

2D Laser Servoing for Precision Motion Control of an ODV Robotic License Plate Recognition System

(Draft 2: After Dr. Moore)

Zhen Song^a, Kevin Moore^a, Yangquan Chen^a, and Vikas Bahl^a

^aCSOIS, 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 $\{ODIS\} \xrightarrow{1} \{\textit{Omni-Directional Inspection System (ODIS)}\}$ 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 this task, 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 2-D laser-based approach to bumper identification and vehicle 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 100 ms. 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, a recursive line-fitting algorithm is used to determine a line for the data in each stall’s “bumper box.” The fitting technique uses $\{Hough\ transform\} \xrightarrow{2} \{\textit{Hough based transforms}\}$, which are robust against segmentation problems and $\{\textit{also applicable to which is also effective with multi-resolution data (the resolution of data varies from stall-to-stall as the laser looks ahead in space)}\} \xrightarrow{3} \{\textit{fast enough for real-time line fitting}\}$. Once a bumper line is fit with 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 results that show the effectiveness of the technique, including its ability to work in real time. The paper also includes discussion about how to increase the performance of the line fitting algorithms and how to reduce the computational cost.

Key Words: UGV; ODV; 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, T4/ODIS system,¹ developed by the Center of Self-Organizing and Intelligent Systems (CSOIS). Both T4 and ODIS are omni-directional vehicle (ODV) $\{robot\} \xrightarrow{4} \{\textit{robots}\}$, or unmanned ground vehicles (UGV). $\{ODIS^2\ stands\ for\ Omni-Directional\ Inspection\ System.\} \xrightarrow{5} \{\}$ Its small size and low profile $\{made\} \xrightarrow{6} \{\textit{make}\}$ it easy to go under chassis $\{\textit{for dangerous objects}\} \xrightarrow{7} \{\textit{of other vehicles for}\}$ inspection. T4 serves as a “marsupial mothership” for ODIS and performances coarse resolution inspection. The working scenario $\{\textit{will be the}\} \xrightarrow{8} \{\textit{is as}\}$ follows: T4 patrols in a parking lot while carrying ODIS by its $\{\textit{under chassis}\} \xrightarrow{9} \{\textit{under-chassis}\}$ elevator $\{\}$. $\{\textit{During the time, T4 triggers}\} \xrightarrow{10} \{\textit{and using}\}$ its $\{\textit{license plate recognition (LPR)}\} \xrightarrow{11} \{\textit{LPR}\}$ system to identify the license numbers of the vehicles in the parking lot. The recognized numbers are transferred to a remote workstation by a wireless TCP/IP connection. The workstation checks in a database if those licenses are permitted in this parking lot, or if they are reported as been stolen $\{\}$ $\xrightarrow{12} \{\textit{, etc.}\}$ The application executing on the workstation can issue “check chassis” $\{\textit{command}\} \xrightarrow{13} \{\textit{commands}\}$ to T4/ODIS $\{\textit{according to certain logic. At the same time, the}\} \xrightarrow{14} \{\textit{. The}\}$ user of the workstation can also manually issue $\{\textit{the same command}\} \xrightarrow{15}$

March 2003. For submission to SPIE 2003. 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>

Figure LaserServingBlock will be here

Figure 1. Laser serving system diagram.

{such commands} to T4/ODIS. After {received the}¹⁶ ⇒ {receiving a} “check chassis” mission, T4 will stop at a proper distance to the target vehicle, and release ODIS from its elevator. Then, ODIS will go under the chassis of the vehicle, take video images on the chassis and transfer the signal back to the workstation. Thus the user can inspect the chassis {in safe place where is far from the}¹⁷ ⇒ {in a safe place far from} possible dangerous objects, e.g., {bombs, under the chassis.}¹⁸ ⇒ {bombs.}

Among all the {tasks above}¹⁹ ⇒ {tasks mentioned above}, a key task for T4 is to identify license plates {by its (LPR)}²⁰ ⇒ {using its LPR} system³ with {}²¹ ⇒ {an} acceptable accuracy rate. However, many commercial LPR systems are required to be mounted on static position near the entrance of the parking lot or at the side of a road, and {}²² ⇒ {to} be triggered by external sensor. Otherwise, the inspection distance and orientation might not be proper, thus the accuracy rate cannot be guaranteed. These two requirements are not satisfied in the application of T4/ODIS system. Firstly, the LPR system has to be mounted on one side of T4 and {}²³ ⇒ {must} move together with the robot; secondly, there is no simple sensor system that can trigger the LPR when the distance and orientation requirements are satisfied. In parking lots, the only reasonable assumption is that all vehicles are parked inside stalls. As {what}²⁴ ⇒ {} we will explain in Sec. 2.4, if T4 simply {stop}²⁵ ⇒ {stops} at a certain distance and orientation {}²⁶ ⇒ {relative} to the centers of stalls, the requirements for LPR system will not be guaranteed, i.e., sometimes it may just “see” half of the plate, thus the accuracy rate drops. The motivation of this paper is to design a laser serving system that can drive T4 to {}²⁷ ⇒ {a} proper position, and trigger its LPR system at the proper time.

