Supplementary Material of LSG-GPD: Coherent Point Drift with Local Surface Geometry for Point Cloud Registration

Weixiao Liu[1,2] Hongtao Wu[1,2] Gregory S. Chirikjian[1,2*]
[1]National University of Singapore, Singapore [2]Johns Hopkins University, USA
{mpewxl, mpehtw, mpegre}@nus.edu.sg

Abstract

Detailed derivations and experiment settings are provided in this supplementary material. In Sec. 1, the derivation of the EM algorithm is presented. In Sec. 2, we show the derivation of the outlier weight. In Sec. 3, concepts and definitions about Lie groups and Lie algebra are briefly reviewed. Then, the derivation of the gradient and the Hessian matrix are discussed. The update of the covariance multiplier σ^2 is also derived in this section. In Sec. 4, detailed experiment settings are given. Finally, in Sec. 5, we provide a visualization of the point-to-plane penalization coefficient α on a point cloud data and more insights on the robustness of the method. The source code is available at https://github.com/ChirikjianLab/LSG-CPD.git.

1. Derivation of EM

Recall the negative log-likelihood function in Sec. 3.1 of the paper. The negative log-likelihood of the transformed observations $g(\mathbf{x}_n)$ is:

$$\begin{aligned} L(g,\sigma^2) &= -\sum_{n=1}^{N}\log p(g(\mathbf{x}_n))) \\ &= -\sum_{n=1}^{N}\log\Big(w_n p_o \\ &\quad + (1-w_n)\sum_{m=1}^{M}\pi(m)p(g(\mathbf{x}_n)|m)\Big) \end{aligned} \tag{1}$$

Because of the summation inside the logarithm, it is intractable to solve this MLE problem directly. Therefore, the EM algorithm [3, 6] is applied. In the EM framework, we treat the correspondence between the m-th mixture component and the observation $\mathbf{x}_n$ as a latent variable $z_{mn} \in \{0,1\}$. $\mathbf{Z}$ is the set containing z_{mn} where $m = 1,2,\ldots,M+1$ and $n = 1,2,\ldots,N$. $z_{mn} = 1$ if $\mathbf{x}_n$ corresponds to the m-th mixture component; $z_{mn} = 0$ if otherwise. Note that $m = M+1$ indicates the uniformly distributed outlier component. The complete negative log-likelihood function L_c is:

$$\begin{aligned} L_c(g,\sigma^2,\mathbf{Z}) &= -\log\prod_{n=1}^{N}\Big((w_n p_o)^{z_{(M+1)n}} \\ &\quad \prod_{m=1}^{M}[(1-w_n)\pi(m)p(g(\mathbf{x}_n)|m)]^{z_{mn}}\Big) \\ &= -\sum_{n=1}^{N}\Big(z_{(M+1)n}\log(w_n p_o) + \\ &\quad \sum_{m=1}^{M} z_{mn}\log[(1-w_n)\pi(m)p(g(\mathbf{x}_n)|m)]\Big) \end{aligned} \tag{2}$$

Ignoring terms independent of g and σ^2, Eq. 2 becomes:

$$L_c'(g,\sigma^2,\mathbf{Z}) = -\sum_{n=1}^{N}\sum_{m=1}^{M} z_{mn}\log\left(p(g(\mathbf{x}_n)|m)\right) \tag{3}$$

The latent variables z_{mn} cannot be observed directly. Thus, it is replaced with its conditional expectation given the transformation g_{old} obtained in the last iteration, *i.e.*, $E(z_{mn}|g_{old}(\mathbf{x}_n))$. According to the Bayes' rule, the posterior correspondence probability is:

$$\begin{aligned} P(z_{mn}=1|g_{old}(\mathbf{x}_n)) &= \frac{P(z_{mn}=1)p(g_{old}(\mathbf{x}_n)|z_{mn=1})}{p(g_{old}(\mathbf{x}_n))} \\ &= \frac{(1-w_n)\pi(m)p(g_{old}(\mathbf{x}_n|m))}{w_n p_o + (1-w_n)\sum_m \pi(m)p(g_{old}(\mathbf{x}_n)|m)} \end{aligned} \tag{4}$$

