Two-Dimensional Laser Servoing for Precision Motion Control of an ODV Robotic License Plate Recognition System

Zhen Song[a], Kevin Moore[a], YangQuan Chen[a], and Vikas Bahl[a]

[a]CSOIS, Utah State University, Logan, UT, US

ABSTRACT

As an outgrowth of series of projects focused on mobility of unmanned ground vehicles (UGV), an omni-directional (ODV), multi-robot, autonomous mobile parking security system has been developed. The system has two types of robots: the low-profile Omni-Directional Inspection System (ODIS), which can be used for under-vehicle inspections, and the mid-sized T4 robot, which serves as a "marsupial mothership" for the ODIS vehicles and performs coarse resolution inspection. A key task for the T4 robot is license plate recognition (LPR). For a successful LPR task without compromising the recognition rate, the robot must be able to identify the bumper locations of vehicles in the parking area and then precisely position the LPR camera relative to the bumper. This paper describes a 2D-laser scanner based approach to bumper identification and laser servoing for the T4 robot. The system uses a gimbal-mounted scanning laser. As the T4 robot travels down a row of parking stalls, data is collected from the laser every 100ms. For each parking stall in the range of the laser during the scan, the data is matched to a "bumper box" corresponding to where a car bumper is expected, resulting in a point cloud of data corresponding to a vehicle bumper for each stall. Next, recursive line-fitting algorithms are used to determine a line for the data in each stall's "bumper box." The fitting technique uses Hough based transforms, which are robust against segmentation problems and fast enough for real-time line fitting. Once a bumper line is fitted with an acceptable confidence, the bumper location is passed to the T4 motion controller, which moves to position the LPR camera properly relative to the bumper. The paper includes examples and results that show the effectiveness of the technique, including its ability to work in real-time.

Key Words: UGV; License Plate Recognition; Hough Transform; 2D-laser Scanner; Laser Servoing; Omni-directional vehicle (ODV).

1. INTRODUCTION

This paper will present a 2D-laser servoing system for precision motion control of an autonomous multi-robot system, the T4/ODIS system,[1] developed by the Center of Self-Organizing and Intelligent Systems (CSOIS), Utah State University. Both T4 and ODIS robots are ODVs and UGVs. ODIS is a low-profile robot whose small size makes it easy to go under the chassis of other vehicles for inspection. T4 serves as a "marsupial mothership" for ODIS and performs a coarse resolution inspection. A working scenario of this multi-robot system is as follows: T4 patrols in a parking lot carrying ODIS in its under-chassis elevator. T4 triggers its LPR system in front of each user-desired parking stall and gets the license plate (LP) number of the corresponding vehicle (if present) parked in it. The obtained LP number is then transmitted wirelessly to a remote workstation, which checks in a database for the following:

1. Is the vehicle with the given LP number permitted in this parking area?
2. Has the vehicle with the given LP number been reported stolen?

On the non-compliance of any of the above mentioned conditions, an "inspect vehicle" mission is issued to the system, upon receiving which T4 will stop at a proper position from the target vehicle and release ODIS to perform an under-chassis inspection. During the inspection, ODIS will transmit video images to the workstation

Zhen Song, Kevin Moore, YangQuan Chen, and Vikas Bahl are with the Center for Self-Organizing and Intelligent Systems (CSOIS), Utah State University, 4160 Old Main Hill, Logan, UT 84322-4160. Phone/Fax: (435)797-2924/2003. Email: `moorek@ece.usu.edu` `http://www.csois.usu.edu`

for the operator viewing. This facilitates the operator to inspect a vehicle at a safe place, away from possible danger.

Among all the tasks mentioned above, a key task for T4 is to identify LP plates using its LPR system[2] with an acceptable accuracy rate. Most of the commercially available and operating LPR systems are required to be mounted on a static platform with their operation triggered by an external sensor. Moreover, to ensure a high accuracy rate, these systems have to be placed at a prescribed distance and orientation with respect to the target license plate. The aforementioned conditions cannot be directly realized on the T4/ODIS robotic platform because it is a mobile system and does not have a simple sensor system that can trigger the LPR when the distance and orientation requirements are met. The only reasonable assumption for the T4/ODIS system about the environment is that the vehicles are parked inside the stalls. As we will explain in Sec. 2.4, if T4 simply stops at a certain distance and orientation relative to the center of a stall, the requirements for LPR system will not be guaranteed, i.e., sometimes it may just "see" half of the plate, thereby reducing the accuracy rate. The motivation of this paper is to design a laser servoing system that can drive T4 to a proper position and trigger its LPR system at the proper time for each stall.