Data serving {are the}²⁸ ⇒ {refers to} techniques to design control laws that track objects, or control positions of robots by processing certain kind of data as feedbacks. If those data are laser data, it is called laser serving. In this paper, a SICK 2D laser scanner⁴ and a {self made}²⁹ ⇒ {self-made} gimbal system were used to provide three dimension laser data to {T4 robot}³⁰ ⇒ {T4’s} position control system. Figure 1 is the laser serving system block diagram. Note: 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: the user selects one or several stalls by clicking mouse on the graphic user interface (GUI) application at the workstation. Then, these click actions are compiled into commands and transferred to T4 and ODIS by wireless TCP/IP connections. After {received}³¹ ⇒ {receiving} these commands, T4 can plan a proper path, drive to proper position, trigger its LPR system, and return the license number back to the workstation. From the control system point of view, the relation {was}³² ⇒ {is} plotted in Fig. 1. The two blocks with dashed lines represent input and output. The input is the “target stall(s)” selected by user, and the output is the license number(s) of the vehicle(s) in the target stall(s). The inner loop has three blocks: “motion controller and planner,” “robot dynamics,” and “odometry system.” {The details of them can be found in publications of CSOIS}³³ ⇒ {Details for these can be found in publications},^{1, 2, 5-8} thus this paper will not focus on them. The block “observable environment” in the outer loop is the observable parts of the real parking lot environment, which {is}³⁴ ⇒ {are} changing with respect to the motion of the robot. The “laser and gimbal system” is the sensor system for the feedback of the outer loop. The whole system was titled “laser serving” 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, and {boarders}³⁵ ⇒ {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 parking lot map, fit bumpers, and compute proper positions for LPR triggering, which are the reference inputs to the robot motion controller and planer, i.e, the bumper fitter directs the robot to proper positions for LPR inspection. Ideally, the laser serving system should fit license plate, instead of bumper, but the field testing demonstrated that the plates are too flat to detect for laser range data. So bumper fitting strategies were 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{.}³⁶ ⇒ {}.³ Since the characteristics of this model {has}³⁷ ⇒ {have} profound effects on the laser serving strategies, we will discuss more on it 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 the time when we wrote this paper}³⁸ ⇒ {at this time}, we

used $\{\}$ $\xRightarrow{39}$ $\{the\}$ T2E robot to simulate T4. Their software architectures are exactly the same, but the hardware $\{are\}$ $\xRightarrow{40}$ $\{is\}$ different. For the rest of this paper, we will not distinguish T4 and T2E by default, because the software and sensor system on T2E will be immigrated to T4 without modification. Figure 3 is a picture of T2E. On the left, the 2D laser and gimbal system is mounted in the front of T2E. In middle of the 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 many sonar sensors, but we will not discuss them since they are not directly related to the laser serving task.

Sec. 2 will describe the platforms of the laser serving system: T4/ODIS robot system. The descriptions will cover the following topics: basic configurations of T4, specifications of laser, gimbal, and LPR system, analysis $\{on\}$ $\xRightarrow{41}$ $\{about\}$ the height of laser, T4LPR application, accuracy rate of LPR, and the system architecture. Sec. 3 will focus on the bumper fitting $\{algorithms\}$ $\xRightarrow{42}$ $\{algorithm\}$ and its implementations. Based on the analysis on the characteristics of 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 experiment results. Sec. 5 will conclude the paper.

Figure T4ODIS will be here

Figure 2. ODIS is docking to T4

Figure T2E will be here

Figure 3. T2E robot, the platform of testing

2. TESTING PLATFORM

2.1. T4 System Overview

The modules of T4 and the workstation that $\{related\}$ $\xRightarrow{43}$ $\{relate\}$ to the laser serving task $\{were\}$ $\xRightarrow{44}$ $\{are\}$ plotted in Fig. 4. $\{On\}$ $\xRightarrow{45}$ $\{In\}$ this figure, $\{those\}$ $\xRightarrow{46}$ $\{\}$ blocks with shadows are $\{the\ concepts\ in\ the\ software\ domain\}$ $\xRightarrow{47}$ $\{software\}$, 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. This figure does not distinguish between hardware $\{connection\}$ $\xRightarrow{48}$ $\{connections\}$ and software coupling. Any line in the figure that has a hardware block on one of its two sides is a physical wire, and the line connects two software blocks is a software coupling. A software coupling might be a “calling” between functions, or a communication channel among different threads.

