Pointcloud with colors

(A project of the 2017 Robotics Course of the School of Information Science and Technology (SIST) of ShanghaiTech University)

Instructor: Prof. Sören Schwertfeger ·
https://robotics.shanghaitech.edu.cn/teaching/robotics2017

Chen Hongyu
School of Information Science and Technology(SIST), ShanghaiTech University

January 19, 2018

Abstract

Robots are typically equipped with multiple sensors,which require calibration in order to present sensed information in a common coordinate system [1]. In order to increase accuracy and robustness in state estimation for robotics, a growing number of applications rely on data from multiple complementary sensors. For the best performance in sensor fusion, these different sensors must be spatially and temporally registered with respect to each other. Most methods for state estimation that fuse data from multiple sensors assume and require that the timestamps of all measurements are accurately known with respect to a single clock. Consequently, the time synchronization of sensors is a crucial aspect of building a robotic system.

Our mapping is equipped with many kinds of sensors such as the cameras, velodyne laser sensor,2D laser sensors as well as the IMU. In order to use the data from theses sensors. The first thing is to calibrate all the sensors mounted on the car.

1 State of the Art

Developing autonomous systems that are able to assist humans in everyday tasks is one of the grand challenges in modern computer science [2]. One example are autonomous driving systems which can help decrease fatalities caused by traffic accidents. Today, visual recognition systems are still rarely em- ployed in robotics applications. Robotic platforms, both autonomous and remote controlled, use multiple sen- sors such as IMUs, multiple cameras and range sensors [3]. Each sensor provides data in a complementary modality. For instance, cameras provide rich color and feature information which can be used by state-of-the-art algorithms to detect objects of interest (pedestrians, cars, trees, etc.). Multiple sensors are employed to provide redundant information which reduces the chance of having erroneous measurements. In the above cases, it is essential to obtain data from various sensors with respect to a single frame of reference so that data can be fused and redundancy can be leveraged Range sensors have gained a lot of popularity recently despite being more expensive and also contain moving parts. These can provide rich structural information and if correspondence can be drawn between the camera and the LiDAR, when a pedestrian is detected in an image, it's exact 3D location can be estimated and be used by an autonomous car to avoid obstacles and prevent accidents. A variety of novel sensors have been used for object recognition in the last few years.However visual sensors have been used in visual sensors are rarely exploited in robotics applications: Autonomous driving systems rely mostly on GPS, laser range finders, radar as well as very accurate maps of the environment. Robots are typically equipped with multiple sensors,which require calibration in order to present sensed information in a common coordinate system [1]. There are a lot of work about how to calibrate the relationship between different sensors. Calibration is basic requirement in multi-sensor platforms where data needs to represented in a common system. All the camera sensors must be calibrated both intrinsically and extrinsically. As for the camera calibration ,there are a lot of work to calibrate the camera. Most of the work like Zhang [4]uses a chessboard to calibrate the camera. Different pose of the chessboard need to be captured. There are a lot of packages about how to calibrate the cameras. With the advent of autonomous vehicles,Lidar and camera have become an indispensable combination of sensors. They both provide rich data which can be used in a lot of applications. Colour points obtained from the camera and Lidar fusion provided to be useful in autonomous driving. A combination of the aligned lidar and camera also successful used in the process of building 3D geometry maps. Method of calibration camera and lidar can be divided into several groups. The first groups of methods required a chessboard marker for autonomous calibration. An automatic alignment of the camera with 2D laser range finder where typical checkerboard marker must be observed from multiple views was proposed by Zhang and Pless [5]. Another group of people they use some specific chessboard to calibrate the camera and Lidar [6]. Some of the recently used methods use a sequence of captured frames and compute the calibration parameters from the motion obtained through the feature tracking or special IMU units. Existing calibration toolboxes still require laborious effort from the operator in order to achieve reliable and accurate results. There are also some work about how to automatic robust calibration of the lidar and the camera.Abdallah Kassir [7] they proposed an algorithm to autonomous calibration the calibration of 2D lidar and the camera.For the camera calibration they use the algorithm which is presented by [4] to get the intrinsic of the camera. Then they use the Pless's [5] algorithm to calibrate the lidar and the camera.

