A Deep-learning based Multi-modality Sensor Calibration Method for USV

Hao Liu[1], Yingjian Liu[1], Xiaoyan Gu[2,*], Yingying Wu[1], Fangchao Qu[1], Lei Huang[1]

[1]*Department of Computer Science and Technology*
Ocean University of China
Qingdao, China

[2,*] *Institute of Information Engineering*
Chinese Academy of Sciences
Beijing, China

liu.hao@ouc.edu.cn, 550758467@qq.com, guxiaoyan@iie.ac.cn, 1059951836@qq.com, 1151958516@qq.com, huangl@ouc.edu.cn

***Abstract*—The automatic obstacle avoidance and other tasks of the unmanned surface vehicle rely on the fusion of multi-modality onboard sensors. The accurate calibration method is the foundation of sensor fusion. This paper proposes an online calibration method based on the deep learning for visual sensor and depth sensor. Through an end-to-end network, we combine feature extraction, feature matching and global optimization process of sensor calibration. After initial training, the network can continuously calibrate multi-modality sensors. It solves the USV calibration challenges under complex operating environment. In the simulation environment and realistic environment, we conducted a fast online calibration of the camera, LIDAR and depth camera, which showed the effectiveness of the algorithm.**

***Index Terms*—multi-modality sensor, calibration, USV, deep learning**

I. INTRODUCTION

USV(Unmanned Surface Vehicle) is one of the most important marine monitoring platforms. For the past few years, automatic obstacle avoidance is increasingly a key research field for the unmanned driving system. This task depends on the fusion of a series of sensors such as camera, LIDAR, radar and so on. Calibration of the multi-modality sensor is a foundation for sensor fusion. Because it is different from the traditional camera calibraation, calibration of the USV sensor will encounter some unique challenges.

- Feature extraction of multi-modality sensors. The sensing subsystem of a USV usually consists of cameras, LIDAR, radar, sonar and other sensors. According to the output data format, there are two categories of these sensors: vision sensors and depth sensors. A calibration algorithm usually requires a batch of the corresponding feature points in camera views. However, conventional handcrafted features (SIFT, image edges, etc.) for camera calibration is not suitable for building the corresponding pattern between image and depth.
- Online calibration. The operation condition of a USV is relatively complicated and usually unstable for calibration. Traditional calibration methods rely on some particular targets of calibration such as a checkerboard panel. Obviously, those offline methods require unrealistic human interaction for USV operation. A suitable method should calibrate based on matching patterns in an unstructured environment.
- Continuous calibration. Since the extremely changing temperature and the violent vibration, the onboard sensors' extrinsic parameters may change. Therefore, the recalibration is necessary to constantly update these extrinsic parameters.

In this paper, we propose a multi-modality sensor calibration method based on deep learning for USV. It realizes a continuous online calibration for different types of onboard sensors and only requires an initial calibration for training. It meets the requirements of USV operation environment.

II. RELATED WORK

Early USVs were manually remote-controlled, and these sensors were only used to support manual observations[1]–[3]. In recent years, a large number of USVs have been equipped with automatic obstacle avoidance such as in an ideal environment the USVs can be driven automatically by GPS or in complex environments often rely on point cloud information formed by radar or other depth sensors[4]–[7]. However, standalone depth sensors cannot distinguish the specific properties of objects[8]. In order to accomplish complex tasks, visual sensors are also widely equipped on modern USVs[9], but traditional USV does not effectively fuse data from multi-modality sensors..

Accurate calibration of sensors is the basis of sensor fusion. According to a batch of letters, camera calibration is the process of determining the internal and external parameters of the camera[10], [11]. The traditional camera calibration method establish the constraints of the camera model parameters by the through the optimization algorithm. Typical representatives include DLT method, nonlinear optimization method, etc[12], [13]. Although the traditional methods have achieved outstanding results[14], they usually require a relatively stable calibration environment for the sensors and apparently are not suitable for the USV operation[15]. F. M. Mirzaei et al. and S. Bileschi and G. Pandey et al. obtained high precision camera and laser radar calibration through convert problem to the minimum solution conditions or inertial measurement unit(IMU) information or mutual information[16]–[18]. J. levinson and s.Thrun used the edge information of the image to match the

This work was financially supported by The Aoshan Innovation Project in Science and Technology of Qingdao National Laboratory for Marine Science and Technolo-gy (No.2016ASKJ07), Key R&D plan of Shandong province (2016ZDJS09A01) and Qingdao Science and technology plan (17-1-1-3-jch).

