The 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2015)
October 28~30, 2015 / KINTEX, Goyang city, Korea

Pedestrian-following service robot applications using chance-constrained target tracking

Jinyoung Choi[1], Sunwoo Lee[2], Yoonseon Oh[3] and Songhwai Oh[4]
[1,2] Department of Electrical Engineering, Seoul National University, Korea
(Tel: +82-10-2011-5349; E-mail: {soapcandy, woodstar1}@snu.ac.kr)
[3,4] Department of Electrical Engineering, Seoul National University, Korea
(Tel : +82-2-880-1512; E-mail: {yoonseon.oh, songhwai.oh}@cpslab.snu.ac.kr)

Abstract This paper presents a mobile robot system where the robot tracks a moving target. The system minimizes the probability of losing the target. If the next position of a moving target has the Gaussian distribution, the proposed system guarantees the tracking success probability. In addition, we minimize the moving distance of the mobile robot based on the chosen bound on the tracking success probability. We built a well-designed system on Robot Operating System for applications such as a smart shopping cart, selfie robot, and transporter. The performance of the proposed system is extensively evaluated in field experiments.

Keywords – Robust target tracking, chance-constrained optimization, ROS

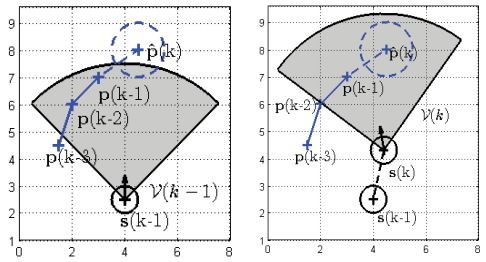
1. Introduction

Mobile robots tracking a person have received considerable attention for a number of applications such as transporting loads, serving customers, and caring seniors or infants. In general, sensors which is mounted on mobile robots have a fan-shaped sensing region with a limited range. Some authors proposed control laws such that risk functions to lose a target are minimized [1]-[3]. However, they assume a simple constant velocity model [3] or do not predict the motion of the target [1], [2]. Other authors proposed control inputs to minimize the uncertainty of the predicted location of the target [4], [5]. Instead, we use the algorithm which minimizes the tracking failure probability for the guaranteed performance [6].

In this paper, we build a fast and stable system by designing the system on Robot Operating System[1]. Our system achieves high speed and compatibility by implementing the algorithm as a ROS package and using a GPU (Graphics Processing Units) for a prediction algorithm. While the preliminary version [6] focuses on evaluating the performance of a tracking method itself, this paper validates its applicability by extensive experiments for a smart shopping cart, selfie robot, and transporter robot.

We briefly introduce a chance-constrained target tracking algorithm [6] in Section 2. Section 3 explains the architecture of the system and shows the performance. In Section 4, we validate our algorithm for several applications for service robots.

[1] http://www.ros.org/



(a) Time *k-1* (b) Time k

Fig. 1. An illustration of the target tracking problem considered in this paper. Gray regions are sensing region of a mobile robot. A target is detected at time k-1 and the predicted new location of the target is p(k) at time k. The blue dashed line represents the variance of the prediction.

2. Preliminaries

This section defines the problem of tracking a target and introduces a chance-constrained target tracking algorithm [6]. We assume that a mobile robot and a target move on a 2D plane. Let $\mathrm{s(k)} = [x_s(k)\ y_s(k)]^{\mathrm{T}}$ be the position of the mobile robot at time k. The heading of the robot is the angle from the x-axis and denoted by $\varphi_s(k)$. Let $\mathrm{u(k)} = [\,u_v\ \ u_w]^{\mathrm{T}}$ be the control input at time k and the dynamic model is a unicycle model. The motion of the robot is described by a discrete-time dynamic system

$$[\mathrm{s(k+1)}\ \ \varphi_s(k+1)]^{\mathrm{T}} = \mathrm{f}\big(\mathrm{s(k)}, \varphi_s(k), \mathrm{u(k)}\big).$$

The moving distance of the robot is denoted by $\mathrm{d(u(k))}$.

We assume that the mobile robot carries a sensor with a finite and fan-shaped sensing region. The sensing region of the mobile robot at time k is denoted by $V(\mathrm{k})$. Its angular field of view is θ_s and the maximum range is R_s. We assume that the sensor is rigidly attached to the mobile robot.

Figure 1 shows an illustration, in which a mobile robot has detected a target from time k-2 to k-1 (figure 1(a)) and moves to a new location to make sure the target is within the sensing range (figure 1(b)). We assume that the distribution of the next position of the target is available using a motion prediction algorithm. Let $\mathrm{p(k)} = [x_T(k)\ \ y_T(k)]^T$ be the position of the target at time k. From the motion prediction algorithm, the target's position at time k has a Gaussian distribution with mean $\hat{p}(k)$ and covariance $\sum_T(k)$.

