Autonomous Vehicle Navigation by Building 3D Map and by Detecting Human Trajectory using LIDAR

Satoshi Kagami[*†‡‖§], Simon Thompson[*§], Ippei Samejima[*‡§], Tsuyoshi Hamada[¶], Shinpei Kato[‖],
Naotaka Hatao[*§], Yuma Nihei[*†§], Takuro Egawa[*†§], Kazuya Takeda[‖], Hiroshi Takemura[†*], Hiroshi Mizoguchi[†*]
∗ Digital Human Research Center, National Institute of Advanced Industrial Science and Technology, † Tokyo Univ. of Science
‡ Nara Inst. of Science and Technology, ¶ Nagasaki University, ‖ Nagoya University, § Japan Science and Technology Agency

***Abstract*—This paper describes an autonomous vehicle navigation system based on Velodyne LIDAR. The system mainly focusing on an autonomy at the car park, and it includes following functions, 1) odometory correction, 2) 3D map building, 3) localization, 4) detecting human trajectories as well as static obstacles, 5) path planning, and 6) vehicle control to a given trajectory. All those functions are developed on ROS. Car park of 70x50[m] area is used for experiment and results are shown.**

I. Introduction

Research and development of autonomous functions for a road vehicle becomes increasingly active in recent years. DARPA grand challenge [1, 2] and urban challenge [3, 4] widely opened this field. Currently Google self-driving car is running in public load over hundred thousand miles.

One particular difficulty for autonomous vehicle navigation is, it needs to be concious with not only static environment, but with others who are also moving around. So, detection and tracking function of others are needed.

In this paper, we are studying an autonomy at car park in urban environment where GPS signal is not reachable because of limited visibility to the sky. In such a place, map building and localization functions are required to plan a path and to control a vehicle. Moreover static obstacle finding and walking human detection are also required. Vertical wall segment based method is proposed to build a map and to localize where the vehicle is. SJPDAFs combined with SVM classifier is proposed to find out and track walking human being.

II. Vehicle Configuration

A. Hardware

In this paper, we adopted ZMP Robocar HV as a vehicle platform, that was modified based on Toyota Prius NHW20. Steering, accel, brake and shift positions can be controlled.

As for an external sensor, Velodyne HDL-32e is adopted. This LIDAR has rotating head with 32 time of flight simultaneously measuring distance sensor of Class 1 eye safe laser at 905[nm] wave length. Therefore, observation becomes multilayered concentric point clouds. View angle is 30.7[deg] for lower side from horizontal plane, and 10.7[deg] for upper side. Distance range is about 70[m] with 2[cm] in 1σ at 25[m]. Measuring cycle of rotation is 10[hz] in this paper.

Fig.1 shows the vehicle with Velodyne LIDAR is attached on the roof.

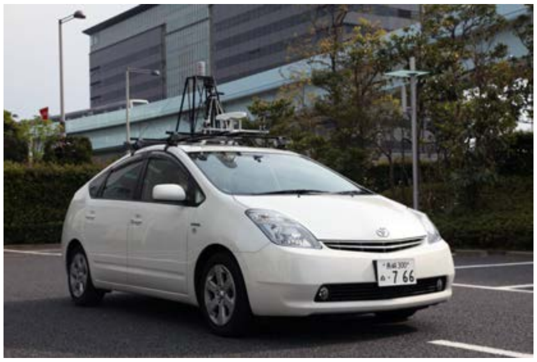

Fig. 1. Vehicle with LIDAR on roof

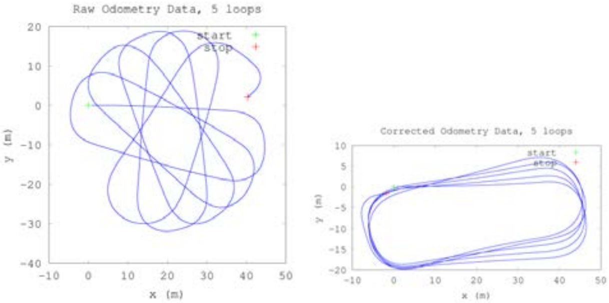

Fig. 2. Raw odometry data from 5 loops of car-park (left) and corrected odometry (right)

Lower boundary of view cone of Velodyne HDL-32e is 30[deg] from horizontal axis. We searched a position in 2D space to minimize self-obsclusion of the car, and result was shown in the figure. Obtained position of LIDAR was 2.27[m] from the ground and 1.0[m] front from rear wheel axis. View cone slightly hit four corners of the car, and the cone hits to the ground with about 7.7[m] diameter. Lower of this cone is not visible.

