An Adaptive Differential Evolution Algorithm with a Point-Based Approach for 3D Point Cloud Registration

Linh Tao[1], Trung Nguyen[1], Tinh Nguyen[1], Toshio Ito[2], and Tam Bui[1,2]
[1] School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi, Vietnam
[2] Department of Machinery and Control Systems, Shibaura Institute of Technology, Japan
Email: linh.taongoc@hust.edu.vn, tambn@shibaura-it.ac.jp

***Abstract*—This paper presents a novel pairwise registration approach, which aligns images of the same object that have different ranges. By using a point search medium instead of a conventional six-dimensional parameter to reduce the number of search dimensions, the new method resulted in a higher convergence rate and robustness in the same search conditions. The approach integrated a hybrid registration strategy, a combination of Iterative Closest Point (ICP) as a local aligning tool and a global search algorithm such as simulated annealing, particle swarm optimization, differential evolution, etc. An adaptive differential evolution algorithm called ISADE was chosen as the best-so-far global search algorithm. Different experiments on different datasets were carried out. In the new method, as compared with the conventional approach, better aligning results in convergence rate and robustness were observed.**

***Index Terms*—hybrid registration, 3D registration, ICP, global optimization algorithm, point-based registration**

I. Introduction

Commercial depth camera productions have moved computer vision research from using 2D images to 3D images or depth data since sufficient information from the surrounding environment enables us to handle complicated tasks more easily. 3D data have become popular in controlling mobile robots and object reconstruction. In mobile robot mapping, 3D data captured from different positions are aligned and used to create a large map. This map could be used for controlling and positioning mobile robots to perform their tasks. For a 3D object reconstruction problem, 3D images of an object are taken from different angles. Those images go through aligning steps if camera positions and transformation are not available, and they result in a completed single 3D object. The process of aligning similar parts of images is known as image registration. Rotation and translation movements of sensors are calculated through movements that align overlapped regions of images. This paper presents work on 3D pairwise registration. From two point clouds, the reference and the current, a pairwise registration algorithm estimates a transformation, which moves the current toward the reference and causes the two datasets to overlap at similar regions.

Iterative Closest Point (ICP) [1], EMICP [2], and generalized-ICP [3] as its variants are key methods in 2D and 3D registration. ICP methods (ICPs) often use L_2 (RMSE) error as in Equation (1) to derive a transformation, which reduces the error between two point clouds. To achieve a final small error, numbers of iterations are applied. The most challenging task is how to calculate numerous L_2 errors in a reasonable amount of time. ICPs are local registration methods with the drawback of requiring a further assumption in which the initial positions of two datasets are close enough to guarantee a correct alignment. The closer the gap between the two datasets, the higher the rate at which ICPs result in correct aligning transformations. Otherwise, the ICPs are often trapped in local convergences. Generally, the registration process is divided into coarse and fine steps. The coarse finds the rough alignment, which brings two point clouds close enough together. Then the fine step completes the fine alignment. The former remains a challenge with two approaches: local and global.

Local methods use local features [4] such as PFH [5] and SIFT [6] descriptions. The coarse movement can be estimated when enough corresponding key points appear in both the reference and current point cloud. Sample consensus algorithms such as RANSAC [7] are widely used in this case to find those corresponding point pairs.

Global methods such as Go-ICP [8] or SAICP [9] use global search algorithms and all points of data instead of feature points only. Global methods are difficult to apply to real-time application because of the high complexity in computing a large number of points in one step. By using other methods such as multilayer and approximation corresponding search, researchers have achieved some improvement. Moreover, integrating global search algorithms and using powerful Graphics Processing Units (GPUs) [10] helps to solve a global registration problem in a reasonable amount of time. Different research with a global approach using a hybrid mechanism between global optimization algorithms and ICP achieved promising results. SAICP uses Simulated Annealing (SA) [11] and Go-ICP [12] uses Branch-and-Bound (BnB) to search for global convergence.

Manuscript received September 7, 2021; revised December 21, 2021.

doi: 10.18178/joig.10.1.1-9

This paper proposes a novel hybrid point-based global registration mechanism using points as search variables. It divides alignment movements into two parts corresponding to two search variables. In this way, the number of search dimensions is reduced from six conventional parameter-based methods to two. This significantly improved the global convergence rate and robustness. As in other hybrid methods, ICP worked as a local convergence estimator, cost function calculator. It also completed the final fine registration steps.

Various experiments were carried out to prove the advantages in terms of accuracy and robustness by using the hybrid-point based integrated with the improved self-adaptive differential evolution (ISADE) algorithm.