2 System description

In this project we have to calibrate all the sensors on our mapping robot. It include calibrate the two cameras in front of our jackal robot. The stereo camera on the top of the top of the jackai robot. The relationship between the two velodyne laser sensors. Also we have to calibrate the relationship between the velodyne sensor and the top camera which on the top of jackal robot. Another thing we have to do is learn how to calibrate the 2D lidar sensor ,as well as the 2D lidar sensor and the camera. To summarize we have to calibrate all the sensors on the jackal robot.

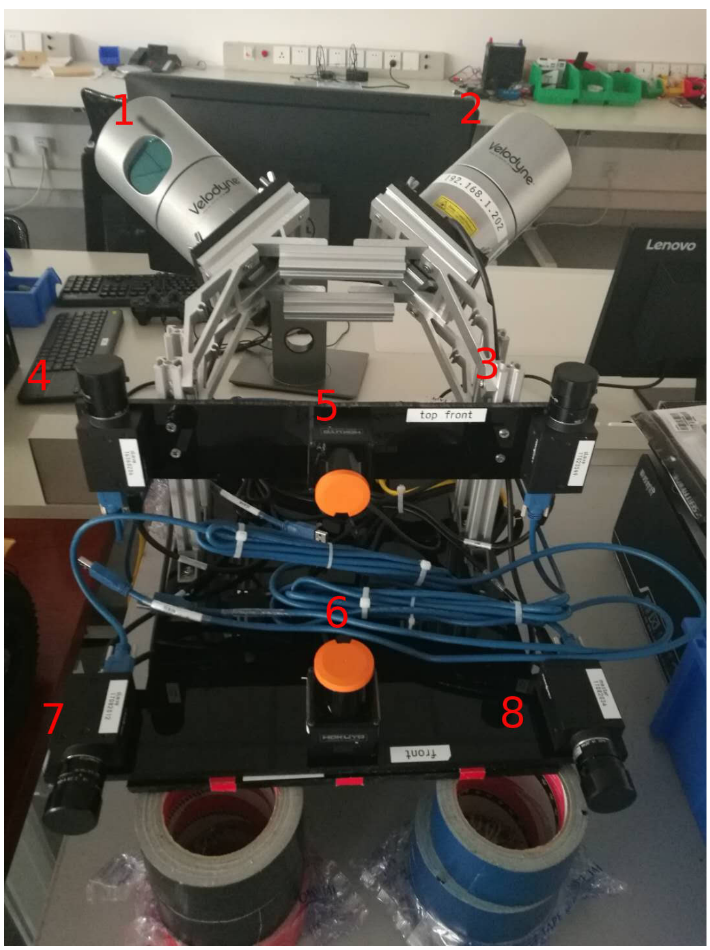

Figure 1: Mapping robot

2.1 Calibrate the stereo camera

The stereo camera we used for calibration is camera 7, 8. To automatically calibrate the intrinsic and extrinsic parameters of the cameras,we recorded different pose of the checkboard. Then we try to calibrate the relationship between them. The algorithm we used is from the Zhengyou Zhang [4]. All parameters are optimized by minimizing the average reprojection error.

