Simulation Of A Human Following Robot With Object Avoidance Function

Nattawat Pinrath
Graduated school of Mechanical Engineering
Shibaura Institute of Technology
3-7-5 Toyosu Kotoku Tokyo
md17422@shibaura-it.ac.jp

Nobuto Matsuhira
Department of Engineering Science and Mechanics
Shibaura Institute of Technology
3-7-5 Toyosu Kotoku Tokyo
matsuhir@shibaura-it.ac.jp

***Abstract*—This study aims to propose an object avoidance algorithm for use in a human-following robot. The algorithm is tested in a well-known robot simulator called virtual robot experiment platform (V-REP), which includes built-in models for realistic sensors and robot platforms. This paper describes a basic path-planning algorithm for following a human target and then describes algorithms for obstacle avoidance and target reacquisition. Simulations are run for each situation that the human-following robot could encounter. Simulation results show that the algorithm is ready to be implemented in a physical robot prototype.**

Keywords—Human-following robot; obstacle avoidance; path planning

I. INTRODUCTION

The introduction of robots in our daily life may add challenges to the standard tasks of autonomous robots. A robot that could follow a person can provide an effective solution for work situations that require an assistant.

In recent years, studies have adopted various sensors for robotic human-following applications, such as light detection and ranging sensors, radio frequency identification modules, laser range finders (LFRs), infrared (IR) sensing modules, thermal imaging sensors, cameras, and wireless transmitter/receivers. These sensors and modules work in unison to detect and follow the target [1]. Many studies have also used image-processing techniques for tracking and identifying target positions. In [2], stereo vision was used to track a human user. Stereo vision systems can acquire various kinds of information, such as color images and depth maps. Wilhelm [3] applied a stereo vision system that focuses on the color of a particular user's skin. Information can be collected from various sensors [4]. However, image processing for these applications is computationally burdensome. Therefore, the proposed study applies the Kinect depth sensor for detecting a human target in a human-following assistant robot [5]. Though the sensor system and target-following algorithms were successful in that study, the robot still needs to detect and avoid obstacles autonomously.

Obstacle avoidance has been addressed by many researchers. Path-planning algorithms are usually based on configurations of known environments or on real-time data collected from unknown environments. In a known environment, the path-planning algorithm will acquire pre-loaded data about the position of obstacles and will create a path without collisions [6]. Various algorithms are effective for path planning in this situation, such as the occupancy grids, A* algorithm, quad-tree algorithm [7], simultaneous localization and mapping (SLAM), and the local planning algorithm. Algorithms typically used for path planning for real-time object avoidance in unknown environments include vector field histogram (VFH) [8][9], bug algorithm [10], potential field algorithm [11] [12], and bubble-band technique [13].

In recent years, many researchers have devised and developed algorithms for human-following and object avoidance. The proposed study tests such an algorithm using the virtual robot experimentation platform (V-REP) from Copella Robotics. The simulated robot is equipped with a vision sensor and scanning laser rangefinder, both modeled in V-REP. This virtual environment allows us to model a robot that follows a human at a safe distance while avoiding obstacles and re-acquiring target positions when they are obstructed in an unknown environment.

II. MODELING

A. V-REP Simulation and model

Robotics simulators are frequently used in the design and testing of control algorithms to reduce the time and cost involved in building physical prototypes. V-REP allows researchers to consider the relevant parameters for a robot-control system in a short period of time. V-REP is one of the most widely used robotics simulators for educational and research purposes and includes a free license for educational use.

V-REP includes an integrated development environment and is based on distributed control architecture so

978-1-5386-5094-3/18/$31.00 ©2018 IEEE

that each modeled object can be individually controlled via an embedded script, a plugin, a ROS node, a remote API client, or a custom solution. V-REP can be used as a stand-alone application or can easily be embedded into a main client application. The simulator includes models of a range of sensors, robots, and actuators that can interact with a simulated scene [14].

This research focuses on a model available in V-REP, the Pioneer 3 DX, as it is widely used in the field.