T4 has three embedded computer systems called $\{\}$ $\xRightarrow{49}$ $\{the\}$ “master node,” “sensor node,” and $\{“LPR\ node”\}$ $\xRightarrow{50}$ $\{“LPR\ node”\}$ respectively. Most of the functions $\{were\}$ $\xRightarrow{51}$ $\{are\}$ implemented on the “master node,” including the supervisory task controller (STC),⁵ a map database, and different threads $\{that\}$ $\xRightarrow{52}$ $\{\}$ dedicated to individual tasks. $\{\}$ $\xRightarrow{53}$ $\{The\}$ “sensor node” mainly $\{pre-process\ the\}$ $\xRightarrow{54}$ $\{pre-processes\}$ raw sensor data in real-time, then $\{send\}$ $\xRightarrow{55}$ $\{sends\}$ the result to the workstation via the master node. Without $\{the\}$ $\xRightarrow{56}$ $\{such\}$ pre-processing, the master node $\{might\}$ $\xRightarrow{57}$ $\{would\}$ not have enough $\{computation\}$ $\xRightarrow{58}$ $\{computational\}$ power to process in real-time. $\{\}$ $\xRightarrow{59}$ $\{The\}$ “LPR node” is $\{dedicate\}$ $\xRightarrow{60}$ $\{dedicated\}$ to the T4LPR application, which can grab images from the LPR header, identify license numbers, and send results to the workstation via $\{\}$ $\xRightarrow{61}$ $\{the\}$ master node. These three nodes are connected by TCP/IP ethernet connections via a hub, which is also $\{connect\}$ $\xRightarrow{62}$ $\{connected\}$ with a wireless TCP/IP node that communicates with the other $\{node\}$ $\xRightarrow{63}$ $\{nodes\}$ on the workstation side. The odometry system that $\{\}$ $\xRightarrow{64}$ $\{is\}$ connected to the sensor node can estimate the robot’s position and orientation in real-time. As shown in Fig. 1, odometry information $\{performances\}$ $\xRightarrow{65}$ $\{acts\}$ as a feedback of the inner loop in the system.

In the master node, $\{\}$ $\xRightarrow{66}$ $\{the\}$ STC is linked with a dashed line block that contains $\{\}$ $\xRightarrow{67}$ $\{the\}$ “bumper fitting thread,” “awareness thread,” and other threads $\{, which\ means\ the\ STC\ can\ communicate\ with\ each\ of\ these\ threads\ individually.\}$ $\xRightarrow{68}$ $\{.\}$ The bumper fitting thread, together with the “LaserComm” thread and the $\{pre-process\}$ $\xRightarrow{69}$ $\{pre-processing\}$ module on the sensor node, are the “bumper fitter” in Fig. 1. Although the bumper fitter is not physically connected to the GUI, which is the reference input 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 $\{that\} \xRightarrow{70} \{a\}$ danger avoidance thread that can use laser and sonar data* to build map of surrounding environment in real-time and halt the vehicle in $\{emergence\} \xRightarrow{71} \{emergency\}$.⁹ Because $\{the\} \xRightarrow{72} \{\}$ danger avoidance has higher priority than $\{the\} \xRightarrow{73} \{\}$ laser servoing, we must consider the requirements together in order to reduce the chance of $\{conflictions\} \xRightarrow{74} \{conflicts\}$ 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. $\{The\} \xRightarrow{75} \{These\}$ details will be presented in Sec. 2.3.

The “xxxComm” blocks are functions of the communication thread. They share the same IP address, but have different TCP sockets. The reason for separating them from task threads, such as the bumper 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, $\{\} \xRightarrow{76} \{a\}$ light duty communication thread is dedicated for packet buffering, so that the packet $\{losing\} \xRightarrow{77} \{lose\}$ is prevented as much as possible.