II. An Overview of 3D Global Point Cloud Registration

A. Iterative Closest Point (ICP)

From the L2 error of two datapoints, a displacement that might reduce the error is derived. It moves the current $X = \{x_i\}$ to the reference $Y = \{y_j\}$, where x_i, y_j are points in the current and the reference respectively. A rotation matrix $R \in SO^3$ and a translation matrix $t \in R^3$ minimize the L_2 error in Equation (1):

$$E(R,t) = \sum_1^n \left|Rx_i + t - y_{j*}\right|^2 \quad (1)$$

where y_{j*} is the corresponding closest point of x_i in the current Y , with j∗ is as shown in Equation (2).

$$j* = \min_{j \in \{1,\dots,n\}} \left|Rx_i + t - y_j\right| \quad (2)$$

The flowchart is shown in Fig. 1 in iterations to achieve the final transformation. There are different ICP based methods using different approaches in determining corresponding point pairs to calculate $E(R,t)$ such as LMICP [13] and SICP [14]. LMICP uses distance from the point in the current scan to a tangent plane of points in the reference scan allowing scans to "slide" against each other. However, in some cases, where there are few geometric constraints such as tunnels or corridors, point-to-plane ICP can wrongly converge with too much sliding. It is a benefit to weight the point pairs by assigning more weight for pairs that are more likely to correspond to each other. However, like point-to-plane ICP, in tunnel or corridor cases, wall or ceiling points are weight higher while corner, far distance, or other feature point influences are overwhelmed. In trying to generalize ICP with a possible error model, the generalized error in Equation (1) becomes a generalized ICP error.

ICP spends most of its calculation time for nearest neighbor searching. A brute-force neighbor search method runtime is of $O(mn)$ complexity with number of points in the reference m and number of points in the current n.

Normally, the points in the reference are stored and constructed in a KD-tree [15] with expected runtime for one neighbor search reduced to $O(n \log m)$ plus $O(m \log m)$ for building a KD-tree. Using a KD-tree structure, a greater improvement could be achieved especially for a greater number m. However, ICP's total runtime remains large. A recent approximate nearest neighbors' search replacing the original KD-tree reduces this burden to some degree. ICP is used in the well-known accumulation registration method KinectFusion, in which every two consecutive frames' transformation is calculated continuously. The current transformation matrix is a multiplication of matrices from the previous registration steps.

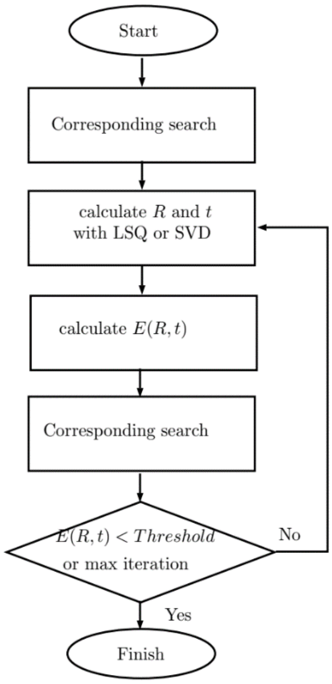

Figure 1. ICP algorithm implementation steps.

B. Global Hybrid Approach

A global search algorithm itself has the ability to search for the global optimal solution. However, to lighten the workload for a global search, in the hybrid approach, ICP works as a fitness function simplifier. Fig. 2 and Fig. 3 show ICP effects by turning a black complex fitness function into a simple red one. With a simpler fitness function, a search algorithm has a higher success rate in finding the global convergence point. The transformation for the current is calculated as in Equation (3).

$$T = T_{ini} * T_{icp} \quad (3)$$

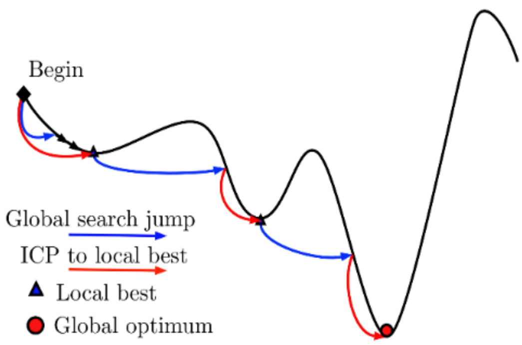

Figure 2. Global hybrid search strategy.

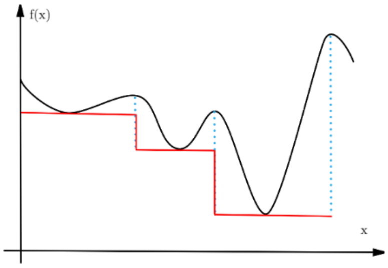

Figure 3. ICP simplification for objective function.

C. Current Hybrid Methods Using Global Heuristic Search Algorithms on Six Dimension

In Equation (3), T_{icp} is calculated in ICP loops. The initial transformation matrix T_{ini}, including one rotation and one translation, is derived as in Roll–Pitch–Yaw movements as in Equation (4):

$$T_{ini} = \begin{bmatrix} R_{\alpha\beta\gamma} & & & t_{xyz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

where $\alpha; \beta; \gamma$ are rotation angles and $x; y; z$ are translation variables.

Simulated annealing simulates a cooling process in which a new trial is created based on a temperature variable. At the beginning, the temperature value is set at a high level, which enables the algorithm to create trials far away from the current point. Temperature reduces in a cooling procedure and the algorithm stops at maximum generation. The trend is mostly downhill with reducing in the cost function value, but in some cases uphill steps are acceptable. When temperature reaches a minimum value, the algorithm results in its best-so-far convergence. SA is employed in SAICP, a hybrid registration method. SA uses a single candidate in one generation and creates the next generation candidate from the current one. Metaheuristic algorithms use multiple candidates at one generation (it) and create new ones for the next generations from combining current generation candidates.

