A smart interface-unit for the integration of pre-processed laser range measurements into robotic systems and sensor networks

Hannes Bistry, Daniel Westhoff and Jianwei Zhang

***Abstract*— In this paper we propose a novel method of connecting laser range finders to a service robot or to an arbitrary sensor network. The developed unit can be connected to two laser range finders and provides a standard Ethernet port. Autonomous computing power enables the unit to carry out data manipulation tasks. Every device in the network will be able to access both the raw measurement data and pre-processed measurements calculated by the smart interface unit. The paper will discuss the technical aspects of the developed unit as well as the implemented novel processing algorithms. These algorithms are specific to tasks many service robots have to carry out. We present a use case where the developed unit has been applied successfully.**

I. Introduction

Incorporating sensor measurements in technical systems is becoming increasingly important in many application areas. Industrial production sites are more flexible using advanced processing techniques for sensor data. Some systems like autonomous mobile robots would not be possible without the use of sensor measurements, since the interpretation of sensor data provides the basic model of the robot's environment, which is in turn the basis for all planning and task execution algorithms that follow a measurement. In the next years, autonomous robots will be used in many applications. These could be assembly, delivery and cleaning tasks, or security services in dynamic and changing environments.

Currently laser range finders are among the most popular and successful sensor devices in robotics. Over the last decade, they have been used for security applications or other high-level applications like robot self localization [1], map building [2], [3] and people tracking [4], [5], [6]. Although the use of cameras in robotics is increasing, laser range finders will remain popular in coming years. With a smart interface for laser range finders, pre-processing algorithms, like those presented in [7], and feature extraction algorithms such as line or edge detection, could be computed directly on the smart interface. Comparisons of line extraction algorithms can be found in [8], [9]. New applications like the people tracking presented in [5] have become possible with smart interfaces.

Integration of laser range finders into a robot system or an intelligent environment is not always easy because these devices provide a proprietary type of interface. Therefore we developed our smart interface unit that makes it possible to connect them via Ethernet and allows a one-to-many type of connection.

The remainder of this paper is organized as follows: In section II we present our research background and discuss the technical challenges that occurred with the laser range finders. We introduce the service robot TASER which will be used as an example throughout this paper. Section III introduces the features and the implementation of the novel smart interface for laser range finders. In section IV several experiments analyze the performance, reliability and functionality of the newly developed device. Section V gives an outlook on future research issues. A conclusion is given in section VI.

II. Research background

The Institute of Technical Aspects of Multimodal Systems (TAMS) is doing research on service robotics using the Service Robot TASER (see figure 1). The platform is a modified version of the MP-L655 by NEOBOTIX. It features a differential drive with wheel encoders. For manipulating tasks it has two Mitsubishi PA10-6C robot arms with a payload of 6 kg each. Each arm is equipped with a 3-finger robot hand from Barrett Technologies Inc. Thus the robot system is capable of grasping and manipulating objects and releasing them at another position. Several camera systems are integrated in the service robot. A stereovision head with a pan-tilt unit provides three-dimensional images of the environment. A high-resolution omnivision camera system gives a 360° view of the robot's surroundings. Each robot hand has its own hand-camera for controlling the approach to objects. Furthermore, the service robot is equipped with two SICK LMS 200 laser range finders, which are used for localization and collision prevention. All the sensor data is processed in an industrial PC with an Intel P4 processor running Linux OS. The laser range finders are connected to a MOXA Multiport Serial Board CP-132. This is a PCI card with two RS-422 interfaces. Due to the unusual baud rate of the SICK laser range finders (500 kBd) some modifications had to be carried out on this card.

During the operation of the service robot several problems used to occur: When the laser range finders were set to their real-time mode, the system load of the PC was about 75 % in its basic working mode, and about 40 % of the measurement data got lost. The reason for these problems was the inefficient implementation of the MOXA card. This device provides only a small 16 byte FIFO, which requires

Hannes Bistry and Daniel Westhoff are with the Institute of Technical Aspects of Multimodal Systems, Department of Computer Science, University of Hamburg, Germany. {bistry, westhoff}@informatik.uni-hamburg.de

J. Zhang is with the Institute of Technical Aspects of Multimodal Systems, Department of Computer Science, University of Hamburg, Germany. He is also the leader of Cognitive Technology Laboratory, Shenzhen Institute of Advanced Technology, Shenzhen, China. (e-mail: zhang@informatik.uni-hamburg.de). zhang@informatik.uni-hamburg.de

