PAPER • OPEN ACCESS

Velodyne LIDAR method for sensor data decoding

IOP Conf. Series: Materials Science and Engineering **516** (2019) 012018 doi:10.1088/1757-899X/516/1/012018

Velodyne LIDAR method for sensor data decoding

M V Okunsky, N V Nesterova

National Research Tomsk State University, Tomsk, Russia
E-mail: iamleftbrain@gmail.com

Abstract. This article discusses the main technical features of the LIDAR technology and work with the output data of the LIDAR sensor. We considered the main technical characteristics of the sensor company Velodyne. The main area of consideration was the output data packet of the VLP-16 sensor. Such a data packet contains a cloud of points, their coordinates and characteristics and is transmitted via the UDP protocol. The main goal of the work was to develop a decoding algorithm for raw data for use in other software.

1. Introduction

Lidar (now used as an acronym of Light Identification Detection and Ranging) is a technology for obtaining and processing information about distant objects by means of active optical systems that use the phenomena of absorption and scattering of light in optically transparent media. To a first approximation, lidar as a device is an active rangefinder of the optical range. In machine vision systems, scanning lidars form a two-dimensional or three-dimensional picture of the surrounding space, which can also be used for mapping of a given space.

Lidar uses a light transmitter that sends a short pulse of a laser beam in one direction. The light wave passes through space until it hits the target, reflects and returns. The receiver records the flight time and light intensity. After that, the flight time is calculated. The relationship between time, speed and the distance to the target can be expressed by the following equation:

$$d = \frac{t-t'}{2} * c, \tag{1}$$

where d is the distance to the target, t is the flight time, t' is the start time of the beam, c is the speed of light propagation in the medium.

In expression (1), the total time is divided by 2 because the light takes the same path twice.

This article discusses LIDAR VLP-16 by Velodyne.

Velodyne VLP-16 is an intelligent sensor that has its own internal processor and network interface. It provides data via the Ethernet interface over UDP protocol.

The VLP-16 has 16 vertically arranged channels with a transmitter / receiver pair on each channel. Lasers rotate through 360° around the vertical axis. The range of the vertical field of view is 30°, which means that the distance between each pair of rays is 2°. The maximum measurement distance is 130 m (found experimentally); this corresponds to obtaining 300,000 scanned points per second.

Table 1 presents the main technical specifications of the VLP-16.

Table 1. Technical Specifications of the VLP-16.

Title of the product	Number of working channels	Measurement range	Measurement accuracy	Angle of view		Rotary speed	Angular resolution	
				Vertically	Horizontally		Vertically	Horizontally
VLP-16	16	Up to 100 m	±3 cm	30°	360°	5–20 hz	2°	0.1°

Each laser has its own angle of shift relative to the vertical axis. This arrangement of angles prevents interference of light and other interference [1–3]. Figure 1 and Table 2 present the values of degrees with the corresponding laser numbers.

Table 2. Numbering of laser beams with the corresponding direction.

Number of laser	Angle of direction shift, °	Number of laser	Angle of direction shift, °
0	-15	8	-7
1	1	9	9
2	-13	10	-5
3	3	11	11
4	-11	12	-3
5	5	13	13
6	-9	14	-1
7	7	15	15

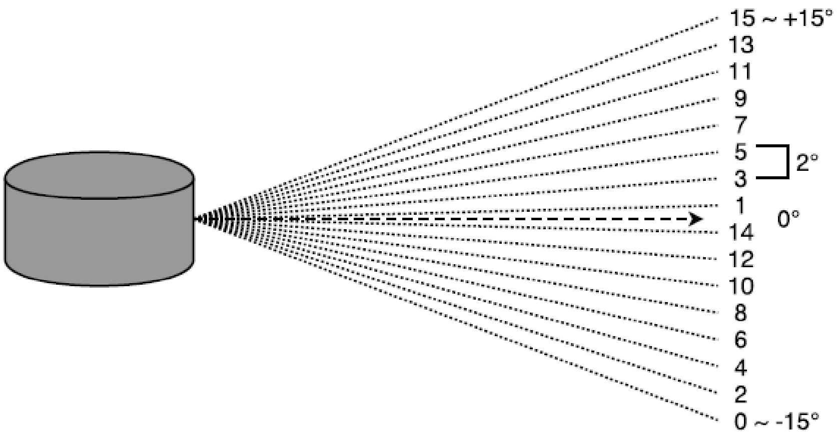

Figure 1. Vertical range of VLP-16.