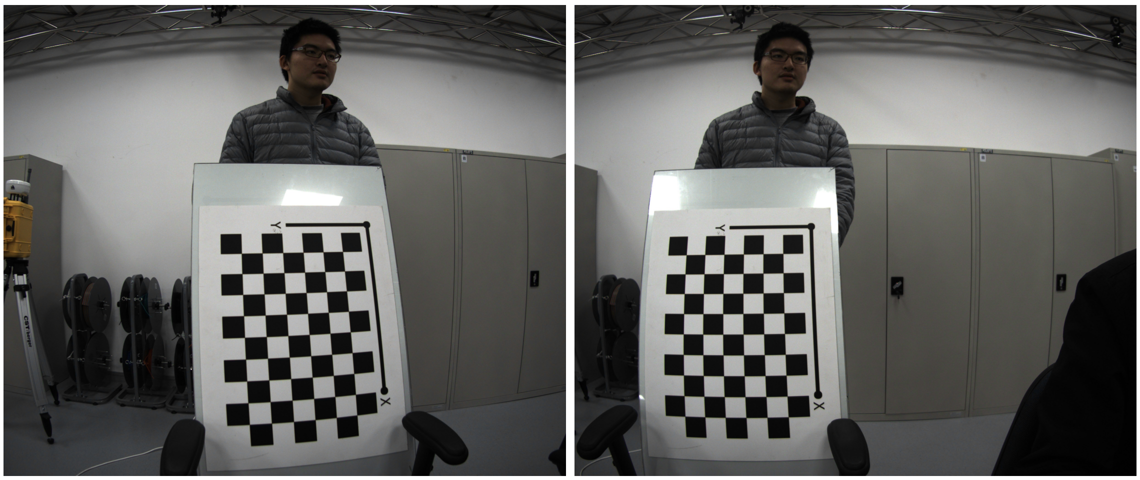

Figure 2: Origin image

Figure 3: Rectify image

Figure 4: Make the left and right image on the same canvas

In the figure 2 is the origin image we collect from the stereo camera 7 and 8. Figure 3 show the result we rectified after calibrated. Figure 4 shows the result we put the two image on same image. Through the green line we can see that the checkboard on the left and the right camera is nearly on the same line. The rotation and translation of the camera 7 with respect to 8 is represent by R and T. The value of R and T is as follows. The intrinsic of left camera 8 is $Rl_{intrinsic}$. The intrinsic of right camera 7 is $Rr_{intrinsic}$.

$$R = \begin{bmatrix} 0.9999920393994229 & 0.0002815963784309728 & 0.003980181059023304 \\ -0.0002699098116537817 & 0.9999956522778934 & -0.002936421973083051 \\ -0.003980990640095292 & 0.002935324307480607 & 0.9999877677175523 \end{bmatrix}$$

$$T = \begin{bmatrix} -0.400258 & -0.00128463 & 0.00534028 \end{bmatrix}$$

$$Rl_{intrinsic} = \begin{bmatrix} 1.7909326755548996e+03 & 0 & 1.2605775401905662e+03 \\ 0 & 1.7918350061251585e+03 & 1.0359958481597453e+03 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Rr_{intrinsic} = \begin{bmatrix} 1.8006493371498443e+03 & 0 & 1.2440324946403066e+03 \\ 0 & 1.7997226582209817e+03 & 1.0513552251972808e+03 \\ 0 & 0 & 1 \end{bmatrix}$$

This is used for vertical stereo. The rotation of camera 4 wrt camera 3 is R_v. The translation is T_v.

$$R_v = \begin{bmatrix} 0.9997879230797189 & -0.0172882810196332 & -0.01119036207315549 \\ 0.01722284482640407 & 0.9998341631943347 & -0.005917746663221277 \\ 0.01129081396657181 & 0.005723761776196699 & 0.999919874825529 \end{bmatrix}$$

$$T_v = \begin{bmatrix} 0.0221912 & -8.15886e-05 & -0.000116287 \end{bmatrix}$$

The intrinsic of camera 3 is $Rvl_{intrinsic}$. The intrinsic of camera 4 is $Rvr_{intrinsic}$.

$$Rvl_{intrinsic} = \begin{bmatrix} 1.7830786994500043e+03 & 0 & 1.2122057966389689e+03 \\ 0 & 1.7923701603935851e+03 & 1.0069665801222835e+03 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Rvr_{intrinsic} = \begin{bmatrix} 1.7926233095691284e+03 & 0 & 1.2168430330507983e+03 \\ 0 & 1.7983992967784427e+03 & 1.0150864159065204e+03 \\ 0 & 0 & 1 \end{bmatrix}$$

2.2 Calibrate the velodyne and camera

As shown in the figure 1 the laser sensor 2 and camera 3 is used for calibration. CalLibration of the LIDAR sensors with RGB camera finds its usage in many application fields from enhancing image classification to the environment perception and mapping. Fusion of the aligned camera and LIDAR sensor was recently used in many tasks of computer version in order to enhance their performance. Colour point clouds obtained though the camera-Lidar fusion proved to be useful for autonomous driving and vehicle tracking. Method of calibration with the lidar sensor can be devided into different groups. The first group requires a chessboard like marker for calibration. Also there are some group that the use a sequenced of captured frames to compute the paramaters from motion through feature tracking or special IMU units. There are alo some methods they use a very special maker to calibration the relation between the parameters [8].