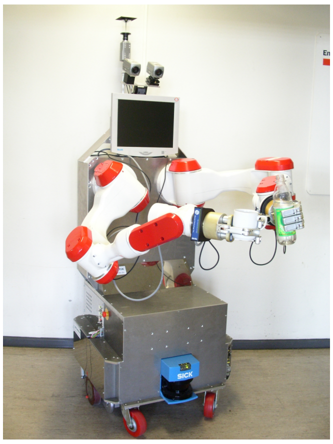


Fig. 1. The Service Robot TASER (TASER stands for TAMS Service Robot)

a whole interrupt service routine to be executed for fetching only a few bytes. Instead of just replacing the device with a RS-422 board featuring a bigger FIFO, we planned to develop a novel method of connecting the laser range finders to the service robot. In addition to the elimination of the problems, the unit should feature smart capabilities like autonomous control of the laser range finders, preprocessing of measurement data and one-to-many type of connections. We decided to use Ethernet as the interface because it is very common and not restricted to a single architecture or operating system of a host PC.

III. Implementation of the smart RS422-to-Ethernet converter

There are several solutions on the market for connecting serial devices to Ethernet. None of the devices known to us can be used to connect SICK laser range finders, because they do not supports the baudrate of 500 kBd. Most of the devices support up to 230 kBd, some up to 460 kBd. In order to receive real-time measurement data, the communication partner has to support 500 kBd.

Therefore a special solution based on a microcontroller system had to be developed. The Rabbit Powercore 3800 is one of the few microcontroller platforms that is capable of asynchronous serial communication at a baudrate near 500 kBd (difference less than 1 %, not critical). This system consists of a small industrial PC featuring a Rabbit 3000 8-bit microprocessor running at 51,6 MHz, 10/100BaseT compatible Ethernet, six asynchronous serial ports, 1 MB flash ROM and 1 MB RAM. Further components can be added on a user-designed motherboard which is connected to the Powercore 3800 by a 50-pin connector. The development kit including a programming interface, an integrated development environment and a power supply is available for about $200.

A. Hardware features

We developed a motherboard with additional parts required for the desired application that is connected to the Powercore. Because the serial input and output ports have CMOS level, a dual RS-422 transceiver chip has to be added to convert the levels. Two SUB-D male connectors are soldered onto the motherboard to connect the laser range finders. Some results of the processing of measurement data are to be visualized with LEDs in the housing of the device, so driver chips are added that connect the LEDs to general purpose I/Os of the Powercore. The layout of the whole interface unit is shown in figure 2.

B. Software features

The basic functionality of the device is the forwarding of measurement data. Several additional functions have been added to make the integration into the robot control software as easy as possible and to reduce the load of the host system.

1) Telegram level synchronization: One important function is the telegram level synchronization. The SICK LMS 200 sends status and measurement data in a special telegram format. The task of the software is to locate the start and end of each telegram in the stream of serial data. One complete telegram has a maximum size of 732 bytes. To achieve a solid synchronization, several criteria of a validly received telegram have to be checked for:

- first byte has to be 0X02 (start byte)
- second byte is 0X80 (address byte)
- third and fourth bytes contain a valid length
- last two bytes contain a valid CRC

The unit is tolerant against byte loss and resynchronizes to the stream of data fast. A six-state automaton is implemented to provide the described functionality (see fig. 3) .

2) Control of the laser range finders: We implemented some functions to automatically synchronize to the laser range finders, to set up the right operation mode and to control the flow of data.

At start-up, the laser range finders could be set to one of the four possible baudrates. These are 9.6 kBd, 19.2 kBd, 38.4 kBd or 500 kBd, but it is unknown which is set. Therefore the developed unit tries to determine the actual speed the following way. It sends a status request telegram with each of the possible speeds and waits for the response. If no response arrives, the next baudrate is tried. After the response has been received, the commands are sent to change the speed to 500 kBd and to transmit all measurement values as a stream of data. The data is received by the unit and forwarded to the host PC. If no measurement data is received for one second, it is assumed that a connection problem has occurred. In this case, the system tries to resynchronize to the laser range finders by repeating the whole process of synchronization described above.

