一般社団法人 人工知能学会
Japanese Society for
Artificial Intelligence

人工知能学会研究会資料
JSAI Technical Report
SIG-Challenge-055-4
(11/22)

マイクロフォンアレイおよびデプスセンサーのオンラインキャリブレーションに関する考察

Online calibration of microphone array and depth sensors

劉 超然[1*]　石井 カルロス[1]
Chaoran Liu[1]　Carlos Ishi[1]

[1] 国際電気通信基礎技術研究所 石黒浩特別研究所
[1] ATR Hiroshi Ishiguro Laboratories

Abstract: RGB-D sensor and microphone array are widly used for providing an instantaneous representation of the current visual and auditory environment. Sensor pose is needed for sharing and combining sensing results together. However, manual calibration of different type of sensors is tedious and time consuming. In this paper, we propose an online calibration framework that can estimate sensors' 3D pose and works with RGB-D sensor and microphone array. In the proposed framework, the calibration problem is described as a factor graph inference problem and solved with a Graph Neural Network (GNN). Instead of frequently used visual markers, we use multiple moving people as reference objects to achieve automatic calibration.

1 Introduction

In a sensor network, the observations from each sensor are measured on sensor's own 3D coordinate and need to be combined together to yield a collective observation. This process requires each sensor's 3D pose in the world coordinate to conduct the coordinate conversion. The calibration of sensors is a crucial but often tedious problem in a sensor network. Especially when the network includes different type of sensors, the observations themself are difficult to be used as references. This fact motivate us to propose a calibration framework that is able to calibrate different type of sensors without human intervention.

Calibration of cameras and RGB-D sensors is a well-studied topic. For cameras, researchers used wand [1], plane [2] or orthogonal planes [3] as calibration objects to calculate intrinsic and extrinsic parameters. Regarding RGB-D sensors, point cloud of a calibration plane was used to generate virtual points and calibrate sensors [4]. Conventional calibration object such as checkerboard is also used for calibrating multiple RGB-D sensors [5]. Other than calibration objects, human bodies are also used in calibration process. In [6], skeleton-based viewpoint invariant transformation (SVIT) is proposed to derive the transformation from human body to RGB-D sensor. A commonly observed human body (skeleton) by two neighboring sensors is used to calculate the relative position and orientation between two sensors. Similarly, an algorithm that calibrate and automatically re-calibrate RBG-D sensors using joints of observed skeleton is proposed in [7].

Microphone arrays are widely used for auditory environment sensing and improving robot audition [8, 9]. In [10], 3 Ω D sound maps are created by a moving 3D microphone array taking into account the prior probability of sound emitting. In [11], multiple calibrated microphone arrays are used to reproduce and/or manipulate auditory environment for people at a remote location. Multiple 3D microphone arrays are also used for hearing support system with the ability of emphasizing the target sound and depressing undesired ones [12]. In [13], a pair of linear placed microphone arrays (Kinect) are used together for sound source localization. Note that all above works using multiple microphone arrays are calibrated manually. There are few works focus on the calibration of multiple microphone arrays [14].

In this paper, we propose a auto-calibration framework works with different type of sensors simultaneously including RGB-D sensors and microphone ar-

*連絡先：株式会社　国際電気通信基礎技術研究所
〒 619-0288 京都府相楽郡精華町光台二丁目 2 番地 2
E-mail: chaoran.liu@atr.jp

rays. We used a factor graph to describe the calibration process. GNN is employed for the parameter inference.

2 Background

In this section, we briefly introduce 3D rotation and translation on Special Euclidian group SE(3), factor graph and Graph Neural network.

2.1 3D rotation & translation

A transformation in a 3D space is usually described as:

$$\boldsymbol{T} = \begin{pmatrix} \boldsymbol{R}^{3\times3} & \boldsymbol{t}^{3\times1} \\ \boldsymbol{0}^{1\times3} & 1 \end{pmatrix}$$

with the top left matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix with 3 degrees of freedom, the top right vector $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ is a translation vector. The set of rotation matrix $\boldsymbol{R}$s form a 3D Spatial Orthogonal group $SO(3)$ with group product the standard matrix product.

$$SO(3) = \left\{ \boldsymbol{R} \in \mathbb{R}^{3\times3} | \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I} | \det(\boldsymbol{R}) = 1 \right\}$$