Differential Evolution (DE) is simple, reliable, and robust. It is one of the most effective global search algorithms. Using the same process as other EAs, unlike other EAs, the DE [16] type algorithms scale current generation populations for creating candidates for the next generation. Those populations can be selected in all members or in mixing elite population members, which derived good results in the current generation. New populations in DE could be created in different mutation schemes. Many DE variants perform better than the original [17], the ISADE [18], [19] method is an example. In [20], ISADE achieved significantly better results than DE in registering 3D range images. By using an adaptive mechanism in different search parameters including the number of populations, crossover, and scaling factor parameters, ISADE converges on good results without tuning search parameters.

By applying wild animal behavior in groups, Particle Swarm Optimization (PSO) [21], [22] creates a mechanism to make both individual and group perform better after generations. The PSO principles are observed in living species and in human societies. People want to improve themselves; we change our ways toward ones that work better. The changing of individuals is influenced by their inertia, a small neighborhood group affects the whole group. This affect drives individuals to test a different possibility for a better result. In [23], Wang *et al.* used PSO in their method to solve the Simultaneous Localization and Mapping (SLAM) task in the mobile robot area.

III. Point Based Method Overview

A. Point-Based Approach Methodology

In the Interactive Point Cloud Registration (IPCR) [24] method, users choose two corresponding points which appear in both point clouds, the rest of the registration work relies on the ICP algorithm. The idea is to break down T_{ini} in Equation (3) into smaller simple parts. The initial movement can be broken down into two steps: the first step makes two corresponding points coincide and the second step makes a final move for the current using ICP. Adding one rotation movement into the first step of IPCR, our registration movement is shown in Fig. 4 with the red reference and the blue current point clouds. Instead of only one degree of freedom as in IPCR, our method has two degrees of freedom, which is enough for aligning point clouds at any initial position.

- **Step 1:** Normal vector alignment. The movement for the current point cloud is a translation and a rotation combination. The translation matrix is as in Equation (5):

$$t_0 = [x_i - x_j \quad y_i - y_j \quad z_i - z_j]^T \tag{5}$$

The rotation moves normal vector n_d to align with vector n_m. The rotation and translation matrices are:

$$R_1(j)= \begin{bmatrix} u^2 + (v^2 + w^2)cos\phi & uv(1 - cos\phi) - wsin\phi & uw(1 - cos\phi) - vsin\phi \\ uv(1 - cos\phi) + wsin\phi & v^2 + (u^2 + w^2)cos\phi & vw(1 - cos\phi) - usin\phi \\ uw(1 - cos\phi) - vsin\phi & vw(1 - cos\phi) + usin\phi & w^2 + (u^2 + v^2)cos\phi \end{bmatrix} \tag{6}$$

$$t_1(j)= \begin{bmatrix} \big(a(v^2 + w^2) - u(bv + cw)\big)(1 - cos\phi) + (bw - cw)sin\phi \\ \big(b(u^2 + w^2) - v(bv + cw)\big)(1 - cos\phi) + (cu - aw)sin \\ \big(c(v^2 + u^2) - w(bv + cw)\big)(1 - cos\phi) + (av - bu)sin\phi \end{bmatrix} \tag{7}$$

$$T_1(j)= \begin{bmatrix} R_1(j) & & & t_1(j) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where $[a, b, c] = [x_{x_i}, y_{x_i}, y_{x_i}]$, $[u \quad v \quad w]^T$ is the normalized vector $(\overrightarrow{n_d} \times \overrightarrow{n_m})$, and ϕ is the angle between vector $\overrightarrow{n_m}$ and $\overrightarrow{n_d}$.

- **Step 2:** Directional alignment. The current is rotated about the normal vector $\overrightarrow{n_m}$) θ angle. The transformation matrix is $R_2(\theta)$ and $t_2(\theta)$ calculated in the same way as above $R_1(j)$ and $t_1(j)$ with $[u \quad v \quad w]^T = \overrightarrow{n_m}$ and $\phi = \theta$.

$$T_2(\theta) = \begin{bmatrix} R_2(\theta) & & & t_2(\theta) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Now, the initial transformation matrix as in Equation (3) includes two variables as in Equation (10).

$$T = T_1(j) * T_2(\theta) \tag{10}$$

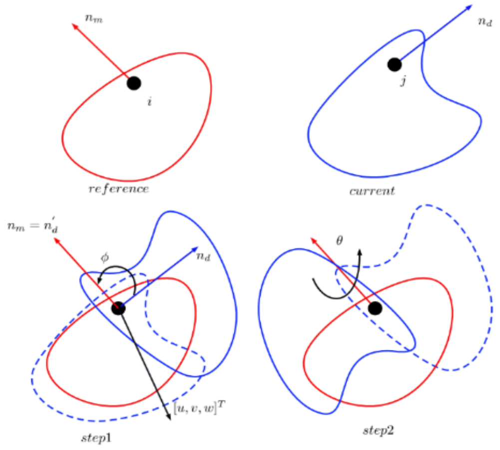

Figure 4. Aligning point clouds in two steps.

The implementation is shown in algorithm 1 which is a common implementation for hybrid registration such as PSO-ICP, SA-ICP, and SaEvo [25].