3) Data transmission: The processed data is sent to the host system as UDP packets. The UDP protocol was chosen for several reasons. There is less overhead compared to a TCP connection and the retransmission of lost packets is unwanted anyway as they would contain old data. The TCP connection would have to be handled as a stream of data, so the previously determined partition into telegrams would be lost. A further advantage of UDP packets is the ability

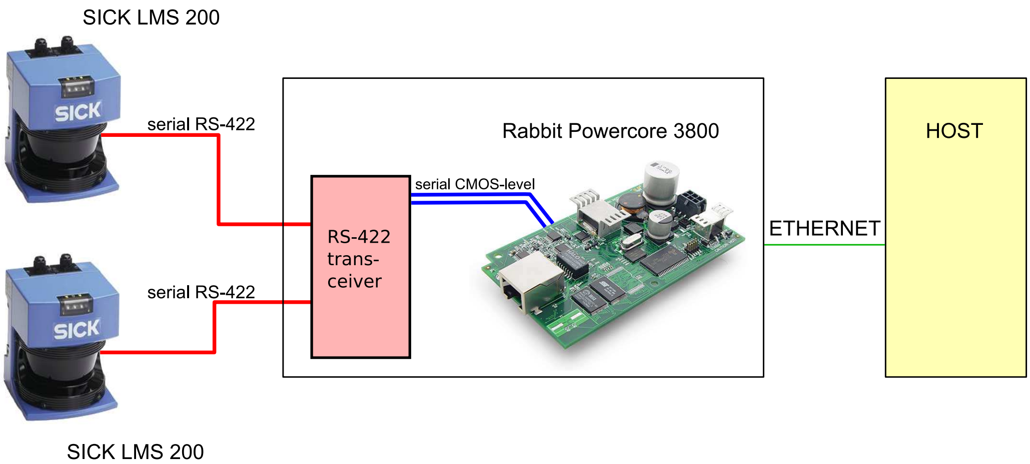


Fig. 2. This figure shows the structure of the whole system. The two SICK LMS 200 are connected to the Powercore 3800 via a dual RS-422 transceiver-chip. Forwarding of the measurement data to the host PC of the service robot is done by UDP packets over Ethernet.

to send them to a multicast address. Every device in the local area network which subscribes to the multicast address receives the packets. The measurement data can be used by many devices. This feature is already used by a project that uses the measurement data from the laser range finders on an external computer. The control PC of the service robot is configured to route the packets from the wired LAN to the wireless LAN. There are several possible applications where sensor data has to be processed by multiple programs on multiple computers. This will be a research topic in upcoming projects.

4) Pre-processing of measurement data: The Powercore 3800 provides enough computing power to carry out some easy processing tasks. We implemented some functions to support the operation of the service robot. These are a search function for reflector marks, a minimal distance check and a background subtraction algorithm.

The SICK LMS 200 checks the intensity of the reflection of each measurement value. If you attach special reflectors in a room at the appropriate height, the laser range finder will be able to recognize them. For each 13-bit distance value, the system outputs a 3-bit intensity value that is typically 0 if no reflector mark is hit. Knowing the position and the arrangement of the marks in a room, a self-localization algorithm for the robot can easily be implemented. Therefore, there is the need for generating a list that contains the position of all recognized marks. One problem that had to be solved is that one mark can appear in several adjacent measurement values. In this case the position of the mark has to be approximated. This is done by averaging the measurement values weighted with the 3-bit intensity value.

Data of the laser range finders is used on the service robot to prevent collisions. If any obstacle comes too close to the robot, the motors are stopped. The system calculates if two circular areas around the center of the robot have been penetrated by any object. This can be done by comparing each value with previously calculated thresholds. The thresholds depend on the actual angle of the measurement. At system

<table>
<tr><td>STX</td><td>ADR</td><td colspan="2">LENGTH</td><td>CMD</td><td>DATA</td><td>STATUS</td><td colspan="2">CRC</td></tr>
<tr><td>1</td><td>1</td><td colspan="2">2</td><td>1</td><td>LENGTH-2</td><td>1</td><td colspan="2">2</td></tr>
<tr><td>state0</td><td>state1</td><td>state2</td><td>state3</td><td colspan="3">state4</td><td>state5</td><td>state6</td></tr>
</table>

