2018 IEEE International Conference on Robotics and Automation (ICRA)
May 21-25, 2018, Brisbane, Australia

AA-ICP: Iterative Closest Point with Anderson Acceleration

Artem L. Pavlov[1], Grigory V. Ovchinnikov[2], Dmitry Yu. Derbyshev[3],
Dzmitry Tsetserukou[4], and Ivan V. Oseledets[5]

***Abstract*— Iterative Closest Point (ICP) is a widely used method for performing scan-matching and registration. Being simple and robust, this method is still computationally expensive and may be challenging to use in real-time applications with limited resources on mobile platforms. In this paper we propose a novel effective method for acceleration of ICP which does not require substantial modifications to the existing code.**

This method is based on an idea of Anderson acceleration which is an iterative procedure for finding a fixed point of contractive mapping. The latter is often faster than a standard Picard iteration, usually used in ICP implementations. We show that ICP, being a fixed point problem, can be significantly accelerated by this method enhanced by heuristics to improve overall robustness. We implement proposed approach into Point Cloud Library (PCL) and make it available online. Benchmarking on the real-world data fully supports our claims.

I. Introduction

Localization is one of the fundamental problems of modern robotics. Robots commonly use the laser scanner data presented in the form of point clouds. To relate one scan to another scan-matching algorithms are used with Iterative Closest Point [1] being one of the most popular approaches. Scan-matching problems also arise outside of the robotics domain, for example, in the context of medical image comparison and registration [2], [3], for which ICP is widely used as well.

In practice, different modifications are implemented to speed up the matching process and to improve the ICP reliability (see section II-A). Nevertheless, underlying iterative structure of ICP in general remains almost unchanged. In this paper, we propose to accelerate ICP through modification of iteration procedure, so we can keep all benefits from the above-mentioned modifications, thus making ICP even faster. Instead of using "state-less" approach which depends only on the last iteration, the proposed idea is to select the next iteration point based on solution of the specific optimization problem over the history of previous iterations. The optimization problem itself is quite simple and, therefore, can be solved on any hardware, which is capable of running ICP in the first place.

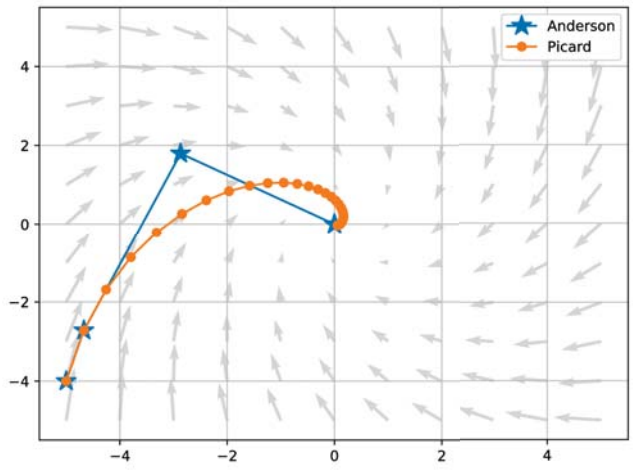

Fig. 1: Comparison of Picard iteration procedure and Anderson acceleration for simple linear 2D mapping $G(u) = A \cdot u$ (translations only). Both iteration processes start from point $(-5, -4)$. Stars and circles denote iteration points, and grey arrows visualize mapping in different points.

This paper is organized as follows. In section 2 we provide theoretical background by describing basic form of ICP and Anderson acceleration (AA), and then highlight the main theoretic properties for both approaches. Next, we propose modifications necessary to make Anderson accelerated version of ICP (AA-ICP) work with the real world data, which often violates underlying assumptions. In section 3 we provide experimental results, which prove that our modification of ICP with AA achieves significant speed-up (more than 30%) and the slight error improvement (0.3%) on the dataset from [4].

In this work we focus on 3D datasets, but proposed method can be applied to 2D ICP variants as well.

We implement AA-ICP as part of the widely used Point Cloud Library (PCL), and the source code is freely available in our fork repository[1]. Finally, section 4 summarizes the paper.