Data servoing refers to techniques to design control laws that track objects, or control positions of robots by processing certain kinds of data as feedback. If the data is from the laser, it is called laser servoing. In this paper, a SICK 2D-laser scanner[3] and a self-made gimbal system were used to provide three-dimensional laser data to T4's position control system. Figure 1 is the laser servoing system block diagram. Note that it is only part of the T4/ODIS system diagram, and can not achieve all the missions described in the above paragraphs. According to this figure, the surveillance procedure is simplified as follows: An operator interfaces with the system via a communicating agent called the "graphical user interface" (GUI), that allows him or her to choose the desired stalls on which the "surveillance" task has to be performed. These tasks are sent in the form of a "mission" to the decision maker, the "supervisory task controller" (STC), on T4. The STC is responsible for the successful realization of a mission and it does so by planning a proper path for it. A mission is recursively decomposed into "command-units *" that drive the robot to the desired positions in front of the chosen stalls, trigger its LPR system, and return the LP numbers back to the workstation. From the control system point of view, the relationship is explained in Fig. 1. The two blocks with dashed lines represent input and output. The input is the "target stall(s)" selected by the user, and the output is the LP number(s) of the vehicle(s) in the target stall(s). The inner loop includes three blocks: "motion controller and planner," "robot dynamics," and "odometry system." Details for these can be found in publications,[1, 4–8] thus this paper will not focus on them. The block "observable environment" in the outer loop is the observable part in the real parking lot environment, which changes with respect to the motion of the robot. The "laser and gimbal system" is the sensor system to provide the feedback of the outer loop. The whole system is titled "laser servoing" because the feedback of the outer loop is laser data set. The block "map" is a database that stores the positions of the stalls, landmarks, the borders of the parking lot, and the target stalls. The "bumper fitter" block is the focus of this paper. It can process the laser data associated with the parking lot map, fit bumpers, and compute proper positions for LPR triggering, which are the reference inputs to the robot motion controller and planner, i.e, the bumper fitter deviates the robot from the planned path and directs it to positions appropriate for LPR inspection. Ideally, the laser servoing system should fit the license plate, instead of the bumper, but field testing demonstrated that the plates are too flat to detect with laser range data. So, bumper fitting strategies are proposed in this paper. There is no feedback from the "LPR" block, which is based on a commercial product called `See/Lane` from Hi-Tech Solutions Inc.[2] Since the characteristics of the LPR system have profound effects on the laser servoing strategies, we will discuss more on these in Sec. 2.4.

Figure 2 is a picture of T4/ODIS system. T4 is the bigger robot on the left. On the right, ODIS is docking into the elevator of T4. Because T4 is not finished at this time, we used the T2E robot to simulate T4. Their software architectures are exactly the same, but the hardware is different. For the rest of this paper, we will not distinguish between T4 and T2E, because the software and sensor system on T2E will be migrated to T4 without modification. T2E with its sensor-pack is shown in Fig. 3. On the left, the 2D-laser and gimbal system is mounted in the front of T2E. In the middle of this figure, there is a camera header, which contains a infrared (IR) camera, an optical filter, and an illumination system, mounted on the left of T2E. T2E also has thirty two sonar sensors, but we will not discuss them since they are not directly related to the laser servoing task.

[*] A command-unit is a set of individual commands that define various robot specific actions to be executed in parallel.

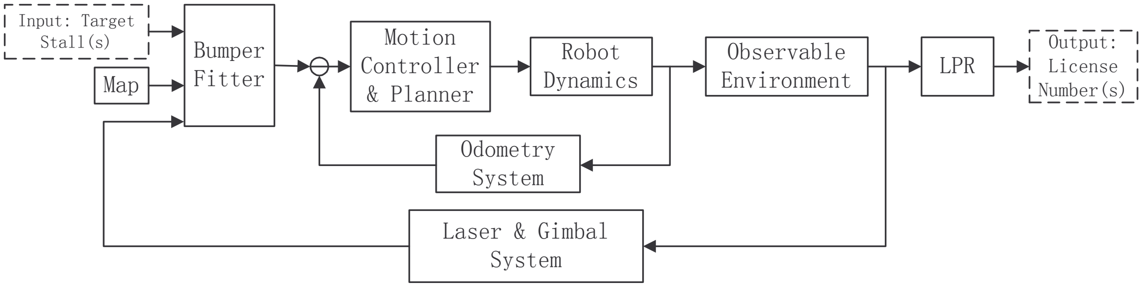

Figure 1. Laser servoing system diagram.

Sec. 2 will describe the platforms of the laser servoing system: the T4/ODIS robot system. The descriptions will cover the following topics: basic configurations of T4, specifications of the laser, gimbal, and LPR system, analysis about the height of laser, `T4LPR` application, the distribution for the accuracy rate of LPR, and the system architecture. Sec. 3 will focus on the bumper fitting algorithms and their implementations. Based on the analysis of the characteristics of the laser data, Hough transform based algorithms, i.e., standard Hough transform (SHT) and sparse Hough transform (SPHT), were implemented on T4. Sec. 4 will present the experimental results. Sec. 5 will conclude the paper.

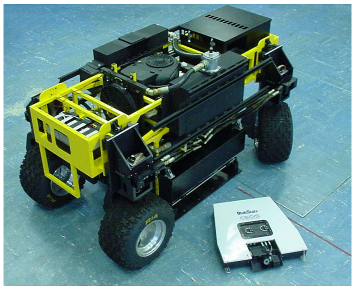

Figure 2. ODIS is docking to T4

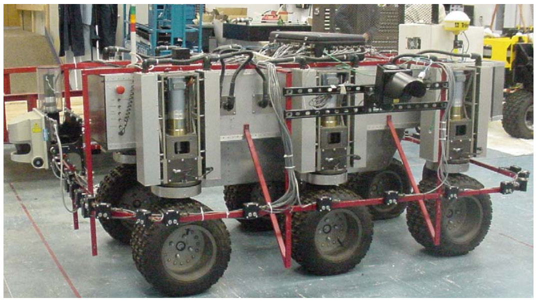

Figure 3. T2E robot, the platform of testing

2. TESTING PLATFORM

2.1. T4 System Overview

The modules of T4 and the workstation that relate to the laser servoing task are shown in Fig. 4. In this figure, blocks with shadows are software modules, and those without shadows are physically existing hardware. All the blocks in the left dash line box are installed on T4, and those on right are on the workstation side. We will not distinguish between hardware connections and software coupling. Any line in the figure that has a hardware block on one of its two sides is a physical wire, and a line connecting two software blocks is a software coupling. A software coupling might be a “function call,” or a communication channel among different threads.