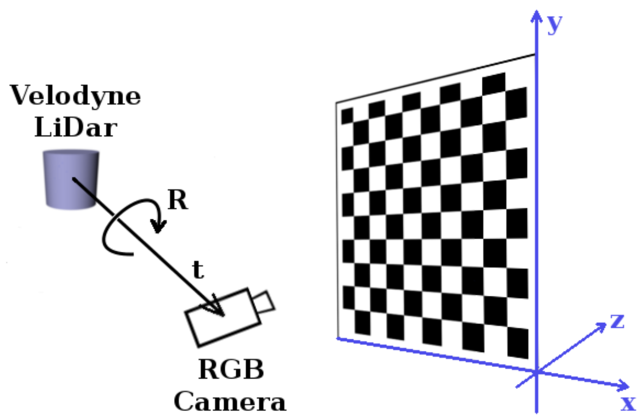

Figure 5: Problem of a camera-Velodyne calibration. The solution of the problem are the vectors t and R describing the translation and rotation of Velodyne Lidar related to the camera. The image is from [9]

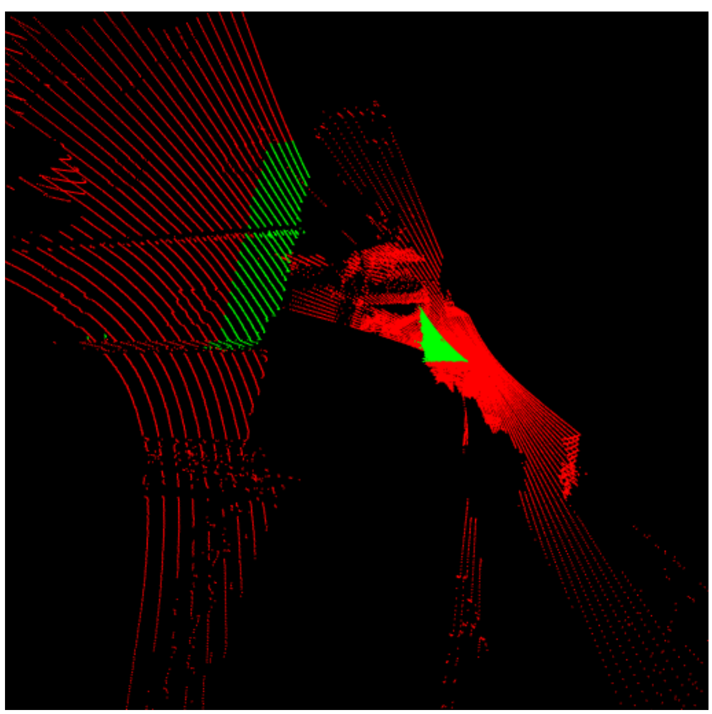

Figure 6: Pointcloud with colors

Because the velodyne points is very sparse. In order to make contrast, in the figure 6 all the green points belong in the camera view. All the red points are not in the camera view.