II. Background

There are many variations of ICP, e.g. presented in [5], [6] and [7], but for our discussion it is important that ICP

*The research was supported by RSF (project No. 17-11-01376)

[1]A. L. Pavlov is a PhD student with Space Center, Skolkovo Institute of Science and Technology, Skolkovo Innovation Center, Moscow 143026 Russia artem.pavlov@skolkovotech.ru

[2]G. V. Ovchinnikov is a Research Scientist with the Center for Computational and Data-Intensive Science and Engineering, Skolkovo Institute of Science and Technology

[3]D. Yu. Derbyshev is a PhD student with Moscow Institute of Physics and Technology

[4]D. Tsetserukou is an Assistant Professor with Space Center, Skolkovo Institute of Science and Technology

[4]I. V. Oseledets is an Associate Professor with the Center for Computational and Data-Intensive Science and Engineering, Skolkovo Institute of Science and Technology

[1]https://github.com/SkoltechRobotics/pcl/tree/anderson

essentially boils down to a fixed point problem:

$$u = G(u), \tag{1}$$

which is usually solved with the simple (or also called Picard) iteration procedure: $u^{k+1} = G(u^k)$. Here and thereafter $u \in \mathbf{R}^n$ describes the roto-translation between two scans (u can be represented by the translation coordinates and Euler angles for rotation).

Anderson acceleration [8] (also known as Anderson mixing in computational chemistry) is a different method of finding u^{k+1} based on the history of $m+1$ latest iterations and residuals. In case of $m = 0$ it is equivalent to Picard iteration.

In general, AA is theoretically and practically superior to Picard in many cases [9], [10] and additional cost of selecting different points is equal to solving least squares problem of size m, which for ICP is negligible compared to the cost of a single iteration.

A. Iterative Closest Point

ICP algorithms are often used in robot navigation for performing scan-matching of data provided by LIDARs, RGB-D cameras or stereo-cameras. Thus, it is quite common underlying algorithm in different simultaneous localization and mapping (SLAM) setups as discussed in [11] and in a two-part survey [12], [13].

ICP operates on two sets of points S and S^{ref}, where $S_i, S_j^{ref} \in \mathbf{R}^n$ (usually $n = 2$ or $n = 3$) , with initial proper rigid transformation guess u^0. The simplest algorithm variant [1] can be described as follows:

1) Transform S using u^0.
2) For every point in S find the closest point in S^{ref}, pairs of such points are called correspondencies.
3) Find such transform u which minimizes the mean distance between correspondencies (i.e. error).
4) Apply transformation u to S.
5) If change in the error falls below a given threshold - then terminate; otherwise - go to the step 2.

As mentioned before, various modifications are usually used in practice, such as:

- Different metrics (point-to-plane, feature based).
- Usage of indexes (e.g. K-d trees).
- Dynamic caching.
- Point sampling.
- Random restarts.
- And others.

But in general those modification still use basic Picard iteration.

B. Anderson acceleration

We notice several properties of ICP, which make finding fixed point of (1) computationally expensive and hard to accelerate with higher-order methods.

The main problem is the unavailability of derivatives because function G in ICP case is not even differentiable. While this limitation can be side-stepped with continuous relaxation (e.g. in [14] authors apply Newton-Raphson method to ICP formulated in terms of differentiable energy functions), even with continuous relaxation, computation of gradients is often prohibitively expensive. Moreover, continuous relaxation is both hard to implement from scratch and to apply to the existing code, e.g., to PCL[2] or CSM[3], which are widely used in the community. The other option could be the usage of numerical differentiation, but its computational cost is also high. Additionally, due to noise in the input data finite-difference approximation of gradients becomes unreliable.

As was outlined above, in this work we propose to use Anderson acceleration to speed up the convergence process of ICP. For differentiable functions it is analogous to pseudo-Newton methods [15] and in the linear case it is equivalent to GMRES [16].

In our perspective, AA is the best iteration scheme we could employ here: it almost always requires less iterations to converge to the same error than simple iteration; it does not require additional calls to the expensive and memory demanding ICP function; it relies on the history of iterations alone with overhead being negligible compared to the cost of a single ICP iteration step. Last, but not least, it could be trivially added into the existing ICP implementations, thus many existing implementations of ICP could potentially benefit from it.

The simple variant of AA is represented by algorithm 1.