Fig. 3. The structure of a SICK LMS 200 telegram and the corresponding state machine. The red colored states perform a check of the received data. The LSB of the length is not checked because the value is not meaningful without knowing the MSB

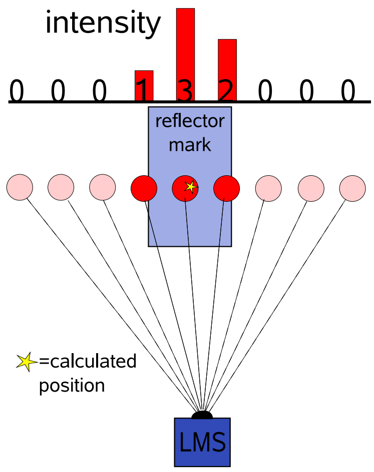


Fig. 4. If one mark is hit by multiple laser rays, the software will calculate the probable position of the mark by averaging the weighted values.

start-up a list of these values is calculated. In the current configuration the two areas are 1.0 m and 0.6 m around the center of the robot. If no object violates the area of 1.0 m, no further calculation has to be performed on the host system of the service robot because the next obstacle is far away. In the case that both areas are violated, the robot has to stop its motion, and there is no need for additional calculations on the robot. Only for the case that the 1.0 m area is violated but the 0.6 m area is not, there have to be calculations on the robot itself, which are mainly adjustments of the driving speed of the robot. The result of the minimal distance check

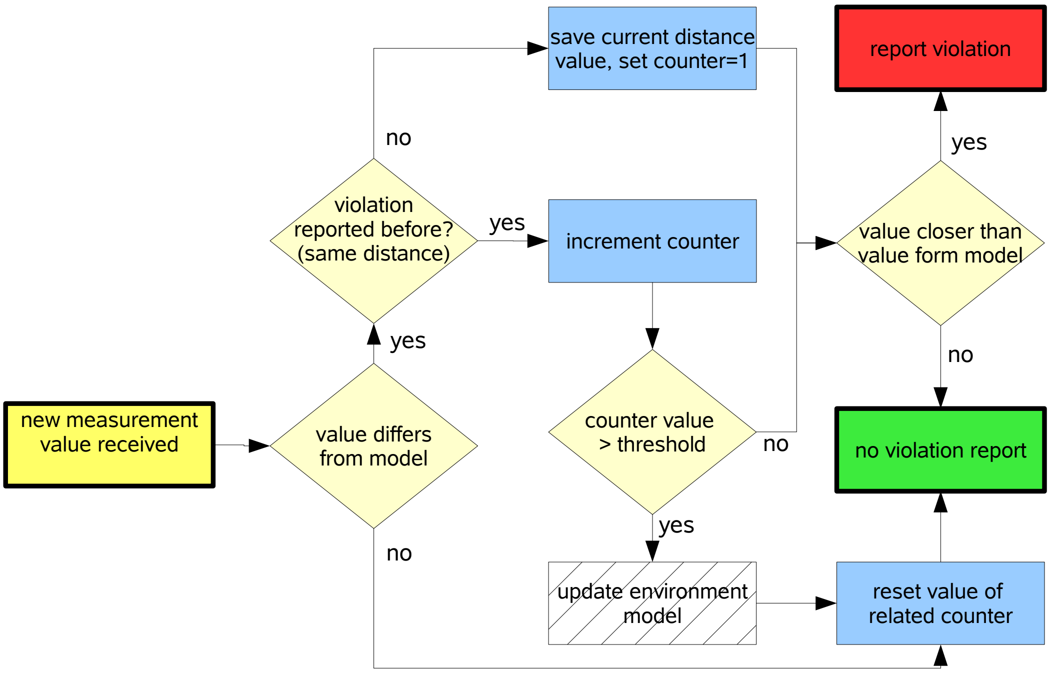


Fig. 5. This figure shows the background subtraction algorithm. It starts with a new measurement value and decides, whether this measurement value is belonging to an object moving between the background and the laser range finders

is visualized by LEDs in the housing of the device.