The T4LPR application executing on the LPR node is based on the See/Car DLL[†], which returns a license $\{\} \xRightarrow{78} \{plate\}$ number after $\{received\} a\} \xRightarrow{79} \{receiving\} an\}$ image of a license plate. On the LPR node, there is a frame $\{grab\} \xRightarrow{80} \{grabber\}$ card that can record $\{signal\} \xRightarrow{81} \{signals\}$ from the LPR header into $\{\} \xRightarrow{82} \{a\}$ bitmap format. The T4LPR application transfers these images to the See/Car DLL for identification, and send 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 comprehensively to get the best fit license number. In the testing $\{of\} \xRightarrow{83} \{for\}$ this paper, T4LPR took 5 images for each license.

On the workstation side, the system is simplified as GUI and CSI. The GUI is an $\{integrate\} \xRightarrow{84} \{integrative\}$ information environment that can display the information, such as position and latest found license number, of T4 and ODIS on the screen. At the same time, it can also receive operations from the keyboard and the mouse, and transfer them to the CSI. The CSI can compile these operations into STC compatible “command units” and send them to T4 and ODIS via the wireless connection. The details are $\{out\} of\ the\ range\ of\} \xRightarrow{85} \{out\} side\ the\ scope\ of\}$ this paper.

Figure BumperFitArch will be here

Figure 4. System architecture of bumper fitting

2.2. LPR Checking Mission

The $\{executing\} \xRightarrow{86} \{\}$ procedures of a typical license plate checking mission are illustrated in Fig. 5. On the left, there are stalls and several cars in them. T4 started from the lower right position, and moved upward follow the dash arrow. The semicircle in front of T4 represents the 2D laser and gimbal system. The arrow on the left of T4 is the camera header. The “bumper box” is drawn in dashed $\{line\} \xRightarrow{87} \{lines\}$ on Fig. 5. Each “bumper box” is a rectangular region of 1/4 of the length of a stall, and with the same width of the stall. Since the stalls are static, these bumper boxes are also static, i.e., they are well known before the mission starts. Cars are illustrated by white boxes with dark edges, which represent the bumper. Cars are dynamic objects in the sense that their positions are not known before the mission, but this does not imply they are moving during the checking license mission. Instead, we assume they will not move once T4 started its mission. The numbers in the figure represent $\{the\} \xRightarrow{88} \{\}$ different $\{position\} \xRightarrow{89} \{positions\}$ on the tracks of T4. The comments for each position are the followings:

1. After T4 $\{just\} finished\} \xRightarrow{90} \{finishes\}$ the $\{one\} \xRightarrow{91} \{\}$ license checking task for the car in Stall 1, new commands came from the workstation side. The $\{planer\} of\ T4\ checked\} \xRightarrow{92} \{planner\} checks\}$ map and the current position measured by the odometry system, then $\{decided\} \xRightarrow{93} \{decides\}$ to go straight forward. Because the LPR system is not sensitive to the inspection distance, $\{which\} will\ be\ explained\ in\ Sec.\ 2.4,\}$

*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.

$\Rightarrow \{\}$ T4 $\{just\ went\}$ $\Rightarrow \{goes\}$ straight in order to simplify the system. T4 $\{queried\}$ $\Rightarrow \{queries\}$ the map and $\{get\}$ $\Rightarrow \{gets\}$ the positions $\{\}$ $\Rightarrow \{of\}$ the $\{associated\}$ $\Rightarrow \{\}$ bumper boxes of the next three stalls, i.e., **Stall 2 to Stall 4, on its left.** $\Rightarrow \{.\}$ Every 100 ms, when a new set of 2D laser data $\{arrived,\}$ $\Rightarrow \{arrives\}$ T4 $\{fitted\}$ $\Rightarrow \{fits\}$ the bumpers by the data points that $\{fell\}$ $\Rightarrow \{fall\}$ into these bumper boxes.

2. With more and more laser data $\{came\}$ $\Rightarrow \{coming\}$ in, the fitted bumper $\{changed\ and\ was\}$ $\Rightarrow \{changes\}$ closer to the real bumper. When T4 $\{moved\}$ $\Rightarrow \{moves\}$ from the position "1" to the position "2," the fitted bumper $\{changed\}$ $\Rightarrow \{changes\}$ from B1 to B2. The position 2 is defined as the intersection on the extension of the common edge of **Stall 1** and **Stall 2** and the robot's track. At the position "2," T4 $\{inquired\}$ $\Rightarrow \{inquires\}$ the best fitted bumper line from the bumper fitting thread, and the position "3" is the best inspection position according to the fitted bumper.
3. At position "3," T4 $\{triggered\}$ $\Rightarrow \{triggers\}$ the LPR system, $\{wait\}$ $\Rightarrow \{waits\}$ for 5 sec., $\{report\}$ $\Rightarrow \{reports\}$ the result to the workstation, and $\{delete\}$ $\Rightarrow \{deletes\}$ the information of the bumper box in **Stall 2**. Then, T4 $\{started\}$ $\Rightarrow \{starts\}$ to track the stall above **Stall 4**, $\{\}$ $\Rightarrow \{\}$ and move toward position "4."
4. At position "4," T4 $\{checked\}$ $\Rightarrow \{checks\}$ the fitted bumper line and $\{set\}$ $\Rightarrow \{sets\}$ the target position as "5."
5. At position "5," T4 $\{did\}$ $\Rightarrow \{does\}$ the same job as it did at "3." $\{Follow\}$ $\Rightarrow \{Following\}$ these procedures, T4 $\{moved\}$ $\Rightarrow \{moving\}$ on until it finished all the stalls.