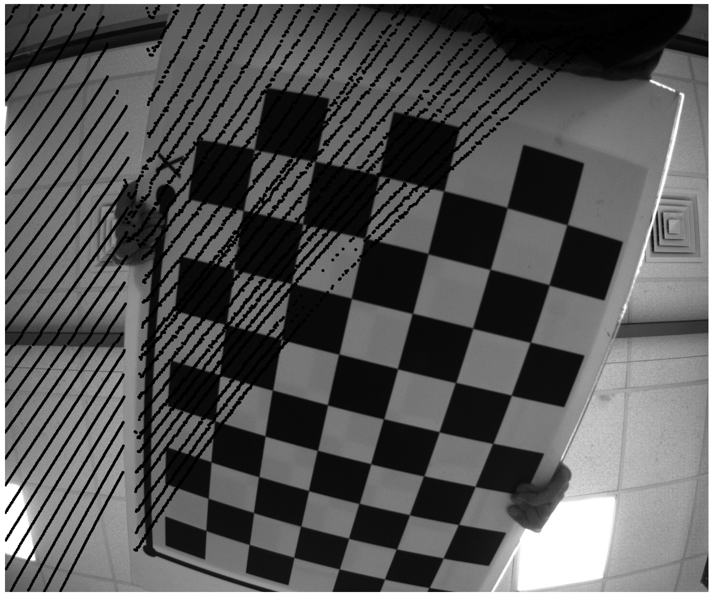

Figure 7: Project points to image

Figure 7 shows the result if we project all the points to the image. Only those points belongs to the image will be displayed on the image. The rotation and translation of the velodyne with respect to the camera is R and T.

$$R = \begin{bmatrix} -0.022018 & -0.688231 & 0.725157 \\ 0.685889 & 0.517317 & 0.511800 \\ -0.727373 & 0.508646 & 0.460660 \end{bmatrix}$$

$$T = \begin{bmatrix} -0.023458 & 0.195192 & 0.098256 \end{bmatrix}$$

2.3 Calibrate the two velodyne laser sensors

Lidars are one of the most important sensors in many robotic applications, mainly because of their accuracy and robustness of their measured distances. I am trying to use the Iterative Closest Point(ICP) [10] algorithm to calibrate the relationship between the two velodyne laser sensors which is mounted on the jackal robot. However I could not get the correct results of the two velodynes.

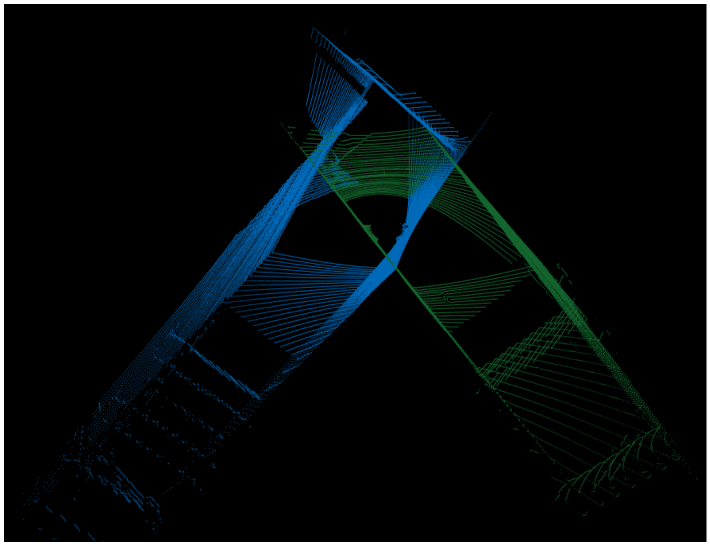

Figure 8: The origin points of the two velodynes

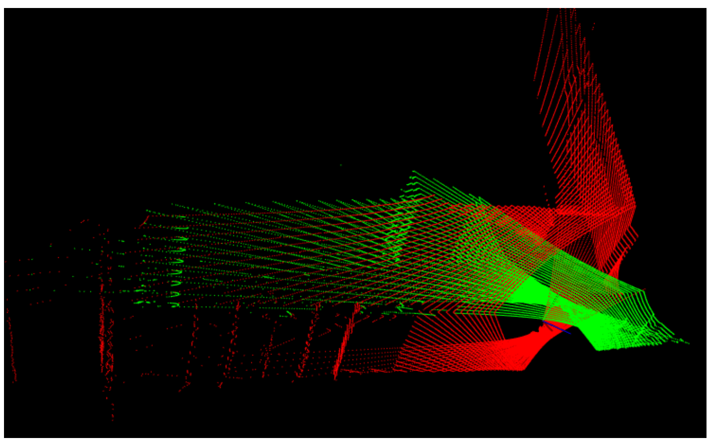

Figure 9: Calibration result of the two velodynes

Because the overlap of the two velodyne is very small. So the translation of the two velodynes does not correct. I need to record a new data set to calibrate the two velodynes. The algorithm I use to calibrate the two velodynes is NDT algorithm. The initial guess of the two velodynes is $Initial - guess$

$$R_{initial_guess} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$T_{initial_guess} = \begin{bmatrix} 0.2 & 0.37 & -0.32 \end{bmatrix}$$