978-1-5386-5321-0/18/$31.00 ©2018 IEEE

point cloud[19]. However, because the traditional self-calibration method relies on manually defined features that require rich details in the scene, it is not suitable for environments with less recognition features and greater interference.

In recent years, deep learning achieves great improvements in many computer vision problems including calibration[20], [21]. A. Kendall et al. proposed six parameters of the camera (6 DOF) that are recovered from the single frame by improving the GoogLeNet network[22]. I. Melekhov et al. estimated the translation and rotation between two cameras using the convolutional neural network[23]. M. Giering et al. used multi-channel CNN to achieve the calibration between radar and video [24]. In general, deep learning performs better on feature extraction and matching stages for calibration.

In this paper, we propose an end-to-end CNN that combine all three procedures of sensor calibration. In our dataset, the network accepted an RGB image, a front view of LIDAR depth points and a depth image from a stereo camera. We test the network in both simulation and real environment.

III. METHOD

A. Sensors

On our experimental USV platform, there are three types of sensors: camera, LIDAR and depth camera. Respectively, they generate RGB image, point cloud and depth map.

The camera is a MindVision industrial camera. And the LIDAR model is VLP-16 which generates 300, 000 points per second. We crop the front side of the scene and project the points to front view. The depth camera is a DJI Guidance. It originally outputs five depth channels and we choose the front channel. The specification of sensors are listed in TABLE I.

TABLE I. THE SPECIFICATION OF SENSORS

Sensor	Specification	*Value*
Camera	Focal length	4mm
	CMOS size	5.86*5.86μm
	Resolution	1920×1200
LIDAR	Scan rate	5—20Hz
	Vertical scan angle	360°
	Horizontal scan angle	-15° - +15°
	Angular resolution	0.1° - 0.4°
	Scan lines	16
Depth Camera	Focal length	2.6mm
	Observation range	0.2m-20m
	Resolution	320×240

B. Geometric Model

1) RGB Camera: For the regular onboard camera, we use pinhole camera model[25]. This basically projects a 3D point in real world to a corresponding 2D point on the image plane. By establishing a certain mathematical relationship in this process of camera calibration, the image plane and the world coordinate system are correspondingly established.

2) LIDAR: The VLP-16 LIDAR represents points about spherical coordinates(γ, ω, α). Before further processing steps, we converte them to Cartesian coordinate system ignoring one meter away points. With the information of the vertical (elevation) angle (α) and the horizontal angle (azimuth) angle (ω), the X, Y, and Z coordinates of each measurement points can be calculated. The conversion is shown inFig. 1. After the conversion, we remove noise points and intercept the point cloud to fit the RGB image view.

3) Depth camera: The depth camera estimates the depth based on the principle of stereo vision. It generates a dense depth map in short range. This property ideally offsets the limitation of LIDAR. The rows and the columns of the depth map can correspond to one of the angles, say ω, α in Fig. 1.

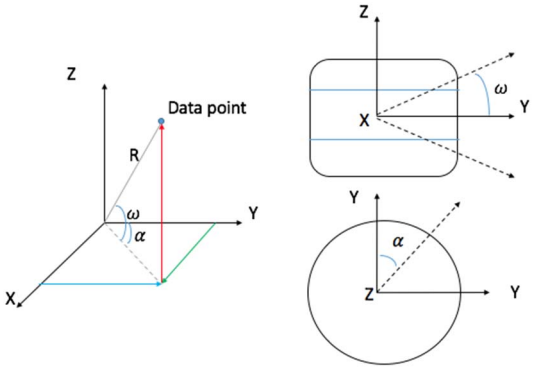

Fig. 1. The conversion of LIDAR

C. Network structure

We have made uniform requirements for the format and size of the input data. And the world coordinate system of the camera is selected as the origin coordinate system of the three sensors to establish the correspondence between the three sensors. Meanwhile, we consider the onboard radar as a 2D LIDAR. Regarding sizes of three datas determines the smallest depth map of the image, with a size of 320*240.

The entire calibration process is divided into two phases. In the first stage, we calibrate two common depth sensors to fuse all point clouds. And in the second stage, we calibrate the camera and the virtual depth sensor. each phase bases on the method from N. Schneider et al[26].