B. Pioneer 3 DX mobile-robot model

The directory for the Pioneer 3-DX can be found with the following path:

…\VREP3\VREP_PRO_EDU\models\robots\mobile\pioneer p3dx.ttm

The Pioneer 3 DX is the world's most popular research mobile robot. Its versatility, reliability, and durability have made it a standard platform for the robotic research. Pioneer 3 DX is a differential wheeled robot whose movement is based on two separately driven wheels and motor placed on each side of its robot. The robot model in V-REP is shown in Fig. 1. The velocity of each wheel is controlled using the simSetJointPosition function in the Lua programming language.

Moreover, we included the vision sensor and the laser rangefinder model named "HokuyoFast." The directory for this model can be located in a typical V-REP installation with the following path:

…\VREP3\VREP_PRO_EDU\models\component\sensor\Hokuyo URG 04LXUG01_Fast.ttm

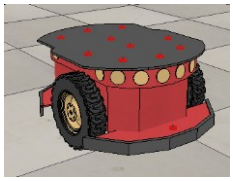
The vision sensor is used for target detection. The vision sensor is set to detect red color with RGB values (100,0,0) and we simulated a situation with only one such object for the sake of simplicity. The screenshot of the vision sensor detecting a red object is shown in Fig. 3. The target position is detected using the SimGetObjectPosition function in Lua script. The value returned by SimGetObjectPosition is a position in x, y, and z coordinates of the target relative to the vision sensor's position.

If an obstacle is detected by the HokyuoFast sensor, the value will be recorded in matrix coordinates [x,y,z] at one time. This model simulates the Hokuyo URG 04lx-ug01 sensor. The maximum scan distance is set to 5 meters and the scanning pans over 240 degrees to match the specifications of the actual sensor.

C. Human model

The human to be followed by the robot was modeled with the "Walking Bill" model in V-REP, as shown in Fig. 2. Walking Bill is available at the following directory:

…\V-REP3\V-REP_PRO_EDU\models\people.



(1)

(2)

Fig. 1 Pioneer 3 DX model in V-REP software
Fig. 2 Walking Bill model in V-REP software

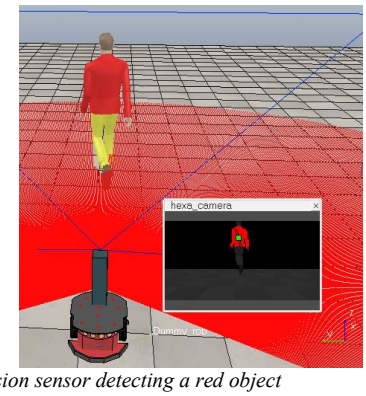


Fig. 3. Vision sensor detecting a red object

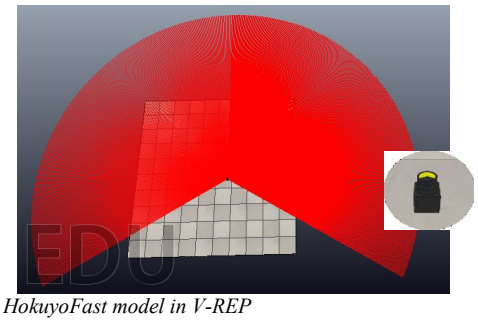


Fig. 4. HokuyoFast model in V-REP

In the original file, Bill walks randomly around the simulation scene. In this experiment, we modified the script to control Bill with a game pad. V-REP gets data from the gamepad and translates it to control Bill's direction, and the model walks at the speed of 2 m/s.

III. FOLLOWING ALGORITHM

A. Human follwing Algorithm

The algorithm for human following and object avoidance is shown in Fig. 3.

In the first stage, the robot finds the target. In this simulation, the target must be red in color so that the vision sensor can detect it. A Lua script records the target coordinates, and then calculates the angle ($\theta_{(r)}$) between the robot heading and the target position using equation (1). $\theta_{(r)}$ is the angle between the target position and the robot heading, y_t is the distance along the y axis, and x_t is the distance along the x axis.