2.4 Calibrate the 2D laser sensor and the camera

The 2D lidar 6 and the camera 8 is used for calibration. Camera and laser calibration is necessary for many robotics and computer vision applications. The paper present a robust and accurate method for the calibration of a perspective camera with a 2D laser range finder. Lidars are one of the most important sensors in many robotics applications because of their accuracy and robustness of their measured distances [5]. In order to effectively use the data from the camera and laser range finder, it is important to know their relative position and orientation from each other. The toolbox I used to calibrate the camera and the 2-D lidar is [5]. The flow chart of the algorithm is in the following chart.

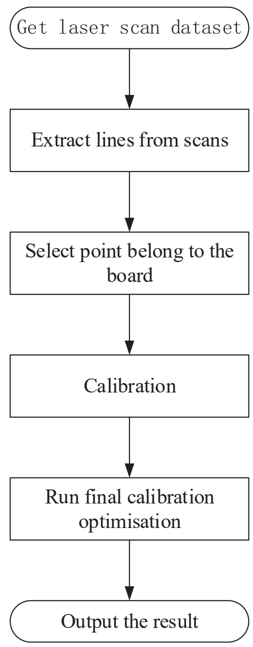

Figure 10: The flow chart of the calibration algorithm

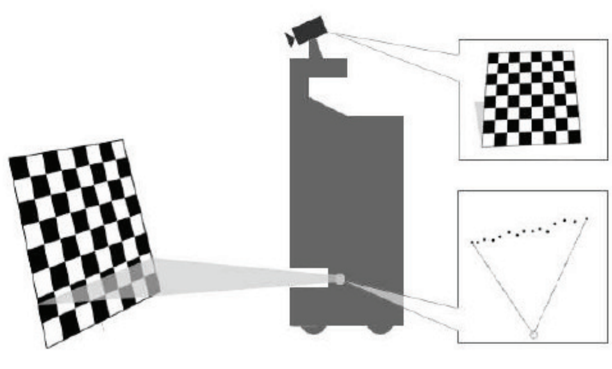

Figure 11: The flow chart of the calibration algorithm. The image is from [4]

From the image 10,11 ,we can know that we should manually selected the points on the checkerboard. Also another important things is that the camera should be calibrated. This requires us to know the camera's internal parameter matrix.

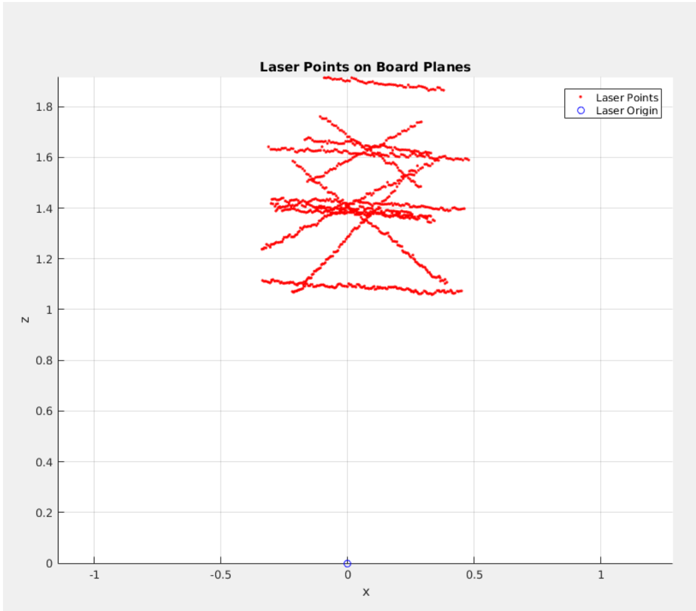

Figure 12: Different pose of the laser points selected to calibrate the laser point.