2. Sensor data packeges

Data packages contain the following components:

1) 3D-data measured by the sensor;
2) the calibrated reflectivity of the surface from which the light pulse returned;
3) a set of azimuths;
4) 4 byte timestamp
5) 2 factory bytes identifying the sensor model and the mode of operation

The sensor transmits information about the distance relative to its position in spherical coordinates (r is the radius, ω is the height, α is the azimuth). Conversion from spherical to Cartesian coordinates is performed using the following formulas:

$$X = R * \cos(\omega) * \sin(\alpha)$$
$$Y = R * \cos(\omega) * \cos(\alpha) \quad (2)$$

IOP Conf. Series: Materials Science and Engineering **516** (2019) 012018 doi:10.1088/1757-899X/516/1/012018

$$Z = R * \sin(\omega).$$

Figure 2 show the relationship of the spherical coordinate system and the Cartesian coordinate system.

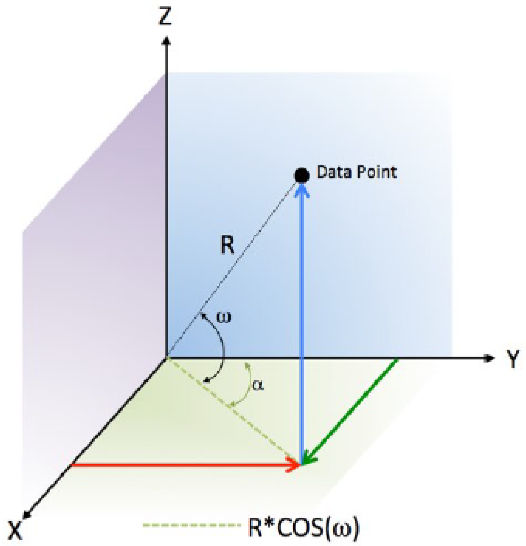

Figure 2. Spherical coordinates of the point.

Azimuth is 2-byte value; it is an angle in hundredths of a degree. The valid azimuth value is from 0 to 35999. One azimuth value is transmitted per data block.

A data block consists of 100 bytes of binary data:

- two-byte flag (0xFFEE);
- two-byte azimuth;
- 32 data points.

The result of measuring the distance with a single laser channel may be called as a data point out of a cloud of points. The data point is represented in a packet in three bytes — two bytes of distance and one byte of the calibrated reflectivity. The distance is an integer without a sign. Calibrated reflectivity is reported using a scale from 0 to 255.

3. Decoding the Velodyne LIDAR data packet

The data packet contains header files, data on distance and angle to the target, and information on the reflectivity; these data are transmitted via the UDP Ethernet protocol. There are three modes of data transfer: strongest, last, and dual [4, 5]. To implement the decoding method, we select the dual mode because in this case the data packets will receive information from two modes of operation at the same time. The data packet will consist of 1248 bytes.

Packet decoding operations can be divided into three parts: azimuth calculation (horizontal angle), distance calculation, and conversion to Cartesian coordinate systems.

After checking the flag file of the block (xFFEE), we can proceed to the calculation of the azimuth. First, the system receives two sequences of hexadecimal values. Second, we swap the bytes of the two hexadecimal values and combine them together into a two-byte value. Third, we convert the hexadecimal value to decimal and divide by 100, thereby obtaining the current value of the azimuth (horizontal angle) [6].