Subsequently, the conditional expectation can be computed: $E(z_{mn}|g_{old}(\mathbf{x}_n)) = 1\cdot P(z_{mn}=1|g_{old}(\mathbf{x}_n)) + 0\cdot P(z_{mn}=0|g_{old}(\mathbf{x}_n)) = P(z_{mn}=1|g_{old}(\mathbf{x}_n))$. And $P(z_{mn}=1|g_{old}(\mathbf{x}_n))$ is $\mathbf{P}_{mn}$ which is computed via Eq. 5 of the paper in the E step. Substituting z_{mn} in Eq. 3 with its conditional expectation $E(z_{mn}|g_{old}(\mathbf{x}_n))$, we obtain the objective function:

$$\begin{aligned} Q &= -\sum_{n=1}^{N}\sum_{m=1}^{M} E(z_{mn}|g_{old}(\mathbf{x}_n))\log\left(p(g(\mathbf{x}_n)|m)\right) \\ &= -\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{P}_{mn}(\log(c_m) - \frac{1}{2}\|g(\mathbf{x}_n)-\mathbf{y}_m\|^2_{\mathbf{\Sigma}_m^{-1}}) \end{aligned} \tag{5}$$

*Corresponding author

where c_m is the normalizing constant and $\mathbf{\Sigma}_m$ is the covariance matrix:

$$\begin{aligned} c_m &= (2\pi\sigma^2)^{-\frac{3}{2}}(\alpha_m + 1)^{\frac{1}{2}} \\ \mathbf{\Sigma}_m^{-1} &= \frac{1}{\sigma^2}(\alpha_m \mathbf{n}_m \mathbf{n}_m^T + \mathbf{I}) \end{aligned} \tag{6}$$

Ignoring terms independent of g and σ^2, the objective function becomes:

$$Q = \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{M} \mathbf{P}_{mn}(3\log(\sigma^2) + \|g(\mathbf{x}_n) - \mathbf{y}_m\|^2_{\mathbf{\Sigma}_m^{-1}}) \tag{7}$$

In the M step, we update g and σ^2 to minimize Q. The algorithm iterates between the E and M step until convergence.

2. Derivation of the Outlier Weight

The outlier component makes CPD [11] and FilterReg [7] robust to outliers (source points with no correspondence to any target points). The role of the outlier weight w_0 is similar to the trimming threshold in TrICP [4]. It makes the update of g in the M step primarily focus on points with strong correspondence. On the one hand, the larger w_0 is, the less sensitive the algorithm is to wrong correspondence; on the other hand, as $w_0 \to 1$, information provided by correct correspondences of the inliers will also be "trimmed" out, resulting in low convergence rate or getting stuck at a local minimum. Therefore, it is important to find a proper w_0 which is large enough to ensure robustness while not too large such that the inliers are submerged.

Recall Eq. 7 in the paper and follow the definition of z_{mn} in the above section. Given a guess of the outlier ratio η, our goal is to estimate the upper bound of the baseline uniform outlier weight w_0 from a posterior prospective. The number of outliers is defined as:

$$N_{outlier} = \sum_{n=1}^{N} z_{(M+1)n} \tag{8}$$

$z_{(M+1)n}$ is a latent random variable. The posterior probability of $z_{(M+1)n}$ can be calculated given the transformation g via the Bayes' rule. Similar to the procedure of E step:

$$P(z_{(M+1)n} = 1|g(\mathbf{x}_n)) = \frac{w_0 p_o}{w_0 p_o + (1-w_0)\sum_m \pi(m) p(g_{old}(\mathbf{x}_n)|m)} \tag{9}$$

w_0 is the baseline outlier weights to be estimated. Therefore, the expectation of the number of outliers given g becomes:

$$E(N_{outlier}|g(\mathbf{x}_n)) = \sum_{n=1}^{N} 1 \cdot P(z_{(M+1)n} = 1|g(\mathbf{x}_n)) \tag{10}$$