B. Software System

We developed our software on robotics middleware ROS. ROS version is *Fuerte*. Introduced packages are *velodyne_driver, velodyne_pointcloud, pointcloud_to_laserscan, map_server, slam_gmapping, amcl and tf.*

C. Odometry

Access to car state information is through the ZMP Robocar HV interface API. Both current velocity [km/h] and a steering angle measure α [rad] can be read. Translational velocity v [m/s] and angular velocity ω [rad] can then be determined. From this, an odometry estimate (x, y, θ, 2D position and yaw), can be calculated, using the bicycle mode of motion:

$$x' = x + v \cdot cos(\theta)$$

$$y' = y' + v \cdot sin(\theta)$$

$$\theta' = \theta + \frac{v}{L} \cdot tan(\frac{\alpha}{N})$$

where, L is the wheel base, and N is the steering ratio.

Raw odometry readings contain some bias. To correct for this bias, an offset in angular velocity and a scalar adjustor of translational velocity were estimated, and used to correct measured velocity data:

$$\omega_c = \omega_r + \epsilon_\omega$$

$$v_c = v_r \cdot \epsilon_v$$

where, ω_c and ω_r refer to corrected and raw angular velocity respectively, and ϵ_ω to the bias in angular velocity. Similarly v_c and v_r refer to the corrected and raw translational velocity and ϵ_v to the translational velocity weighting co-efficient.

To estimate ϵ_v and $epsilon_\omega$ raw odometry was captured from various circular motions of known radius, and constant angular velocity, with the motion starting and stopping on the same location. A combination of raw odometry and candidate values of epsilon can be used to predict a stop position for the car. An error measure was defined as the difference between predicted odometric path and the real path of the final stop position and the total distance travelled. A search of the bias values was performed to minimize this error measure. For the low speeds used in these experiments $(5-20)$[km/h], ϵ_v was estimated to be 1.02 ϵ_ω to 0.61 [deg].

Odometry was captured over 5 loops around a parking lot. The raw odometry data and corrected odometry using ϵ_v and ϵ_ω as defined above, are shown in Fig.2.

III. Mapping, Localization and Static Obstacle Detection

Point cloud is obtained by Velodyne LIDAR by using *pointcloud_to_laserscan* package. Vertical wall segment is searched from the data, and appropriate height data is projected into 2D plane and converted into laserscan data. In order to avoid walking human and other vehicles, above 2[m] from the ground is searched.

Fig.3 shows the map built by manually driving one round at our car park. In this experiment, 3.0–3.5[m] height is used. Grey region shows objects in map. Velodyne LIDAR input shows concentric points where orange is low and green is high in elevation. Localized vehicle trajectory is shown in yellow line.

Fig.4 shows reprojected 3D points by localizing vehicle trajecotry. Again point color shows elevation by the gradation from orange to green. Parking vehicles and trees can be seen.

Fig.6 shows a static obstacle map calculated from 3D map. Objects observed above ground and under vehicle hight is statistically updated. Black pixels are observed multiple times during one round observation. There were seveal human walking in the car park, and their trajectories are also remained.