Figure Track will be here

Figure 5. LPR checking when moving straight.

2.3. 2D Laser Scanner and Gimbal System

Specifications

The specifications of the 2D laser scanner is listed in Table 1, $\{and\}$ $\Rightarrow \{\}$ Table 2 $\{is\}$ $\Rightarrow \{shows\}$ 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	45°/sec. $\{\}$ $\Rightarrow \{check!\}$

2D Laser Height

Figure 6 plots $\{\}$ $\Rightarrow \{the\}$ heights of some vehicles. Each vertical line in this figure represents one vehicle. The short bars $\{of\}$ $\Rightarrow \{at\ the\}$ 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 $\{\}$ $\Rightarrow \{the\}$ top of $\{\}$ $\Rightarrow \{the\}$ bumpers, are the engine covers. Although the laser data points $\{\}$ $\Rightarrow \{that\}$ 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 the bumper fitting algorithm, the desired height of the 2D laser scanner is between the second bar and the third bar, and the desired $\{\text{tile}\} \xRightarrow{128} \{\text{tilt}\}$ angle is 0. This configuration can greatly simplify the bumper fitting algorithm and $\{\text{the}\} \xRightarrow{129} \{\text{more}\}$ easily guarantee the real-time property.

Consider the case when the laser scanner is mounted higher than the $\{\text{bottoms}\} \xRightarrow{130} \{\text{bottom}\}$ of windows. Then the $\{\text{tile}\} \xRightarrow{131} \{\text{tilt}\}$ angle has to be adjusted according to the distance between $\{\} \xRightarrow{132} \{\text{the}\}$ T4 robot and the target vehicle in order to shine the laser beam roughly to the bumper $\{\} \xRightarrow{133} \{\text{height}\}$ 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 $\{\} \xRightarrow{134} \{\text{to}\}$ adjust the $\{\text{tile}\} \xRightarrow{135} \{\text{tilt}\}$ angle. Another disadvantage of the configuration is the complexity. For this approach, the laser will “see” the ground, bumpers, engine covers, $\{\} \xRightarrow{136} \{\text{and}\}$ windows $\{\} \xRightarrow{137} \{\}$ 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 2 dimensions, since the differences between their heights are neglectable.

For the case when the 2D laser is mounted lower than the bottom of $\{\} \xRightarrow{138} \{\text{the}\}$ bumpers, the 2D laser and gimbal system might be damaged by stones on $\{\} \xRightarrow{139} \{\text{the}\}$ road. $\{\text{Still}\} \xRightarrow{140} \{\text{However}\}$ since the tilt angle is positive, and the measured data are in 3 dimensions, 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 $\{\text{chose}\} \xRightarrow{141} \{\text{choose}\}$ this height, we need to consider constrains from the awareness thread^{1,9} and $\{\} \xRightarrow{142} \{\text{the}\}$ localization thread,^{1,10} since these two threads also need laser data.

Figure VehicleHeight will be here

Figure 6. Vehicle heights in a parking lot.