Most gradient-based optimization algorithms such as gradient descent, Gauss-Newton and Levenberg-Marquart are designed to work on Euclidian space but not on a $SO(3)$ since addition is not defined on this manifold. The associated Lie algebra $\mathfrak{so}(3)$ of group $SO(3)$ is used instead of the matrix form $\boldsymbol{R}$ for calibrating the Jacobians on $SO(3)$ manifold. Following exponential and logarithm functions are used as $\mathfrak{so}(3) \mapsto SO(3)$ mapping function and its inverse.

$$\begin{aligned} \boldsymbol{R} &= \exp(\xi^\wedge) = \boldsymbol{I} + \frac{\sin|\xi|}{|\xi|}\xi^\wedge + \frac{1-\cos|\xi|}{|\xi|^2}(\xi^\wedge)^2 \\ \xi &= \log(\boldsymbol{R}) = \left(\frac{\theta}{2\sin(\theta)}(\boldsymbol{R} - \boldsymbol{R}^\top)\right)^\vee \\ \theta &= \arccos((\mathrm{tr}(\boldsymbol{R}) - 1)/2) \end{aligned}$$

where $\boldsymbol{R} \in SO(3)$, $\xi \in \mathfrak{so}(3)$, $|\cdot|$ is the length of a vector, $^\wedge$ indicates conversion from vector to skew symmetric matrix, and vice versa.

$$\boldsymbol{a}^\wedge = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}^\wedge = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} = \boldsymbol{A}$$
$$\boldsymbol{A}^\vee = \boldsymbol{a}$$

In order to avoid complex calculation of Jacobian, derivative of $\mathfrak{so}(3)$ could be used as an approximation in many on-manifold optimization algorithms. Give a rotation $\boldsymbol{R} = \exp(\psi^\wedge)$ and an initial position $\boldsymbol{p}$, the derivative with respect to the increment $\Delta\boldsymbol{R} = \exp(\phi^\wedge)$ can be written as:

$$\begin{aligned} \frac{\partial \boldsymbol{R}\boldsymbol{p}}{\partial\phi} &= \lim_{\phi\to0} \frac{\exp(\phi^\wedge)\exp(\psi^\wedge)\boldsymbol{p} - \exp(\psi^\wedge)\boldsymbol{p}}{\phi} \\ &\approx \frac{(\boldsymbol{I} + \phi^\wedge)\exp(\psi^\wedge)\boldsymbol{p} - \exp(\psi^\wedge)\boldsymbol{p}}{\phi} = -(\boldsymbol{R}\boldsymbol{p}^\wedge) \end{aligned}$$

On the basis of $SO(3)$, Special Euclidian group $SE(3)$ and associated Lie algebra $\mathfrak{se}(3)$ are defined similarly:

$$SE(3) = \left\{ \boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^\top & 1 \end{bmatrix} \in \mathbb{R}^{4\times4} | \boldsymbol{R} \in SO(3), \boldsymbol{t} \in \mathbb{R}^3 \right\}$$

The exponential, logarithm and pseudo-derivative on $\mathfrak{se}(3)$ can be derived accordingly.

In this work, we use a c++ implementation named *Sophus* [1] for Lie algebra computation.

2.2 Calibration factor graph

In a real world problem, we cannot observe a true position of calibration objects with sensors due to measurement uncertainty. Instead, a probabilistic representation can be inferred from observed noisy data. Factor graph is a convenient graphical language for modeling such an inference problem. Assume we have two sensors $\boldsymbol{s} = (s_1, s_2)$ make measurements for a moving object $\boldsymbol{x} = (x_1, x_2, x_3)$. The observations are described as $\boldsymbol{z} = (z_{12}, \cdots, z_{23})$. Fig. 1 shows the factor graph for this sensor calibration problem.

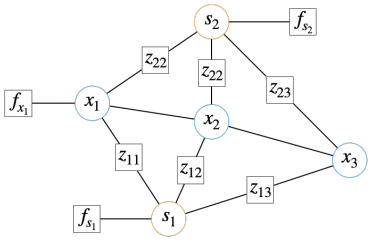

図 1: A factor graph for sensor calibration.