T4 has three embedded computer systems called the “master node,” “sensor node,” and “LPR node,” respectively. Most of the functions are implemented on the “master node,” including the STC, a map database, and different threads dedicated to individual tasks. The “sensor node” mainly pre-processes raw sensor data in real-time, then sends the result to the workstation via the master node. Without such pre-processing, the master node would not have enough computational power to process in real-time. The “LPR node” is dedicated to the `T4LPR` application, which can grab images from the LPR header, identify LP numbers, and send results to the workstation via the master node. These three nodes are connected by TCP/IP and exist on the same subnet, which in turn is connected to the workstation. The odometry system that is connected to the master node can estimate the robot’s position and orientation in real-time. As shown in Fig. 1, odometry information acts as feedback for the inner loop in the system.

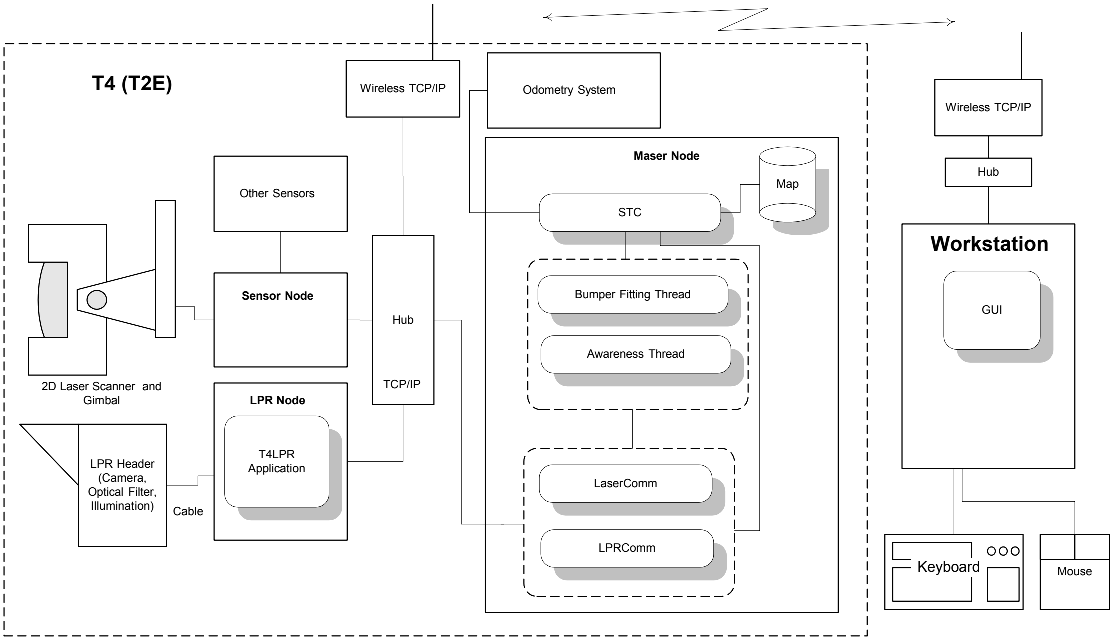

Figure 4. System architecture of bumper fitting

In the master node, the STC is linked with a dashed line block that contains the "bumper fitting thread," "awareness thread," and other threads. The bumper fitting thread, together with the "LaserComm" thread and the pre-processing module on the sensor node, constitute the "bumper fitter" in Fig. 1. Although the bumper fitter is not physically connected to the GUI or map, they are connected in logic, because many other modules, such as hubs, perform as relays of the communication flow.

The awareness thread is a danger avoidance thread that can use laser and sonar data[†] to build a local map of the surrounding environment in real-time and halt the vehicle in emergency.[9] Because the danger avoidance has a higher priority than the laser servoing, we must consider the requirements together in order to reduce the chance of conflicts when the awareness thread has to preempt the bumper fitting thread and halt the laser servoing procedure. The result of this consideration is that the 2D-laser must scan horizontally at a proper height. These details will be presented in Sec. 2.3.

On the master node, there are also other communication threads, which are not shown in Fig. 4. They share the same IP address, but have different TCP sockets. The reason for separating communication threads from task threads, such as the bumper fitting thread, is that the operating systems (Linux and Windows 2000) of these three nodes are not real-time operating systems, but the application requires real-time capability, i.e., each node has to process received data packets in time, otherwise the robot might be in danger. In the system, these light duty communication threads are dedicated for packet buffering, so that the packet loss is prevented as much as possible.

The `T4LPR` application executing on the LPR node is based on the `See/Car` DLL[‡], which returns an LP number after receiving an image of a license plate. On the LPR node, there is a frame grabber card that can record signals from the LPR header into a bitmap format. The `T4LPR` application transfers these images to the `See/Car` DLL for identification, and sends the result to the "LPRComm" thread with a TCP/IP connection. In order to enhance the accuracy rate, `T4LPR` can take up to 10 images of one license and consider the results

[†] The communication channel between the awareness thread and sonars is not shown in Fig. 4, because it is not directly related to the laser servoing system.

[‡] "DLL" stands for the dynamic link library on Microsoft Windows. `See/Car` DLL is a module of the `See/Lane` system.

comprehensively to get the best fit LP number. In the testing for this paper, T4LPR took 5 images for each license.

On the workstation side, the most important module of the system is the GUI. The GUI is an integrative information environment that can display the information, such as positions of T4 and ODIS, together with the LP numbers found by T4, on the screen. At the same time, it can also receive operations from the keyboard and the mouse, compile these operations into STC compatible missions and send them to T4 and ODIS via the wireless connection. The details are out side the scope of this paper.