After calibration we project the lidar points on to the image to see the calibration result.

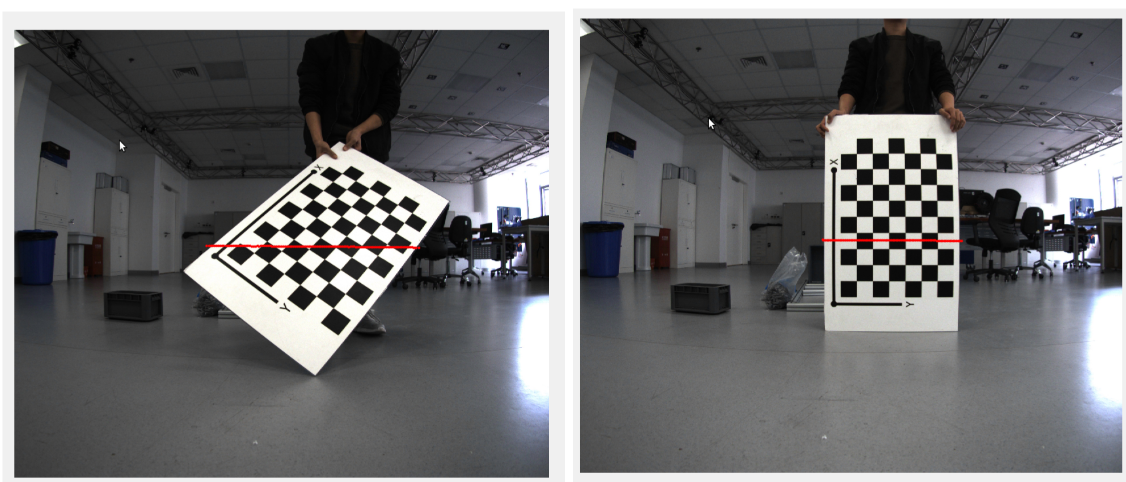

Figure 13: Project laser points to image

The image 13 shows that when we want to project the laser points onto the image. From the image we can see that we take different pose of the image then we project the laser points on to the image. The red points represent the laser points.

The rotation of the laser with respect to the camera is matrix R. The translation is vector T.

$$R = \begin{bmatrix} 0.9986 & 0.0052 & 0.0527 \\ -0.0053 & 1.000 & 0.0028 \\ -0.0526 & -0.0031 & 0.9986 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.177 & -0.037 & -0.0589 \end{bmatrix}$$

The rms error is 0.00523. the relationship of the camera 12 with respect to the honrizontal lidar

$$R = \begin{bmatrix} 0.9982 & -0.0429 & 0.0411 \\ 0.0419 & 0.9988 & 0.0247 \\ -0.0421 & -0.023 & 0.9988 \end{bmatrix}$$

$$T = \begin{bmatrix} -0.168 & 0.01 & -0.0533 \end{bmatrix}$$

The rms error is 0.00675.

2.5 Calibrating the robot IMU

there are several works about how to calibrate the camera and the IMU. The toolbox we used to calibrate the IMU and the camera is [11],which is published by ETH.

Figure 14: The camera and the IMU we used to calibrate their relationship

We mount a pointgray camera an Xsense IMU on the board to calibrate them. The rotation and translation of the IMU with respect to camera is in the following. The square size of the checkboard we use is 0.065 m.

$$R = \begin{bmatrix} 0.01194582 & -0.99644329 & -0.08341498 \\ -0.05056707 & 0.0827122 & -0.99528974 \\ 0.99864922 & 0.01610761 & -0.04939915 \end{bmatrix}$$

$$T = \begin{bmatrix} 0.10103464 & 0.0167043 & 0.05343886 \end{bmatrix}$$

3 System evaluation

In the previous section ,we have known the relation between the camera 7 and camera 8. Also through calibration of the camera and the 2D lidar. We can known the relation between camera 7 wrt lidar 6,as well as the relation of 6 wrt to camera 8. we try to analysis the error.

