2016 IEEE International Conference on Robotics and Automation (ICRA)
Stockholm, Sweden, May 16-21, 2016

Learning Anisotropic ICP (LA-ICP) for Robust and Efficient 3D Registration

Bhoram Lee and Daniel D. Lee
GRASP Laboratory
University of Pennsylvania
Philadelphia, PA 19104

***Abstract*— This paper presents an online learning approach to 3D object registration that vastly improves the performance of Iterative Closest Point (ICP) methods. Our approach achieves better robustness and stable convergence by learning generalized distance functions directly from a stream of object depth data. The proposed algorithm, Learning Anisotropic ICP (LA-ICP), parameterizes the point uncertainty of the underlying object surface as an anisotropic Gaussian, and estimates the covariance parameters of the likelihood function for ICP from data. Our learning scheme does not require manual tuning and the parameters of the algorithm are continually updated from observed data. Experiments on various RGB-D object datasets demonstrate the effectiveness of our approach in terms of convergence and pose accuracy as well as robustness to initial conditions.**

I. Introduction

Object perception has long been one of the most challenging problems in robotics and computer vision. Recent advances in RGB-D sensing technology have accelerated studies on object recognition and interactions using the depth channel [1]. Among its sub-problems, obtaining a good pose is crucial in scenarios that involve robot interactions with the world, such as grasping or manipulation. However, precise object pose recognition is not a simple task for machine intelligence, due to partial views, inherently symmetric shapes, and a large search space of $SE(3)$.

ICP [2] has been a popular registration algorithm at a very precise level that appears in various robotic applications, some examples of which include 3D SLAM [3] [4] or object tracking [5]. It is an iterative algorithm that gives a suboptimal registration solution of two point sets, which we will call a model set and a data set. The iterative procedure of ICP decomposes into two steps: 1) find *best* correspondences given a transformation prior and 2) estimate the *best* transformation given the correspondences. As in any optimization problem, the ICP algorithm tries to find the *best* answer based on *some* error metric. This error metric determines the way the algorithm converges as well as the quality of the pose estimate.

The proposed learning approach, Learning Anisotropic ICP (LA-ICP), automatically determines the parameters of the error metric for ICP-based registration (Fig. 1) without pre-training. We discuss how to model the point representation uncertainty of the object surface as an anisotropic Gaussian in a learnable form, and how to estimate the uncertainty parameters from data. We consider pose estimation under viewpoint changes in a stream of noisy data, which is a challenging yet common situation in robotic applications. Hence, a flexible learning algorithm is more desirable than one with fixed parameters. Unlike previous approaches, LA-ICP is able to adapt while learning the parameters by utilizing moderate changes in such settings. We also demonstrate that our online learning approach improves the performance of ICP in terms of robustness and efficiency on real object point clouds of various shapes from varying view points.

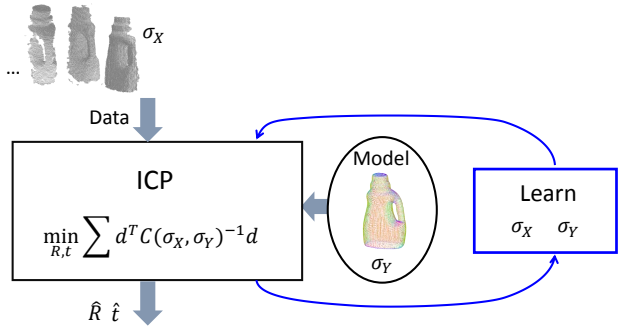


Fig. 1. Learning Anisotropic ICP (LA-ICP): LA-ICP learns the anisotropic covariance parameters in the ICP algorithm from converged registration of the data and the model. It keeps updating them without manual tuning or pre-training.

In our work, a model set consists of cleaned points with pre-computed surface normals. The data set is assumed to consist solely of points. It is usually hard to obtain geometric properties around missing values or boundaries, which arises frequently when dealing with range sensors or RGB-D cameras. Thus, this is a practical assumption that simplifies the problem and speeds up the computation.

After reviewing previous work in the next section, the ICP-based registration problem is summarized in Section III. Section IV describes our learning strategy to find the anisotropic covariance parameters of the algorithm. Then, in Section V, our experiment design is explained and the results are shown.

