Proceedings of the 33rd Chinese Control Conference
July 28-30, 2014, Nanjing, China

A Laser Scanning Data Acquisition and Display System Based on ROS

WANG Rui, LI Xiaoge, WANG Shuo

State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences,

Beijing 100190, P. R. China
E-mail: rwang5212@ia.ac.cn, lixiaoge2012@ia.ac.cn, shuo.wang@ia.ac.cn

Abstract: In this paper, a 3D environmental data acquisition and display system is designed based on laser scanning technology and ROS. We set up the data acquisition platform using a 2D laser range finder mounted on a pan-tilt, the acquired 3D point cloud data will be processed and visualized based on ROS. Pan-tilt control module is designed to drive pan-tilt with laser range finder to specified location. The laser scanning data acquisition, processing and visualization module is developed based on ROS to fulfill the functions like communicating with laser range finder, filtering point cloud data, transforming laser scanning data to point clouds in base coordinate system and visualizing it in RVIZ. Finally, the developed system was used in 3D environmental scanning experiments, and the results show the effectiveness of the system.

Key Words: ROS, Laser Scanning, Three-dimensional Reconstruction, Point Cloud, Laser Range Finder

1 Introduction

For robots to work in unknown environments, they need to be able to perceive the world by various sensors, among which 3D sensors such as laser range finder attract many researchers' attention because of its rapidity and accuracy [1]. The data obtained from laser range finder is called Point Cloud, which is a set of data points representing the external surface of an object in some coordinate system [2].

Compared with traditional modeling approaches, laser scanning technology allows obtaining objects' surface information, both in range and intensity without contact directly. The collected data will further be used for conducting three-dimensional reconstruction. This method can be little-affected by surface complexity of the object and have high precision if the sampling density is intensive enough. So the laser scanning technology and its related data modeling techniques developed fast over the last decade, many of which have been applied into practical engineering applications. For example, in 2006, Rabbani T. has studied the algorithms for automatic reconstruction of industrial installations using point cloud data of large-scale equipment in a factory acquired by a 3D laser scanning device [3]. However, it's too expensive to use 3D laser scanner directly. Fu has designed a high-precision stepper motor control system, which can turn the pan-tilt to rotate the 2D laser range finder in order to achieve 3D scanning data of the surrounding environment. This system is mainly used to meet the requirements of visual navigation of mobile robots [4]. Feng of China Agricultural University applied the laser vision system into apple picking robots by driving laser range finder using a linear motion unit [5].

In this paper, taking advantages of the open-source feature of Linux and ROS's set of software libraries and tools specialized in dealing with meta-operations of robots, we designed a laser scanning data acquisition and display system.

*This work is supported by National Key Technology Research and Development Program of China under Grant No.2013BAK03B00.

2 ROS

ROS (Robot Operating System, http://wiki.ros.org/) was released by Willow Garage in 2010 to provide libraries and tools for meta-operations of robots. ROS can provide services like an operation system, including hardware abstraction, low-level device control, frequently-used functions, communications between programs, package management and some libraries. Code based on ROS can be edited compiled and run on computer with various types of operation system. ROS provides a standard mechanism to implement code-sharing, to improve the modularity and reusability of the code. Thanks to this open source motto and the community that is developing the state-of-the-art algorithms and providing new functionalities, ROS is growing every day [6].

ROS is a distributed processing framework. Nodes can be individually designed and run simultaneously, and they can establish contact with each other via ROS communication module.

Communication between nodes in ROS is divided into two ways: Message and Service.

Before the introduction of these two ways of communication, let's have a quick overview of some graph concepts:

Nodes: A node is an executable that runs particular tasks. A system can contain multiple nodes. Each node has a unique identifier, and communicates with other nodes by calling ROS library functions.

Messages: Data in special ROS data type, such as int, float, struct, array.

Topics: Nodes can publish messages to a topic and subscribe to a topic to receive messages.

Services: A service is similar to a remote procedure call, including a request and a response.

Master: Master is a node, essentially. It records topics nodes publish messages to or subscribe to, and services nodes provide or call. Nodes could not find other nodes without Master.