978-1-4673-7971-7/15/$31.00 ©2015 IEEE

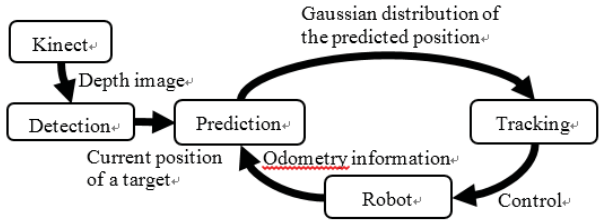


Fig. 2. Flow of the system. The black arrows represent the transmition of the ROS messages.

Our goal is to find a control $\mathrm{u(k-1)}$ which minimizes the tracking failure probability $\mathrm{P}\big(\mathrm{p(k)} \notin V(\mathrm{k})\big)$ and the moving distance of a mobile sensor. Using the chance-constrained optimization method [7], we formulate the tracking problem as the following multi-objective optimization problem:

$$\begin{aligned} \Pi_0 : \min_{\varphi(k)} \quad & \varepsilon^*(k) \\ \text{subject to } & \mathrm{P}\big(\mathrm{p(k)} \notin V(\mathrm{k})\big) \le \varepsilon(k), \\ & 0 \le E_1 \le \varepsilon(k) \le E_2 \le 1, \\ & [\mathrm{s(k)} \quad \varphi_s(k)]^{\mathrm{T}} \\ & = \mathrm{f}(\mathrm{s(k-1)}, \varphi_s(k-1), u(k-1)), \end{aligned} \tag{1}$$

where $\varepsilon(k)$ is the upper bound on the tracking failure probability.

To solve this multi-objective optimization problem Π_0, we optimize $\varepsilon(k)$ first and optimize d(u(k-1)) under the optimized $\varepsilon(k)$. The optimization problem is first formulated by a non-convex problem. By approximating the sensing region and fixing the next heading of the sensor, we find the analytical solution $\varepsilon^*(k)$. This procedure is repeated for each $\varphi_s \in H$, a set of candidate headings, to find the optimal control which minimizes the tracking failure probability and the moving distance.

3. System Architecture

This section presents the architecture of the robot system and its performance.

3.1 Overview of the system

The system architecture is shown in Figure 2. The system consists of three modules for detection, prediction, and tracking. The whole system is implemented by C++ on ROS (Robot Operating System) and each module is executed by a separated program (node). Each node communicates with each other by ROS messages.

The first module is a detection module. It detects a person using a depth image of a Kinect sensor and sends detected positions to a prediction module. The prediction module predicts the next position of a target based on measurements. Since these measurements are generated in different coordinate systems, the odometry information of the robot is required to find the global positions of the target. Then the module sends the distribution of the next position to a tracking module. The tracking module calculates the optimal control of the robot and sends it to the robot.

3.2 Detection module

There are many algorithms and open sources to detect a pedestrian such as a HOG pedestrian detector [8], part based model [8], particle filter based tracker [9], and OPENNI [2]. A HOG pedestrian detector [8] is sensitive to changes in a human pose and the whole human body should be in the sensing region. In contrast, since OPENNI is robust against changes in a human pose and its computation speed is fast, we employ OPENNI. However, the method does not identify the target when there are multiple detections. We choose the closest pedestrian to the previous position of the target and it results lower failure probability.

3.3 Prediction module

For the prediction module, a Gaussian process regression [6] is chosen as a prediction algorithm. The next position is predicted based on three recent positions of the target. The Gaussian process regression is trained using 3480 trajectories of pedestrians. The prediction module takes about 0.8 seconds when we use only CPU. It is not acceptable to a real-time system. So we use NVIDIA CUDA (Computer Unified Device Architecture) for parallel computation. It takes only 0.06 seconds after applying CUDA.

3.4 Tracking module

A tracking module computes the optimal control of the mobile robot. A candidate set of the angular velocity u_w has uniformly spaced discrete values in the range of $W_{min} \le u_w \le W_{max}$, where $W_{min} = -140°/s$ and $W_{max} = 140°/s$. We set the maximum and minimum linear velocity to $V_{min} = -700\ mm/s$ and $V_{max} = 700\ mm/s$, respectively. We set $\mathrm{R_s} = 3500\,mm$ and $\theta_s = 57°$. Parameters for the tracking failure probability are set to $E_1 = 0.0001$ and $E_2 = 1$. If the ROS messages from the prediction module are not received for 0.15sec due to target loss or malfunction on the detection and prediction modules, the system stops tracking.