If the angle between robot heading and target position is not zero, the robot will rotate to the point toward the target position. The direction of rotation depends on the angle, if $\theta_{(r)} > 0$ the robot will turn right, and if $\theta_{(r)} < 0$ the robot will turn left.

$$\theta_{(r)} = \tan^{-1}(y_t/x_t) \quad (1)$$

$$D_{tb} = \sqrt{(x_t^2 + y_t^2)} \quad (2)$$

In the next step, the robot calculates the distance between the robot and target, (D_{tb}). The robot then moves in the shortest path toward the target as long as the distance between robot and target is more than 1 meter. If distance between robot and target is less than 1 meter, the robot stop moving, and continue to rotate to point toward the target. This prevents the robot from detecting the target as an obstacle and ensures that the robot maintains a safe distance from the human user.

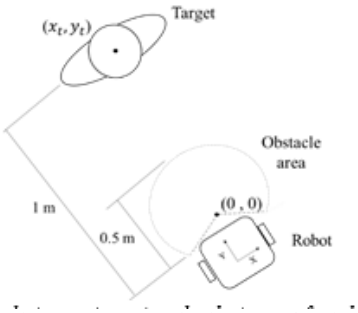


Fig. 5. Distance between target and robot, area for obstacle detection, and coordinate system used

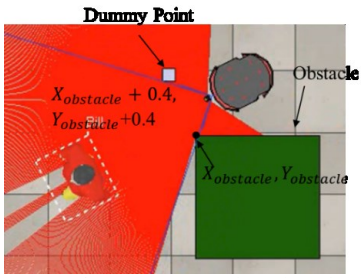


Fig. 4. Dummy point for moving around an obstacle

B. Human-following Algorithm and avoidance object

The algorithm for object avoidance is activated while the robot follows a human user. The HokuyoFast model detects an obstacle within the range of 0.5 meters from the robot. The algorithm retrieves the obstacle's coordinates and calculates the distance to the obstacle and the angle between the robot heading and the obstacle. The algorithm then creates a dummy point to direct the robot around the obstacle. The position of the dummy point depends on the minimum and maximum coordinates of the obstacle with respect to the direction of the robot, as shown in Fig. 4.

Once the dummy point has been created, the robot will move toward it. First, the robot rotates until it points toward the dummy point, as the robot matches heading with the human target above. Once the robot's heading matches with the dummy position, the robot will move toward the dummy position at fast speed until the robot reaches the dummy position. Then the robot scans for the target again and resumes the human-following algorithm.

During obstacle avoidance, the vision sensor will lose contact with the target if the obstacle is too large or the robot has to move away from the target heading to avoid the obstacle. Furthermore, some environments obstruct the robot's view of the human target, such as when the human walks behind a wall, as seen in Figs. 4(a) and 4(b).

The algorithm used for re-acquiring the target when contact is lost is shown in Fig. 5. When the robot loses contact with the target, the robot will acquire the last target position (x_l, y_l), and the robot takes the shortest path to the last target position. This process uses the same path-planning algorithm described above. However, while the robot moves to the last detected target position, the robot will switch back to the target-following algorithm if the vision sensor detects the target.

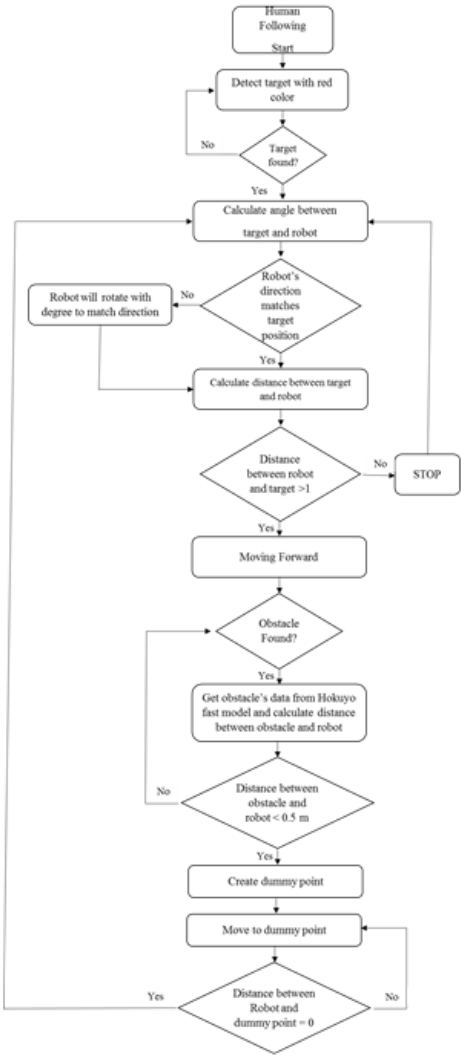


Fig. 3. Decision flow chart for target-following algorithm with obstacle avoidance