$\{\} \xRightarrow{143} \{\text{Constrains: localization and awareness thread}\}$ The awareness thread can process laser and sonar data in real-time, build a HIMM map, estimate positions of $\{\text{potential}\} \xRightarrow{144} \{\text{potentially}\}$ dangerous objects, and halt $\{\} \xRightarrow{145} \{\text{the}\}$ T4 robot if necessary. In order to detect objects as far $\{\} \xRightarrow{146} \{\text{away}\}$ as possible, the best laser tilt angle for the awareness thread $\{\text{0}\} \xRightarrow{147} \{\text{is } 0^\circ\}$, which means the laser should scan $\{\text{horizontal}\} \xRightarrow{148} \{\text{horizontally}\}$. Since security is more important than any mission, the awareness thread has the highest priority $\{\} \xRightarrow{149} \{\}$ which $\{\text{enable}\} \xRightarrow{150} \{\text{enables}\}$ it to preempt laser/sonar data and the gimbal control. If 2D laser is mounted $\{\text{on a certain height}\} \xRightarrow{151} \{\text{at a height}\}$ such that horizontal scanning can support both awareness thread and the bumper fitting, then resource $\{\text{confection is}\} \xRightarrow{152} \{\text{confects are}\}$ 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 confliction with the bumper fitter, since the bumper fitting algorithm does not need to work when T4 stops. $\{\text{For the real-time localization, it}\} \xRightarrow{153} \{\text{Real-time localization}\}$ is still $\{\text{on}\} \xRightarrow{154} \{\text{under}\}$ development. $\{\text{We}\} \xRightarrow{155} \{\text{Based on the considerations, we}\}$ propose to keep the tilt angle $\{\text{as } 0\} \xRightarrow{156} \{\text{at } 0^\circ\}$ in order to accommodate the laser serving system.

According to the above considerations, the 2D laser is mounted at 28in with default tilt angle $\{\text{as } 0\} \xRightarrow{157} \{\text{at } 0^\circ\}$.

2.4. LPR System

LPR System Specifications

The LPR system $\{\} \xRightarrow{158} \{\}$ including software and hardware, on T4 is based on the **See/Lane** system of Hi-Tech Solutions Inc. $\{\} \xRightarrow{159} \{\}$ On the software $\{\text{part}\} \xRightarrow{160} \{\text{side}\}$, the **See/Lane** system includes a license identification

DLL called “See/Car DLL,” and an GUI application call “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 $\{integrated\} \Rightarrow \{integrates\}$ an IR camera, an optical filter, and an illumination system, a frame $\{grab\} \Rightarrow \{grabber\}$ card, cables, a license plug[‡], and a PCI I/O $\{cards\} \Rightarrow \{card\}$ 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 $\{, thus\} \Rightarrow \{ to make\}$ the identification result $\{is\} \Rightarrow \{ \}$ more reliable. Because the embedded computer on T4 has no PCI slot, the testing of this paper is based on $\{ \} \Rightarrow \{a\}$ constant illumination level.

The Accuracy Rate of LPR

$\{ \} \Rightarrow \{Factors that effect the accuracy rate: sunshine, mode, distance, angle. \}$ There are several factors that effect the accuracy rate of the LPR system: sunlight, distance, angle, and identification mode.

If there is no protection against $\{the sunshine\} \Rightarrow \{sunlight\}$, the accuracy rate of the LPR system can be virtual zero. To reject the disturbance of the sunshine, we $\{chose\} \Rightarrow \{used\}$ 940nm narrow band IR optical filter, and the illumination system with 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 of the US licenses are reflective to IR, while the paint of the vehicles and backgrounds, etc., are non-reflective, $\{so\} \Rightarrow \{ \}$ 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 optical IR filter, while the lower one was taken without the filter. The accuracy rates on the images with $\{the\} \Rightarrow \{ \}$ IR filter are much higher, since $\{here is\} \Rightarrow \{there are\}$ not many objects in this type of image to disturb the license identification procedure.

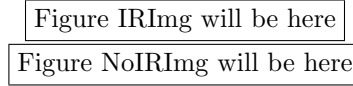


Figure 7. Images with and without IR optical filter

The LPR system has $\{ \} \Rightarrow \{a\}$ “single mode” and $\{ \} \Rightarrow \{a\}$ “multiple mode.” The former $\{one\} \Rightarrow \{ \}$ implies the T4LPR application returns one license number per image. In the latter mode, the T4LPR may take up to 10 images, consider the identification result of each one, and return one 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, $\{where normally is\} \Rightarrow \{which is normally\}$ 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. $\{While\} \Rightarrow \{When\}$ the distance CA $\{is increasing\} \Rightarrow \{increases\}$, the accuracy rate of the LPR system $\{firstly dropped, then became zero, because part of the license was invisible\} \Rightarrow \{decrease to zero\}$. Figure 9 has two plots: the upper one is the accuracy rate vs. inspection angle $\angle CDA$, the lower one is the accuracy rate vs. inspection distance CD. $\{ \} \Rightarrow \{demo plots now!\}$ According to these testing results, the positioning angular resolution for LPR should be $\pm 9^\circ$, and the positioning distance resolution $\{ \} \Rightarrow \{should\}$ be less than $\pm 50\text{cm}$. $\{ \} \Rightarrow \{It is observed in field testing that the accuracy rate is not very sensitive to the inspection distance compared with the inspection angle.\}$