Algorithm 1: Anderson acceleration

Data: initial guess u^0, contraction mapping G, maximum iterations limit n_{max}
Result: fixed point u
$g^0 = G(u^0)$;
$f^0 = g^0 - u^0$;
$u^1 = g^0$;
for n *in* $1..n_{max}$ **do**
  $f^n = G(u^n) - u^n$;
  Find $\alpha \in \mathbf{R}^{n+1}$ which minimizes $\| \sum_{j=0}^{n} \alpha_j f^j \|_2$ subject to $\sum_j \alpha_j = 1$;
  $u^{n+1} = \sum_{j=0}^{n} \alpha_j G(u^j)$;
  if *convergence criteria is true* **then**
    break;
  end
end
return u^n;

Here $a..b$ denotes a sequence of integers from a (including) to b (excluding), therefore $1..4$ implies the following sequence: 1, 2, and 3.

The minimization problem for α can be reformulated as a linear least-squares problem. Constraint can be replaced by substitution $\alpha_0 = 1 - \sum_{j=1}^{n} \alpha_j$, which leads to the following

[2] http://pointclouds.org/
[3] ttps://censi.science/software/csm/

unconstrained problem:

$$\min_{\alpha_1,\ldots,\alpha_n} \|f_0 + \sum_{i=1}^{n} \alpha_i (f_i - f_0)\|.$$

The behavior of AA for 2D translations (without rotation) is presented in Fig. 1. It clearly showcases "jumpy" nature of the algorithm, which enables much faster convergence compared to the simple Picard iteration. Such jumps can be viewed as attempts to predict the most plausible convergence point based on the history of previous iterations.

C. Heuristics

The main problem of pure AA is the serious instability when working with non-contractive mappings, which is quite common when processing real-world data. For example, existence of several convergence points automatically makes related mapping non-contractive. Even the existence of a single convergence point does not generally make mapping contractive.

Due to this instability, in practice we cannot use simple AA demonstrated in alg. 1. Usually various heuristics and modifications can be added, such as: limiting history length; introduction of dumping factors; linear search using easier to compute but less precise approximations. In this work we propose the specific set of heuristics developed considering ICP properties.

Our method with incorporated heuristics can be represented by alg. 2.

In this algorithm we use u for denoting concatenated vector of translation coordinates and Euler angles for rotation, e.g. $u = (x, y, z, \phi, \theta, \psi)$, thus for 3D case $u \in \mathbf{R}^6$ and for 2D case $u \in \mathbf{R}^3$. Due to assumption that two given scans are spatially close, singularity in Euler representation of rotation does not impact us. It could be argued that reduction of roto-translation for 3D case to such vector and operating with it as described here is not strictly correct, since the addition of rotations is not a commutative operation. Unfortunately, to our best knowledge, generalization of AA to roto-translation groups does not yet exist. Nevertheless, experiments show that for small rotation angles (up to 10 degrees in our work) use of Euler angles produces good results.

The first heuristic in the algorithm checks whether the error estimate returned by ICP step is not considerably larger than the error for previous iteration (the most common error estimate is the mean distance between correspondences). If this condition is true, we fully reset the history of iterations and start the process from the last known "trustworthy" point. It's a safeguard against cases, when the second heuristic fails to filter out bad jumps. This heuristic results in "empty" iteration, i.e., only error estimate is used from ICP step call, but such cases are relatively rare ($3 - 5\%$ of iterations) and associated costs are out-weighted by provided robustness.

The second heuristic loops over the incremented history length, which is used for AA and checks whether computed alphas fall within the specified range. Similar heuristics are widely used and in some cases they have theoretical guaranties [17].

Algorithm 2: AA-ICP

Data: initial guess u^0, history length limit m, alpha limit α_l, maximum iterations limit n_{max}
Result: convergence point u^{n+1}

```
h = 0;                    // history cut-off cursor
g^0 = G(u^0);
f^0 = g^0 - u^0;
u^1 = g^0;
for n in 1..n_max do
    g^n = G(u^n);
    f^n = g^n - u^n;
    if ICP error is too big then
        h = n;            // "resetting" history
        u^{n+1} = g^{n-1};
        continue;
    end
    u^{n+1} = g^n;
    for i in 1..min(m, n - h) do
        α_{1,...,n} = min ||f_0 + Σ_{j=1}^{n} α_j (f_j - f_0)||;
        α_0 = 1 - Σ_{j=1}^{n} α_j;
        if (-α_l ≤ α_j ≤ α_l, ∀j) and (α_0 > 0) then
            u^{n+1} = Σ_{j=0}^{i} α_j g^{n-j};
        else
            break;
        end
    end
    if convergence criteria is true then
        break;
    end
end
return u^{n+1};
```