2.2. LPR Mission

A typical LPR mission is illustrated in Fig. 5. With no loss of generality, let us assume that the user wishes to perform this LPR mission on a row of stalls (consisting of four stalls in our case). Also, let us assume that the mission starts at point "S." The main motive of this mission is to identify the LP numbers of all the vehicles that are currently parked in the chosen row. Upon finding an LP number, the same has to be relayed to the GUI for the operator viewing. It is also desired that if a stall is devoid of any vehicle, the T4 does not trigger the LPR system and proceeds to the next stall. On receiving the mission, the STC plans a path for the same by adding intermediate nodes at points A, B, C, and D respectively. These nodes correspond to the "entry-points[4]" of the individual LPR tasks in front of each stall. Note that at the beginning of a mission, the STC does not distinguish between the stalls that have a vehicle parked in it with those without a vehicle and plans for the complete row. Once the mission is started, the STC either retains or deletes the task to be performed on a particular node based on the updates provided by the bumper fitter. The bumper fitter is designed to look-ahead (in space) for three stalls (this number was chosen for computational simplicity) and for each of these stalls it computes a corresponding "bumper box [§]" wherein it fits the bumper lines. Every control cycle, the laser data that falls into these boxes is used to "progressively fit" a line, thereby approximating the fitted line to the real bumper and simultaneously affecting the confidence of the fit. Upon arriving at the entry-point of each LPR task, the STC queries the bumper fitter asking for a "bumper line" and the confidence of its fit. If the bumper fitter replies in an affirmative (at nodes A, B, and D in the chosen example), the STC positions the robot according to the obtained bumper line and issues an LPR command-unit to it. On the other hand, if the reply from the bumper fitter is negative (implying that the obtained data has a confidence less than the minimum threshold), as is the case at node C, the STC does not issue an LPR command-unit and instead instructs the robot to go to the next node.

2.3. 2D-laser Scanner and Gimbal System

Specifications

The specifications of the 2D-laser scanner are listed in Table 1. Table 2 shows the specifications for the gimbal system.

Table 1. Specifications of SICK 2D-laser scanner.

Type	Sick LMS 220
Dimensions	(352 × 228.5 × 266mm), 9kg
Distance Error	< 1cm
Angular Resolution	0.25° / 0.5° / 1° (selectable)
Scanning Angle	180°
Detection Range	Maximum 80m, and 10m for 10% reflectivity objects
Response Time	53ms/ 26ms / 13ms (resolution dependent)

Table 2. Specifications of gimbal system.

Type	Self made
Tilt Angle	-90° to 45° (upward is positive)
Resolution	1°
Speed	50°/sec.

[§] A bumper box is rectangular polygon formed with four points. The width of this polygon is equal to the stall-width and the length is a fraction of the stall-length.

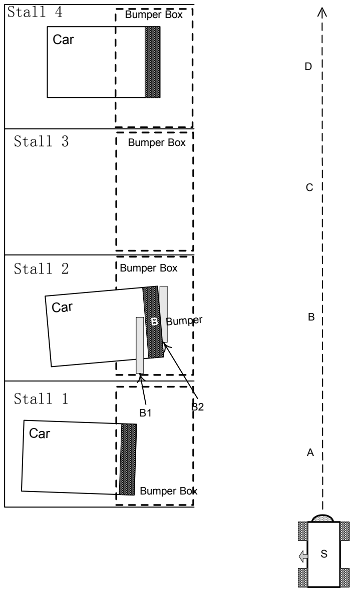

Figure 5. LP checking when moving straight.

2D-laser Height

Figure 6 plots the heights of some vehicles. Each vertical line in this figure represents one vehicle. The short bars at the top of these lines is the bottom of the window. Since the laser beam can go through the glasses of the windows, and "see" inside of the vehicles, the data points higher than this line can not indicate the position of the license plate of a vehicle. Between the first bars and the second bars, which represent the top of the bumpers, are the engine covers. Although the laser data points that fall in these regions will be further than the position of the bumpers, the error will be less than those points above the first bar, since the length of the engine covers are limited. From the second bars to the third bars are the bumpers. Data points in this region can indicate the exact distances and orientations of bumpers.

For bumper fitting algorithms, the desired height of the 2D-laser scanner is between the second bar and the third bar, and the desired tilt angle is 0°. This configuration can greatly simply the bumper fitting algorithm and more easily guarantee the real-time property.

Consider the case when the laser scanner is mounted higher than the bottom of windows. Then the tilt angle has to be adjusted according to the distance between the T4 robot and the target vehicle in order to shine the laser beam roughly to the bumper height and measure the orientation. One disadvantage of this configuration is the limited speed. As shown in Tables 1 and 2, the 2D-laser scanner is very fast compared with the gimbal system. So it is better to continuously scan the laser beam than to adjust the tilt angle of the gimbal system. Another disadvantage of the configuration is the complexity. For this approach, the laser will "see" the ground, bumpers, engine covers, and windows and might "see" some thing inside the windows, etc. Then the bumper fitting algorithm has to segment data points, and fit the bumper only. On the other hand, if the tilt angle is 0, the data points are actually in two-dimension, since the differences between their heights are negligible.

For the case when the 2D-laser is mounted lower than the bottom of the bumpers, the 2D-laser and gimbal system might be damaged by stones on the road. However since the tilt angle is positive, and the measured data is in three-dimension, the complexity due to segmentation is still not avoidable.

So, for the requirements of the bumper fitting algorithm, the desired height of the 2D-laser is about 28in. But before we choose this height, we need to consider constrains from the awareness thread[1, 9] and the localization thread,[1, 10] since these two threads also need laser data.

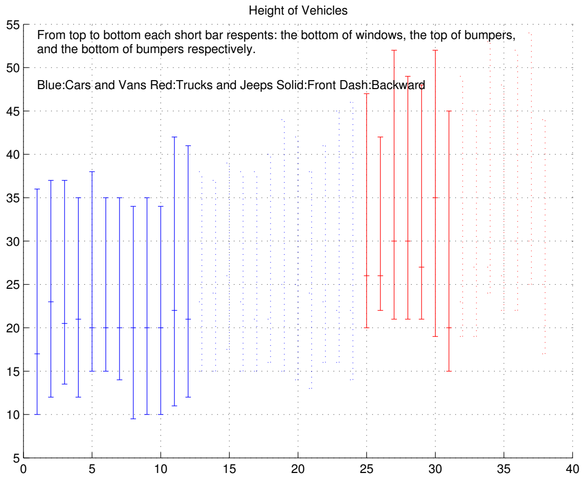

Figure 6. Vehicle heights in a parking lot.