One of the popular use cases of laser range finders are tracking algorithms. Tracking means that the motion of an object is observed and evaluated. These algorithms generate a background model and then they determine whether each measurement value belongs to the background or not. Implementing background substraction capabilities on the smart interface unit has the advantage that you can configure the unit to transmit only measurement data from objects in the foreground. The current model of the background is only sent on demand. This feature reduces the load of the network and eases the implementation of higher-level applications on the control PC. We developed a completely new algorithm for the background subtraction task that is adapted to the limited resources of the Powercore 3800. Our algorithm combines on-the-fly map generation, map update and the check for objects moving inside this environment. The algorithm tries to determine whether a recognized object is only temporally moving through the scene or if it has been placed there for a longer period of time. We only change the environment model when the measurement value differs from the model and stays constant within a tolerance limit for some time. The implementation of the algorithm is shown in fig. 5. For each angular value we store three pieces of information:

- The current distance of the background (environment model).
- The last measured distance that differed from this value.
- A counter value; the amount of scans where the distance has been constant and different from the model.

Each measurement value is compared to the corresponding value from the environment model. If the deviation is smaller than a threshold that has to be chosen according to the noise of the laser range finders, it is assumed that no object is located between the background and the laser range finders. If a deviation is determined, it is checked whether this object has been recognized before. The measurement value is compared to the value of the last deviant value of the same measurement angle. Each time an object of the same distance is recognized, a counter is increased by one. If this counter reaches a threshold, we assume that the background has changed and save the current measurement value as the new background value for this angle. This update of the value does not affect the values for the other angles. The counter will only count up if the measurement value at the angle is constant within a tolerance limit for a certain time. Otherwise a model update could be initiated when there are many moving objects for a longer period of time. Each time the current measurement value is smaller than the value from the environment model, the recognized object is reported.

We designed this algorithm to run on low-cost hardware in realtime, so it has to be very efficient. For the common case that no object occurs between the background and the laser range finder, the current measurement value is compared to the distance of the background and the counter is reset. These two operations consume little CPU time.

IV. Tests of the Smart Interface

Many tests were done to examine whether the developed unit is working properly. We analyzed the reliability, performance and functionality of the unit. The tests were carried out on the service robot TASER, but the results are significant for other applications, too.

connection	tel. dropped	tel. loss ratio	byte loss ratio
MOXA-PCI	4590	40.8 %	0.135 %
Powercore	0	0 %	0 %

Fig. 6. The dropped telegrams were counted within a period of 5 min. A total amount of 11250 telegrams was send by the sensor. The number of lost bytes was determined by checking how many bytes of a bad telegram were missing.

A. Reliability

One reason for the development of the system was to avoid the loss of telegrams. To verify if this goal was achieved, we did some tests comparing the developed unit to the connection via the MOXA PCI-board. The telegram parsing software of the two systems was modified to count the telegrams that had to be dropped due to interference on the transmission line or due to lost bytes because of an interrupt service routine that was executed too late. The system was operating for 5 minutes and the values have been evaluated for one of the laser range finders. Results of this test are shown in fig. 6.

According to the communication protocol of the laser range finders, one measurement data telegram consists of 732 bytes. If only one of these bytes is lost, the whole telegram is not valid anymore and has to be dumped. Therefore the 0.135 % lost bytes lead to a telegram loss of 40.8 %. When the laser range finders were connected to the Powercore, not one telegram got lost during the test period, even when the test period was extended to several hours. On the robot system, the received telegrams from the Powercore were also counted, because there could be a loss of UDP packets. In our case we received telegrams at a constant rate of 37.5 telegrams per second. This matches the telegram rate of the laser range finders. In the test setup, the Powercore was connected to the PC directly with a crossed Ethernet cable, so no collisions could occur.

B. Performance

In another test, the system load of the control PC of the service robot was measured while reading and processing the measurement data. This was done over a period of one minute by analyzing the virtual file /proc/stat. On Linux systems you access some system information provided by the kernel by reading this file. The results of this test are shown in fig. 7. The load is divided into system load (drivers, Linux kernel) and user load (running programs). You can see a strong reduction of the system load from 23,4 % to 3,9 % compared to the MOXA interface. This reduction occurs because the processing of network traffic is highly optimized within Linux systems. The user load increases slightly. This is due to the fact that 69 % more telegrams have to be processed by the localization algorithm. (100 % instead of 59.2 % lead to an increase of 69 %) The computing time per telegram instead has been reduced to 64.5% (100% = connection with MOXA) due to the pre-processing of the data.