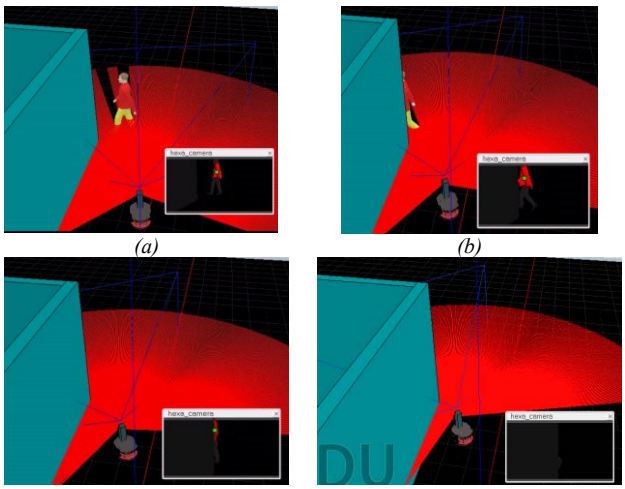


Fig. 4. Situation in which the robot cannot detect the target

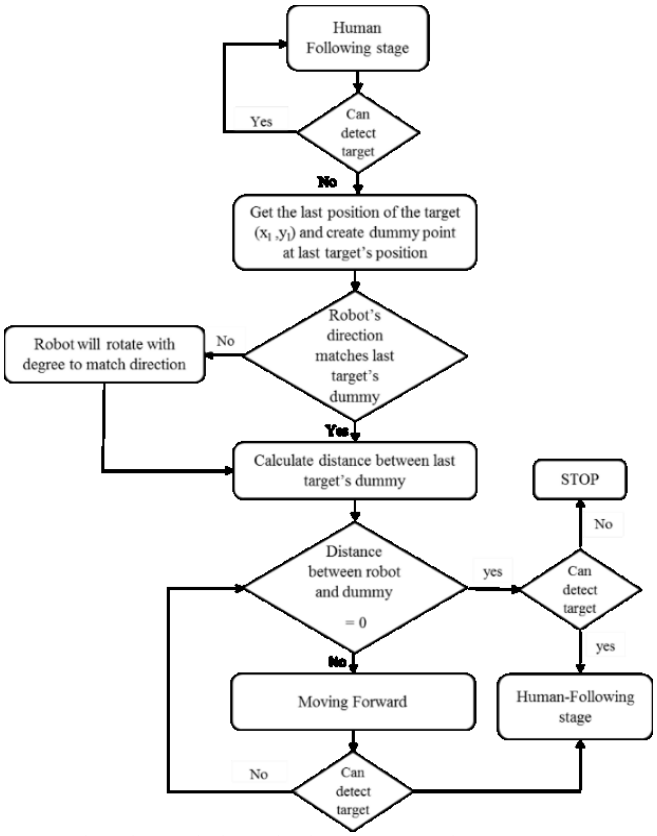


Fig. 5. Algorithm used when robot loses contact with target

IV. SIMULATION RESULTS AND DISCUSSION

A. Human-following behavior

The results of a simple human-following simulation are shown in Fig. 6. In this case, the human model moves from left to right of the pictured scene and the range of vision sensor is 130 degrees, and is marked with blue lines. In the first stage, in Fig. 6(a), the sensor cannot detect the target because the human is outside the detection area of the vision sensor. In Fig. 6(b), the human target in a red shirt walks through the sensor area, so the sensor can detect the target. Next, the robot calculates the angle between the target and itself, and turns until it points toward the human. In Fig. 6(c), the distance between the robot and target is less than 1-meter, so the robot does not move forward but continues to rotate to the point toward the human. In Fig. 6(d), when the distance between robot and human is again more than 1 meter, the robot will move forward to follow the human target.

