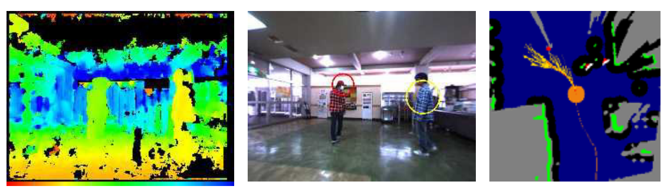


Figure 6. Person following result in an experimental room.

Figure 7. Snapshots of a person following experiment at the cafeteria.

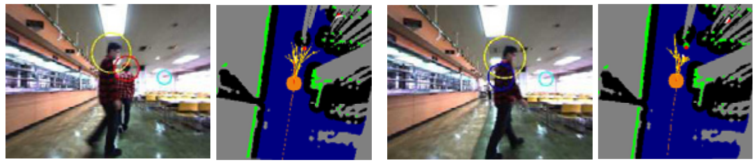


(a) stereo data (b) detected persons (c) recognition result and path planning

Figure 8. An example of environment recognition and motion planning.

the right (see Fig. 8(b)). Fig. 8(c) shows the result of environment recognition and motion planning. The rightmost non-target person is outside the field of view of the camera; the EKF-based tracker predicted his position. Fig. 9 shows an occlusion scene. If the target person is not detected, the robot set his/her latest detected position as a subgoal. The robot makes a plan to avoid the front person.

Visual recognition is sometimes difficult in a real world due to, for example, a great variety of environments and illumination conditions. The proposed person detection and tracking method is relatively robust thanks to stereo-based range information, but sometimes fails to recognize persons correctly. Fig. 10 shows some recognition failure cases. Cases (a) and (b) are due to the failure of SVM-based person verification. Case (c) is the failure of target identification using color due to a bad illumination. Use of other visual features (e.g., HOG) and/or other sensors (e.g., a laser range finder) is a possible



(a) just before occlusion (b) during occlusion

Figure 9. An occlusion scene.

(a) false positives by SVM (b) false negative by SVM (c) target identification error

Figure 10. Visual person detection errors.

improvement of the system.

Even if such improvements are made, however, it is still difficult to completely avoid visual recognition failures. In the case of our current system, since the robot moves to the previous target position when he/she is not detected, it can continue to follow him/her after a short non-detection period. In the case of our experiments with ten runs, the robot did not correctly localize the target for about 15% of the total frames, but could follow the target person without failure. It is, therefore, necessary to consider not only the robustness of each functional module but also that of the whole system.

4.4. On reusability of the modules

Appropriate functional decomposition is a key to simultaneously achieving a high reusability of modules and a high performance of the system. We considered several issues in the decomposition such as hardware dependency, unified interface between modules, cycle time and traffic of communication between modules.

To show the reusability of the developed modules, we exhibited an extended version of the proposed person following robot system, which additionally had an SLAM capability, at the International Robot Exhibition 2009 held in Tokyo on November 25-28, 2009. At the exhibition, we demonstrated two configurations of the person following robot which were different only at the mobile base and the stereo vision with corresponding software modules. The other parts of the system were reused without any modifications. This indicates that a certain level of reusability has been achieved.

5. Conclusion

This paper has described an implementation of person following robot system. The system integrates the necessary functions including person detection and tracking, map generation, motion planning, all of which run on-line, in a self-contained mobile robot. We tested the system in a reasonably complex environment to show the feasibility of the system and to analyze the remaining problems. Each functional module has been developed as an RT-component, which can easily be reused and connected with other components running on RT-middleware.

For more robust and efficient person following, several improvements are possible such as the use of other sensory features, the use of a more accurate motion model of persons specific to each environment, and speeding up each part of the system for coping with more dynamic situations.

Acknowledgment

We would like to thank Dr. Naoyuki Sawasaki and Mr. Manabu Nakao of Fujitsu Laboratories Ltd. for their help in developing RT-components used in our system. This work is supported by NEDO (New Energy and Industrial Technology Development Organization, Japan) Intelligent RT Software Project.

References

[1] Ando, N., Suehiro, T., Kitagaki, K., Kotoku, T., Yoon, W.K.: RT-middleware: Distributed component middleware for RT (robot technology). In: Proceedings of 2005 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3555–3560 (2005)

[2] Beymer, D., Konolige, K.: Tracking people from a moving platform. In: Proceedings of 2002 Int. Symp. on Experimental Robotics, pp. 234–244 (2002)

[3] Calisi, D., Locchi, L., Leone, R.: Person following through appearance models and stereo vision using a mobile robot. In: Proceedings of VISAPP-2007 Workshop on Robot Vision, pp. 46–56 (2007)

[4] Dalal, N., Briggs, B.: Histograms of oriented gradients for human detection. In: Proceedings of 2005 IEEE Conf. on Computer Vision and Patttern Recognition, pp. 886–893 (2005)

[5] Ess, A., Leibe, B., Schindler, K., Gool, L.: A mobile vision system for robust multi-person tracking. In: Proceedings of the 2008 IEEE Conf. on Computer Vision and Pattern Recognition (2008)

[6] Hassouna, M., Abdel-Hakim, A., Farag, A.: Pde-based robust robotic navigation. Image and Vision Computing **27**, 10–18 (2009)

[7] Howard, A., Matthies, L., Huertas, A., Bajracharya, M., Rankin, A.: Detecting pedestrians with stereo vision: Safe operation of autonomous ground vehicles in dynamic environments. In: Proceedings of the 13th Int. Symp. of Robotics Research (2007)

[8] Hsu, D., Kindel, R., Latombe, J., Rock, S.: Randomized kinodynamic motion planning with moving obstacles. Int. J. of Robotics Research **21**(3), 233–255 (2002)

[9] Itoh, K., Kikuchi, T., Takemura, H., Mizoguchi, H.: Development of a person following mobile robot in complicated background by using distance and color information. In: Proceedings of IECON 2006, pp. 3839–3844 (2006)

[10] Lau, B., Sprunk, C., Burgard, W.: Kinodynamic motion planning for mobile robots using splines. In: Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2427–2433 (2009)

[11] Lee, J., Abe, K., Tsubouchi, T., Ichinose, R., Hosoda, Y., Ohba, K.: Collision-free navigation based on people tracking algorithm with biped walking model. In: 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 2983–2989 (2008)

[12] Miura, J., Negishi, Y., Shirai, Y.: Mobile robot map generation by integrating omnidirectional stereo and laser range finder. In: Proceedings of 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 250–255 (2002)

[13] Rodriguez, S., Lien, J.M., Amato, N.: A framework for planning motion in environments with moving obstacles. In: 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 3309–3314 (2007)

[14] Satake, J., Miura, J.: Multiple-person tracking for a mobile robot using stereo. In: Proceedings of IAPR Conf. on Machine Vision and Applications, pp. 273–277 (2009)

[15] Sawano, Y., Miura, J., Shirai, Y.: Man chasing robot by an environment recognition using stereo vision. In: Proceedings of the 2000 Int. Conf. on Machine Automation, pp. 389–394 (2000)