If w_0 is chosen properly, $E(N_{outlier}|g^*(\mathbf{x}_n))$ should equal to ηN, where g^* is the true transformation that aligns the point clouds. However, g^* is not available *a priori*. Thus, w_0 cannot be solved from this relationship directly. But a proper choice of w_0 can be bounded by setting $p(g_{old}(\mathbf{x}_n)|m)$ equal to its peak density c_m:

$$\begin{aligned} \eta N &= E(N_{outlier}|g^*(\mathbf{x}_n)) \\ &\geq \sum_{n=1}^{N}\left(\frac{w_0 p_o}{w_0 p_o + (1-w_0)\sum_m \pi(m) c_m}\right) \end{aligned} \tag{11}$$

c_m is also the normalizing constant defined in Eq. 2 in the paper. Its value is given in Eq. 6. Rearranging terms and substituting p_o with $1/V$, we get:

$$w_0 \leq \frac{\eta V \sum_m \pi(m) c_m}{(1-\eta) + \eta V \sum_m \pi(m) c_m} \tag{12}$$

That is, from a probabilistic prospective, regardless of the true transformation g^*, if w_0 is chosen larger than this upper bound, it is guaranteed that more points are considered as outliers than the truth. Therefore, we choose w_0 to be this upper bound to strike a balance.

3. Optimization on Matrix Lie Group

3.1. Mathematical Preliminaries

In this section, we show the detailed definitions and derivations in Sec. 3.5 of the paper. The 3-d rigid transformation $g = (\mathbf{R}, \mathbf{t})$ forms a a group called the special Euclidean group $SE(3)$. The homogeneous form of g and point $\mathbf{x}$ are defined respectively as:

$$\widetilde{g} = \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{pmatrix} \qquad \widetilde{\mathbf{x}} = \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix} \tag{13}$$

i.e., group elements are matrices and the group operation is matrix multiplication. The image of a point x transformed by g is $\widetilde{g}\widetilde{\mathbf{x}}$. $SE(3)$ is a matrix Lie group. The basis elements of the corresponding Lie algebra $se(3)$ are:

$$\begin{aligned} \mathbf{E}_1 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{E}_2 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{E}_3 &= \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{E}_4 = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \\ \mathbf{E}_5 &= \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{E}_6 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \end{aligned} \tag{14}$$

For any $\mathbf{X} \in se(3)$, it can always be represented as:

$$\mathbf{X} = \sum_{i=1}^{6} x_i \mathbf{E}_i \tag{15}$$

The matrix exponential maps $\mathbf{X}$ to an element in $SE(3)$, *i.e.*, $\exp(\mathbf{X}) \in SE(3)$, $\forall \mathbf{X} \in se(3)$. Two operators $\vee$ and $\wedge$

are defined to map between $\mathbf{X} \in se(3)$ and a vector $\mathbf{x} = [x_1\ x_2\ x_3\ x_4\ x_5\ x_6]^T \in \mathbb{R}^6$:

$$\mathbf{X}^\vee = \mathbf{x} \qquad \mathbf{x}^\wedge = \mathbf{X} \tag{16}$$

For a function defined on a matrix Lie group $f(\widetilde{g})$, the right derivative on the direction of a basis element $\mathbf{E}_i$ is defined as [5]:

$$E_i^r f(\widetilde{g}) \doteq \frac{d}{dt} f(\widetilde{g} \circ \exp(t\mathbf{E}_i))\Big|_{t=0} \tag{17}$$

According to the definition of matrix exponential

$$\exp(t\mathbf{E}_i) = \mathbf{I} + t\mathbf{E}_i + \frac{1}{2!}t^2\mathbf{E}_i^2 + \frac{1}{3!}t^3\mathbf{E}_i^3 + \dots \tag{18}$$

and thus for $i = 1, 2, \dots, 6$

$$\begin{aligned}\frac{d\exp(t\mathbf{E}_i)}{dt}\Big|_{t=0} =& \mathbf{E}_i + t\mathbf{E}_i^2 + \frac{1}{2!}t^2\mathbf{E}_i^3 + \dots\Big|_{t=0} \\ =& \mathbf{E}_i\end{aligned} \tag{19}$$

The gradient vector $\nabla f(\widetilde{g})$ and the Hessian matrix $\mathbf{H}(\widetilde{g})$ of $f(\widetilde{g})$ are defined as:

$$\nabla f(\widetilde{g}) = \begin{pmatrix} E_1^r f & E_2^r f & \dots & E_6^r f \end{pmatrix}^T \tag{20}$$

$$\mathbf{H}(\widetilde{g}) = \begin{pmatrix} E_1^r E_1^r f & E_1^r E_2^r f & \dots & E_1^r E_6^r f \\ E_2^r E_1^r f & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ E_6^r E_1^r f & \dots & \dots & E_6^r E_6^r f \end{pmatrix} \tag{21}$$