Email: {bhorlee,ddlee}@seas.upenn.edu

II. Related Work

As reported in [6], many variants of ICP have been suggested since the standard ICP was introduced in [2] and the point-to-plane version was proposed in [7]. Earlier works explored different aspects of the algorithm ranging from how to sample to how to accept/reject, to how to find better correspondences. Standard ICP and most of its variants use mean-squared Euclidean distance as the error metric to be minimized. Other cases include the point-to-plane ICP that uses projected distance on the tangent plane [7] as well as feature-aided variants that utilized other feature space with the geometric distance [8]. Some studies on obtaining closed-form solutions of ICP are also found in literature [9] [10] [11]. Implementation of some of these algorithms is readily accessible from the Point Cloud Library [12].

More recently, researchers started to use the Mahalanobis (or generalized) distance for registration. The registration problem is now seen as optimizing under the uncertainty of possibly anisotropic distributions for both sets. A well-organized probabilistic view of the ICP algorithm, named Generalized-ICP, can be found in [13]. Even before that, [14] ran their experiments with different settings of eigenvalues for the model uncertainty covariance in the anisotropic form. [15] used the Mahalanobis distance metric for aligning multiple data sets and evaluated their methods in 2D applications.

Within the generalized ICP framework, we need to have some strategies to determine the values of covariances. The authors of [13] suggested plane-to-plane ICP as a special case of generalized ICP. Along the same line, covariance matrix computation methods based on Principal Component Analysis (PCA) and Voronoi Analysis were suggested in [16]. They adopted the anisotropic distance for both steps of ICP. [17] further proposed using exact likelihood instead of the Mahalanobis distance for matching points. They modeled measurement noise, modeling uncertainty, and matching uncertainty in separate terms. However, many studies tested their algorithm with some arbitrary values of noise covariances. Our effort is made to contribute a principled yet flexible method that learns the right anisotropic covariance values for ICP from data online.

To provide a complete and broader perspective, there has been another generalized view of ICP, especially in terms of finding correspondences [18]. Whereas, most of the previous works mentioned above are based on hard assignment of correspondences, this framework allows many possible ways to assign or score correspondences among multiple pairs of points. Even though we do not directly address this issue, LA-ICP can be applied to general alignment cases.

III. Background

We briefly review the ICP algorithm in the EM framework in this section. Similar derivation can be also found in [18] [19] for further details.

Let $Y = \{\mathbf{y}_j\}_{j=1}^{N_y}$ be the model and $X = \{\mathbf{x}_i\}_{i=1}^{N}$ be the observed point set to be aligned with the model. Our goal is to find the transformation $R \in SO(3)$ and $t \in \mathbf{R}^3$ that maximizes the log-likelihood,

$$\begin{aligned} L(R,t) &= \sum_{i=1}^{N} \ln p(x_i|R,t) \\ &= \sum_{i=1}^{N} \ln \sum_{z_i} p(x_i, z_i|R,t), \end{aligned} \tag{1}$$

where z_i is a hidden variable indicating the association of x_i with a point in Y, $p(z_i = k)$ meaning the probability that x_i corresponds to y_k. However, because this is intractable, we may solve it iteratively in an EM framework [20]. Thus, we are going to optimize the expected complete log-likelihood,

$$L_{ec}(R,t) = \sum_{i=1}^{N} E_{z_i}[\ln p(x_i, z_i|R,t)]. \tag{2}$$

Considering that the distribution of z_i is discrete, the function that we are practically going to optimize can be boiled down to, 归结为

$$L_{ec}(R,t) = \sum_{i=1}^{N} \sum_{j=1}^{N_y} \alpha_{ij} \{\ln p(x_i|y_j, R, t) + \ln \pi_j\}. \tag{3}$$

Note that π_j is the prior distribution of y and $\alpha_{ij} := E[\mathbb{1}(z_i = j)]$ can be written as $p(z_i = j|\hat{R}, \hat{t})$. Without any particular prior, we can set π_j to be a uniform distribution. (Refer [21] for incorporating non-uniform priors).

If we can take the Gaussian model for the distribution $p(x_i|y_j, R, t)$ with the covariance C_{ij}, then the problem of maximizing L_{ec} over R and t is equivalent to minimizing the generalized distance as, Cij：高斯分布的方差

那不就是CPD么?

$$\min_{R,t} \sum_{i=1}^{N} \sum_{j=1}^{N_y} \alpha_{ij} (Rx_i + t - y_j)^\top {C_{ij}}^{-1} (Rx_i + t - y_j). \tag{4}$$

The ICP variants share the goal (i.e., to find at least a local optimal R and t) but have different strategies on how to determine α_{ij} and C_{ij}. Let us review some of them from special cases to more general ones.