Algorithm 1:

1: **Procedure** Search algorithm.
2: Initialize Xi = with the center point of the reference.
3: Initialize for populations with random values of $(\theta;\ j)$
4: **while** (Not reached stop criterion) do
5: **for** the whole populations do
6: Move the data surface to model surfaces using point-based steps. After applying ICPs, remaining error is calculated.
7: **end for**
8: Sort all populations in decreasing error order.
9: Update the best-so-far solution until the current step.
10: Update for the next generation population from the current generation using suitable search strategies (SA, PSO, DE, ISADE, etc.).
11: **end while**
12: **end procedure**
13:
14: **procedure** Fine registration.
15: Use ICP for best-so-far-solution from the above search procedure.
16: **end procedure**

B. Robust Objective Function For Global Registration

The objective function has the global smallest value at the correct alignment of two point clouds. We used a modified fitness function $F\ (R;\ t)$ derived from $E(R;\ t)$, similar to [20].

$$F(R,t) = f(k)E(R,t) \tag{11}$$

The function $f(k)$ reduces the cost function at a larger number of inlier cases as in Equation (12). It also eliminates an incorrect convergence possibility where L_2 is small but the number of inliers is small too.

$$f(k) = \begin{cases} \infty & if\ k < n/10 \\ 1 - \frac{k}{n} & if\ k > n/10 \end{cases} \tag{12}$$

here, n is the number of points in the current and k is the number of inliers.

IV. Experiment and Results

A. Experimental Setup

In this section, 3D data were scanned; surface data were available on the internet. The first group of 3D scanned data was from the Stanford 3D Scanning Repository website http://graphics.stanford.edu/data/3Dscanrep. Armadillo, Dragon, Stanford Bunny, and Happy Buddha scans were used. The second group was in the Queen's Range Image and 3-D Model Database which is available at: https://code.engineering.queensu.ca/rcvlabdatabases/qr3d. Old Gnome and Angel scans were used. The scanned data used in experiments are shown in Fig. 5. The experiments aimed to show the advantage of the new approach in coarse registration in comparison with the conventional six-dimensional method on different scanned data. To enable algorithms to finish in a reasonable amount of time on a normally-equipped computer, all input data were subsampled, and the size reduced to 2000 points. In ICP loops, only 100 points in the current point cloud were used to significantly reduce runtime. Fig. 6 shows the subsampled input cloud data.

Figure 5. Stanford and Queens objects.

Figure 6. Stanford and Queens subsampled scans.

In the point-based approach, all points in the current were set in the search pool, the angle θ search range was $[-\pi/2; \pi/2]$. Meanwhile, the search ranges in conventional methods were set at $[-\pi/2; \pi/2]$ and $[-0.15; 0.15]$ for $(\alpha;\ \beta;\ \gamma)$ and $(x;\ y;\ z)$, respectively. All algorithm codes were written in C++ language with reference from [26]. We used a laptop installed with an Intel core i3, 1.3 GHz processor. The results were from each combination of data scanned 500 times. The number of populations was N=50 and the number of iterations was 50 for DE, ADE, and PSO algorithms. The iteration number was set at 2500 for the SA algorithm. FLANN [26], an approximate KD-Tree structure library, was used to reduce neighbor searching time. The names of algorithms were written as follows:

- PSA-ICP and SA-ICP are abbreviations for point-based and original combinations of SA and ICP, respectively.
- PPSO-ICP and PSO-ICP are abbreviations for point-based and original combinations of PSO and ICP, respectively.
- PDE-ICP and DE-ICP are abbreviations for point-based and original combinations of DE and ICP, respectively.
- PADE-ICP and ADE-ICP are abbreviations for point-based and original combinations of Adaptive DE (ISADE) and ICP, respectively.

Parameters for each algorithm are in Table I.

TABLE I. PARAMETERS FOR GLOBAL SEARCH ALGORITHMS

SA	DE	ISADE	PSO
T_0=100 β= 0.003	F_0= 0.8 C_r= 0.9	F_{max}= 1.6 F_{min}= 0.4 n_{min}= 0.2 n_{max}= 6.0	c_1 = 2.1 c_2 = 2.1 K = 0.5

B. *Alignment Results with Different Objects and Algorithms*

At first, different algorithms were used to register different objects. The currents and the references in all cases were at 100 percent of overlap with the arbitrary initial position, which means they were subsamples from the same scans. The box plot error of all results are shown in Fig. 7, Fig. 8.

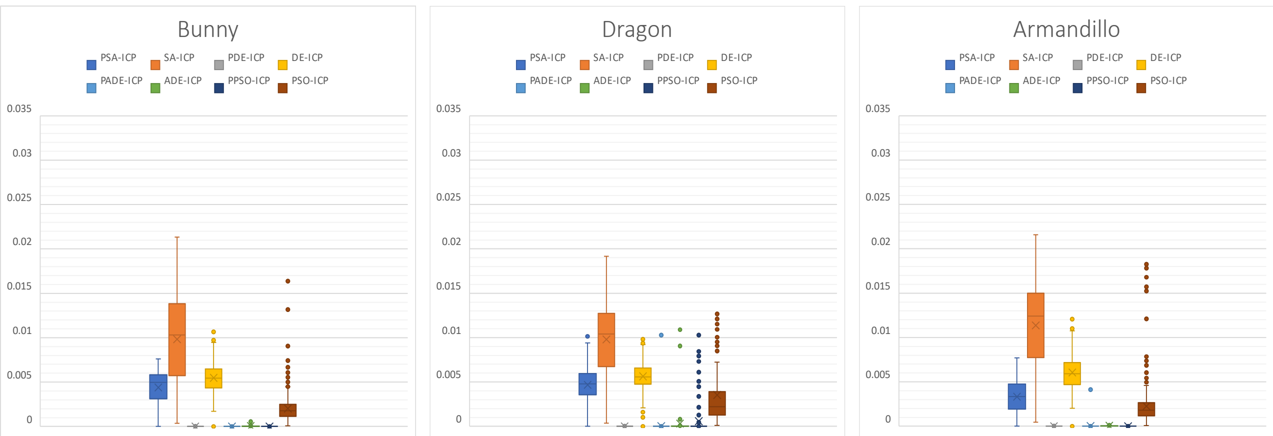

Figure 7. Results from Bunny, Dragon, and Armadillo scans on different algorithms.