B. Human following and object avoidance

While the robot followed its target during the simulation, two obstacles were placed in the space, marked by the green boxes in Fig. 7. The human model walks from the right of Fig. 7 to the bottom left. After the human model passes the lower green box, it will turn 45 degrees to the left and continue walking forward. In this case, the shortest path between the human and robot is obstructed.

The simulation results show the point at which the laser rangefinder detects an obstacle. A dummy point will then be created as shown in Fig. 7.

In Fig. 8(a), the robot changes direction and moves to the dummy point. Fig. 8(b) shows that when the robot reaches the dummy point, the robot acquires the target position and changes the direction, as in Fig. 8(c). The robot then continues following as in Fig. 8(d).

Lastly, we simulated a complex environment that resembles a maze, as shown in Fig. 9. This area includes high green walls. The human model walks following the wall to the end position marked in Fig. 9. The robot follows human and must avoid the obstacle wall that is vertical and near the middle of Fig. 9.

In Fig. 10(a), the human model is walking forward and once the distance between robot and human is more than 1 meter, the robot starts to follow the human. In Fig. 10(b), the human model stops in the corner to decide whether to turn left or

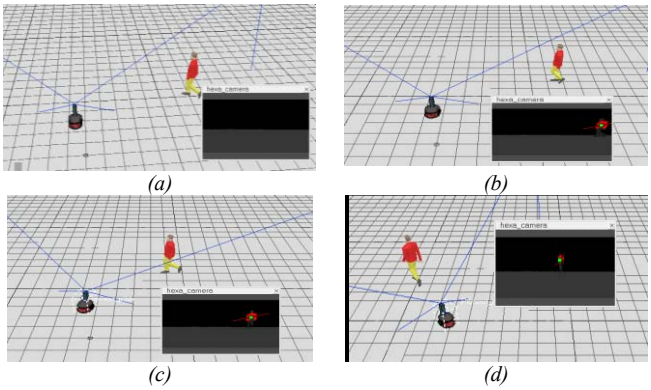


Fig. 6 show the robot detected human and following in V-REP simulation

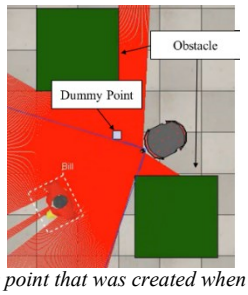
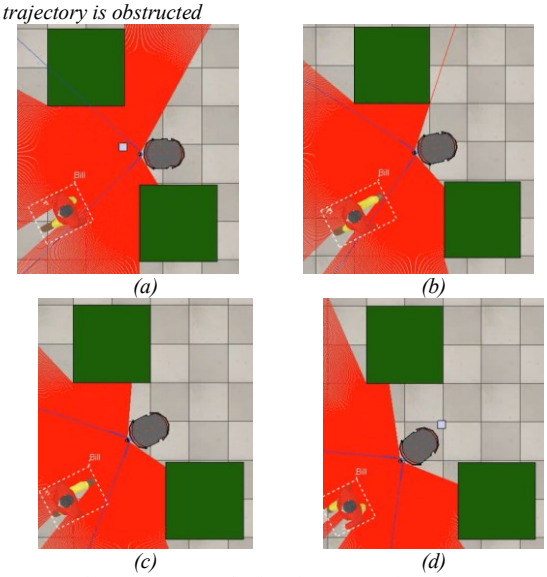


Fig. 7 show dummy point that was created when the robot trajectory is obstructed



(a) (b) (c) (d)

Fig. 8 show movement of robot during avoidance stage

right. The robot also stops once the distance between the robot and human is 1 meter, and it waits for the human model to move. In Fig. 10(c), the robot trajectory must adjust for the side wall, so the robot turns until the sensor no longer detects the obstacle, as in Fig. 10(d). In Fig. 10(e), the robot continues to follow the human model. The trajectories of the robot and human in this simulation are shown in Fig. 10. The red line is the human trajectory and the pink line is the robot trajectory.