The awareness thread can process laser and sonar data in real-time, build a HIMM map, estimate positions of potentially dangerous objects, and halt the T4 robot if necessary. In order to detect objects as far away as possible, the best laser tilt angle for the awareness thread is 0°, which means the laser should scan horizontally. Since security is more important than any mission, the awareness thread has the highest priority, which enables it to preempt the bumper fitter and the robot motion. If the 2D-laser is mounted at a height such that the horizontal scanning can support both the awareness thread and the bumper fitting, then resource conflicts are reduced, and system security is improved. The localization thread can locate T4 and calibrate its odometry system. There are two kinds of landmarks in the parking lot: curbs and lampposts, and two types of localization strategies: real-time localization, i.e., locate while moving, and static localization, i.e., stop and locate. For the static localization, the landmark can be a curb, thus the tilt angle of the gimbal system will change inside a range and get a 3D point cloud for the localization procedure.[10] It has no conflict with the bumper fitter, since the bumper fitting algorithms do not need to work when T4 stops. The real-time localization is still under development. Based on the considerations, we propose to keep the tilt angle at 0° in order to accommodate the laser servoing system.

According to the above considerations, the 2D-laser is mounted at 28in with the default tilt angle at 0°.

2.4. LPR System

LPR System Specifications

The LPR system, including software and hardware, on T4 is based on the `See/Lane` system of Hi-Tech Solutions Inc. On the software side, the `See/Lane` system includes a license plate identification DLL called the "`See/Car` DLL," and a GUI application call the "`See/Lane` application," which is not installed on T4 because it can not communicate with "LPRComm" on the master node. Instead, we programmed an application called `T4LPR` that calls `See/Car` DLL and communicates with the "LPRComm." The hardware of the `See/Lane` system includes a camera header, which integrates an IR camera, an optical filter, and an illumination system, a frame grabber card, cables, a license plug[¶], and a PCI I/O card that controls the level of illumination. The illumination system has three levels of brightness, and the `See/Car` DLL prefers to get images of different levels to make the identification result more reliable. Because the embedded computer on T4 has no PCI slot, the testing of this paper is based on a constant illumination level.

The Accuracy Rate of LPR

There are several factors that effect the accuracy rate of the LPR system: sunlight, distance, angle, and identification mode.

[¶] The license plug is a plug on the parallel port that enables the `See/Car` DLL.

If there is no protection against sunlight, the accuracy rate of the LPR system can be virtual zero. To reject the disturbance of the sunshine, we used 940nm narrow band IR optical filter, and set the illumination system to the same wave length. Although the sunshine has much more energy than the illumination system for all the spectrum, the bandwidth of the illumination is very narrow, thus in this specific bandwidth the illumination is much stronger than the sunshine. Then, the disturbance of the sunshine is not significant. Because most US license plates are reflective to IR, while the paint of the vehicles and backgrounds, etc., are non-reflective, the license plates are very bright compared with the dark backgrounds in the IR images. In Fig. 7, the image on top was taken with an 940nm narrow band optical IR filter, while the lower one was taken without any filter. The accuracy rates on the images with IR filter are much higher, since there are not many objects in this type of image to disturb the license identification procedure.

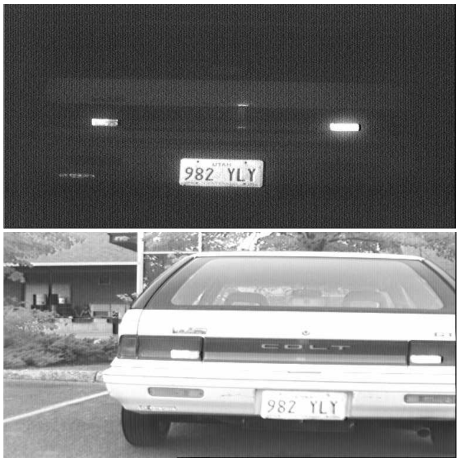

Figure 7. Images with and without IR optical filter

The LPR system can be operated in two modes, namely, the "single shot mode" and the "multiple shot mode." The former implies the T4LPR application returns one LP number per image. In the latter mode, the T4LPR may take up to 10 images, consider the identification result of each one, and return the best fit number in the end. After field testing, we chose the multiple mode with 5 images per identification.

See Fig. 8. D is the center of a bumper, which is normally also the center of the license plate. The segment CD is perpendicular to the bumper. Then, CD is the inspection distance and $\angle$CDA is the inspection angle. When the distance, CA, increases, the accuracy rate of the LPR system decreases to zero. In Fig. 9, there are two plots. On the upper part of the left plot, there is a car in a stall. The contour box at the lower part of the same plot is the distribution of the accuracy rate. The color bar on the right indicates the associated relationships between the accuracy rate and the level of darkness. The right plot is an enlarged contour box. The triangles are the positions where we have tested the accuracy rate. In these two plots, the center of the license plate is the origin of the coordinate. According to these testing results, the positioning angular resolution for LPR should be ±9°, and the positioning distance resolution should be less than ±50cm. It is observed in field testing that the accuracy rate is not very sensitive to the inspection distance compared with the inspection angle.

The inspection distance and angle are important to mobile applications, such as robots. By default, See/Lane system used a 16mm focus lens whose suggested inspection distance is 8m. However, due to the limitation of the lane width of the parking lot, the common inspection distance of T4 is about 3m. Thus we used a 9mm lens.