where R represent the rotation of camera 7 wrt camera 8. R_2 represent the rotation of lidar 6 wrt camera 7. R_1 represent the rotation of lidar 6 wrt camera 8.

$$R = \begin{bmatrix} 0.9999920393994229 & 0.0002815963784309728 & 0.003980181059023304 \\ -0.0002699098116537817 & 0.9999956522778934 & -0.002936421973083051 \\ -0.003980990640095292 & 0.002935324307480607 & 0.9999877677175523 \end{bmatrix}$$

$$R_1 = \begin{bmatrix} 0.9986 & 0.0052 & 0.0527 \\ -0.0053 & 1.000 & 0.0028 \\ -0.0526 & -0.0031 & 0.9986 \end{bmatrix}$$

$$R_2 = \begin{bmatrix} 0.9982 & -0.0429 & 0.0411 \\ 0.0419 & 0.9988 & 0.0247 \\ -0.0421 & -0.023 & 0.9988 \end{bmatrix}$$

We try to turn the rotation matrix R into axis–angle representation (R_{axis}, R_{angle}).

$$R_{axis} = \begin{bmatrix} 0.2557 & 0.3491 & -0.0241 \end{bmatrix}$$

$$R_{angle} = 0.0114$$

Similarly we try to turn the rotation matrix $R_1 * R_2^{-1}$ into axis–angle representation $(R_{axis_1}, R_{angle_1})$.

$$R_{axis_1} = \begin{bmatrix} 0.4345 & 0.2245 & -0.8674 \end{bmatrix}$$

$$R_{angle_1} = 0.0493$$

We try to normalize the R_{axis} and R_{axis_1} using the following formula $n_{normlize} = \frac{n}{n^T n} * angle$. Then we can get the normalized angle .

$$R_{angle} = \begin{bmatrix} 0.0155 & 0.0212 & -0.0055 \end{bmatrix}$$

$$R_{angle_1} = \begin{bmatrix} 0.0216 & 0.0111 & -0.0430 \end{bmatrix}$$

$$\delta = R_{angle} - R_{angle_1} = \begin{bmatrix} -0.0060 & 0.0101 & 0.0416 \end{bmatrix}$$

where δ means that the angle difference in the different axis.

4 Conclusions:

If we want to fuse al the data ,the first thing is to calibrate all the sensors on the robot. So the main goal of this project is to calibrate all the sensors mounted on the jackal robot. In this project I have calibrated most of the sensors. Use the same way ,we can calibrate all the sensors on the mapping robot. This is an very important step for fusing all the sensors' data on the mapping robot. Also we can use optimize method to optimize the relationships between two sensors. Also there are still a lot of work to do. For example, all the calibration are calibrated offline which needs a lot of time to calibrate all the sensors.

References

[1] A. Geiger, F. Moosmann, O. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," pp. 3936–3943, 2012.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," pp. 3354–3361, 2012.

[3] A. Dhall, K. Chelani, V. Radhakrishnan, and K. M. Krishna, "Lidar-camera calibration using 3d-3d point correspondences," 2017.

[4] Z. Zhang, "A flexible new technique for camera calibration," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pp. 1330–1334, 2000.

[5] P. R. Zhang Q, "Extrinsic calibration of a camera and laser range finder (improves camera calibration)," intelligent robots and systems, pp. 2301–2306, 2004.

[6] Z. M. A. H. M. Velas, M. Spanel, "Calibration of rgb camera with velodyne lidar)," international conference on robotics and automation, 2014.

[7] A. Kassir and T. Peynot, "Reliable automatic camera-laser calibration," 2010.

[8] Z. M. Martin Velas, Michal Spanel, "Calibration of rgb camera with velodyne lidar," pp. 3936–3943, 2012.

[9] http://wiki.ros.org/but_calibration_camera_velodyne.

[10] http://pointclouds.org/documentation/tutorials/interactive_icp.php/.

[11] https://github.com/ethz-asl/kalibr/.