3.5 System specification

In our implementation, we mounted the Kinect sensor on the Pioneer mobile robot platform. A Pioneer mobile robot platform is popular due to its excellent maneuverability and stability. We choose ROS Indigo on Ubuntu 14.04 as an OS for our system. The specifications of the computer for our system are i5 quad core 2.6GHz CPU, 8GB RAM, and NVIDIA Geforce940m GPU.

We use ROS packages to control the pioneer robot and the Kinect sensor with ROS messages. ROSARIA [3] package provides interface for Pioneer and sends the robot a desired velocity received from a tracking module. We extract the position of the target from a Kinect using Openni_launch package.

3.5 Results

The tracking module calculates the optimal angular and linear velocity and sends them to the robot. It takes about 0.15 s to detect a target and compute a control input for tracking. The average speed of the robot is 1.2m/s.

4. Applications

We suggest three applications using the chance-constrained tracking algorithm. First, we introduce a shopping cart robot which follows customers and helps their convenient shopping. In Figure 3, the shopping cart is

[2] http://wiki.ros.org/openni_launch
[3] http://wiki.ros.org/ROSARIA

Fig. 4. A selfie robot. First four images are snapshots from a third-view camera and the last image is a photo taken by a mobile robot at 25s.

Fig. 3. A smart shopping cart robot. The person with a blue shirt is a target and a mobile robot with a gray basket is a shopping cart.

a robot to which a basket is attached. The shopping cart robot needs to track a user in close distance. So we change the radius of the sensing region to 90cm. Since complex environments decrease quality of detection, we lower the probability of sensing other objects by shortening the sensing region. We launch robot on the various location of the local market and validate the algorithm (see Figure 3).

Selfie robot is the robot that follows a user and takes pictures of the user without any manual control. When the user stops at the same position for 5s, it takes pictures and records important events. In Figure 4, the robot captures the RGB image of the Kinect sensor at 25s. If we set the robot to record whole video instead of the image, it can be used as a cameraman robot or care robot for seniors or infants.

The tracking algorithm can be used as a transporter robot. In the storage, the robot follows a user and moves the box along the user like Figure 5. Our robot successes in all five trials. It was much faster and easier than moving the box manually.

5. Conclusion

We suggest a robot system which tracks a pedestrian. The tracking algorithm minimizes both the probability of losing a target and a traveling distance of the target. By considering the distribution of predicted position of a target, we get robust performance against the uncertainty of prediction. In addition, we validate the system for three applications in the real world and it shows high performance.

Acknowledgement

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2013R1A1A2009348). Y. Oh and S. Oh are with the Department of Electrical and Computer Engineering, ASRI, Seoul National University, Seoul 151-744, Korea.

Fig. 5. A transporter robot. The person with a blue shirt is a target and a robot carries a yellow box.

References

[1] C.Y. Lee, H. Gonzalez-Banos, and J.C. Latombe, “Real-time tracking of an unpredictable target amidst unknown obstacles,” in 7th International Conference on Control, Automation, Robotics and Vision (ICARCV), Dec 2002.

[2] H. Gonzalez-Banos, C.Y. Lee, and J.C. Latombe, “Real-time combinatorial tracking of a target moving unpredictably among obstacles,” in Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2002.

[3] T. Bandyopadhyay, N. Rong, M. Ang, D. Hsu, and W. S. Lee, “Motion planning for people tracking in uncertain and dynamic environments,” Workshop on people detection and tracking, IEEE International Conference on Robotics and Automation (ICRA), 2009.

[4] E. Frew and S. Rock, “Trajectory generation for constant velocity target motion estimation using monocular vision,” in Proc. of the IEEE International Conference on Robotics and Automation (ICRA), Sept 2003.

[5] K. Zhou and S. Roumeliotis, “Multirobot active target tracking with combinations of relative observations,” IEEE Transactions on Robotics, vol. 27, no. 4, Aug 2011.

[6] Y. Oh, S. Choi, and S. Oh, "Chance-Constrained Target Tracking for Mobile Robots," in Proc. of the IEEE International Conference on Robotics and Automation (ICRA), May 2015.

[7] L. Blackmore and M. Ono, “Convex chance constrained predictive control without sampling,” in Proc. of the AIAA Guidance, Navigation and Control Conference, August 2009.

[8] N. Dalal and B. Triggs. "Histograms of oriented gradients for human detection," Computer Vision and Pattern Recognition, 2005.

[9] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," Pattern Analysis and Machine Intelligence, IEEE Transactions on, Sept. 2010.

[10] K. Nummiaro, E. Koller-Meier, and L. V. Gool. "Object tracking with an adaptive color-based particle filter." Pattern Recognition. Springer Berlin Heidelberg, 2002.