3. LASER SERVOING TECHNIQUES

3.1. Mission Requirements Analysis

The basic requirements on the bumper fitting algorithm are: fast, progressive, and robust. The algorithm must be able to fit bumper in real-time on an embedded computer, whose computation capability is limited.

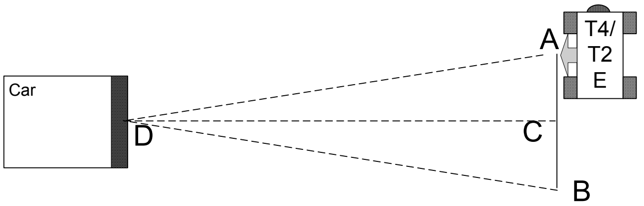

Figure 8. Inspection distance and inspection angle of LPR system.

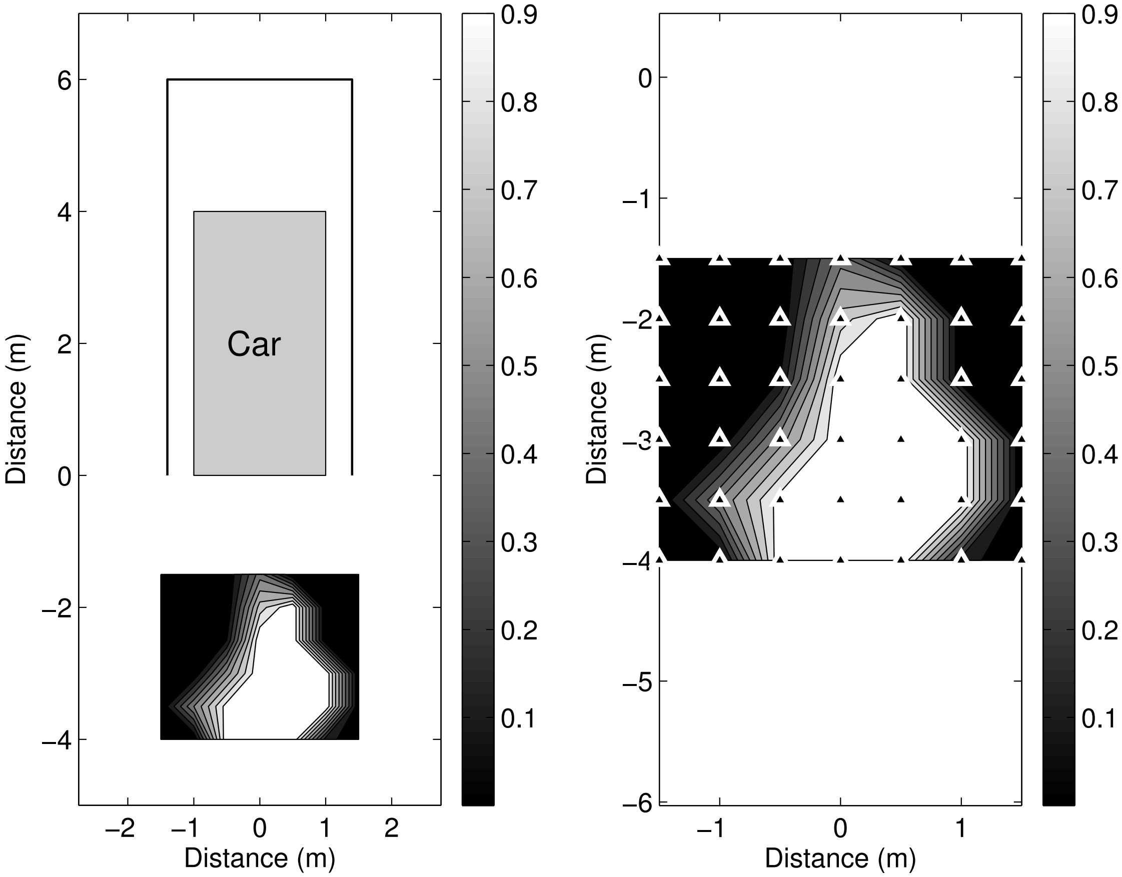

Figure 9. The accuracy rate distribution of LPR system.

Progressive is actually derived from the requirement of "fast." Because each 100ms the 2D-laser scanner can sweep once and get new data, it is inefficient if the algorithm has to fit again from the beginning every time new data comes in. So the feature of progressive is important. Robust in also an important issue. If the fitting result is wrong, the planner might try to direct the robot to "run over" other vehicles. Although there is an awareness thread that is supposed to prevent collisions, a "misleading" bumper fitter still might lead the robot system to unpredictable dangerous states. Before we chose the fitting algorithm, we used the laser scanner to collect 3D data with the tilt angle swept between ±30°. After cropping data points further than 9m, the result was plotted in Fig. 10. This figure demonstrates that the laser beam might go through the windows and "see" inside a vehicle (look at the van on right). Though the height of the 2D-laser scanner was carefully chosen such that the possibility of "seeing" though windows is very small, it is still a consideration in the algorithm selection due to its high demand on the safety. Another kind of disturbance that is not shown in this plot is pedestrians. They might walk in the bumper boxes and greatly interfere with the bumper fitter.

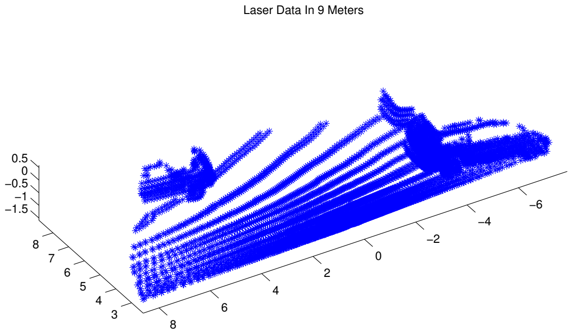

Figure 10. Collected laser data with tilt angle.