The fusion of LIDAR and depth image. Firstly, the LIDAR is mapped to a depth image plane using an $\boldsymbol{H_{init1}}$, and the two inputs are respectively subjected to feature extraction, feature matching and global regression through the fusion layer. The output is converted to $\boldsymbol{\theta_1}$ for rotation and translation matrix. The matrix $\boldsymbol{\theta}$ is composed of a 3*3 rotation matrix $\boldsymbol{R}$ and a 3*1 translation vector $\boldsymbol{t}$:

$$\theta = \begin{bmatrix} R & t \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{1}$$

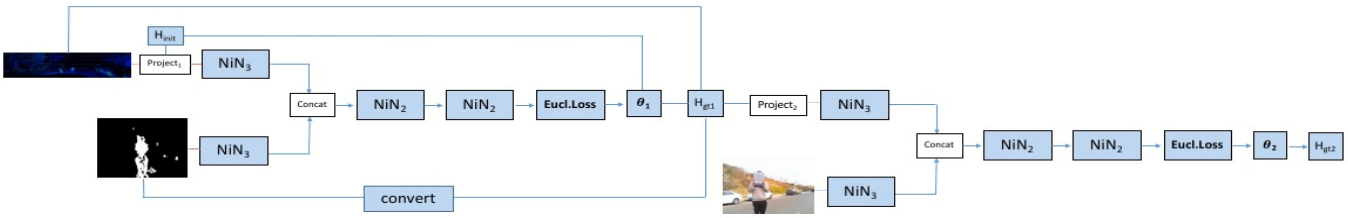

Fig. 2. The diagram of network structure

The fusion of point cloud and RGB image. Point clouds from LIDAR and the depth images in the first stage are merged to one single point clouds. Here, depth images need to be converted into point cloud data. After that, we consider this entire point cloud is generated by one virtual depth sensor. We take it and RGB as the input data in the second stage. Once again, the new point cloud data by mapping $\boldsymbol{H_{init2}}$ to the RGB image, extraction, matching and global regression. Finally, we can get the output parameter $\boldsymbol{\theta_2}$.In order to reduce the parameters, we use a quaternion to represent the rotation matrix. The network will output seven parameters to represent the calibration result.

In our network, the feature extraction and the feature matching phase are implemented with NiN blocks. The first stage is the step of parallelizing the two inputs in order to extract the parts that we are interested in. In the feature matching phase, we merge the feature maps through the association layer[27]. The association layer is based on the method proposed by A. Dosovitskiy et al[20]. Global regression, which separates the rotation matrix from the panning matrix through a fully connected layer, resulting as the final result. *Fig. 1.* shows the network structure and channel used in our paper.

IV. EXPERIMENTS

A. Dataset

We use 3ds Max to generate the simulation data needed for the test, and use the built-in MAXScript script to directly generate the extrinsic parameters which regards as ground truth. The real world dataset comes from a series of real environments that we collect ourselves. All the sensors are mounted to a frame. We check the result by the calibration method of F. M. Mirzaei[16].

B. Simulation

We constructed a series of simulation scenes by using 3ds Max. One scene consists of two cameras, a dummy object that is used as LIDAR center and several objects. The render output of one camera is RGB image and that from the other one is depth image. The depth image is generated by configuring render e We implement LIDAR by a MAXScript program that records the nearest intersection of the ray and object surface. Several virtual rays are beamed from the dummy object and scan the scene. This is a direct simulation of a real LIDAR. shown in Fig. 3.

C. Result

In order to verify our method, we mainly use the extrinsic parameters derived from the simulation scenario as ground truth and compare the translation and the rotation MAE respectively. The primitive rotation parameters are represented by quaternions and we convert them to euler angles. All the results are listed in TABLE II.

In addition, we fuse the three sensor data of the real world scenes to visually display the results of the calibration, as shown in Fig. 4 and Fig. 5.

The USV application has strict performance requirements for the algorithm. In our experiment, the algorithm needs 10ms to process one frame. The maximum design speed of our USV is 25 knots, and the performance of the algorithm can meet the driving safety requirements of the USV.

V. CONCLUSION

We proposed a calibration method for onboard multi-modality sensors of USV. An end-to-end deep learning network combines feature extraction, matching and global optimization procedures of calibration. The method is applied to our experimental USV platform that equipped with camera, LIDAR and depth camera. The experiment shows the accuracy and the performance of the method meets the requirement of USV operation.