Message is an asynchronous communication based on a publish/subscribe model. The publisher Node should specify

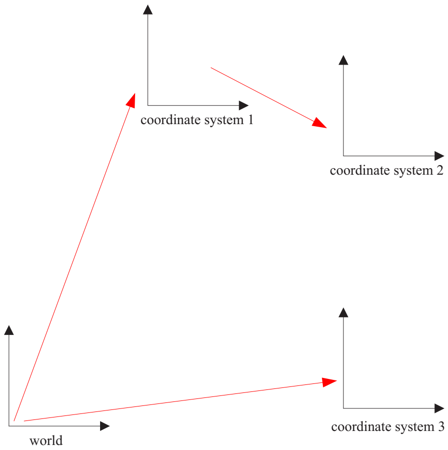


Fig. 1: Transformation of coordinate systems

a topic messages are published to, meanwhile, the subscriber node also specify the topic it gets interested in. It's possible that multiple nodes publish message or subscribe to the same topic. The connection between nodes is direct and peer-to-peer, the master only provides name service for nodes. Namely, the master helps nodes with the same topics find each other.

Service is another way that nodes can communicate with each other. Services allow nodes to send a request and receive a response, which is similar with client and server in TCP/IP communication model.

To facilitate the development and application of robot system, ROS provides many packages to implement different functions. We'll introduce tf, rivz, and some point cloud processing packages used in our laser scanning data acquisition and display system in the following section briefly.

2.1 Coordinate transformation

Tf package is mainly used for coordinate transformation between different coordinate systems. Tf provides a mechanism to broadcast the transformation relationship between two coordinate systems.

The tf package allows us define our own space coordinate systems, and then broadcasts transforms between each two coordinate systems. Transforms can be represented by rotational transfer matrix, quaternion, etc. Furthermore, other nodes can subscribe to the transforms and extract necessary relationships of coordinate systems in the robot system.

Tf creates a tree structure to represents the relationships between coordinate systems. Cyclic structure is not allowed in this tree, i.e., each coordinate system can have only one parent but can have multiple child coordinate systems.

In Fig. 1, coordinate system 1 and 3 are child of world, coordinate system 2 are child of 1, once we broadcast the transforms between them by tf, the coordinate in 2 can be easily converted to coordinate in world.

2.2 RVIZ

RVIZ is one of the huge number of small tools ROS provides to achieve a variety of functions. RVIZ provides a lightweight open source 3D visualization environment of robots, which can display coordinate systems, 3D objects,

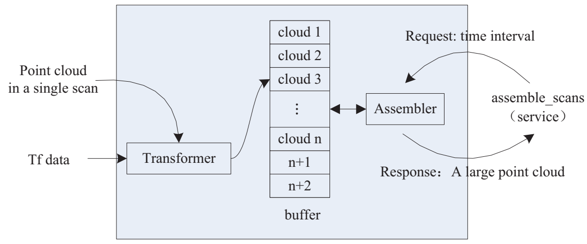


Fig. 2 The point cloud synthesis schematic

point clouds, etc. by some simple settings. 3D modeling results of the environment will be showed in RVIZ later in this paper.

2.3 Point cloud processing

The raw point clouds obtained from the laser range finder should be filtered for further use, Laser_filters package works well in this job.

Laser_filters package provides some common filters for point clouds in sensor_msgs/LaserScan type, such as the median filter, mean filter. Users can also define the relationship between the input and output just as they wish. Sensor_msgs/LaserScan is a data type similar to the "struct" in C. It can be used to indicate the data contained in a scan by 2D laser range finder, including the interval between two scan, the start and end angle of the scan, resolution, ranges array, and some other member variables.

After filtering, package Laser_geometry could convert point clouds in sensor_msgs/LaserScan type to point clouds in sensor_msgs/PointCloud type, whose member variables include points coordinates in 3D Cartesian coordinate system.

Furthermore, Laser_assembler package synthesizes point clouds obtained in multiple scans into a large point cloud in sensor_msgs/PointCloud type.