3.2. Closed Form Gradient and Hessian

In the first step of M step, we want to find the optimal g that minimizes the objective function Eq. 7. Ignoring terms independent of g in Eq. 7 and substituting terms with their homogeneous form, it is equivalent to minimize the following:

$$Q(\widetilde{g}) = \sum_{n=1}^{N}\sum_{m=1}^{M} \mathbf{P}_{mn}(\widetilde{g}\widetilde{\mathbf{x}}_n - \widetilde{\mathbf{y}}_m)^T \widetilde{\mathbf{\Sigma}}_m^{-1}(\widetilde{g}\widetilde{\mathbf{x}}_n - \widetilde{\mathbf{y}}_m) \tag{22}$$

where $\widetilde{\mathbf{\Sigma}}_m^{-1} = \mathbf{\Sigma}_m^{-1} \oplus 0$ is the augmented inverse covariance matrix and $\oplus$ denotes the direct sum. According to Eq. 17 and Eq. 19, we obtain:

$$\begin{aligned} E_i^r Q(\widetilde{g}) =& \frac{d}{dt} Q(\widetilde{g} \circ \exp(t\mathbf{E}_i))\Big|_{t=0} \\ =& 2\sum_{n=1}^{N}\sum_{m=1}^{M} \mathbf{P}_{mn}(\widetilde{g}\exp(t\mathbf{E}_i)\widetilde{\mathbf{x}}_n - \widetilde{\mathbf{y}}_m)^T \widetilde{\mathbf{\Sigma}}_m^{-1} \\ & \frac{d}{dt}\left(\widetilde{g}\exp(t\mathbf{E}_i)\widetilde{\mathbf{x}}_n - \widetilde{\mathbf{y}}_m\right)\Big|_{t=0} \\ =& 2\sum_{n=1}^{N}\sum_{m=1}^{M} \mathbf{P}_{mn}(\widetilde{g}\exp(t\mathbf{E}_i)\widetilde{\mathbf{x}}_n - \widetilde{\mathbf{y}}_m)^T \widetilde{\mathbf{\Sigma}}_m^{-1} \\ & \widetilde{g}\frac{d}{dt}\left(\exp(t\mathbf{E}_i)\right)\widetilde{\mathbf{x}}_n\Big|_{t=0} \\ =& 2\sum_{n=1}^{N}\sum_{m=1}^{M} \mathbf{P}_{mn}(\widetilde{g}\widetilde{\mathbf{x}}_n - \widetilde{\mathbf{y}}_m)^T \widetilde{\mathbf{\Sigma}}_m^{-1}\widetilde{g}\mathbf{E}_i\widetilde{\mathbf{x}}_n \end{aligned} \tag{23}$$

It is worth noting that $\sum_{m=1}^{M}\mathbf{P}_{mn}\widetilde{\mathbf{\Sigma}}_m^{-1}$ and $\sum_{m=1}^{M}\mathbf{P}_{mn}\widetilde{\mathbf{y}}_m^T\widetilde{\mathbf{\Sigma}}_m^{-1}$ can be pre-calculated to speed up the calculation of the derivatives. Similarly, $E_i^r E_j^r Q(\widetilde{g}) = E_i^r(E_j^r Q(\widetilde{g}))$ can be derived in closed form following this procedure. In practice, in Eq. 12 in the paper, the second order derivatives can be approximated by omitting the last term.