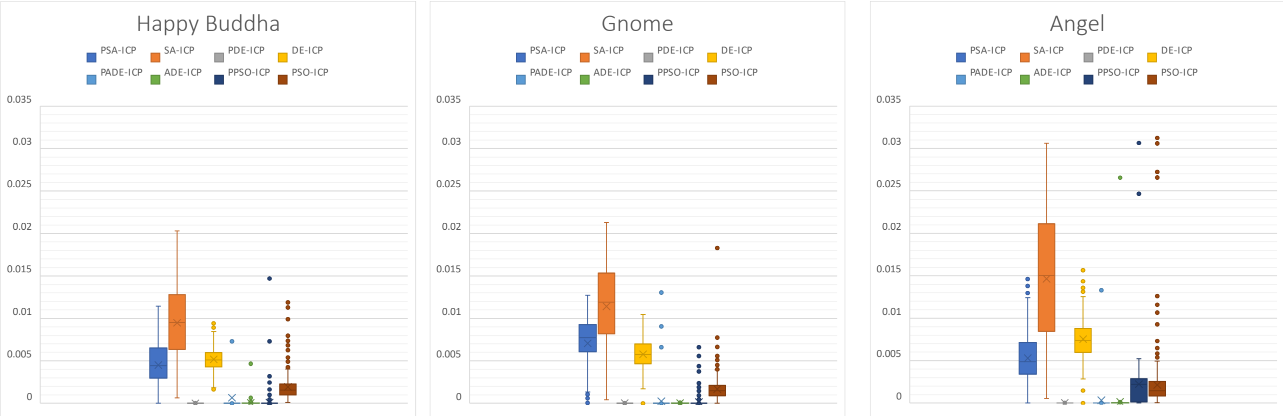

Figure 8. Results from Happy Buddha, Gnome, and Angel scans on different algorithms.

From Fig. 7 and Fig. 8, a similar result pattern of different algorithms on different scans can be observed. PDE-ICP, PADE-ICP, and ADE-ICP rank as the best algorithms; their error boxes are almost invisible in all figures of Fig. 7 and Fig. 8.

Multiple agents search methods (PSO, DE, and ADE) performed significantly better than SA, a single agent method in almost all objects.

For further comparison between the point-based approach and their counter parts, Fig. 9 and Fig. 10 show results in pairs of search algorithms.

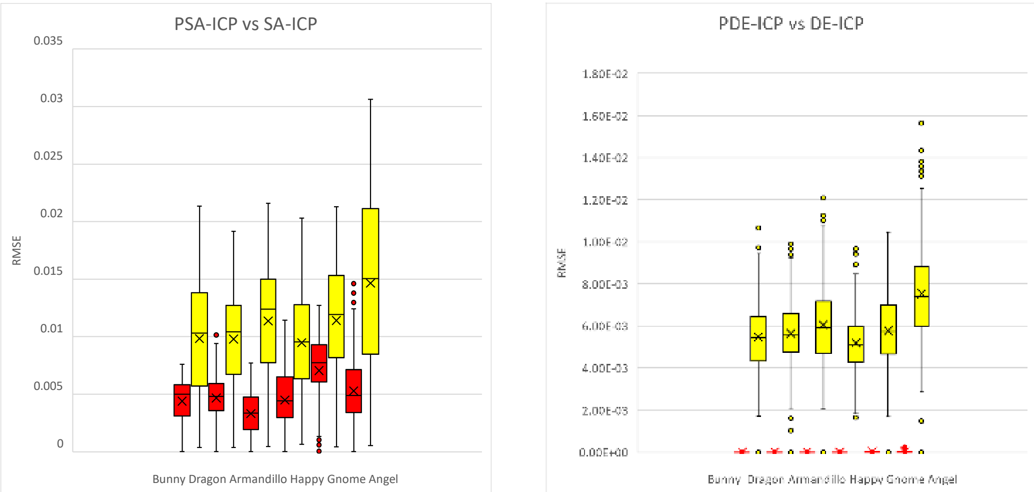

Figure 9. Comparison between PSA-ICP, PDE-ICP in red boxes and SA-ICP, DE-ICP in yellow boxes.