Fig. 1 defines a factor graph $F = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ where circles $\mathcal{V}$ describe variables, squares $\mathcal{U}$ describe factors and edges $\mathcal{E}$ are always between variables and factors. Like Bayesian networks, factor graph can describe joint probability as a product of factors. For

[1] https://github.com/strasdat/Sophus

the example in Fig. 1, the conditional probability $p(\boldsymbol{x}, \boldsymbol{s}|\boldsymbol{z})$ can be written as:

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{s}|\boldsymbol{z}) \propto & p(x_1)p(x_2|x_1)p(x_3|x_2) \\ & \times p(s_1)p(s_2) \\ & \times l(s_1, x_1; z_{11})l(s_1, x_2; z_{12}) \cdots l(s_2, x_3; z_{23}) \end{aligned}$$

where $l(s_i, x_j; z_{ij})$ is the pseudo-likelihood factor of s_i and x_j given observation z_{ij}. Note that in this equation, p and l are denoted as those in Bayesian networks, but in factor graph, they are not necessarily probability distributions and can be replaced with other more generalized function f.

2.3 Graph neural network

Graph Neural Network (GNN) is a class of neural networks that process graph-structured data. Recently, it shows high ability in relation inference [15] and multi-agent interacting system [16, 17]. In [18], GNNs have been considered as performing local message passing on pairwise graphs. They generalized GNN to a Message Passing Neural Network (MPNN) architecture. In a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $v \in \mathcal{V}$ denote vertices and $e \in \mathcal{E}$ denote edges two adjacent vertices, the message pass operations in MPNN are defined as following:

$$\begin{aligned} v \to e: \quad & \boldsymbol{h}_{i,j}^{l} = \mathrm{NN}_{v2e}^{l}([\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \boldsymbol{x}_{i,j}]) \\ e \to v: \quad & \boldsymbol{h}_{j}^{l+1} = \mathrm{NN}_{e2v}^{l}([\sum_{i \in \mathcal{N}_j} \boldsymbol{h}_{i,j}^l, \boldsymbol{x}_j]) \end{aligned}$$

where $v \to e$ and $e \to v$ denote vertex to edge and edge to vertex message passing, $\boldsymbol{h}_{i,j}^l$ and $\boldsymbol{h}_i^l$ are the embeddings (i.e. hidden layer) of edge $e_{i,j}$ and vertex v_i in layer l respectively, $\boldsymbol{x}_{i,j}$ and $\boldsymbol{x}_i$ are features for edge $e_{i,j}$ and v_i in the initial layer, $\mathrm{NN}([\cdot])$ is a full connected neural network takes $[\cdot]$ as input, $[\cdot, \cdot]$ denotes concatenate of vectors. $\mathcal{N}_j$ denotes the set of all adjacent vertices of vertex v_j. These operations allow message passing between vertices and edges multiple rounds (depends on the depth of the neural network). Fig. 2 depicts these message passing neural networks.

3 Proposed method

In this work, we use human head positions as calibration objects to estimate position and orientation

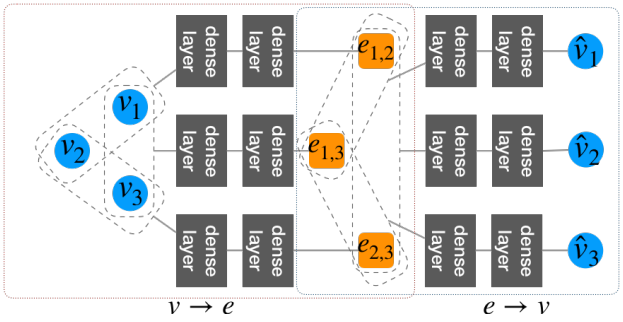

図 2: Vertex to edge and edge to vertex message passing.

of two *Intel RealSense*[2] cameras and one 16-channel microphone array. For cameras, head positions are extracted by *OpenPose* [19] and depth from RGB-D sensor. For microphone array, sound direction is calculated every 100ms.

One of our purpose in this work is to achieve online calibration need not human intervention. The calibration algorithm will run in the background and update the estimation of sensors continuously. That means the algorithm has to deal with the situation that multiple heads detected simultaneously. To this end, we first use a graph auto-encoder to perform unsupervised human identification, and then optimize the calibration factor graph on $\mathfrak{se}(3)$ manifold and estimate sensors' 3D position and orientation.

3.1 Human tracking in discontinuous periods

As shown in the extremely simplified Fig. 1, sensor id i and object id j are needed for each measurement $z_{i,j}$ to construct a proper factor graph. When there are multiple human observed simultaneously, we need to identify them before performing factor graph inference. In a continuous time period, this can easily achieved by a tracking system like Kalman filter. However, microphone array is not able to detect the direction of human all the times if he/she is not keep voicing. Consequently, a collection of audible time periods are used to perform graph inference. It is difficult to carry out human identification in discontinuous time periods. Fig. 3 shows a collection of sensor snapshots.