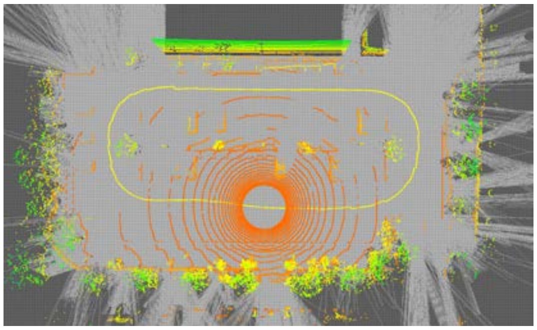

Fig. 3. One round to build a map, localized trajectory and LIDAR is also shown

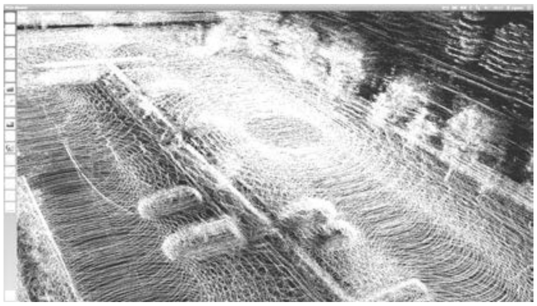

Fig. 4. Enlarged partial 3D point cloud map

Such a human passage has a potential danger, so such changing observation place is also statistically updated. Path planner can utilize this obstacle map to find out collision free path to the goal.

IV. Tracking and Classifying of Multiple Moving Objects

Tracking and classifying multiple moving objects from LIDAR input was conducted with frameworks of SJPDAFs (Sample-based Joint Probabilistic Data Association Filters), the method is robust against occlusions or false segmentation of LIDAR input. The method divides tracking targets and corresponding LIDAR segments into clusters, and is capable of classifying each cluster as a car or a group of pedestrians.

A. Cluster Based SJPDAFs in Outdoor Environments

Suppose that T moving objects are being tracked and m_k features are measured at time k. $\boldsymbol{X}^k = \{\boldsymbol{x}_1^k \cdots \boldsymbol{x}_T^k\}$ and $\boldsymbol{Z}(k) = \{\boldsymbol{z}_1(k) \cdots \boldsymbol{z}_{m_k}(k)\}$ denote the state vectors of moving objects and the measurements, respectively, at time k. In addition, $\boldsymbol{Z}^k$ denotes the sequence of all measurements up to time k.

The likelihood of $\boldsymbol{x}_i^k$ is updated by the following expressions using a Bayes Filter Algorithm [5].

$$p(\boldsymbol{x}_i^k|\boldsymbol{Z}^{k-1}) = \int p(\boldsymbol{x}_i^k|\boldsymbol{x}_i^{k-1}) p(\boldsymbol{x}_t^{k-1}|\boldsymbol{Z}^{k-1}) d\boldsymbol{x}_i^{k-1} \quad (1)$$

$$p(\boldsymbol{x}_i^k|\boldsymbol{Z}^k) = \alpha p(\boldsymbol{Z}(k)|\boldsymbol{x}_i^k) p(\boldsymbol{x}_i^k|\boldsymbol{Z}^{k-1}) \quad (2)$$

Here, α represents a normalizer. (S)JPDAFs introduce "Reliability" to $\boldsymbol{z}_j$ from $\boldsymbol{x}_i$ according to

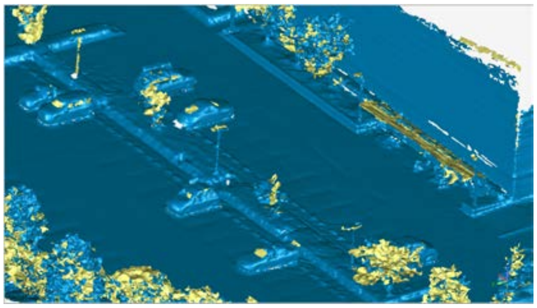

Fig. 5. Enlarged partial 3D polygon map constructed from point cloud map

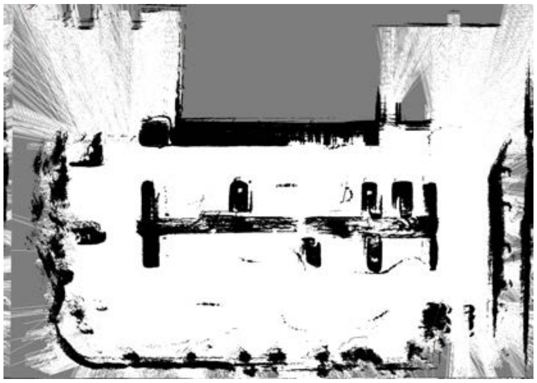

Fig. 6. Detected obstacles shown in black pixel

$$\beta_{ji} = \sum_{\theta \in \Theta_{ji}} P(\theta | \boldsymbol{Z}^k) \tag{3}$$