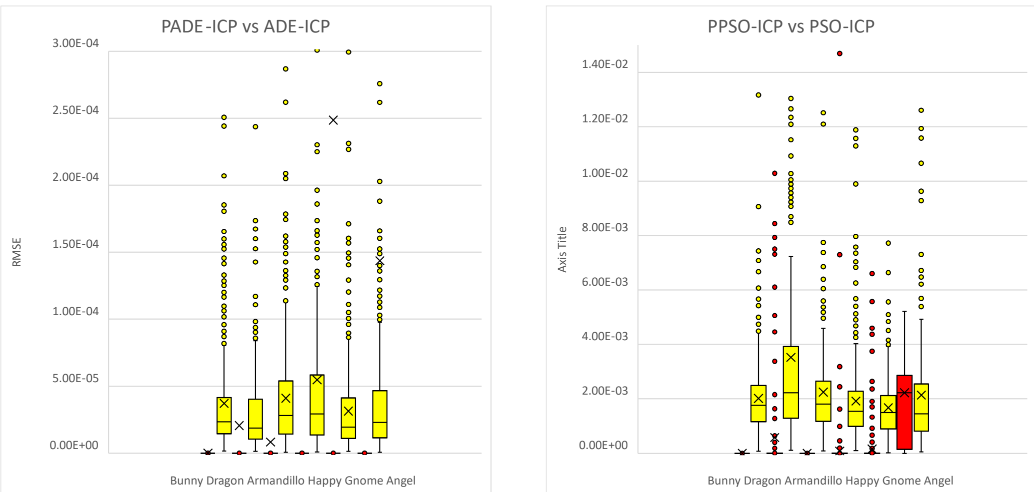

Figure 10. Comparison between PADE-ICP, PPSO-ICP in red boxes and ADE-ICP, PSO-ICP in yellow boxes.

Of all figures in Fig. 9 and Fig. 10, point-based algorithms outperformed their parameter-based ones. Only the PPSO-ICP Angel scan had worse results than PSO-ICP.

From Fig. 11, PADE-ICP and PDE-ICP both gained small final convergence results. PADE-ICP outperformed PDE-ICP with significantly smaller errors.

From all comparisons, we came to the conclusion that our chosen combination, PADE-ICP was the best-so-far point-based algorithm.

Fig. 12 shows the aligned results with different scans, mixed and matched color of currents in red and references in green dots are observed.

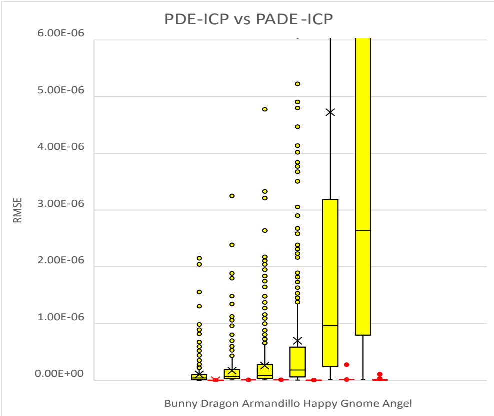

Figure 11. PDE-ICP results in yellow and PADE-ICP results in red.

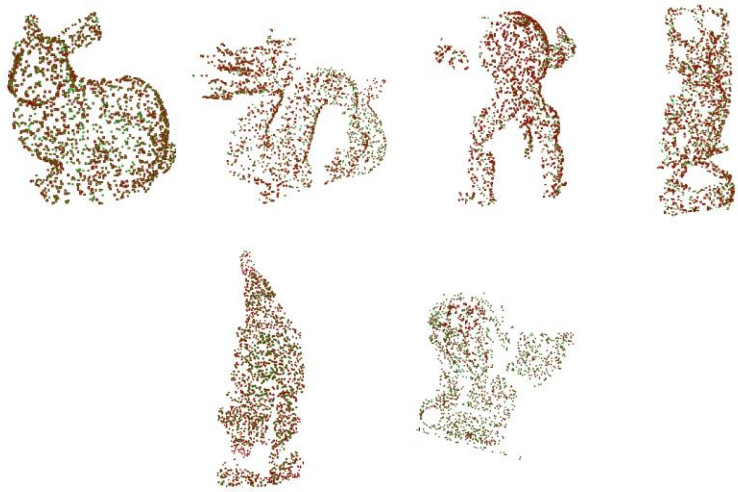

Figure 12. Aligned results with different scans, mixed and matched color of currents in red and references in green dots are observed.

C. Alignment Results with PADE-ICP at Different Overlap Rate

The new algorithm was tested with different overlap rates for different objects. The rate percentages were 75, 87, 83, and 87 for Bunny, Dragon, Armadillo, and Happy Buddha, respectively. Happy Buddha scans with smaller overlap rates of 73 and 60 were also investigated. The results are shown in Fig. 13.

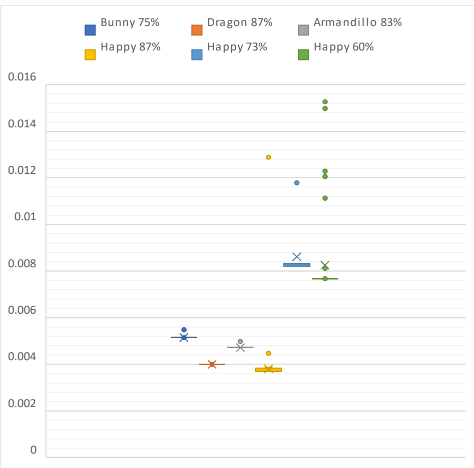

Figure 13. PADE-ICP on different objects and overlap rates.

The success rate for different scans of Bunny, Dragon, Armadillo, and Happy Buddha were 100, 100, 100 and 99.8 percent, respectively. The success convergence in the experiment means PADE-ICP achieved the smallest error.