It also checks whether $\alpha_0 > 0$, thus ensuring that jumps occur in general direction defined by the last G call. In case those conditions are not met, the last result is returned. This way only points most probably residing inside a local contractive area are selected for the calculations. It also ensures that the next jump does not occur in the opposite direction relatively to the last ICP step.

The overall cost of solving several small linear systems is negligible compared to the cost of G, as in practice i in alg. 2 rarely exceeds 5-10.

III. Experimental Results

To measure the performance of AA-ICP, we implemented it as a part of Point Cloud Library (PCL). In our work AA-ICP works with point-to-point ICP. The modified code is freely available in our fork repository[4] and is intended for the inclusion into the upstream.

We used two datasets for our experiments: "RGB-D SLAM Dataset and Benchmark" [4] and the "Stanford Bunny" [18]. The first one was collected using RGB-D camera and the second one – with the laser range scanner.

[4] https://github.com/SkoltechRobotics/pcl/tree/anderson

Fig. 2: Frame examples from Freiburg dataset

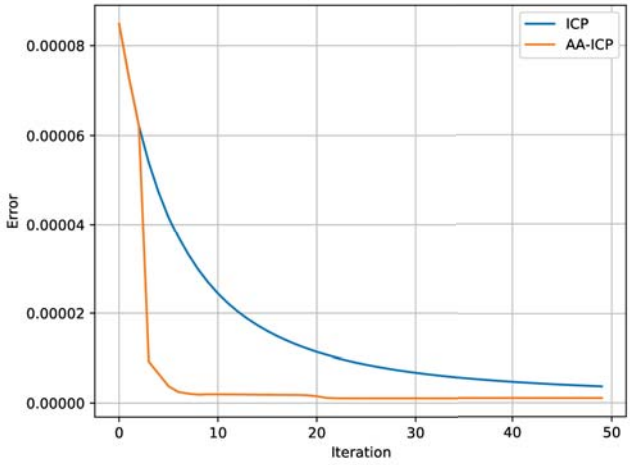

Fig. 3: Example of error behaviour for simple ICP and AA-ICP.

A. RGB-D SLAM Dataset and Benchmark

The following sequences from freiburg1 set were used: room, desk, and xyz. To emulate keyframes we matched not subsequent frames, but the 5th scan from the current one. In total, 2738 scan pairs were processed without any filtering or sampling.

The examples of RGB images are presented in Fig. 2. In our experiments only depth channel was used.

Example of error behaviour for AA-ICP and default PCL ICP is shown in Fig. 3. The error is estimated as the sum of the differences between correspondences in an Euclidean sense, divided by the number of correspondences. Both methods generally converge to the same point.

However, these criteria in approximately 1% of cases terminated AA-ICP too early, thus algorithm was improved by requiring the convergence criteria to hold true for two iterations in a row. It adds one iteration to AA-ICP (although with smarter convergence criteria it can be removed). Nevertheless, even with such handicap, statistically AA-ICP converged faster than simple ICP. There was one exception though: if criteria were satisfied earlier than 4th iteration, the confirming iteration was not required, since this indicated that camera movement between scans was negligible. The maximum number of iterations was limited to 100.

Fig. 4 demonstrates the statistical properties of acceleration over the number of iterations required for convergence,

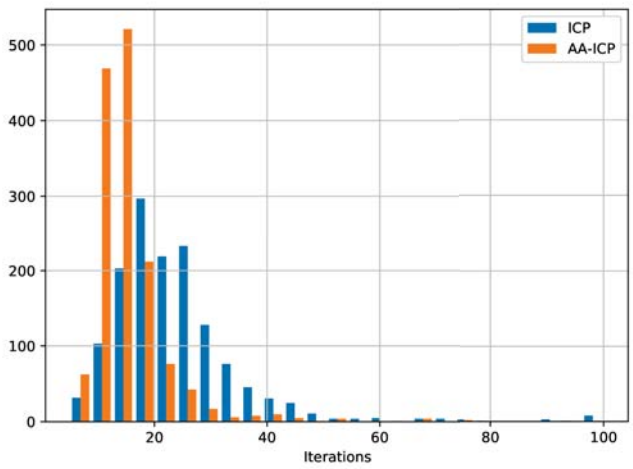

Fig. 4: The number of iterations required for the simple ICP and AA-ICP to converge