A. Special Case I: Standard ICP

The standard algorithm uses ${C_{ij}}^{-1} = I$ and hard-assigns a single correspondence as a delta function, $\alpha_{ij} = \delta(j - k)$ for $k = \arg\min_j \|x_i - y_j\|$. delta function: 0/1

B. Special Case II: Point-to-Plane ICP

The point-to-plane ICP algorithm only takes account of the projected distance along the surface normal. This is based on the idea that the points are actually representing a *surface*. Thus, the covariance can be seen as,

valuable:
1. 解决共性问题;
2. 开发可复用模块;
3. 对传统问题有新见解;
4. 推动原有问题向前发展;

$$ {C_{ij}}^{-1} = M_{y_j}^\top \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} M_{y_j} = \bar{n}_j \bar{n}_j^\top , $$

这个表达是只取法向量;切向的不管

where M_{y_j} is an orthogonal basis around the point y_i with the third row corresponding to the local surface normal $\bar{n}_j^\top$. This is known to converge very quickly [4] [22], but having this covariance is equivalent to having the ranges of local plane extended to infinity and can cause unstable convergence for many cases. faster but narrow converge basin

Model是fixed set, observed set 是moving set;

C. General Distance: Generalized-ICP Variants

The Generalized-ICP (G-ICP) [13] provides a unified model of the error metric from a probabilistic perspective. Also, it allows decomposition of the errors into those generated by the model and by the data.

$$C_{ij} = RC_{x_i}R^\top + C_{y_j} \tag{5}$$

Each point p may have the local covariance of Σ_p, which can be transformed into the world coordinate as $C_p = M_p \Sigma_p M_p^\top$. Although we have a nice framework to work with, there have been only a few studies that mentioned principled ways to determine the covariances, including plane-to-plane ICP [13], PCA-based or Voronoi-based anisotropic ICP [16]. In this work, instead of manual design or computationally expensive analysis, we suggests a new method that *learns* the covariance.

D. General Assignment: EM-ICP Variants

While G-ICP is about the error metric, the EM-like point of view generalizes the assignments of correspondence between the two point sets. Thus, α is not a delta function anymore but it has a distribution of the form, $0 \leq \alpha_{ij} \leq 1$ and $\sum_i \alpha_{ij} \leq 1$. Practically, it results in finding correspondences on some smoothed surface. (Refer to [18] for further reading.)

IV. Learning Anisotropic Distance

Now that we have examined the role of the generalized distance parameter C_{ij}, let us proceed to introduce the idea of learning anisotropic distance. Assume we have obtained reasonably good pose estimates $\hat{R}$ and $\hat{t}$ and consider the uncertainty in the local coordinate. Let $d_{ij} = M_{y_j}(\hat{R}x_i + \hat{t} - y_j)$ and $C_{ij} = M_{y_j}\Sigma_{ij}M_{y_j}^\top$. Now we want to know Σ_{ij}'s, if any, that is most likely given the converged data and the transformation estimate. To this end, consider (3) as a function of Σ_{ij}

$$\begin{aligned} J(\Sigma_{ij}) :&= \sum_i \sum_j \alpha_{ij} d_{ij}^\top \Sigma_{ij}^{-1} d_{ij} \\ &+ \sum_i \sum_j \alpha_{ij} \ln |\Sigma_{ij}|. \end{aligned} \tag{6}$$

Minimizing this over Σ_{ij} will provide the empirical distribution of the points. Let us call it Σ_{obs} with the subscript to denote it is an observed value. This is the information we utilize to improve the current C_{ij}'s for the next round of ICP.

Yet, we should be careful about our interpretation of Σ_{obs}. First, let us consider what shape the covariances would have. A model point will have some variance σ_y on the tangential plane and some smaller variance σ_{yn} along the normal direction as Fig. 2 (a) shows. On the other hand, a measurement of a point is regarded as being be noisy in any direction. An isotropic Gaussian with σ_x may still be a reasonable model for it. Second, the x_i's variance tangential to the local plane of y_j will never be captured by Σ_{obs}, regardless of whether it is smaller or larger than y_j's. This can be seen because if the tangential component of Σ_{obs} is