Eq.(2) can be transformed using β_{ji} as following.

$$p(\boldsymbol{x}_i^k | \boldsymbol{Z}^k) = \ \alpha \sum_{j=0}^{m_k} \beta_{ji} p(\boldsymbol{z}_j(k) | \boldsymbol{x}_i^k) p(\boldsymbol{x}_i^k | \boldsymbol{Z}^{k-1}) \tag{4}$$

θ and $P(\theta|\boldsymbol{Z}^k)$ denote a hypothesis and its likelihood respectively. θ_{ji} denotes the set of all hypotheses for which $\boldsymbol{z}_j$ corresponds to $\boldsymbol{x}_i$.

Normal SJPDAFs estimate the number of people within the entire sensor area by assuming a Poisson process. However, in outdoor environments, the density of moving objects varies considerably, so that a Poisson process is no longer suitable for representing the change in the number of moving objects. To overcome this, the proposed method employs cluster-based SJPDAFs. A “cluster” in this paper means a set of particle filter components of SJPDAFs and corresponding LRF segments. Estimation of the number of pedestrians and classification are performed for each cluster.

The proposed tracking method is performed as follows:

1) Moving object candidates are extracted from the latest LRF scan, and grouping of candidates is performed.
2) Particles in existing SJPDAFs clusters are updated according to $p(\boldsymbol{x}_{i,n}^k | \boldsymbol{x}_{i,n}^{k-1})$, and corresponding groups of candidates are enumerated.

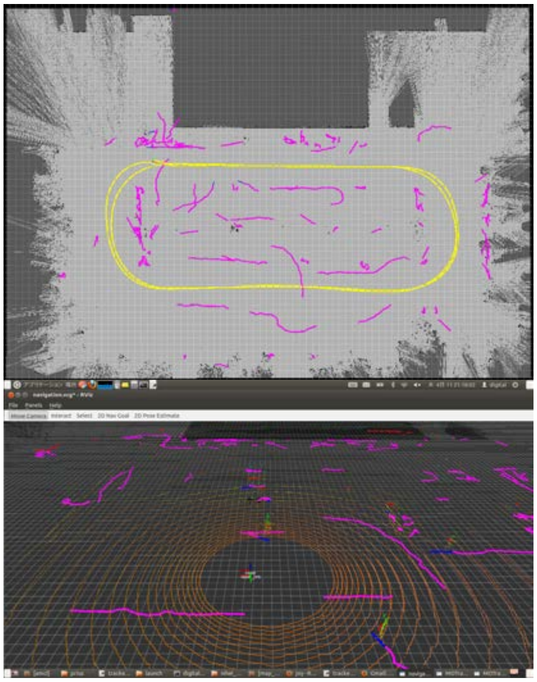

Fig. 7. Walking Human Trajectories (top) and one shot LIDAR scan (bottom)

3) Merging and splitting of SJPDAFs clusters are performed.
4) Each SJPDAFs cluster is updated independently.
5) Classification and estimation of the number of pedestrians are performed for each cluster.
6) New SJPDAFs clusters are initialized for groups of candidates that are not associated with existing SJPDAFs clusters.

As for 5) Classification and estimation of the number of pedestrians, linear SVM(Support Vector Machine) is adopted. As the shapes of LRF scan segments are not stable, the method adopts a time-series estimation.

B. *Experiment*

Experiment were conducted when many person is walking around. Result is shown in Fig.7. In this case, vehicle did two round. Vehicle trajectory (yellow) and all observed walking human trajectories (magenta) are shown in Fig.7(top). Fig.7(bottom) shows one shot LIDAR scan shown in graduation color for elevation as like previous figures. There are false positives occured around trees. Since we adopted cylindorical model as for human shape, there are many places that the trees can fit in that model, it is difficult to avoid this mismatch.