The point cloud synthesis schematic is shown in Fig. 2, the node subscribes to point cloud messages and tf data, converts the point clouds to coordinates in specified coordinate system, stores n scans in the buffer, and finally synthesizes them into a larger point cloud, other nodes can obtain the synthesized 3D point clouds by calling the "assemble_scans" service.

3 System design

3.1 Laser Range finder

We choose SICK LMS400 as our range sensor. According to the phase shift principle of continuous wave, the light propagation time and its wavelength will cause phase shift between laser transmitter and receiver. The phase will shift as the light propagates between transmitter and receiver. The laser range finder can acquire the range information between targets and the origin of the range finder using the frequency information converted from aforementioned phase difference. LMS400 can scan in the scope of 70° and in the range of 0.7-3m. With angular resolution of 0.1°-1.0°, its scan frequency can reach 500HZ and the response time is below 5ms. LMS400 also provides RS232/422 serial port and Ethernet interface to communicate. Using Ethernet interface, the communication speed can reach 10Mbps, which ensures real-time and reliable data acquisition.

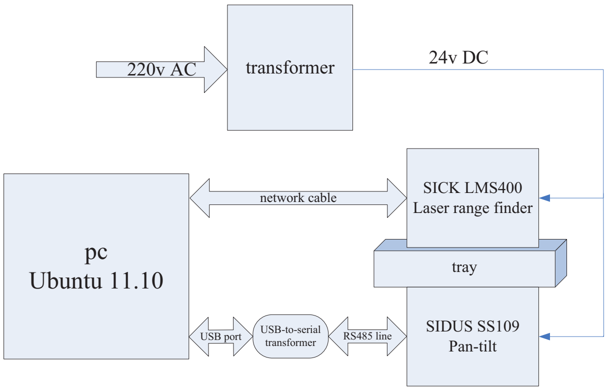


Fig. 3 The hardware block diagram

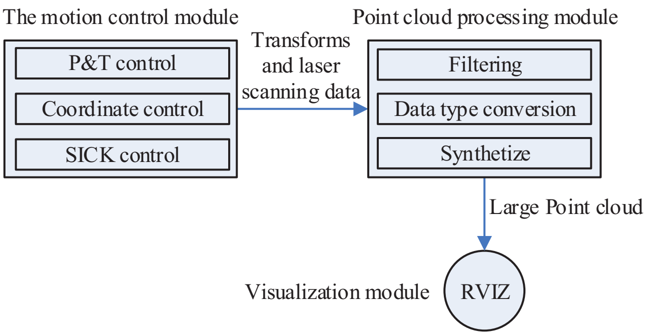


Fig. 4 Software modules diagram

3.2 Pan-tilt

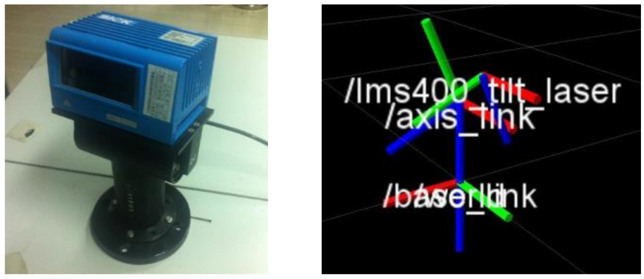
SICK LMS400 is a 2D laser range finder. We attach it to a pan-tilt to achieve 3D scanning. SS109 could realize the good precision of position control with reversible synchronous motor, low backlash harmonic transmission gear, and precision bearings. The motor within pan-tilt can provide torque 13.6Nm at most. The pan-tilt can rotate 360° around horizontal axis and 180° around vertical axis continuously at a top speed of 10°/s. In other aspects, the pan-tilt supports device addressing and RS485 communication protocol with 9600bps, and all commands and responses are 12-bit ASCII format.

3.3 Hardware design

Fig. 3 shows the hardware block diagram of the laser Scanning data acquisition and display system. Transformer of 220v AC to 24v DC powers the laser range finder and pan-tilt respectively. The computer running Ubuntu11.10 controls pan-tilt through serial interface and laser range finder through Ethernet interface. At the same time, the range finder mounted on the pan-tilt via a tray will be driven to rotate and scan environment ahead. The acquired laser scanning data will be transmitted to pc through Ethernet interface and visualized after being processed.