The first steps to calculate the distance are similar to the azimuth calculations. After receiving two sequences of hexadecimal values, we swap them and combine them into a two-byte value and convert from hexadecimal to decimal values. The final step is to multiply the decimal value by 2.0 mm. It is necessary to decode the distance value for each channel in order to build a cloud of points [7].

VLP-16 uses a spherical coordinate system; therefore, it is necessary to convert the values into the Cartesian system. To do this, we use expressions (2). The angle value ω (vertical angle) is a fixed value. Table 2 shows the angle vales according to the channel number.

The value of the angle α corresponds to the found value of the azimuth angle. The value of R also corresponds to the already found value of the distance. Thus, it is possible to decode data packets transmitted via the UDP protocol [8,9].

4. Results

Let us consider an example of data decoding.

Using the VLP-16 sensor, we scanned the room and saved the data in pcap format. Figures 3 and 4 show point cloud and data packets.

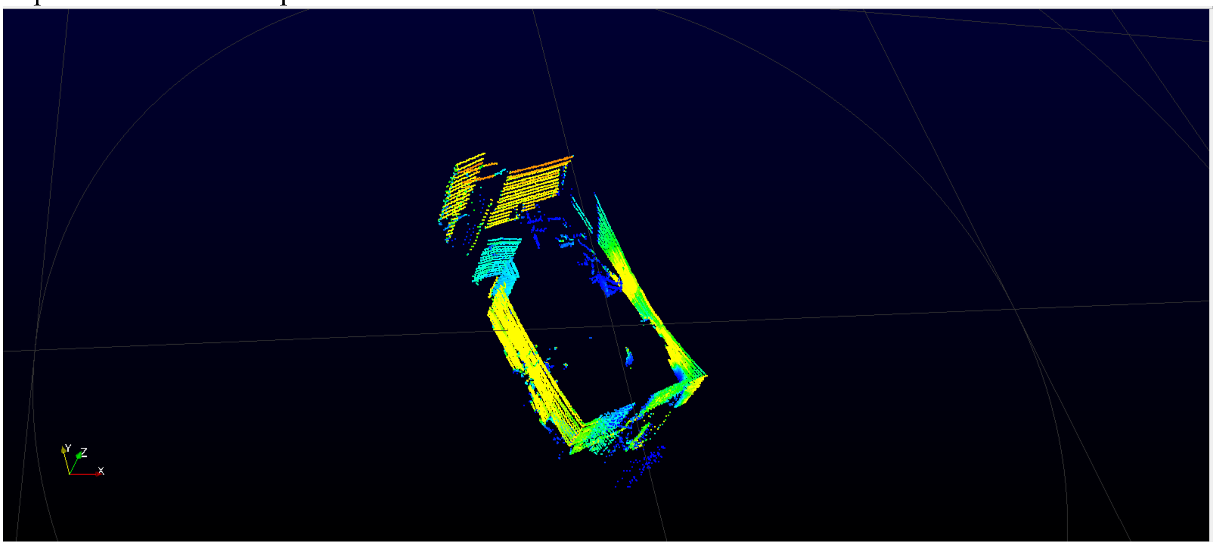

Figure 3. Point cloud received using VLP-16.