In Fig. 3, circles x and y denote two human detected simultaneously by sensors. Since we do not

[2]https://www.intelrealsense.com

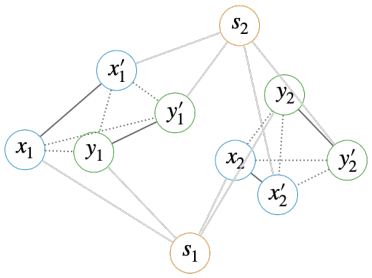

図 3: A complete graph that illustrate three snapshots of two people observed by two sensors.

have enough information about sensors, human are detected by each sensor independently. A complete graph (i.e. each pair of vertices is connected by an edge) is used as start point. However, some of the edges should not actually exist which we want to eliminate during the network training process.

A Variational Graph Auto-Encoder (VGAE) [20] is used for edges elimination. VGAE applies the idea of variational auto-encoder to graph structured data. The input of VGAE is a set of snapshots from each sensor. Sensor positions and orientations are randomly initiated. The measurements are converted to world coordinate using these randomly generated parameters. Fig. 4 shows the architecture of VGAE we used.

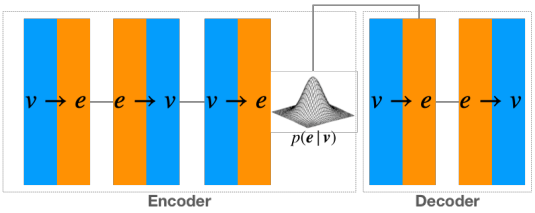

図 4: A variational graph auto-encoder used for edge elimination.

The target of encoder is to predict hidden distribution $p(\boldsymbol{z}|\boldsymbol{x})$ which is a discrete distribution with three states: non-existent, hard connected and loosely connected. The hard connected state indicates same object observed by different sensor, while the loosely connected state indicates objects share similar features (e.g. two human moved in same velocity side by side). In the decoder, the output of the first $v \to e$ neural network was multiplied by a sample extracted from $p(\boldsymbol{z}|\boldsymbol{x})$ and then used as the input of next $e \to v$ network. The decoder was trained to predict next snapshot on the timeline. The training process of VGAE is unsupervised as of a traditional VAE.

3.2 On-manifold 3D-2D calibration

Once a consistent human ID has been estimated, we can minimize the reprojected position error of same ID to calibrate sensors. Give RGB-D camera's pose, reprojected head position in world coordinate can be calculated easily:

$$\begin{bmatrix} x_i^{world} \\ y_i^{world} \\ z_i^{world} \end{bmatrix} = \exp(\xi_j^\wedge) \begin{bmatrix} x_i^{local} \\ y_i^{local} \\ z_i^{local} \\ 1 \end{bmatrix}$$

with i and j are human ID and camera ID respectively, ξ_j is camera j's pose in $\mathfrak{se}(3)$. The last dimension in the right hand side is omitted.

Assume that we use first RGB-D camera's coordinate as world coordinate, the pose of the second camera ξ_2 can be estimated by minimize:

$$\hat{\xi}_2 = \arg\min_{\xi_2} \frac{1}{2} \sum_i \left\| \boldsymbol{x}_i^1 - \exp(\xi_2^\wedge) \boldsymbol{x}_i^2 \right\|^2$$

where $\boldsymbol{x}_i^1$ is the head i's position in camera 1's coordinate (world coordinate), $\boldsymbol{x}_i^2$ is the same head in camera 2's local coordinate.

Regarding the microphone array, since it only detect the direction of sound source with azimuth angle θ_1 and elevation angle θ_2, the optimization has to be performed in 2D. The cost function is the same one as in camera-camera calibration except we picked the second and third dimension as optimization target since it can be written as $\tan(\theta_1)$ and $\tan(\theta_2)$.

3.3 Experimental results

In order to test the proposed calibration algorithm, we modified an open dataset used in [21]. A Gaussian noise with standard deviation of 15cm was added into every camera measurements. Target angles to a randomly chosen microphone array position are also generated with 0 mean 2 standard deviation Gaussian noise. First, 5 sets of measurements with 100 time steps are used to test the VGAE. The training process started with fully connected graph. Experimental results show that the VGAE was able to eliminate 98.3% non-existent connections between different IDs. For the optimization on $\mathfrak{se}(3)$, we used *Ceres Solver*[3] to achieved a mean error of 22mm for RealSense and 57mm for microphone array.