3.4 Software design

Fig. 4 shows the software modules of our system. The motion control module consists of a pan-tilt control node named tiltpan_control, a SICK control node named lms400_node, and a coordinate system control node named lms400_tf_broadcast.



a. actual system　　b. coordinate system in rviz

Fig. 5 Coordinate systems schematic

The tiltpan_control node drives pan-tilt with laser range finder to pitch in the scope of 60° via RS485 serial port. In addition, when the pan-tilt rotates to the both ends of the scope of movement, it will publish a message on topic /current_angle, which includes the current position and direction of motion of the pan-tilt.

The lms400_node controls the range finder and publishes acquired laser scanning data. The concrete process is shown as follows: first, lms400_node reads some necessary initialization data such as scanning resolution, scan rate, etc. from ROS parameter server. Then it creates a socket with the IP address and port of laser range finder to connect and control the range finder. Further we can change the user level and get access to change some settings of the range finder, such as scan frequency, resolution, the start and end angle with password. After the configuration information is saved, the range finder will keep on scanning the environment ahead and publishing point clouds in sensor_msgs/LaserScan type (on /lms400_node/laser_scan topic) until user terminates current process with Ctrl-C, in this case, the node will run the destructor, make the range finder stop scanning and disconnect it.

The lms400_tf_broadcast node is designed to control the transforms among the world, pan-tilt and range finder coordinate systems, which are shown in Fig. 5 (The x, y, z axis was colored with red, green and blue respectively). World is the fixed coordinate system in space. We can get the pan-tilt base coordinate system (base_link) by rotating the world coordinate system 180° around y-axis and then 90° around the new z-axis. The pan-tilt coordinate system (axis_link) is above the base_link by 20.5cm. At last, translating the axis_link along the vector (-0.031, 0, -0.09) cm and rotating 90° around the new x-axis, we get the initial position of the laser range finder coordinate system (lms400_tilt_laser). However, Because of the continuous rotation of the scanner, lms400_tilt_laser coordinate system should keep pace with the actual position of the laser range finder. So the lms400_tf_broadcast node should also subscribe to /current_angle topic tiltpan_control publishes messages of position of pan-tilt to.

Point Cloud data processing module includes three nodes, they are point cloud filtering node, point cloud data format conversion node, point cloud data synthesis node respectively.

The filtering node (scan_filtered) is a simple threshold filter to remove rough point cloud data beyond the scope of 0.3-0.7m. As to the point cloud type conversion node, the aforesaid Laser_geometry package will convert the point

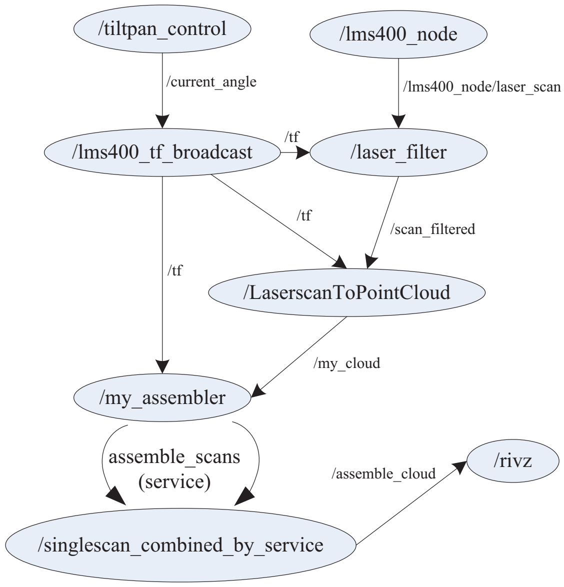


Fig. 6 Nodes relation diagram in ROS

clouds in sensor_msgs/LaserScan type to point clouds in sensor_msgs/PointCloud type, that is to say, it converts the range data acquired from the laser range finder to 3D coordinates under the world coordinate system. At last, we should establish a buffer of size 4000 to store maximum 4000 scan lines data since it takes about 8s for pan-tilt to rotate 60° and the scan frequency of range finder can reach 500HZ. Then ROS will call assemble_scans service to obtain the synthesized large point cloud in a specified time interval.