The inspection distance and angle are important to $\{the\} \Rightarrow \{ \}$ 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. $\{So we replaced with\} \Rightarrow \{Thus we used\}$ a 9mm lens $\{instead\} \Rightarrow \{ \}$. Easy to see, the license images of T4LPR system is larger than the suggestion. $\{It is observed in the field testing that the accuracy rate is not very sensitive to the inspection distance compared with the inspection angle.\} \Rightarrow \{ \}$

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

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. 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 beginning every time new data come. So the feature of progressive is important. Robust is 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 $\{\}$ ¹⁸⁷ \Rightarrow $\{\text{is}\}$ 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 use the laser scanner collect 3D data with the tilt angle swept between $\pm 30^\circ$. After $\{\text{cropped}\}$ ¹⁸⁸ \Rightarrow $\{\text{cropping}\}$ data points $\{\text{farther then}\}$ ¹⁸⁹ \Rightarrow $\{\text{further than}\}$ 9m, the result $\{\text{of them were}\}$ ¹⁹⁰ \Rightarrow $\{\text{was}\}$ plotted in Fig. 10. This figure $\{\text{demonstrated}\}$ ¹⁹¹ \Rightarrow $\{\text{demonstrates}\}$ that the laser beam might go through the windows and “see” inside a vehicle $\{\}$. $\{\text{look at the van on right.}\}$ ¹⁹² \Rightarrow $\{\text{(look at the van on right).}\}$ Though the height of the 2D laser scanner was carefully chosen such that the possibility of “ $\{\text{see}\}$ ¹⁹³ \Rightarrow $\{\text{seeing}\}$ ” though windows is very small, it is still a $\{\text{consider}\}$ ¹⁹⁴ \Rightarrow $\{\text{consideration}\}$ in the algorithm selection due to its high demand on the safety. Another kind of disturbance that did not shown in this plot is pedestrians. They might walk in the bumper boxes and greatly $\{\text{interference}\}$ ¹⁹⁵ \Rightarrow $\{\text{interfere with}\}$ the bumper fitter.

Figure LPRAng will be here

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

DEMO
DEMO

[scale=0.45]DemoLPRAccuracy

Figure 9. The accuracy rate of LPR system.

$\{\}$ ¹⁹⁶ \Rightarrow $\{\text{later: fit bumper}\}$

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 $\{\}$ ¹⁹⁷ \Rightarrow $\{\text{the}\}$ Hough transform. If LMS was implemented on T4, a segmentation algorithm that can distinguish bumper and pedestrians, or “in window objects” is necessary for laser data pre-processing. To $\{\text{reduce the}\}$ ¹⁹⁸ \Rightarrow $\{\text{eliminate the need for such}\}$ complexity, Hough type transforms $\{\text{was implemented on the real system}\}$ ¹⁹⁹ \Rightarrow $\{\text{were implemented}\}$.

$\{\}$ ²⁰⁰ \Rightarrow $\{\text{The}\}$ Hough transform (HT) is a popular method for the extraction of geometric primitives.^{11, 12} Initially, it is only an approach for line detection. Later, many variants are developed to detect circle,¹² ellipse,¹³ or more complex binary patterns.¹⁴ A survey on $\{\}$ ²⁰¹ \Rightarrow $\{\text{the}\}$ Hough transform can be found in.¹¹

Generally speaking, $\{\}$ ²⁰² \Rightarrow $\{\text{the}\}$ 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. A sparse Hough Transform (SPHT) algorithm was proposed in¹⁰ for laser data fitting. In this paper, we will review it and compare the $\{\text{experiment}\}$ ²⁰³ \Rightarrow $\{\text{experimented}\}$ results of SPHT and SHT.

Figure NearPts2 will be here

Figure 10. Collected laser data with tilt angle.

Figure PartOfBumper will be here

Figure 11. On Laser data set of a bumper.

Figure OneSweepBumper will be here

Figure 12. One laser sweep on a bumper.