With the same scan data of Happy Buddha, the success rates were 99.8, 89, and 85.2 percent for 87, 73, and 60 percent of overlap rate, respectively. The smaller success rate is expected as the percentage of overlap rate was reduced.

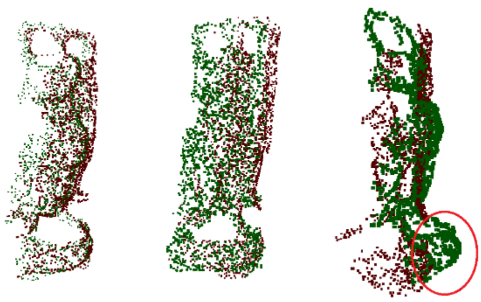

Figure 14. Alignment results at 87, 73, and 60 overlap rates from left to right. The red oval shows the region aligned apparently incorrectly.

Failed convergences were observed even with the smallest error. Fig. 14 shows the correct convergence for the Happy Buddha scan at 87 and 73 percent of overlap rate and an incorrect convergence at 60 percent of overlap rate. To remove incorrect convergences, the stronger outlier rejection for error function should be investigated in the next stage of this research. Incorrect convergence could be a result of the center point selection of the reference when it does not appear in the current. A method to choose a stable center point should be applied in future work.

D. Runtimes and Success Rate on Population Size and Iteration Number

In this experiment, Happy Buddha scans with 73 percent overlap rate were used since it showed the lowest success rate (90.2%) but was correctly aligned. The results are shown in Table II and III.

TABLE II. SUCCESS RATE OF PADE-ICP ON HAPPY BUDDHA SCAN WITH 73 PERCENT OVERLAP RATE AT 50 ITERATIONS

Population size	30	40	50	60	70
Success rate	83.4%	86.2%	89.0%	89.6%	86.4%
Runtime(s)	0.71	0.95	1.19	1.45	1.66

TABLE III. SUCCESS RATE OF PADE-ICP ON HAPPY BUDDHA SCAN WITH 73 PERCENT OVERLAP RATE AT 50 POPULATION SIZE

Iteration	30	40	50	60	70
Success rate	85.6%	87.2%	89.0%	90.2%	90.0%
Runtime(s)	0.76	0.97	1.2	1.4	1.64

From Table II and III, it is clear that an increasing number of iterations or populations does not always achieve better results. When they reach a certain number, 60 for population and 60 for iteration number, the convergence rate almost stops improving.

V. DISCUSSION AND CONCLUSION

The new method was proposed to solve a 3D point cloud data registration problem with a combination of a novel point-based approach and an adaptive differential evolution search method (ISADE). Results from various experiments on different scanned data showed improvements in the robustness and accuracy of the new approach over the conventional six-dimensional search. Moreover, using the ISADE algorithm, PADE-ICP achieved the best-so-far results.

Currently, the center point of the reference is used as the based point for searching for the corresponding point on the current. This assumption limits the overlap rate to fifty percent. The mechanism of having not only one center point should be considered. The quality of center points need further investigation. In addition, normal vectors were estimated from a number of nearest points on the subsampled data by using the SVD method. By preserving normal vectors in the subsampling step, a more correct normal vector could bring better results.

PADE-ICP worked well on low overlap rate cases and in scanned data with a high number of identical features. More investigation on the high overlap rate and low feature number registration should be carried out in the future work.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

AUTHOR CONTRIBUTIONS

Linh Tao conducted research and wrote the paper. Trung Nguyen and Tinh Nguyen performed the experiment and analyzed the data. Ito Toshio and Tam Bui wrote the paper and tested experiments. All authors approved the final version.

ACKNOWLEDGMENT

This work was supported by the School of Mechanical Engineering, Hanoi University of Science and Technology under the T2020-PC-010 project.

REFERENCES

[1] S. Wang, H. Y. Sun, H. C. Guo, L. Du, and T. J. Liu, "Multi-view laser point cloud global registration for a single object," *Sensors*, vol. 18, no. 11, 2018.

[2] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, pp. 239-256, Feb. 1992.

[3] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized ICP," *Robotics: Science and Systems*, 2009.

[4] Y. He, B. Liang, J. Yang, S. Li, and J. He, "An iterative closest points algorithm for registration of 3D laser scanner point clouds with geometric features," *Sensors*, vol. 17, no. 8, 2017.

[5] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Persistent point feature histograms for 3D point clouds," in *Proc. the 10th International Conference on Intelligent Autonomous Systems*, 2008.

[6] A. Sehgal, D. Cernea, and M. Makaveeva, "Real-time scale invariant 3D range point cloud registration," in *Image Analysis and Recognition*, Springer Berlin Heidelberg, Book Section, 2010.

[7] F. L. Wu and X. Y. Fang, "An improved RANSAC homography algorithm for feature based image mosaic," in *Proc. the 7th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision*, Athens, Greece, 2007.

[8] J. L. Yang, H. D. Li, and Y. D. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *Proc. IEEE International Conference on Computer Vision*, 2013, pp. 1457-1464.

[9] J. Luck, C. Little, and W. Hoff, "Registration of range data using a hybrid simulated annealing and iterative closest point algorithm," in *Proc. IEEE International Conference on Robotics and Automation*, 2000, vol. 4, pp. 3739-3744.