The visualization module mainly contains RVIZ node, in which 3D modeling results of the environment will be showed.

In summation, we can draw a ROS nodes relation diagram of the laser scanning data acquisition and display system (Fig. 6), wherein the tilt_control node controls pan-tilt to rotate up and down, and publishes the current position of pan-tilt to the topic /currentangle. Lms400_tf_broadcast node subscribes this message, and broadcasts the transforms of coordinate systems. The lms400_node communicates with the laser range finder and publishes point clouds in type sensor_msgs/LaserScan. The laser_filter node receives and filters the acquired point clouds from lms400_node. Then it publishes the filtered point clouds to LaserScanToPointCloud node, which converts point clouds in sensor_msgs/LaserScan type to point clouds in sensor_msgs/PointCloud type. My_assembler node synthesizes point clouds obtained in multiple scans into a larger point cloud. Finally, Single_combined_by_service node calls the assemble_scans service to extract synthesized large point cloud in specified time interval and visualize it in RVIZ.

4 Experimental results

Fig. 7 shows the actual hardware components of the laser scanning data acquisition and display system. We set LMS400 to scan with angular resolution of 0.1° and scan frequency of 500HZ. The speed of pan-tilt is set to 7.6°/s, so it takes about 8s to complete one scan of the environment ahead.

Fig. 7 Hardware components

Fig. 8 Blackboard to be scanned

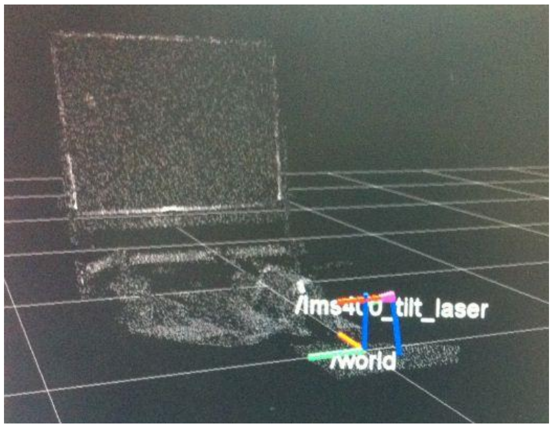


Fig. 9 3D modeling result (Blackboard)

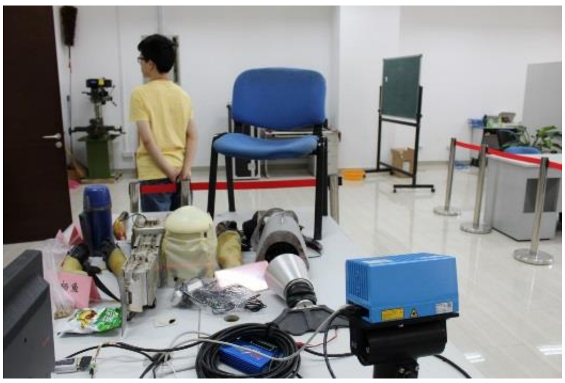


Fig. 10 Environment to be scanned

We scanned the laboratory environment showed in Fig. 8 using this system. 3D modeling result is shown in Fig. 9. Then we scanned environments in Fig. 10 and Fig. 12, the results are shown in Fig. 11 and Fig. 13.