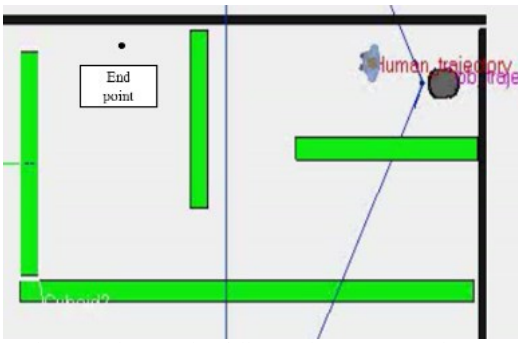
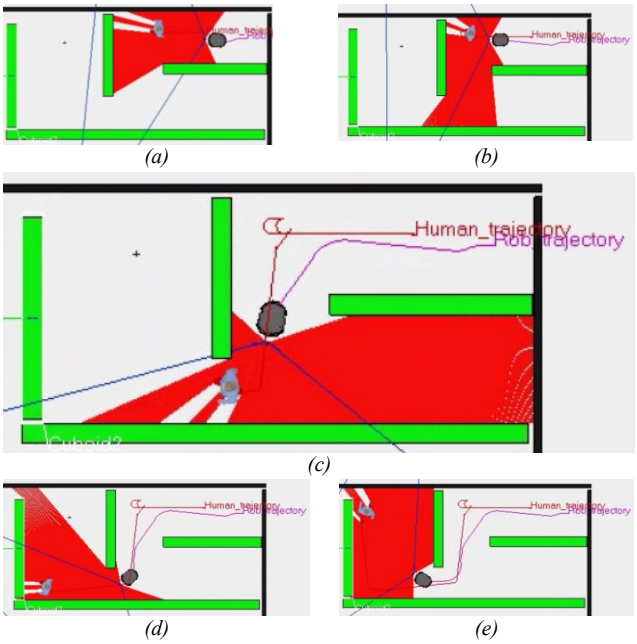


Fig. 9 show simulation robot following human in complex environments



(a) (b) (c) (d) (e)

Fig. 10. Steps in the robot following process

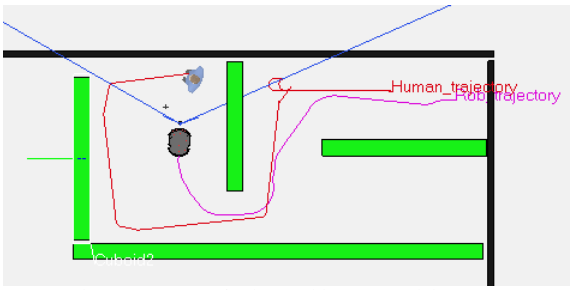
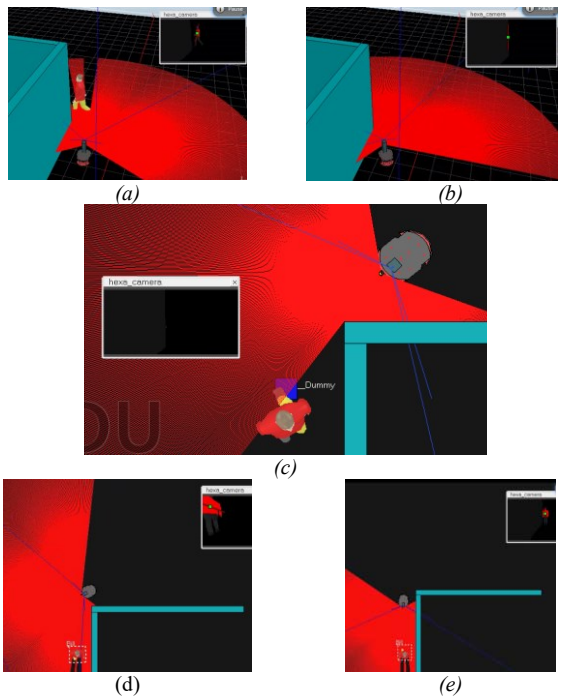


Fig. 11. The trajectories of robot and human model



(a) (b) (c) (d) (e)