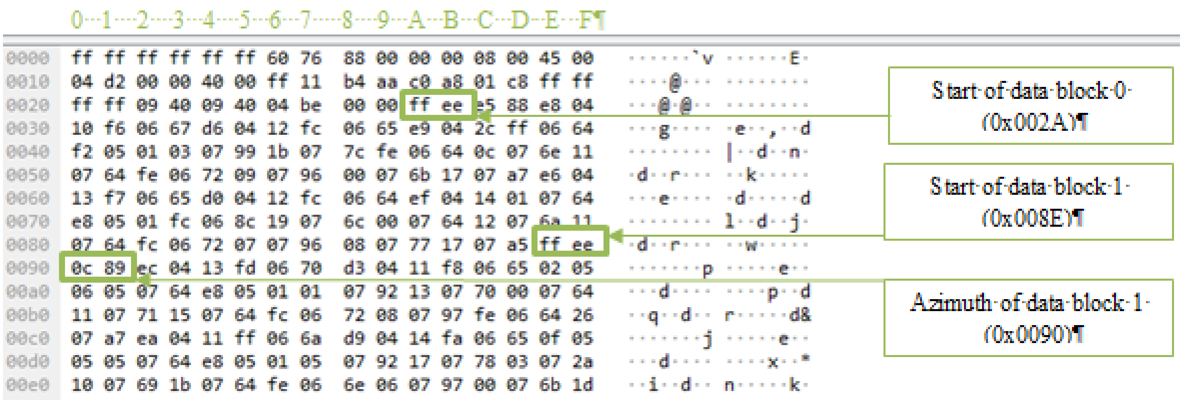

Figure 4. Data saved in pcap file.

Azimuth calculation:

-find azimuth values: 0×0C and 0×89 after flag-bytes 0×FF and 0×EE;
-swap azimuth values: 0×89 and 0×0C;
-unite bytes in one value: 0×890C
-convert the value to decimal: 35084;
-divide the resulting value by 100;
As a result, we get the value: 350.84°.

Calculation of the distance to the point:
-find the distance data: 0×EC and 0×04;
-swap the values found: 0×04 and 0×EC;
-unite data in one value: 0×04EC;
-convert the value to decimal: 1260;
-multiply the resulting value by 2 mm;
As a result, we get the value: 2520 mm or 2.52 m;
Reflectivity is equal to 0×13.
Conversion to Cartesian coordinate system:
-horizontal angle is found: $\alpha = 350.84°$;
-distance to the point is found: $R = 2.52$;
-using Table 2, we find the vertical angle corresponding to the 0th channel of the sensor: $\omega = -15°$;
-using formulas (2), we calculate values of coordinates XYZ:

$$X = R * \sin(\propto) * \cos(\omega) = -0.382$$
$$Y = R * \cos(\propto) * \cos(\omega) = 2.403$$
$$Z = R * \sin(\omega) = -0.652$$

5. Summary
As a result of the work, the space was scanned using the VLP-16 sensor and data were obtained in the form of a cloud of points and hexadecimal values containing information about this cloud of points. Using the developed algorithm, one can obtain the coordinates of each point at any time in the Cartesian coordinate system. In the results section, step-by-step calculation of the parameters of points for obtaining the final coordinates is shown. In further work, the device will be designed for decoding and converting LIDAR data in real time.

Acknowledgments
The research was carried out in Tomsk State University with the financial support of the Ministry of Education and Science of Russia, a unique project identifier: RFMEFI57817X0241.

The authors are grateful to Tatiana B. Rumyantseva fromTomsk State University for English language editing.

References
[1] Yu-Cheng Fan *et al.* 2018 *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)* 1–4 doi: 10.1109/I2MTC.2018.8409759
[2] Shashev D. *et al.* 2015 *IOP Conference Series: Materials Science and Engineering* **81** doi: 10.1088/1757-899X/81/1/012101
[3] Li-Juan Zheng *et al.* 2017 *IEEE International Conference on Consumer Electronics* 17062572 doi:10.1109/ICCE-China.2017.7990982
[4] Shikhman M., Shidlovskiy S. 2018 *IOP Conference Series: Materials Science and Engineering* **363** (1) doi: 10.1088/1757-899X/363/1/012019
[5] Velodyne LIDAR VLP-16 user manual. [Electronic resource]. URL: https://velodynelidar.com/downloads.html.
[6] Shashev D.V., Shidlovskiy S.V. 2017 *Journal of Physics: Conference Series* **881** doi:10.1088/1742-6596/881/1/012029
[7] Shidlovskiy S. 2016 *MATEC Web of Conferences* **79** N01014. doi: 10.1051/matecconf/20167901014
[8] Shidlovskii S. 2006 *Journal of Computer and Systems Sciences International* **45** (2) 282–286 doi: 10.1134/S1064230706020122
[9] Sukhanov D., Zavyalova K. 2016 *MATEC Web of Conferences* **79** 01040