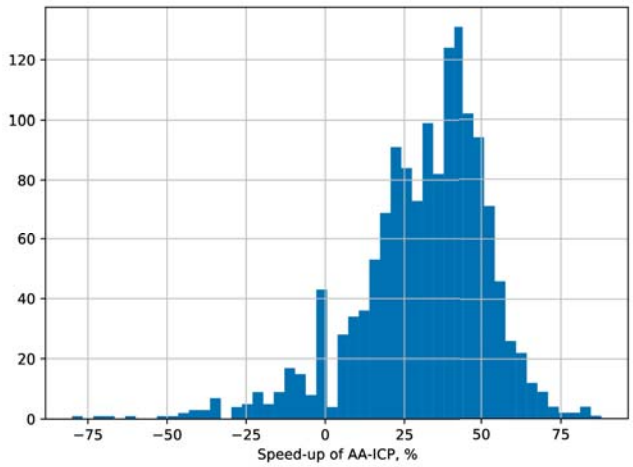

Fig. 5: AA-ICP speed-up relative to the simple ICP

which were calculated for $\varepsilon = 0.001$ [5] and $\alpha_l = 10$. Fig. 5 shows the relative change between the number of iterations for the same pair of scans. Median speed-up in this case is equal to 35%, mean to 30%, and overall more than 90% of AA-ICP runs got accelerated relative to the simple ICP.

In addition to the smaller number of iterations required for convergence, AA-ICP generates results with better quality. This can be seen from Fig. 6, which depicts relative final errors produced by AA-ICP and the simple ICP. More than 97% of runs converged to smaller errors with AA-ICP. The median improvement is equal to approximately 0.3% and mean to 0.4%. Note that convergence criteria modification does not provide a big contribution to those values, as a single iteration of nearly converged AA-ICP usually results in error change with order of magnitude equal to $0.001\% - 0.01\%$.

Finally we demonstrate the dependence between ε and

[5] See documentation for `setEuclideanFitnessEpsilon` method of `pcl::Registration` class

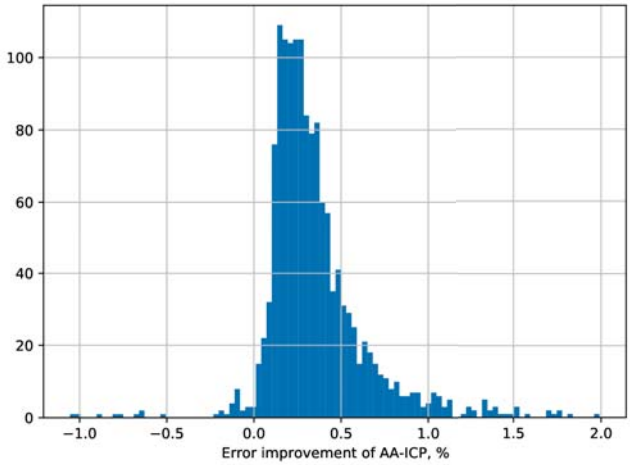

Fig. 6: Final error improvement of AA-ICP relative to the simple ICP

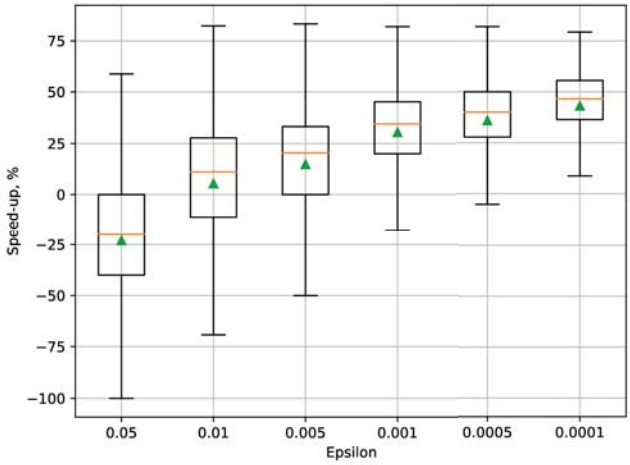

Fig. 7: Box-plot of relative AA-ICP speed-up depending on ε value for RGB-D dataset. Triangle mark denotes mean value.

acceleration improvement in Fig. 7. As we can see, relative speed-up of AA-ICP increases with rising quality criteria for final results, while with overly-relaxed convergence criteria AA-ICP results in bigger number of iterations.