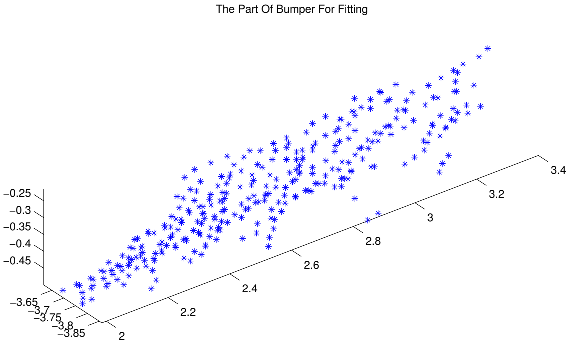

Figure 11. On Laser data set of a bumper.

3.2. Bumper Fitting Algorithms

There are basically two fitting options: least mean square (LMS) type algorithms and Hough type transforms. In general, LMS is fast, but not as robust as the standard Hough transform (SHT). If LMS was implemented on T4, a segmentation algorithm that can distinguish between bumpers and pedestrians, or "in window objects," is necessary for laser data pre-processing. To eliminate the need for such complexity, Hough type transforms were implemented.

The Hough transform (HT) is a popular method for the extraction of geometric primitives.[11, 12] Initially, it was only an approach for line detection. Later, many variants were developed to detect circles,[12] ellipses,[13] or more complex binary patterns.[14] A survey on the HT can be found in.[11] Generally speaking, the HT is robust to sensor noise at the expense of slow computation. The computational cost of the traditional HT is $\mathcal{O}(n^3)$ where n is the number of data points. Though some efforts were made to speed up the HT algorithm,[15] those fast strategies were developed generally for image data processing applications only. We proposed the use of the sparse Hough Transform (SPHT) algorithm for laser data fitting in[10] and compared it with the standard Hough transform (SHT) and Log Hough transform (LHT). The interested reader may refer[10] for details. In the remainder of this paper we briefly compare these HT algorithms and present experimental results using the SPHT.

4. EXPERIMENTAL RESULTS

Figures 12 and 13 are examples of SPHT bumper fitting. Figure 12 is one sweep of laser data on a vehicle bumper. After line fitting by the SPHT, the result is plotted in Fig. 13. The two circles represent the two laser data points that are estimated to be the two ends of the bumper edge. When processing this example, the distance resolution was 6cm and the angular resolution was 9°. Except for the array that stores the laser points, the SPHT needs 3 float variables, while the SHT needs 46×46 integer variables. When the C++ implementations of the two executing on a 700MHz, 256MB desktop, the SPHT took 0.1sec., and the SHT took less than 0.02sec. Though the performances of both the SHT and SPHT algorithms are acceptable in the experiment, we chose SPHT finally. Because it is fast enough, and uses less memory. We did not implement LHT on T2E, because its variant resolution feature is complex for implementation, while its major advantage, fast speed, was not very urgent for our application. The SPHT and the SHT are fast enough. For the LHT approach, when the robot was far from a specific bumper box, it needed fewer cells in its accumulation space for bumper fitting. But while it was approaching the bumper box, more cells were required, which takes time for implementation.

Figure 14 is a screen shot of part of the GUI on the workstation. The user had clicked `Stall 6` to `Stall 10` and sent T2E on the LPR checking mission. This screen shot was taken when T2E was almost finished with the mission. The line in the middle of the figure is the track of T2E during the mission. The white edges in `Stall 6`, `Stall 7`, and `Stall 10` are the fitted bumper lines. T2E had stopped before each of these stalls and triggered the `T4LPR` application.

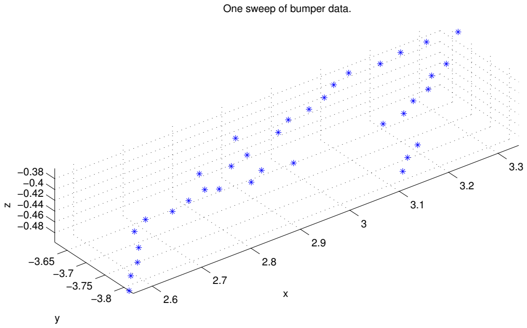

Figure 12. One laser sweep on a bumper.

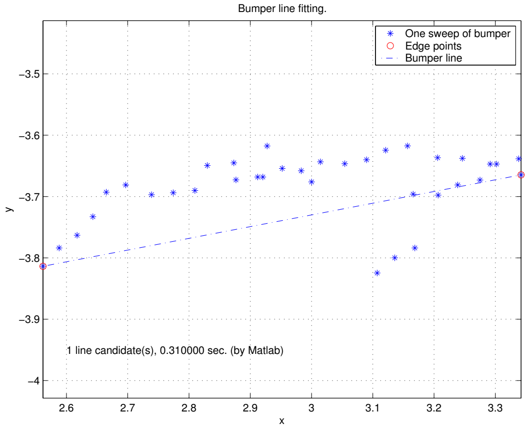

Figure 13. One laser sweep on a bumper and its fitted bumper line.

Figure 15 overlays a GUI screen shot (upper right) on a picture (background) in which T2E was moving in a parking lot. The returned LP numbers were stored on the hard disk of the workstation and were also displayed on a 3D perspective view in the GUI.