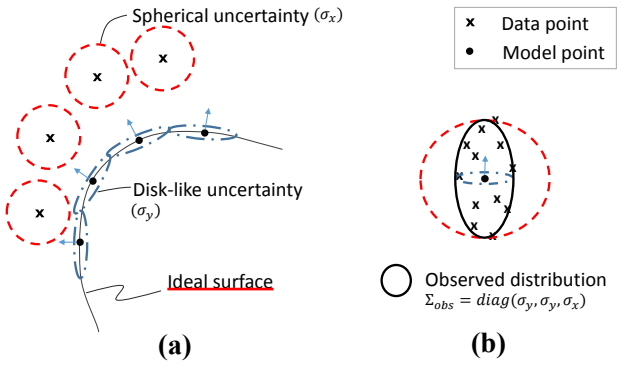


Fig. 2. (a) Model and Data Point Uncertainty Representation: Model points are disk-like distributions parameterized by σ_y. Data points are spherical distributions with uncertainty σ_x (b) Observed Covariance: After converged correspondences are established, we can estimate the data noise (σ_x) from the distribution of associated data points along the normal of the local coordinate frames and σ_y from the tangential distribution. Refer the text for details. (Best viewed in color)

larger than σ_{y_j}, some x_i may become assigned to another y point and thus it would not have the same a_{ij}'s, which would be a contradiction. Hence, the normal variance of Σ_{obs} is due to the data points and the tangential variance is due to the model points (Fig. 2 (b)).

Let us look at the simplest example. Let $d_i = M_{z_i}(\hat{R}x_i + \hat{t} - y_{z_i}) = [\ d_{i,1} \quad d_{i,2} \quad d_{i,3}\ ]^\top$ and take $\alpha_{ij} = \delta$ and $\Sigma_{ij} = \Sigma$ for simplicity. [1] Then we obtain Σ_{obs} as,

$$\begin{aligned} \Sigma_{obs} &= \arg\min_\Sigma \sum_i d_i^\top \Sigma^{-1} d_i + N \ln |\Sigma| \\ &= \frac{1}{N}\sum_i d_i d_i^\top. \end{aligned} \tag{7}$$

With this insight explained, we can project the observed covariance onto the diagonal matrix space to have $\hat{\sigma}_x^2 = \sum_i {d_{i,3}}^2/N$ and $\hat{\sigma}_y^2 = \sum_i ({d_{i,1}}^2 + {d_{i,2}}^2)/2N$. Then we obtain,

$$\hat{\Sigma} = diag(\hat{\sigma}_y^2 + \hat{\sigma_x}^2, \hat{\sigma}_y^2 + \hat{\sigma_x}^2, \hat{\sigma_{yn}}^2 + \hat{\sigma_x}^2). \tag{8}$$

from Eq. 5. We may assign $\sigma_{yn} = r\sigma_y$ for some $r \ll 1$.

Let us take a more general example. Consider the same assignment ($\alpha_{ij} = \delta$) but $\Sigma_{ij} = \Sigma_j = \Sigma_x + \Sigma_{y_j}$ where Σ_{y_j} could vary for each j. From the empirical distribution along the local normal directions, $\hat{\sigma}_x$ may be estimated the same as the previous example. On the other hand, $\hat{\sigma}_{y_j}$ could be estimated over time (k) as $\hat{\sigma}_{y_j}^2 = \sum_{k'}\{d_{i,1}(k')^2 + d_{i,2}(k')^2\}/2N$ for $k' \in \{k | z_i(k) = j\}$. Then $\hat{\Sigma}_j$ can be obtained in the same way as Eq. (8).

Now we are left to update the covariance for the next round in an online fashion. In general, it can have the form,

$$\Sigma \leftarrow (1-w)\Sigma + w\hat{\Sigma}. \tag{9}$$

As an example, we can take a learning model with a weight parameter $w = 1/\min(m, k)$ for the time step $k = 1, 2, \ldots$ and the windowed-memory size $m \geq 1$.

[1] Note that Σ_{ij} is not necessarily constant or diagonal in general and α_{ij} can be generalized as well.

V. Results

A. Experiment

We evaluated our method on a subset of two large-scale object recognition datasets, the BigBIRD dataset [23] and the YCB dataset [24]. They include the RGB-D data, processed model, and pose, of hundreds of objects, Although the number of candidate objects were very large, a subset was chosen because many of them have similar or same shapes. Also, very small objects, transparent objects, or revolved symmetric objects were excluded. We tried to include a variety of shapes since the performance of ICP could largely vary depending on specific shapes. The target objects used in our evaluation are shown in Fig. 3.