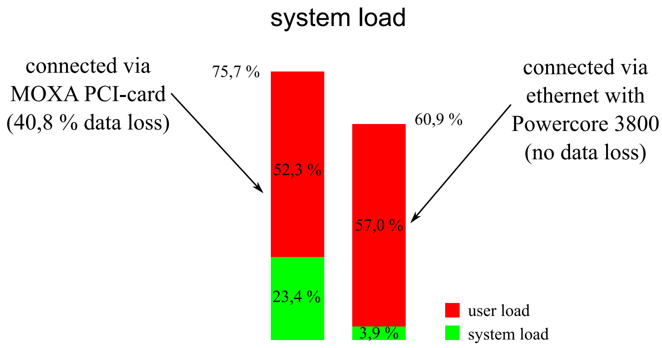


Fig. 7. The smart sensor reduces the system load on the control PC while slightly increasing the user load, because more packages are served.

C. Functionality

The test of the functionality was done by integrating the developed unit into the service robot TASER. The minimal distance calculation and the list of reflector marks were used by the modified software of the service robot. The navigation and collision detection worked without any problems. A description of the navigation functionality of the robot system can be found in [10].

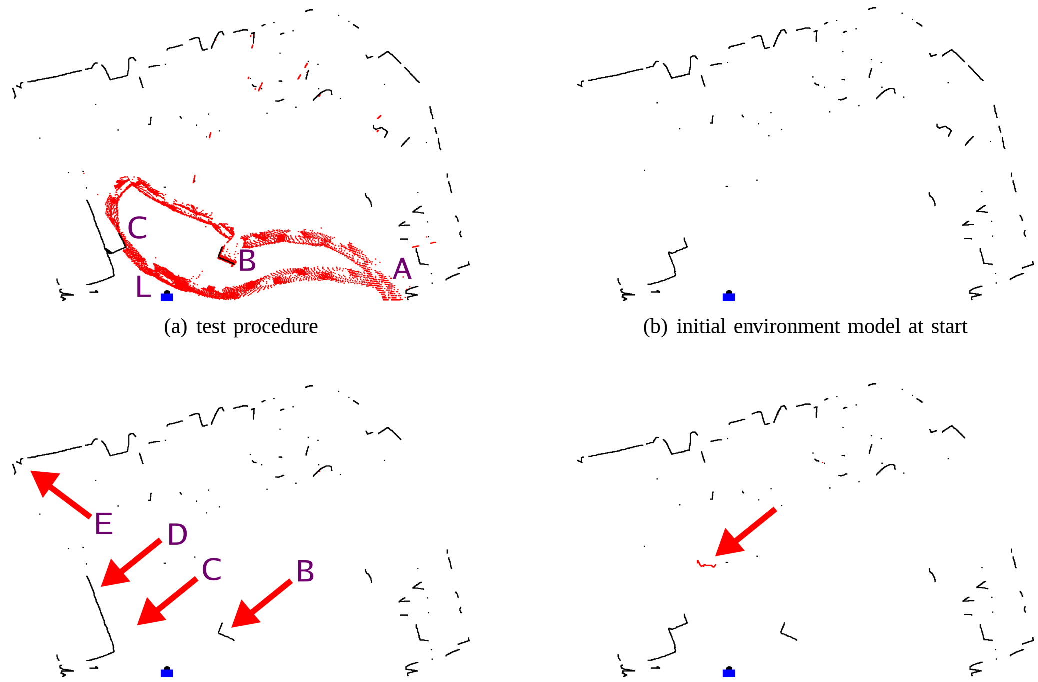
The background subtraction function was checked in a special test. We wanted to examine whether the model of the environment changed when changes in the real environment occurred. A laser range finder was placed in an arbitrary position inside a room. A person entered the room and changed the positions of some objects. We configured the unit to output the current model of the environment and the list of measurement values belonging to a moving object after processing each measurement data set. The amount of 2000 measurement values (53 seconds) was analyzed. For the host system we obtained 2000 not necessarily different models of the environment and 2000 lists of the recognized moving objects. The result of the test is shown in figure 8. The algorithm is capable of differentiating between background and moving objects. Due to noise in the measurement data some points are classified as foreground. These would have to be filtered out in a higher-level algorithm. It can be seen that this algorithm provides a very exact model of the environment.

V. Perspective

One future project is to integrate laser range finders into an intelligent environment using the developed unit. Many laser range finders could be connected to a sensor network. Multi-room-tracking, where an object leaves the measurement range of one system and enters the range of another system becomes possible when the information of many units can be accessed.