Figure OneSweepBumperLine will be here

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

4. $\{EXPERIMENT\} \Rightarrow \{EXPERIMENTAL\}$ RESULTS

Figures 12 and 13 are $\{the\ example\ for\} \Rightarrow \{examples\ of\}$ SPHT bumper fitting. Figure 12 is one sweep of laser data on a vehicle bumper. After line fitting by $\{\} \Rightarrow \{the\}$ SPHT, the result is plotted in Fig. 13. The two circles represent the two laser data points that are believed as two ends of the bumper edge. When processing this example, the distance resolution was 6cm and angular resolution was 9° . Except for the $\{an\} \Rightarrow \{\}$ array that stores the laser points, the SPHT just need 3 float variables, and $\{\} \Rightarrow \{the\}$ SHT needs 46×46 integer variables. $\{With\} \Rightarrow \{When\}$ the C++ $\{implemented\} \Rightarrow \{implementations\}$ of the two executing on a 700MHz, 256MB desktop, the SPHT took 0.1sec., and $\{\} \Rightarrow \{the\}$ SHT took less than 0.02sec. 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. $\{\} \Rightarrow \{The\}$ SPHT and $\{\} \Rightarrow \{the\}$ SHT are fast enough. For the LHT approach, when robot was far from a specific bumper box, it needed less cells in accumulation space for bumper fitting. But while it was approaching the bumper box, more cells are $\{requires\} \Rightarrow \{required\}$. This 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 $\{\} \Rightarrow \{on\}$ the LPR checking mission. This screen shot was taken when T2E $\{\} \Rightarrow \{was\}$ almost finished $\{\} \Rightarrow \{with\}$ 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 these stalls and triggered the T4LPR application.

Figure 15 overlapped a GUI screen shot (upper right) on a picture (background) in which T2E was moving in a parking lot. The returned license numbers were $\{store\} \Rightarrow \{stored\}$ in the hard disk of the workstation, $\{\} \Rightarrow \{and\}$ in the mean time $\{\} \Rightarrow \{\}$ they were displayed on a 3D perspective GUI.

Figure BumperLine will be here

Figure 14. Bumper line on GUI.

Figure T2EPlusGUI will be here

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

5. CONCLUSION

5.1. Comments

In this paper $\{we\ have\} \Rightarrow \{\}$ presented a laser servoing system for precision motion control. Performed as the outer loop, this laser servoing system $\{enhance\} \Rightarrow \{enhances\}$ 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 $\{the\ resources\ confliction\} \Rightarrow \{resources\ conflict\}$, balance among different requirements, $\{\} \Rightarrow \{and\}$ mission scheduling strategies. This paper carefully analyzed all requirements on the laser servoing system from those constrains, then proposed $\{\} \Rightarrow \{the\}$ Hough $\{transform\ based\} \Rightarrow \{transform-based\}$ algorithms as the feedback processor. $\{\} \Rightarrow \{The\}$ SHT and $\{\} \Rightarrow \{the\}$ SPHT were implemented as the kernel algorithms for the bumper fitter. For this application, $\{\} \Rightarrow \{the\}$ SHT is faster but takes more memory, and $\{\} \Rightarrow \{the\}$ SPHT is vice versa: takes less memory but $\{\} \Rightarrow \{is\}$ slower. Another bumper fitting algorithm, LHT, was discussed but not implemented.

5.2. Future Work

The concept of laser servoing might not be limited to LPR accuracy rate enhancement $\{\text{only}\} \Rightarrow \{\cdot\}$. Instead, $\{\text{the}\} \Rightarrow \{\cdot\}$ similar $\{\text{algorithm}\} \Rightarrow \{\text{algorithms}\}$ might help $\{\cdot\} \Rightarrow \{\text{the}\}$ robot to avoid danger, follow objects, or $\{\cdot\} \Rightarrow \{\text{carry out}\}$ other autonomous missions. In $\{\cdot\} \Rightarrow \{\text{the}\}$ future, the concept of laser servoing might be greatly expanded and extended.

REFERENCES

1. V. Bahl and K. L. Moore, "Multi-robot autonomous parking security system," in Proc. of the IEEE Conf. on Decision and Control, 2003.
2. K. Moore, N. Flann, S. Rich, M. Frandsen, Y. Chung, J. Martin, M. Davidson, R. Maxfield, and C. Wood, "Implementation of an omni-directional robotic inspection system (ODIS)," in Proc. of SPIE Conf. on Robotic and Semi-Robotic Ground Vehicle Tech., SPIE, (Orlando, FL.), May 2001.
3. "SeeLane technical information." Hi-Tech Solutions Inc. <http://www.htsol.com/Download.html>, Oct. 2001.
4. "Measurably more economical, LMS200/LMS220 laser measurement systems." SICK Inc. <http://www.sick.de>, 2003.
5. 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.
6. 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.
7. K. L. Moore and N. S. Flann, "A six-wheeled omnidirectional autonomous mobile robot," IEEE Control Systems Magazine **20**, pp. 53–66, Dec. 2000.
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 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 Computation **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.