We took a sequence of point cloud data of each object taken by a static camera while the object is rotated (Fig. 4). The camera view angles θ_c were roughly $\theta_c = (c-1)\pi/8$ for $c = 1, 2, \ldots, 5$, ranging from 0 to $\pi/2$, as provided in the dataset. Even though it is assumed that the algorithm starts given a reasonable initial pose as in other locally optimal ICPs, we perturbed the initial poses with a certain range of variability. Specifically, we added uniform rotation angle and translation error to the ground truth pose. The sampling ranges for two different conditions were 10mm/10deg and 20mm/20deg. Every experiment was repeated 10 times to avoid bias.

The following error metrics are considered to measure robustness and efficiency:

- Number of iteration (count)
- Failure rate (%)

The number of iterations matters because it shows how fast each algorithm converges as well as whether it converges. We defined a state as converged when one of the following happens: (1) the larger change in rotation or translation estimate is smaller than a certain threshold ($1e^{-4}$) or (2) the value of change itself is small enough ($5e^{-4}$) and the rate of change is slow enough ($5e^{-4}$).

In order to evaluate robustness, three types of failure were examined: (1) failure to satisfy the convergence criteria before 100 iterations, (2) rotation error larger than 1 deg, or (3) translation error larger than 1 cm. The translational failure criteria was used in [17], and we extended this to iteration and rotational error. We found these metrics useful to observe the performance of ICP, not only because the difference of pose errors of different methods were subtle for converged cases[2], but also because diverged cases render the statistics of mean and variance of error less informative. Fig. 5 shows visual examples of converged data and diverged data.

For comparison, we chose three other methods: standard ICP, point-to-plane ICP, Anisotropic-ICP based on PCA, which are labeled as 'pt2pt', 'pt2pl', and 'PCA' respectively in Fig. 6 through Fig. 8. The PCA-based covariance matrix computation was implemented by us as explained in [16]. [3] We did not consider the plane-to-plane ICP because we only took data as a point cloud without further processing such as computing surface normal. Other aspects than modeling the point uncertainty to represent the shape for solving Eq. (4) were kept simple in our experiment. For instance, we took hard assignments using the kd-tree structure [25] [26] for finding correspondences and we set a threshold for the maximum distance for correspondences to prevent extreme outliers.

We tested the simplest example of LA-ICP as explained in Section IV, with memory size $m = 10$ (fading memory) for the update Eq. (8). The initial covariances of our methods were set to be the same to standard ICP. All of the ICP variants in this test were implemented mainly in C++ and partially in Matlab. We used linearized least square estimation to solve Eq. (4) for all methods. Absolute computation time was not of our interest for evaluation in this study because it will depend on specific implementations.

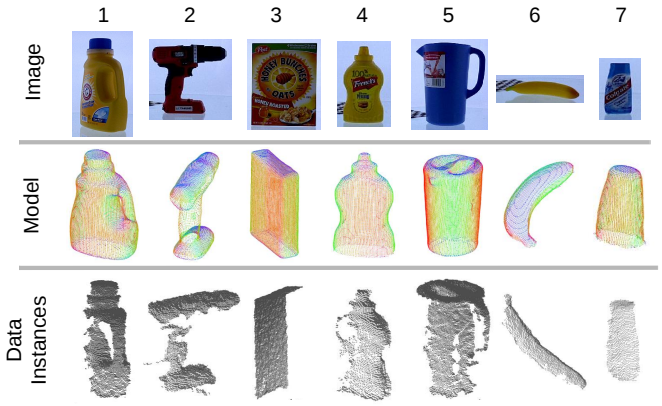


Fig. 3. Chosen Test Objects: Various shapes were used to examine the performance. The object names from the dataset labels are 1-detergent [23], 2-black and decker lithium drill driver unboxed [24], 3-honey bunches of oats honey roasted[23], 4-frenchs classic yellow mustard 14oz [24], 5-rubbermaid ice guard pitcher blue [24], 6-melissa doug farm fresh fruit banana [24], and 7-colgate cool mint [23]