Fig. 12. Simulation of robot losing contact with target

C. *Robot loses contact with target*

Finally, we simulated a situation in which the robot loses contact with the target, as seen in Fig. 12(a). The robot loses the target when the target goes around the corner in Fig. 12(b). In Fig. 12(c), the robot marks the last target position and creates a dummy point at the last position (blue area). The robot will then move to the dummy position. In Fig. 12(d), the vision sensor reacquires the target as the robot moves to the dummy point, so the robot changes heading to follow the human, moving forward again in Fig. 12(e)

V. CONCLUSIONS

In this paper, we propose a robot that autonomously follows a human target [5]. This study tests the obstacle detection and avoidance algorithms for the robot. The algorithm tracks only one object marked with red clothing, and the code written in Lua language retrieves the position of the target from sensor data. These strategies reduce the CPU usage for the simulation and simulation time.

For target following and obstacle avoidance, the algorithm uses vision sensor data to detect the human target and a laser rangefinder to detect obstacles. The range for obstacle detection must be less than the minimum distance between the robot and human because the laser rangefinder would otherwise constantly detect the human as an obstacle. Simulations show that the algorithms developed in this paper are ready for implementation in a physical prototype, which will be a subject for future work.

ACKNOWLEDGMENT

We would like to thank Dr. Marc Freese from Coppelia Robotics for his suggestions about the models in V-REP.

REFERENCES

[1] M. s. Hassan, M. W. Khan and A. F. Khan, “Design and development of human following robot,” in Student Research Paper Conference., Vol 2, No 15, July 2015.

[2] D. calisi, L. Locchi, and G. R. Leone, “Person following through appearance models and stero vision using a mobile robot,’in Proc. Int. workshop Robot Vis., 2007, pp: 46–56

[3] W. Chung, H. Kim, Y. Yoo, C. Moon and J. Park, “The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder,” IEEE Trans. Ind. Eletron., vol. 59 no 8, Aug. 2012.

[4] T. Wilhelm, H. J. Bohme, and H. M, Gross, “Sensor fusion for vision and sonar based people tracking on amobile service robot,’in Proceedings of the International Workshop on Dynamic Perception, 2002, pp: 315–320

[5] Shimoyama Mirai, Mutsuhira Nobuto, Suzuki Kaoru, ‘Human Characterization by a Following Robot Using a Depth Sensor(I),” in IEEE/SICE International Symposium on System Integration, 2017.

[6] Charles W.Warren, “Fast path-planning method using modified A*method,”in IEEE International Conference on Robotics and Automation, pp: 662–667

[7] Borenstein, J.,Koren, Y., “The Vector Field Histogram–Fast Obstacle Avoidance for Mobile Robots.”IEEE Journal of Robotics and Automation, 7, pp: 278–288, 1991

[8] Borenstein, J., Koren, Y., “The Vector Field Histogram–Fast Obstacle Avoidance for Mobile Robots.” IEEE Journal of Robotics and Automation, 7, pp: 278–288, 1991

[9] Ulrich, I., Borenstein, J., “VFH*: Local Obstacle Avoidance with Look-Ahead Verification,” in Proceedings of the IEEE International Conference on Robotics and Automation, San Francisco, May 24–28, 2000.

[10] Kamon, I., Rivlin, E., Rimon, E., “A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robots,” in Proceedings of the IEEE International Conference on Robotics and Automation, Minneapolis, April 1996

[11] Khatib, O., 1985, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots."1985 IEEE International Conference on Robotics and Automation, March 25-28, St. Louis, pp: 500–505

[12] Koren, Y., Borenstein, J., “High-Speed Obstacle Avoidance for Mobile Robotics,” in Proceedings of the IEEE Symposium on Intelligent Control, Arlington, VA, August 1988, pp: 382–384

[13] Khatib O., Quinlan S., Ëlastic Bands: Connecting Path Plainng and Control,”in Proceedings of IEEE International Conference on Robotics and Automation, Atlanta, GA, May 1993

[14] Marc Freese, Fumio Ozaki, Shigeo Hirose and Nobuto Matsuhira., “Virtual Robot Experimentation Platform–A Versatile Small Robot Simulator,” Journal of Robotics and mechatronics, Vol.20, No.1 2008