Currently, our method requires an initial calibration for training. We will introduce other sensor data such as the accelerometer or gyro data to obtain the scene structure directly. Meanwhile, the network is not flexible for different sensor configurations. To overcome this, we will update the network structure.

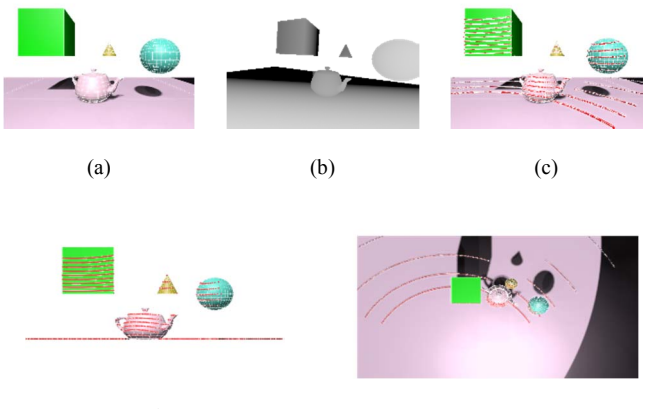

Fig. 3. The simulation scene. (a) RGB image. (b) Depth image. (c) Point cloud visualization from RGB camera. (d) The front view of the scene. (e) The top view of the scene.

TABLE II. MAE OF EXTRINSIC PARAMETERS

Dataset	*Translation Error in m*			*Rotation Error In°*		
	X	***Y***	***Z***	***Yaw***	***Pitch***	***Roll***
Simulation data	0.15	0.17	0.1	0.5	0.9	0.2
Real world data	0.3	0.23	0.15	0.4	0.78	0.1

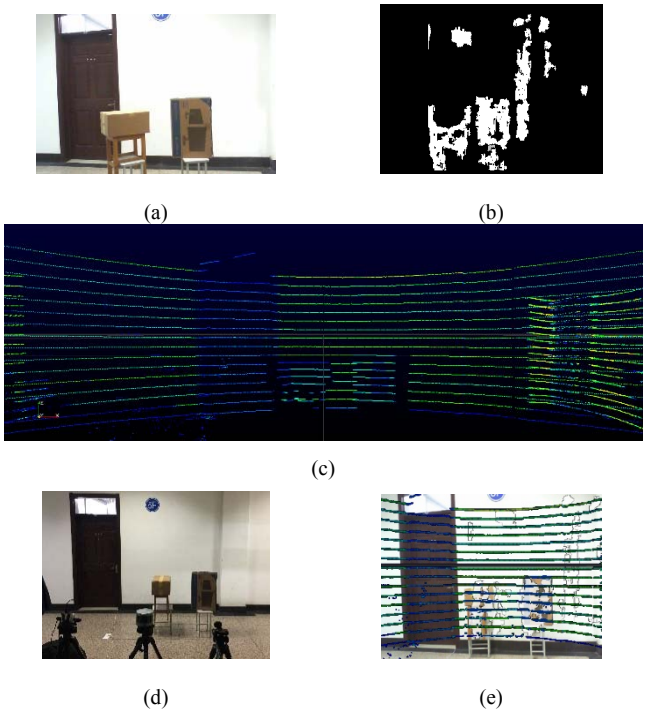

Fig. 4. The result of the calibration (a) RGB image. (b) Depth image. There are lots of fragments. (c) Point cloud from LIDAR. We retained a wider view. (d) Layout of sensors. (e) Fusion result.

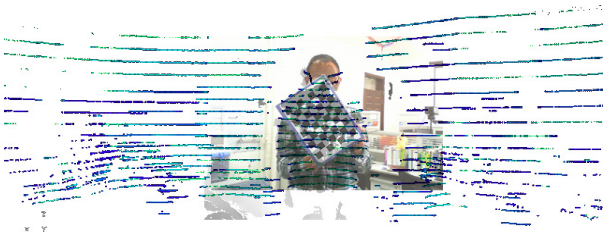

Fig. 5. The fusion result of the calibration II. Indoor background generates a brighter fusion data. The chessboard is utilized for generating a ground truth.

ACKNOWLEDGMENT

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. The authors thanks all anonymous reviewers for the valuable comments and suggestions.