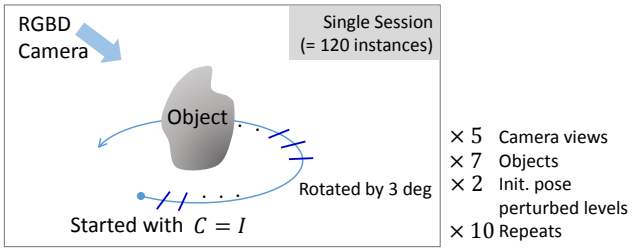
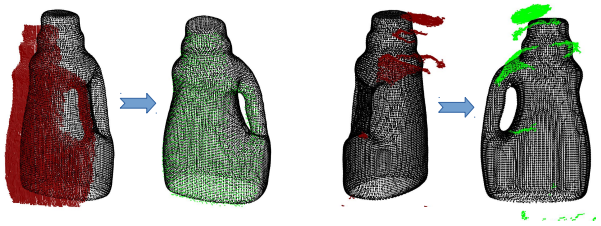


Fig. 4. Experiment Design: We tested thousands of point cloud data instances of the chosen objects from varying view points and with perturbed initial poses.

[2]For curious readers, the median rotation errors of the five methods were 0.0402 (deg), 0.0370, 0.0357, and 0.0300 for 20mm/20deg noise in the order of pt2pt, pt2pl, PCA, and LA. The median translation errors were 5.72 (mm), 5.93, 5.64, and 5.57 for 20mm/20deg noise in the same order.

[3]It should be noted that their experiment included PCA on the model set and the data set, whereas ours only applies to the model set. (As mentioned in Introduction, we treat the data set as only points.) To decide the data point noise for this method, we tested constant isotropic covariances of 1,2,3 mm and final results are from the best case, 2mm.



(a) (b)

Fig. 5. Visual Examples of Initial Poses (Red) and Convergence (Green in (a)) and Failure (Green in (b)) (Best viewed in color)

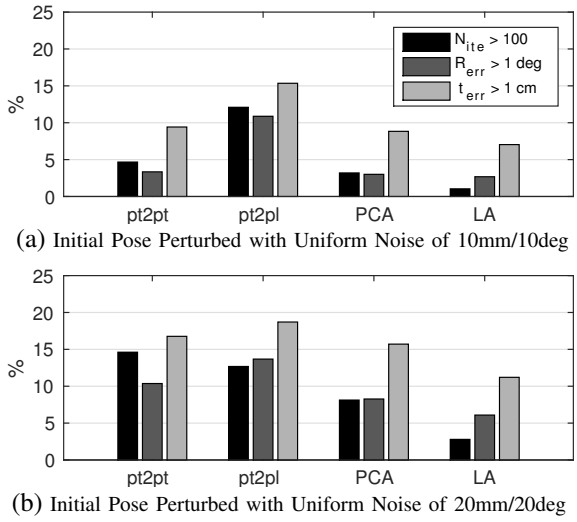
B. Discussion

The overall performance is shown in Fig. 6 and Fig. 7. First, Fig. 6 (a) and (b) show the failure rate for two sets of different initial pose variability. The black columns indicate how frequently it did *not* converge before the maximum iteration number (100). The dark (or light) gray ones represent the percentage of the pose estimate errors *larger* than $1\,\mathrm{deg}$ (or 1 cm). These two figures shows that the LA-ICP performed better than the others in all three terms of robustness. The failure rates of all methods increased as the initial pose error level increased, but the LA-ICP still outperformed the others.

Next, the number of iterations for the two sets are shown in Fig. 7. As mentioned it could vary from case to case, but the mean and median values tell how many iterations each method needed for typical converged cases. The statistics shows that LA-ICP required a smaller number of iterations on average with smaller variance. It is worth noting that the point-to-plane ICP either converged very quickly (Fig. 7) or did not converge. This unstable behavior of point-to-plane ICP resulted in higher total failure rates as shown in Fig. 6.

Beside the evaluation for all data, we were able to observe what affected the performance while analyzing individual object cases. As shown in Fig. 8, a dominant tendency we observed across all objects was the convergence rates in decreasing order, pt2pl, pt2pt, PCA, and LA-ICP. The failure rates of object No. 1 (detergent, Fig. 8 (a)) and 4 (mustard) displayed similar results. One interesting observation within the same two objects was that the failure rate tended to increase with the elevation angle. The performance seems to be degraded because the smaller number of observed points led to fewer constraints, which then led to unstable registration. This is also reflected by the smallest object, object No. 7 (toothpaste, Fig. 8 (c)). The point-to-plane ICP even failed to converge up to 40% of all the instances for this object and other methods' performance was also relatively poor.