The developed unit based on the Powercore 3800 has some limitations because of the performance of the 8-bit CPU and the small memory. Higher-level algorithms have to be implemented in an additional control PC. We plan to implement the unit with a stronger CPU like ARM.

Planned software features of further versions could be:



(a) test procedure

(b) initial environment model at start

(c) environment model at end

(d) snapshot of a recognized moving object

Fig. 8. This figure shows some results of the test of the background subtraction with an adaptive model of the environment. Figure (a) shows all 2000 environment models(black) and the recognized objects(red) of the 2000 lists all overlapped in one picture. The room is cluttered due to several chairs and desks. The laser range finder is positioned beside the mark L. During the test procedure a person already carrying a box entered the room at position A. The box was placed at position B. The person moved to position C, picked up another box and left the room. Figure (b) and (c) show the initial environment model and the model after the test. Differences are marked in figure (c). At position B the contours of the recently placed box already belong to the background model. At position C the box disappeared from the model. The outlines that had been occluded by the box are added to the model (D and E). Figure (d) shows an arbitrary snapshot of the environment model and the recognized object in the foreground where the person has already placed the first box and is walking towards the second box (marked).

- background subtraction while laser range finders are moving, combined with motion estimation based on analyzing the movement of some characteristic points between two scans
- three-dimensional measurement with laser range finders mounted on a tilt-unit, background subtraction and obstacle detection in three-dimensional space

VI. Conclusion

In this paper we presented a low-cost smart interface unit to integrate laser range finders into service robots and intelligent environments. The laser range finders are coupled via standard network technology to obtain high flexibility for robot system design. Applied on the service robot TASER, the unit reduced the load of the control PC compared to the direct connection of the laser range finders. The reliability of the connection was increased noticeably. Due to the possibility of using one-to-many connections it became possible to process data on further computers in the workgroup. Various pre-processing algorithms were implemented on the smart device. The idea of adding smart capabilities will be applied to further types of sensors like cameras.

References

[1] Sebastian Thrun, Dieter Fox, and Wolfram Burgard. *Probabilistic Robotics*. MIT Press, 2005.

[2] Jose A. Castellanos and Juan T. Tardos. *Mobile Robot Localisation and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, 2000.

[3] Hartmut Surmann, Andreas Nüchter, and Joachim Hertzberg. An autonomous mobile robot with a 3d laser range finder for 3d exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3–4):181–198, 2003.

[4] Maren Bennewitz, Wolfram Burgard, Grzegorz Cielniak, and Sebastian Thrun. Learning motion patterns of people for compliant robot motion. *The International Journal of Robotics Research*, 24(1):31–48, 2005.

[5] Martin Weser, Daniel Westhoff, Markus Hüser, and Jianwei Zhang. Real-time fusion of multimodal tracking data and generalization of motion patterns for trajectory prediction. In *Proc. of the IEEE Int. Conf. on Information Acquisition (ICIA)*, Shandong, China, August 2006.

[6] Katsuyuki Nakamura, Huijing Zhao, Ryosuke Shibasaki, Kiyoshi Sakamoto, Tomowo Ooga, and Naoki Suzukawa. Tracking pedestrian by using multiple laser range scanners. In *Proc. of the XXth ISPRS Congress, Commision IV, IAPRS Vol. XXXV*, Istanbul, Turkey, 2004.

[7] Steffen Gutmann. *Robust Navigation for Autonomous Mobile Systems (in German)*. PhD thesis, University of Freiburg, Akademische Verlagsgesellschaft Aka, Berlin, 2000.

[8] Daniel Sack and Wolfram Burgard. A comparison of methods for line extraction from range data. In *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles (IAV2004)*, Lissabon, Portugal, 2004.

[9] Viet Nguyen, Agostino Martinelli, Nicola Tomatis, and Roland Siegwart. A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics. In *Proc. of the 2005 Int. Conference on Intelligent Robots and Systems (IROS2005)*, Edmonton, Canada, 2005.

[10] Torsten Scherer. *A mobile service robot for automisation of sample taking and sample management in a biotechnological pilot laboratory*. PhD thesis, Universitaet Bielefeld, AG Technische Informatik, Technische Fakultaet, Bielefeld, April 2005.