[3] http://ceres-solver.org

4 Conclusion

In this paper, we proposed a sensor calibration framework that can automatically calibrate different type of sensors without any human intervention. It uses detected human head positions as calibration objects. The framework chooses suitable snapshot and concatenate them as time-discontinuous trunk of measurements used for calibration process. The human ID is predicted with a Variational Graph Auto-Encoder in an unsupervised manner. After corresponding human ID has been estimated, an optimization process is performed on Special Euclidian group with the associated Lie algebra $\mathfrak{se}(3)$. The experiment results on synthesized data with additional noise show that the proposed framework can predict human IDs with 100% accuracy and accurately estimate sensor positions and orientations simultaneously.

Acknowledgment

This work was partly supported by the Tateishi Science and Technology Foundation, JST-Mirai Program Grant Number JPMJMI18C6, and JST, ERATO, Grant Number JPMJER1401.

参考文献

[1] Zhengyou Zhang, "Camera calibration with one-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892–899, July 2004.

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

[3] O. Faugeras, *Three-dimensional Computer Vision: A Geometric Viewpoint*. Cambridge, MA, USA: MIT Press, 1993.

[4] E. Auvinet, J. Meunier, and F. Multon, "Multiple depth cameras calibration and body volume reconstruction for gait analysis," in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, July 2012, pp. 478–483.

[5] R. Macknojia, A. Chávez-Aragón, P. Payeur, and R. Laganière, "Calibration of a network of kinect sensors for robotic inspection over a large workspace," in *2013 IEEE Workshop on Robot Vision (WORV)*, Jan 2013, pp. 184–190.

[6] Y. Han, S.-L. Chung, J.-S. Yeh, and Q.-J. Chen, "Localization of rgb-d camera networks by skeleton-based viewpoint invariance transformation," vol. 63, 10 2013, pp. 1525–1530.

[7] K. Desai, B. Prabhakaran, and S. Raghuraman, "Skeleton-based continuous extrinsic calibration of multiple rgb-d kinect cameras," in *Proceedings of the 9th ACM Multimedia Systems Conference*, ser. MMSys '18. New York, NY, USA: ACM, 2018, pp. 250–257. [Online]. Available: http://doi.acm.org/10.1145/3204949.3204969

[8] Kazuhiro Nakadai, Shunichi Yamamoto, H. G. Okuno, Hirofumi Nakajima, Yuji Hasegawa, and Hiroshi Tsujino, "A robot referee for rock-paper-scissors sound games," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 3469–3474.

[9] K. Nakamura, K. Nakadai, F. Asano, and G. Ince, "Intelligent sound source localization and its application to multimodal human tracking," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2011, pp. 143–148.

[10] J. Even, Y. Morales, N. Kallakuri, J. Furrer, C. T. Ishi, and N. Hagita, "Mapping sound emitting structures in 3d," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 677–682.

[11] C. Liu, C. T. Ishi, and H. Ishiguro, "Bringing the scene back to the tele-operator: Auditory scene manipulation for tele-presence systems," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, March 2015, pp. 279–286.

[12] C. T. Ishi, C. Liu, J. Even, and N. Hagita, "Hearing support system using environment sensor network," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 1275–1280.

[13] L. A. Seewald, L. Gonzaga, Jr., M. R. Veronez, V. P. Minotto, and C. R. Jung, "Combining

srp-phat and two kinects for 3d sound source localization," *Expert Syst. Appl.*, vol. 41, no. 16, pp. 7106–7113, Nov. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2014.05.033

[14] D. Su, T. Vidal-Calleja, and J. V. Miro, "Towards real-time 3d sound sources mapping with linear microphone arrays," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 1662–1668.

[15] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4967–4976.

[16] S. Sukhbaatar, a. szlam, and R. Fergus, "Learning multiagent communication with backpropagation," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2244–2252.

[17] P. Battaglia, R. Pascanu, M. Lai, D. J. Rezende, and K. kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, ser. NIPS'16. USA: Curran Associates Inc., 2016, pp. 4509–4517.

[18] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, pp. 1263–1272.

[19] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, 2017.

[20] T. N. Kipf and M. Welling, "Variational graph auto-encoders," 2016.

[21] S. Agarwal, N. Snavely, S. M. Seitz, and R. Szeliski, "Bundle adjustment in the large," in *Proceedings of the 11th European Conference on Computer Vision: Part II*, ser. ECCV'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 29–42.