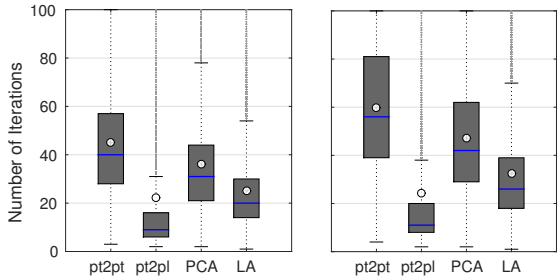
In addition, as for object No. 3 (cereal box, Fig. 8 (b)), all methods had relatively small failure rates ($\leq 5\%$) of iteration and rotation error criteria, but larger rates of translational error ($\geq 10\%$ for all but LA-ICP). Apparently, the shape factor, i.e., flat planes, seemed to allow slipping along the symmetric planes and caused larger translational errors. This kind of unstability of ICP has been analyzed in [27].



(a) Initial Pose Perturbed with Uniform Noise of 10mm/10deg

(b) Initial Pose Perturbed with Uniform Noise of 20mm/20deg

Fig. 6. Overall Failure Rates: LA-ICP showed the least failure rate even with the increased perturbation of initial pose.



(a) 10mm/10deg (b) 20mm/20deg

Fig. 7. Number of Iterations Statistics (best viewed in color): Blue lines are median values and circles are mean values. LA-ICP shows stable convergence tendency with a smaller variance in number of iterations. The point-to-plane ICP was fast if it ever converged but failed to converge in many cases (Fig. 6).

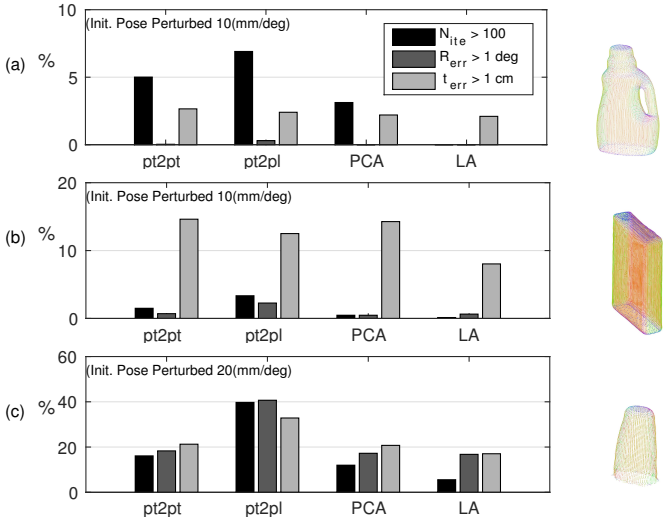


Fig. 8. Failure Rates for Individual Objects: Having symmetric or small partial views, some objects were not favorable to the ICP algorithm. Although LA-ICP was also affected, it still outperformed other methods in most cases.

Nonetheless, LA-ICP exhibited the most stable and robust performance relative to the other algorithms, both for difficult objects and under increased perturbation in initial poses. We attribute the improved performance to covariances that better represent the spatial uncertainty of given model points and the measurement noise of data points. In order to provide a sense of what values LA-ICP actually learned, $\sigma_y^2 = 0.5442, \sigma_x^2 = 1.626$ is an example for object No. 1 (detergent) with the third camera view. Note that the covariance varied since it was updated online after every convergence. Thus, we see that parameters of the error metric affect the convergence trend and the estimated result. LA-ICP was able to learn them from previously converged data, which sped up the tardy convergence of standard ICP without the unstable behavior that point-to-plane ICP demonstrates. Also, LA-ICP handled the uncertainty term for the data set without guessing or expensive processing, which the PCA-based method required.

As for possible extensions, we can apply the learning method to the individual point covariance of each model point as explained in Section IV, or even to generalized correspondences. In our preliminary test, LA-ICP also have shown improved robustness in ICP with soft assignments.

VI. CONCLUSIONS

We have proposed LA-ICP, a new online learning approach for ICP that improves registration performance. We described how the suggested LA-ICP determines the anisotropic covariance parameter from converged data and gave simple implementation examples. The robustness and efficiency of this method was demonstrated on tens of thousands of point cloud instances that span a wide range of view angles of daily objects. In the experiments, LA-ICP showed the most stable performance over other methods for determining the parameter, even under increased initial pose perturbation and decreased number of data points.

In the future, we will investigate the effect of learning memory size of LA-ICP under various dynamic view changes. Also, it will be interesting to combine our learning scheme for determining the point uncertainty with other strategies, for example, trimming outliers or using other features.