REFERENCES

[1] R. Sutton, Ed., *Advances in Unmanned Marine Vehicles*. Institution of Engineering and Technology, 2006.

[2] J. Dufek and R. Murphy, "Visual pose estimation of USV from UAV to assist drowning victims recovery," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 147–153.

[3] X. Xiao, J. Dufek, T. Woodbury, and R. Murphy, "UAV assisted USV visual navigation for marine mass casualty incident response," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 6105–6110.

[4] A. Watanabe, M. Kuri, and K. Nagatani, "Field report: Autonomous lake bed depth mapping by a portable semi-submersible USV at Mt. Zao Okama Crater lake," in *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2016, pp. 214–219.

[5] E. Simetti, G. Casalino, A. Turetta, E. Storti, and M. Cresta, "Towards the Use of a Team of USVs for Civilian Harbour Protection: USV Interception of Detected Menaces," *IFAC Proc. Vol.*, vol. 43, no. 16, pp. 145–150, 2010.

[6] J. Grenestedt, J. Keller, S. Larson, J. Patterson, J. Spletzer, and T. Trephan, "LORCA: A high performance USV with applications to surveillance and monitoring," in *2015 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2015, pp. 1–6.

[7] I. Ohya, A. Kosaka, and A. Kak, "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing," *IEEE Trans. Robot. Autom.*, vol. 14, no. 6, pp. 969–978, 1998.

[8] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-D lidar," in *2015 IEEE International Conference on Information and Automation*, 2015, pp. 1648–1653.

[9] Z. Han, Q. Ye, and J. Jiao, "Combined feature evaluation for adaptive visual object tracking," *Comput. Vis. Image Underst.*, vol. 115, no. 1, pp. 69–80, 2011.

[10] W. Qi, F. Li, and L. Zhenzhong, "Review on camera calibration," in *2010 Chinese Control and Decision Conference*, 2010, pp. 3354–3358.

[11] Y. Furukawa and C. Hernández, "Multi-View Stereo: A Tutorial," *Found. Trends® Comput. Graph. Vis.*, vol. 9, no. 1–2, pp. 1–148, 2015.

[12] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, 2000.

[13] Z. Hu and F.-C. Wu, "A Review on some active vision based camera calibration techniques," *Chin. J. Comput.*, vol. 25, no. 11, pp. 1149–1156, 2002.

[14] Z. Pusztai and L. Hajder, "Accurate Calibration of LiDAR-Camera Systems Using Ordinary Boxes," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 394–402.

[15] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 1, p. 16, 2017.

[16] F. M. Mirzaei, D. G. Kottas, and S. I. Roumeliotis, "3D LIDAR–camera intrinsic and extrinsic calibration: Identifiability and analytical least-squares-based initialization," *Int. J. Robot. Res.*, vol. 31, no. 4, pp. 452–467, 2012.

[17] S. Bileschi, "Fully automatic calibration of LIDAR and video streams from a vehicle," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, 2009, pp. 1457–1464.

[18] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, "Automatic Extrinsic Calibration of Vision and Lidar by Maximizing Mutual Information," *J. Field Robot.*, vol. 32, no. 5, pp. 696–722.

[19] J. Levinson and S. Thrun, "Automatic Online Calibration of Cameras and Lasers," in *Proceedings of Robotics: Science and Systems*, Berlin, Germany, 2013.

[20] A. Dosovitskiy *et al.*, "FlowNet: Learning Optical Flow with Convolutional Networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2758–2766.

[21] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1647–1655.

[22] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 2938–2946.

[23] I. Melekhov, J. Ylioinas, J. Kannala, and E. Rahtu, "Relative Camera Pose Estimation Using Convolutional Neural Networks," in *Advanced Concepts for Intelligent Vision Systems*, 2017, pp. 675–687.

[24] M. Giering, V. Venugopalan, and K. Reddy, "Multi-modal sensor registration for vehicle perception via deep neural networks," in *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, 2015, pp. 1–6.

[25] G. Xu and Z. Zhang, *Epipolar Geometry in Stereo, Motion and Object Recognition: A Unified Approach*. Springer Netherlands, 1996.

[26] N. Schneider, F. Piewak, C. Stiller, and U. Franke, "RegNet: Multimodal sensor registration using deep neural networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, pp. 1803–1810.

[27] M. Lin, Q. Chen, and S. Yan, "Network In Network," 2013.