V. Control

To perform autonomous navigation the control system employs a path following method known as the pure pursuit algorithm [6]. The path is defined by a series of way points, and line segments connecting the way points form the reference path. The pure pursuit algorithm attempts to follow the path by computing a circular arc from the estimated robot position to a point some distance along the reference path. This ”look ahead” distance varies depending on current velocity and

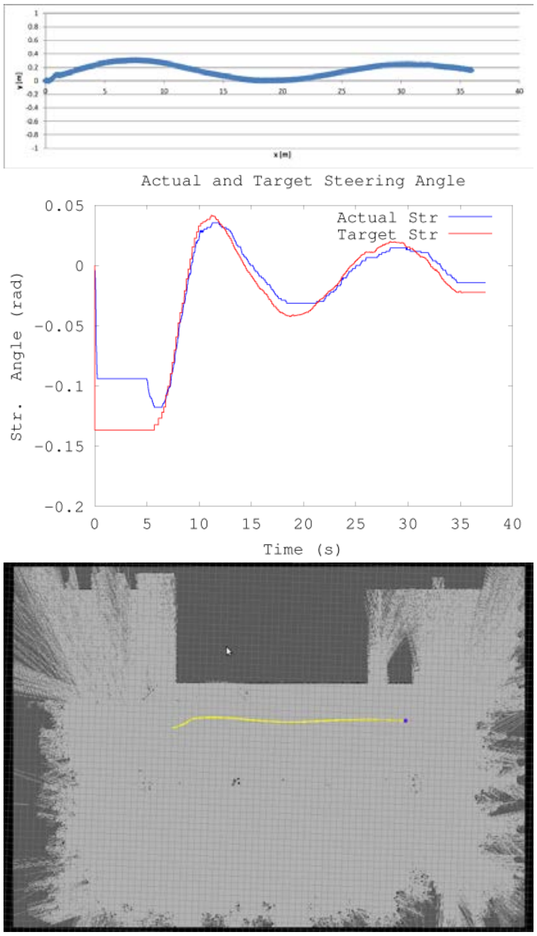

Fig. 8. Results on autonomous run in straight

weather a way-point or goal location is near. An example of a mobile robot being controlled by pure pursuit path following is given in [7]

Fig.8 shows experimenal result of running straight line. Fig.8(top) shows vehicle trajectory with lateral axis shows a line from start to goal position, and vertical axis shows perpendicular to that line. Therefore, vertical axis shows how vehicle moves sideway. As seen from the figure, with about 35 [m] run, lateral movement was about 0.3[m] in maximum. Localization result is shown in Fig.8(bottom). Yellow line shows vehicle trajecotry. Vehicle moves from left to right in the figure. Fig.8(center) shows commanded and actual steering angle. Since vehicle went off from the given line, feedback control were made by using pure pursuit model at maximum 0.15[rad]. Control cycle of this experiment was about 130[ms].

VI. Concluding Remarks

In this paper, we studied autonomous functions at car park in urban environment where GPS signal is not reachable. In such a place, map building and localization functions are required to plan a path and to control a vehicle. Moreover static obstacle finding and walking human detection are also required. Vertical wall segment based method is proposed to build a map and to localize where the vehicle is. SJPDAFs combined with SVM classifier is proposed to find out and track walking human being.

In order to achive autonomy, we developed an autonomous vehicle navigation system using LIDAR that has following functions: 1) map building, 2) localization, 3) static obstacle detection, 4) human tracking, 5) path planning, 6) vehicle control of path following. All those functions are developed on ROS. Car park of 70x50[m] area is used for experiment and results are shown.

References

[1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, “Stanley: The robot that won the darpa grand challenge,” *Journal of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[2] M. Montemerlo, S. Thrun, H. Dahlkamp, D. Stavens, and S. Strohband, “Winning the darpa grand challenge with an ai robot,” in *Proceedings of the national conference on artificial intelligence*, vol. 21, no. 1. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006, p. 982.

[3] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. “ . Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson, “Autonomous driving in urban environments: Boss and the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 8, pp. 425–466, 2008.

[4] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun, “Junior: The stanford entry in the urban challenge,” *Journal of Field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[5] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press Cambridge, 2005, vol. 1.

[6] G. Heredia and A. Ollero, “Stability of autonomous vehicle path tracking with pure delays in the control loop,” in *Advanced Robotics, Vol 21, No. 1-2*, 2007.

[7] S. Thompson, S. Kagami, and M. Okajima, “Constrained 6dof localisation for autonomous vehicles,” in *Proc. of the Int. Conf. on Systems, Man and Cybernetics,*, 2010.