[10] D. Neumann, F. Lugauer, S. Bauer, J. Wasza, and J. Hornegger, "Real-time RGB-D mapping and 3-D modeling on the GPU using the random ball cover data structure," in *Proc. IEEE International Conference on Computer Vision Workshops*, 2011, pp. 1161-1167.

[11] L. Ingber, "Simulated annealing: Practice versus theory," *Math. Comput. Modelling*, vol. 18, pp. 29-57, 1993.

[12] C. Yu and D. Y. Ju, "A maximum feasible subsystem for globally optimal 3D point cloud registration," *Sensors*, vol. 18, p. 544, 2018.

[13] K. Low, "Linear least-squares optimization for point-to-plane ICP surface registration," Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill, February 2004.

[14] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Journal Computer Graphics Forum (Symposium on Geometry Processing)*, vol. 32, no. 5, pp. 1-11, 2013.

[15] C. Sharat, "Introduction to kd-trees," Course note cmsc420-0401, University of Maryland Department of Computer Science, 2002.

[16] R. Storn and K. Price, "Differential evolution? A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[17] M. A. Gunen, *et al.*, "Camera calibration by using weighted differential evolution algorithm: A comparative study with ABC, PSO, COBIDE, DE, CS, GWO, TLBO, MVMO, FOA, LSHADE,

ZHANG and BOUGUET," *Neural Comput. & Applic.*, vol. 32, pp. 17681-17701, 2020.

[18] T. Bui, H. Pham, and H. Hasegawa, "Improve self-adaptive control parameters in differential evolution for solving constrained engineering optimization problems," *Journal of Computational Science and Technology*, vol. 7, no. 1, 2013.

[19] T. Bui, T. Nguyen, and H. Hasegaw, "Opposition-based learning for self-adaptive control parameters in differential evolution for optimal mechanism design," *Journal of Advanced Mechanical Design, Systems, and Manufacturing*, vol. 13, no. 4, 2019.

[20] L. Tao, T. Bui, and H. Hasegawa, "Global ray-casting range image registration," *IPSJ Trans. Comput. Vis. Appl.*, vol. 9, no. 1, 2017.

[21] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, Nov./Dec. 1995, vol. 4, pp. 1942-1948.

[22] M. A. Günen, P. Ç. Beşdok, and E. Beşdok, "Keypose synthesis from 3D motion capture data by using evolutionary clustering," *Concurrency Computat. Pract. Exper*., vo. 34, no. 1, 2022.

[23] J. Wang and Y. Fujimoto, "Combination of the ICP and the PSO for 3D-SLAM," in *Proc. 43rd Annual Conference of the IEEE Industrial Electronics Society*, Beijing, 2017, pp. 8261-8266.

[24] T. J. Chin, A. Parra, B. S. Michael, and D. Suter, "Fast rotation search for real-time interactive point cloud registration," in *Proc. the Symposium on Interactive 3D Graphics*, 2014, pp. 55-62.

[25] J. Santamaria, S. Damas, O. Cordon, and A. Escamez, "Self-adaptive evolution toward new parameter free image registration methods," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 4, pp. 545-557, Aug. 2013.

[26] FLANN. Fast library for approximate nearest neighbors. [Online]. Available: http://www.cs.ubc.ca/research/flann/

Tao Ngoc Linh received his B.E. (2010) at the Hanoi University of Science and Technology, Vietnam and M.E. (2013) from the Taiwan University of Science and Technology, Taiwan. He graduated with a Ph.D. in Functional Control Systems from the Shibaura Institute of Technology, Japan. His research interests include 3D computer vision, intelligent algorithm, and autonomous vehicles localization and tracking.

Thanh-Trung Nguyen received a B.E. in 2008 at the Hanoi University of Science and Technology, Vietnam. He received an M.E. in 2012 and Dr. Eng. in 2015 in Functional Control Systems from the Shibaura Institute of Technology, Japan. His field of research in the masters and doctoral courses was Bio-science and Technology.

Since 2008, he has been a lecturer in the Department of Manufacturing Technology, School of Mechanical Engineering, Hanoi University of Science and Technology. His interest is in Biomechanics in rehabilitation, modeling, simulating, and optimizing mechanisms, and artificial intelligence.

Van-Tinh Nguyen received a B.E. (2012) from the Hanoi University of Science and Technology (HUST), Vietnam, an M.Sc. (2016) and PhD. (2019) in Functional Control Systems from the Shibaura Institute of Technology, Japan. Currently, he works as a lecturer at the School of Mechanical Engineering of HUST. His research interests include optimization system design, multibody systems, humanoid robots, and evolutionary algorithms.

Toshio Ito received B.E. (1982) and D.E. (1995) degrees in systems engineering from Kobe University. He is a Professor in the Department of Machinery and Control Systems, Shibaura Institute of Technology. His current interests include driver behavior and machine vision for advanced driving support systems and automated driving.

Tam Bui received an M.E (2008) degree in mechanical engineering from the Hanoi University of Science and Technology (HUST). He received an M.E (2011) and Ph.D. (2015) in the field of optimal system design at the Shibaura Institute of Technology. He is an assistant professor at the Shibaura Institute of Technology. His current interests include intelligent optimization algorithms, neural networks, and artificial intelligence.