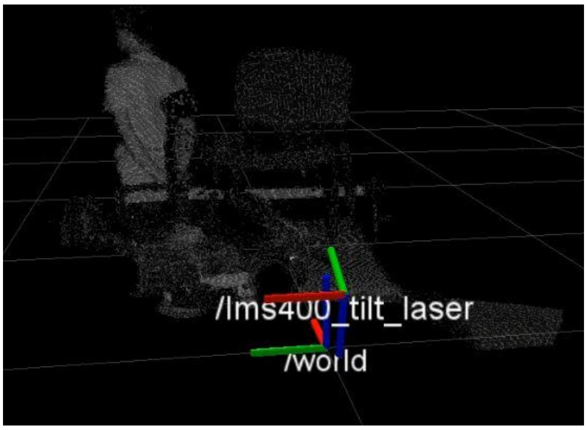


Fig. 11 The 3D modeling result of Fig. 10


Fig. 12 The tank, chair and people

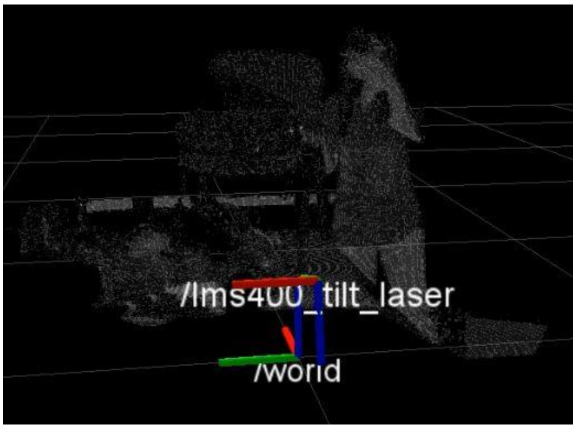


Fig. 13 3D modeling result of Fig. 12

From the experimental results, the system can achieve 3D modeling of the environment with fast scan speed and good real-time performance. But the modeling effect is not very good for objects with complex surface or edge.

5 Conclusion

In this paper, we designed a laser scanning data acquisition and display system for three-dimensional modeling of environment. In this system, we use SIDUS SS109 pan-tilt to drive 2D laser range finder SICK LMS400 to rotate in the scope of 60° in order to achieve 3D scanning of the surrounding environment. Then we convert the acquired laser scanning data to 3D coordinates in the fixed coordinate system in space and visualize it in RVIZ, thus achieving the 3D modeling of the environment. All the tools used are open source software, and it's cost-efficient to achieve 3D modeling only by using 2D laser range finder. This system can be applied to robotic systems which need visual information.

References

[1] Rusu R B, Cousins S, 3d is here: Point cloud library (pcl), *IEEE International Conference on Robotics and Automation*, 2011: 1-4.

[2] Tombari F, Salti S, Di Stefano L, Unique signatures of histograms for local surface description, *European Conference on Computer Vision*, 2010: 356-369.

[3] Tahir Rabbani, Automatic reconstruction of industrial installations using point clouds and images, *Journal of Geodesy*, 6(2), 2006.

[4] Fu Meng-yin, Li Chao, Wang Mei-ling, Yang Yi, A 3D scanning system based on Range data from mobile robot's laser scanner, *Chinese Journal of Scientific Instrument*, 29(4): 434-438, 2008.

[5] Feng Juan, Liu Gang, Si Yongsheng, Construction of laser vision system for apple harvesting robot, *Transactions of the Chinese Society of Agricultural Engineering*, 29(Supp.1): 32-37, 2013.

[6] Martinez, Aaron, and Enrique Fernández, *Learning ROS for Robotics Programming*. Packt Publishing Ltd, 2013, chapter 1.

[7] Quigley, Morgan, ROS: an open-source Robot Operating System, *ICRA workshop on open source software*, 3(3.2), 2009.

[8] Levoy, Marc, The digital Michelangelo project: 3D scanning of large statues, *International conference on Computer graphics and interactive techniques*, 2000: 131-144.

[9] M. Tan, D. Xu, Z. G. Hou, S. Wang, Z. Q. Cao, *Advanced Robotic Control*. Beijing: Higher Education Press, 2007, chapter 2.

[10] Strouth A, Burk R L, Eberhardt E, The afternoon creek rockslide near Newhalem, Washington, *Landslides*, 3(2): 175-179, 2006.

[11] Gao Zhi, Zhong Sidong, Song Lihua, 3D-model Reconstruction Based on Data of Laser Radar, *Chinese Journal of Scientific Instrument*, 25(4): 495-499, 2004.

[12] Alferes, Janelcy, Eduardo Zalama, Reconstruction of three dimensional models of environments with a mobile robot, *IEEE International Conference on Robotics and Automation*, 2005:1865-1870.