B. Stanford Bunny

To test acceleration properties of AA-ICP depending on misalignment of scans we chose to use the "Stanford Bunny" [6], probably one of the most well known 3D test models taken by laser scanner. You can see its photo in Fig. 8 and point clouds in Fig. 9. It provides highest quality of ground truth data for performing initial scan alignment.

In this work we used two scans taken under 0 and 45 degrees (each contains approximately 40000 points). First, we aligned them, then applied random translations and rotations, and measured acceleration of the AA-ICP compared

[6] http://graphics.stanford.edu/data/3Dscanrep/
[7] https://youtu.be/fkEA1H76uCg

Fig. 8: Photo of "Stanford bunny"

Fig. 9: Visual comparison of convergence speed for ICP and AA-ICP. Green point clouds show scan taken under 0 degrees and red under 45 degrees. Both methods have started from the same roto-translation shift for the red point cloud. On 8th iteration AA-ICP performed long jump in the correct direction. Other iterations can be seen in the video attachment[7].

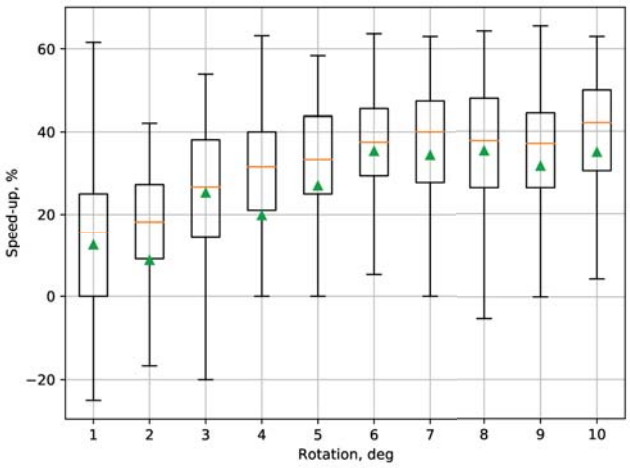

Fig. 10: Box-plot of relative AA-ICP speed-up versus random rotation angle for partial bunny scans.

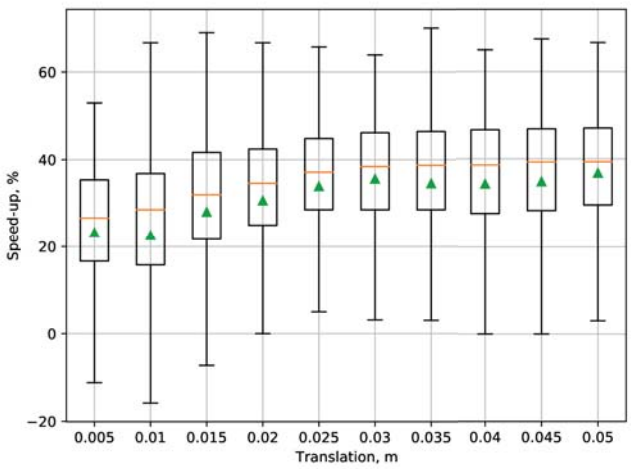

Fig. 11: Box-plot of relative AA-ICP speed-up versus random translation distance for partial bunny scans.

to unmodified ICP. The same parameters have been used as in the previous section: $\varepsilon = 0.001$ and $\alpha_l = 10$.

To test the acceleration properties of AA-ICP we introduced 1000 random iterations per given degree and recorded relative acceleration compared to simple ICP. The results of this experiment can be seen in Fig. 10. As we can see, relative speed-up of AA-ICP rises up to 5 degrees and then stays on the slope of approximately 35%.