REFERENCES

[1] K. Lai, L. Bo, X. Ren, and D. Fox, A large-scale hierarchical multi-view rgb-d object dataset, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[2] P. J. Besl and N. D. Mckay, A method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 14(2), pp. 239–256, 1992.

[3] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D Mapping: Using depth cameras for dense 3D modeling of indoor environments. In *the Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2010.

[4] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, Comparing ICP variants on real-world data sets, *Autonomous Robots*, 34(3), pp. 133–148, 2013.

[5] C. Choi and H. I. Christensen, 3D pose estimation of daily objects using an RGB-D camera, In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.

[6] S. Rusinkiewicz and M. Levoy, Efficient variants of the ICP algorithm, In *Proceedings of the International Conference on 3-D Digital Imaging and Modeling (3DIM)*, 2001.

[7] Y. Chen and G. Medioni, Object modeling by registration of multiple range images, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1991.

[8] G. C. Sharp, S. W. Lee, and D. K. Wehe, ICP registration using invariant features, *IEEE Transactions on Pattern Analysis and MachineIntelligence (PAMI)*, 24(1), pp. 90–102, 2002

[9] K. Arun, T. Huang, and S. Blostein, Least-squares fitting of two 3D point set, *IEEE Transactions on Pattern Analysis and MachineIntelligence (PAMI)*, 9(5), pp. 698–700, 1987.

[10] B. Horn, Closed-form solution of absolute orientation using unit quaternions, *Journal of the Optical Society of America A*, 4(4), pp. 629–642, 1987.

[11] K. Low, Linear least-squares optimization for point-to-plane ICP surface registration, *Tech. Report TR04-004 University of North Carolina*, 2004.

[12] R. B. Rusu and S. Cousins, 3D is here: Point Cloud Library (PCL), in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[13] A. Segal, H. Dirk, and S. Thrun, Generalized-ICP, In *Proceedings of Robotics: Science and Systems (RSS)*, 2009.

[14] R. S. J. Estépar, A. Brun, and C. F. Westin, Robust generalized total least squares iterative closest point registration, In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2004.

[15] M. F. Hansen, M. R. Blas, R. Larsen, Mahalanobis distance based iterative closest point, In *Proceedings of SPIE 6512*, 65121Y, 2007.

[16] L. Maier-Hein, A. M. Franz, T. R. dos Santos, M. Schmidt, M. Fangerau, H. Meinzer, and J. M. Fitzpatrick, Convergent iterative closest-point algorithm to accomodate anisotropic and inhomogenous localization error, *IEEE Transactions on Pattern Analysis and MachineIntelligence (PAMI)*, 34(8), pp. 1520–1532, 2012.

[17] S. D. Billings, E. M. Boctor, and R. H. Taylor, Iterative Most-Likely Point Registration (IMLP): A Robust Algorithm for Computing Optimal Shape Alignment, *PLoS ONE 10(3): e0117688*, doi:10.1371/journal.pone.0117688, 2015.

[18] S. Granger, X. Pennec, and A. Roche, Rigid point-surface registration using oriented points and an EM variant of ICP for computer guided oral implantology, *RR-4169, INRIA*, 2001.

[19] S. Granger and X Pennec, Multi-scale EM-ICP: A fast and robust approach for surface registration, In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002.

[20] C. Bishop, *Pattern Recognition and Machine learning*, Springer, 2006.

[21] B. Combès and S. Prima, An efficient EM-ICP algorithm for symmetric consistent non-linear registration of point sets, In *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*, 2010.

[22] H. Pottmann, Q. Huang, Y. Yang, and S. Hu, Geometry and convergence analysis of algorithms for registration of 3D shapes, *International Journal of Computer Vision*, 67(3), pp.277–296, 2006.

[23] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, Bigbird: A large-scale 3d database of object instances, In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[24] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, Benchmarking in Manipulation Research: The YCB object and model set and benchmarking protocols, in *IEEE Robotics and Automation Magazine*, 22(3), pp. 36–52, 2015.

[25] J. L. Bentley, Multidimensional binary search trees used for associative searching, *Communications of the ACM*, 18(9), pp. 509–517, 1975.

[26] J. Elseberg, S. Magnenat, R. Siegwart, and A. Nüchter, Comparison of nearest-neighbor-search strategies and implementations for efficient shape registration, *Journal of Software Engineering for Robotics*, 3(1), pp. 2–12, 2012.

[27] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, Geometrically Stable Sampling for the ICP Algorithm, In *Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2003.