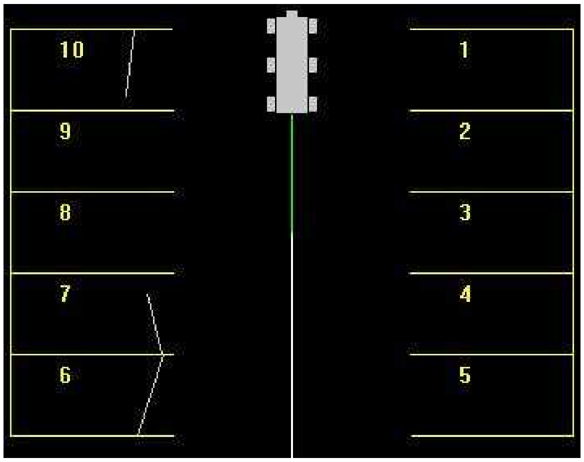

Figure 14. Bumper line on GUI.

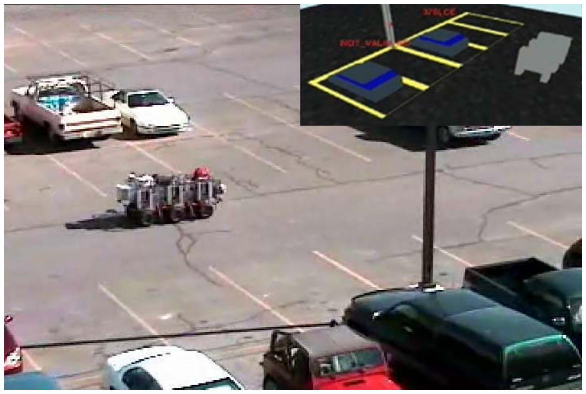

Figure 15. T2E executing bumper fit and check LPR mission.

5. CONCLUDING REMARKS

In this paper we have presented a laser servoing system for precision motion control. Performed as an outer loop, this laser servoing system enhances the accuracy rate of an ODV robot LPR system by fitting bumpers. Since the platform T4 (T2E) robot is a complex autonomous system, there are many considerations for implementation details, such as resources conflict, balance among different requirements, and mission scheduling strategies. This paper carefully analyzed all requirements on the laser servoing system from those constrains, then proposed the Hough transform-based algorithms as the feedback processor. The SHT and the SPHT were implemented as the kernel algorithms for the bumper fitter. For this application, the SHT is faster but takes more memory. On the contrary, SPHT takes less memory but is slower.

The concept of laser servoing might not be limited to LPR accuracy rate enhancement. Instead, similar algorithms might help the robot to avoid danger, follow objects, or carry out other autonomous missions. Our future efforts lie in expanding and extending the proposed laser servoing in other application scenarios.

REFERENCES

1. V. Bahl and K. L. Moore, "Multi-robot autonomous parking security system," in *Submitted to Proc. of the IEEE Conf. on Decision and Control*, 2003.
2. "SeeLane technical information." Hi-Tech Solutions Inc. `http://www.htsol.com/Download.html`, Oct. 2001.
3. "Measurably more economical, LMS200/LMS220 laser measurement systems." SICK Inc. `http://www.sick.de`, 2003.
4. H. Shah, V. Bahl, J. Martin, N. S. Flann, and K. Moore, "Intelligent behavior generator for autonomous mobile robots using planning-based AI decision making and supervisory control logic," in *Conference on Unmanned Robotic Vehicles*, (Orlando, FL), Apr. 2002.
5. W. Smuda, P. Muench, G. Gerhart, and K. L. Moore, "Autonomy and manual operation in a small robotic system for under-vehicle inspections at security checkpoints," in *Conference on Unmanned Robotic Vehicles*, (Orlando, FL), Apr. 2002.
6. K. L. Moore and N. S. Flann, "A six-wheeled omnidirectional autonomous mobile robot," *IEEE Control Systems Magazine* **20**, pp. 53–66, Dec. 2000.
7. K. Moore, N. Flann, S. Rich, M. Frandsen, Y. Chung, J. Martin, M. Davidson, R. Maxfield, and C. Wood, "Implementation of an omni-directionsl robotic inspection system (ODIS)," in *Proc. of SPIE Conf. on Robotic and Semi-Robotic Ground Vehicle Tech.*, SPIE, (Orlando, FL.), May 2001.
8. M. Davidson and V. Bahl, "The scalar ϵ-controller: A spatial path tracking approach for ODV, Ackerman, and differentially-steered autonomous wheeled mobile robots," in *Proc. of IEEE Int. Conf. on Robotics and Automation*, IEEE, (Seoul, Korea), 2001.
9. L. Ma and K. L. Moore, "Sonar and laser-based HIMM map building for collision avoidance of mobile robots," in *Submitted to Proc. of the IEEE International Symposium on Intelligent Control*, 2003.
10. Z. Song, Y. Chen, L. Ma, and Y. C. Chung, "Some sensing and perception techniques for an omni-directional ground vehicles with a laser scanner," in *Proc. of the IEEE International Symposium on Intelligent Control*, IEEE, (Vancouver, British Columbia), Oct. 2002.
11. J. Illingworth and J. Kittler, "A survey of the Hough transform," *J. on Computer Vision, Graphics, and Image Processing* **44**, pp. 87–116, 1998.
12. T. J. Atherton and D. J. Kerbyson, "Size invariant circle detection," *J. Image and Vision Compution* **17**, pp. 795–803, 1999.
13. N. Guil and E. L. Zapata, "Lower order circle and ellipse Hough transform," *J. Pattern Recognition* **30**, pp. 1729–1744, Oct. 1997.
14. N. Guil and E. L. Zapata, "A new invariant scheme for the generalized Hough transform," in *IASTED Int. Conf. on Signal and Image Processing*, pp. 88–91, (Orlando, Fl), Nov. 1996.
15. J. Matas, C. Galambos, and J. Kittler, "Progressive probabilistic Hough transform," in *Proc. British Machine Vision Conference*, Sept. 1998.