3.3. Update of Covariance Multiplier σ^2

After g gets updated, σ^2 is inferred by setting the corresponding derivative of Eq. 7 to zero. The derivative is:

$$\begin{aligned} \frac{d}{d\sigma^2}Q(\sigma^2) =& \frac{d}{d\sigma^2}\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{P}_{mn}\left(\frac{3}{2}\log(\sigma^2) + \right. \\ & \left.\frac{1}{2\sigma^2}\|g(\mathbf{x}_n) - \mathbf{y}_m\|^2_{(\alpha_m\mathbf{n}_m\mathbf{n}_m^T + \mathbf{I})^{-1}}\right) \\ =& \sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{P}_{mn}\left(\frac{3}{2\sigma^2} - \right. \\ & \left.\frac{1}{2(\sigma^2)^2}\|g(\mathbf{x}_n) - \mathbf{y}_m\|^2_{(\alpha_m\mathbf{n}_m\mathbf{n}_m^T + \mathbf{I})^{-1}}\right) \end{aligned} \tag{24}$$

Subsequently, σ^2 is updated by:

$$\sigma^2 = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{P}_{mn}\|g(\mathbf{x}_n) - \mathbf{y}_m\|^2_{(\alpha_m\mathbf{n}_m\mathbf{n}_m^T + \mathbf{I})^{-1}}}{3\sum_{n=1}^{N}\sum_{m=1}^{M}\mathbf{P}_{mn}} \tag{25}$$

4. Experiment Details

4.1. Baseline Methods

The source codes of the baseline methods we used for comparison in the paper are listed here. They are either provided directly by the author or taken from the popular open source library with various performance optimization. The source code of MATrICP [10] and EMPMR [14] in Sec. 4.2 of the paper are provided directly by the authors of the papers. The rest are listed as follows:

CPD [11]: https://www.mathworks.com/help/vision/ref/pcregistercpd.html

FilterReg [7]: https://bitbucket.org/gaowei19951004/poser/src/master/

TrICP/ICP [2, 4]: https://github.com/ethz-asl/libpointmatcher

ECMPR [9]: https://team.inria.fr/perception/research/ecmpr/#code

GICP [12]: https://github.com/SMRT-AIST/fast_gicp

The parameters for running the experiments are either provided directly by the authors/libraries or well tuned by ourselves. We are grateful to the authors of the papers and libraries for providing the code for usage.

4.2. Proposed Method

In this section, we provide the setting of the proposed method in each experiment. Similar to CPD [11], the initial value of σ^2 is computed autonomously from the data in all the experiments. In Sec. 4.1 of the paper, $\eta = 0.5$ for the outlier experiment and $\eta = 0.001$ for the noise experiment. No artificial outliers are added to the data in the noise experiment, thus η is set relatively low. $\eta = 0.5$ in Sec. 4.2 of the paper. Adjacent scans have relatively low overlapping ratio. Thus, many points from the source set should be regarded as outliers/missing points due to lack of correspondence, resulting in the choice of a large η. $\eta = 0.07$ and $\eta = 0.05$ in Sec. 4.3 and Sec. 4.4 of the paper, respectively. Adjacent frames in these two datasets are captured relatively close. The overlapping ratios are higher than that in Sec 4.2 of the paper. Thus, η are set relatively small accordingly. The outlier weight w_0 is computed via the procedure introduced in Sec. 3.2 of the paper. In all the experiments, $\lambda = 0.2$ (Eq. 3 in the paper). $\alpha_{\max} = 30$ (Eq. 3 in the paper) for all experiments except the outlier and noise experiment in Sec. 4.1 of the paper. $\alpha_{\max} = 5$ in these two experiments. The data in these two experiments contains a large amount of artificial noise and outlier which far exceeds those in data directly acquired from the real world. The normal estimation for many points will be ineffective. Thus, we set $\alpha_{\max}$ relatively small to make the maximum weight of the point-to-plane penalization relatively small.

4.3. Confidence Filtering in Registration on RGBD Dataset

Confidence Filtering is only used in the experiment on the Stanford Lounge Dataset [13] (Sec. 4.3 of the paper). We use the error model $e(\mathbf{x})$ (Sec. 3.3 of the paper) provided by [8]. $e(\mathbf{x}) = e(d(\mathbf{x}))$, where $d(\mathbf{x})$ denotes the depth of point $\mathbf{x}$, is a second order polynoimal function of $d(\mathbf{x})$. It is measured specifically for the sensor used to capture this dataset [13], *i.e.*, Asus Xtion Pro, via statistical error evaluation experiments [8]. It represents the precision* of the measurement at this point. Precision quantizes the standard deviation of the camera depth measurements. Within the sensor range, $e(d(\mathbf{x}))$ is an increasing function which increases as $d(\mathbf{x})$ increases. Intuitively, that is the noise of the measurement increases as $d((x))$ increases. Thus, the confidence $\phi(\mathbf{x}) = e_{\min}/e(\mathbf{x})$ decreases as $d(\mathbf{x})$ increases. As a result, the points captured far away from the camera optical center (*e.g.*, points in the green circle of Fig. 1), which contain a lot of noise, will have small confidence and thus be truncated or have little contribution to the registration.