Next experiment was conducted for translations. In it we introduced 1000 random translations per given distance. Results are shown in Fig. 11, they look quite similar to the experiment with the random rotations. Here, relative speed-up rises up to 2.5 cm and then stays on the slope of approximately 35%.

This behaviour can be explained by the fact that the bigger misalignment the more iterations are needed to perform scan-matching, thus, more room exists for AA-ICP to show its acceleration property. Otherwise, with small number of iterations (be it due to small initial misalignment or relaxed convergence criteria) its behaviour will be more similar to the simple ICP.

IV. Conclusions

In this work we proposed and analyzed AA-ICP, the novel modification of Iterative Closest Point algorithm based on Anderson acceleration. This method can be easily applied to existing implementations and has negligible runtime cost. It substantially reduces the number of iterations required for achieving desired scan-matching quality compared to the unmodified version.

We also implemented AA-ICP as a part of Point Cloud Library and benchmarked it against unmodified ICP on real-world data, acquired by RGB-D camera and laser scanner. In addition to the successful demonstration of the acceleration properties of AA-ICP, we showed that with the same convergence criteria our algorithm converges to the better final point. Thus for the same quality of convergence it can be used with less strict convergence criteria.

Further work will be done on the following topics: improving heuristics; benchmarking on other datasets and ICP modifications; applying method to 2D case; and development of better theoretical foundation of Anderson acceleration for roto-translation group with not-fully contractive mappings.

Acknowledgment

The research was supported by RSF (project No. 17-11-01376).

The authors are also grateful to Dr. Evgeny Mironov from Center for Design, Manufacturing and Materials, Skolkovo Institute of Science and Technology, and Pavel Karpyshev.

References

[1] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):239–256, February 1992.

[2] Charles V Stewart, Chia-Ling Tsai, and Badrinath Roysam. The dual-bootstrap iterative closest point algorithm with application to retinal image registration. *IEEE Transactions on Medical Imaging*, 22(11):1379–1394, 2003.

[3] Yaorong Ge, Calvin R Maurer Jr, and J Michael Fitzpatrick. Surface-based 3d image registration using the iterative closest-point algorithm with a closest-point transform. In *Medical Imaging 1996*, pages 358–367. International Society for Optics and Photonics, 1996.

[4] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

[5] Dmitry Chetverikov, Dmitry Svirko, Dmitry Stepanov, and Pavel Krsek. The trimmed iterative closest point algorithm. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 3, pages 545–548. IEEE, 2002.

[6] Andrea Censi. An icp variant using a point-to-line metric. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 19–25. IEEE, 2008.

[7] Hao Men, Biruk Gebre, and Kishore Pochiraju. Color point cloud registration with 4d icp algorithm. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1511–1516. IEEE, 2011.

[8] Donald G Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM (JACM)*, 12(4):547–560, 1965.

[9] Homer F Walker and Peng Ni. Anderson acceleration for fixed-point iterations. *SIAM Journal on Numerical Analysis*, 49(4):1715–1735, 2011.

[10] Alex Toth and CT Kelley. Convergence analysis for anderson acceleration. *SIAM Journal on Numerical Analysis*, 53(2):805–819, 2015.

[11] Udo Frese. A discussion of simultaneous localization and mapping. *Autonomous Robots*, 20(1):25–42, 2006.

[12] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[13] Tim Bailey and Hugh Durrant-Whyte. Simultaneous localization and mapping (slam): Part ii. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[14] Peter Biber and Wolfgang Strasser. nscan-matching: Simultaneous matching of multiple scans and application to slam. In *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pages 2270–2276. IEEE, 2006.

[15] Haw-ren Fang and Yousef Saad. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications*, 16(3):197–221, 2009.

[16] Florian A Potra and Hans Engler. A characterization of the behavior of the anderson acceleration on linear problems. *Linear Algebra and its Applications*, 438(3):1002–1011, 2013.

[17] S.A. Matveev, V.I. Stadnichuk, E.E. Tyrtyshnikov, A.P. Smirnov, N.V. Ampilogova, and N.V. Brilliantov. Anderson acceleration method of finding steady-state particle size distribution for a wide class of aggregationfragmentation models. *Computer Physics Communications*, 224:154 – 163, 2018.

[18] Greg Turk and Marc Levoy. The stanford bunny, 2005.