*The definition of precision follows the official definitions of precision according to ISO 5725-1 [1].

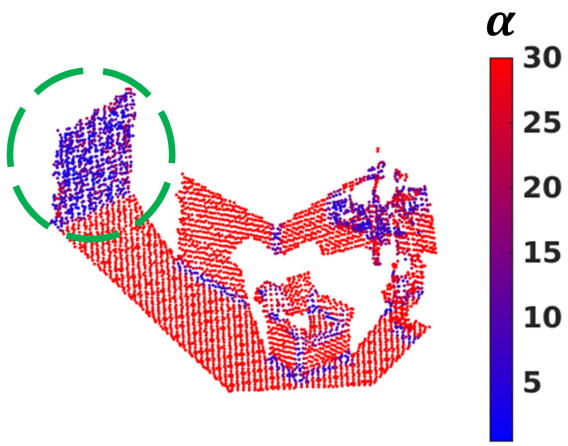

Figure 1. Alpha Visualization. Red indicates large α; blue indicates small α. The green circle indicates a plane captured far away from the camera optical center.

5. Additional Experimental Results

In this section, we visualize α on a point cloud from the Stanford Lounge dataset [13] (Sec 4.3 of the paper). The visualization is shown in Fig. 1. For the region where the local surface is flat, α is large. If the local surface is not flat, *e.g.*, at the intersection between two planes, α is small. If the measurement is noisy, *e.g.*, the green circle in Fig. 1, even though the plane is flat in the real world, the noisy measurement makes the point cloud locally non-flat. And the estimated surface normal from this region will be ineffective. α is also small in this case. The GMM components will be sphere-like. In this way, the robustness of the algorithm is enhanced as the registration will not be misled by ineffectively estimated normals.

References

[1] Accuracy (trueness and precision) of measurement methods and results — part 1: General principles and definitions. Standard ISO 5725-1:1994, International Organization for Standardization, 1994. 4

[2] P. J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE transactions on pattern analysis and machine intelligence*, 14(2):239–256, 1992. 4

[3] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006. 1

[4] D. Chetverikov, D. Svirko, D. Stepanov, and P. Krsek. The trimmed iterative closest point algorithm. In *Pattern Recognition, International Conference on*, volume 3, pages 30545–30545, 2002. 2, 4

[5] G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2*. 01 2012. 3

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B*, 39:1–38, 1977. 1

[7] W. Gao and R. Tedrake. Filterreg: Robust and efficient probabilistic point-set registration using gaussian filter and twist parameterization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11095–11104, 2019. 2, 4

[8] G. Halmetschlager-Funek, M. Suchi, M. Kampel, and M. Vincze. An empirical evaluation of ten depth cameras: Bias, precision, lateral noise, different lighting conditions and materials, and multiple sensor setups in indoor environments. *IEEE Robotics Automation Magazine*, 26(1):67–77, 2019. 4

[9] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang. Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):587–602, 2010. 4

[10] Z. Li, J. Zhu, K. Lan, C. Li, and C. Fang. Improved techniques for multi-view registration with motion averaging. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 713–719. IEEE, 2014. 3

[11] A. Myronenko and X. Song. Point set registration: Coherent point drift. *IEEE transactions on pattern analysis and machine intelligence*, 32(12):2262–2275, 2010. 2, 4

[12] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. In *Proceedings of Robotics: Science and Systems*, Seattle, USA, June 2009. 4

[13] Q.-Y. Zhou and V. Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (ToG)*, 32(4):1–8, 2013. 4

[14] J. Zhu, R. Guo, Z. Li, J. Zhang, and S. Pang. Registration of multi-view point sets under the perspective of expectation-maximization. *IEEE Transactions on Image Processing*, 29:9176–9189, 2020. 3