Global Hybrid Registration for 3D Constructed Surfaces Using Ray-Casting and Improved Self Adaptive Differential Evolution Algorithm

Linh Tao, Tinh Nguyen, and Hiroshi Hasegawa

***Abstract*—As a fundamental task in computer vision, registration has been a solution for many application such as: world modeling, part inspection and manufacturing, object recognition, pose estimation, robotic navigation, and reverse engineering. Given two images and set ones as the model, the aim is to find the best possible spatial transformation matrix causing 3D reconstruction of original object. The paper presents a new hybrid algorithm which improves both speed and convergence guarantee in comparison recently proposed methods of registering structured pointcloud surfaces by using a fast error calculation ray-casting based closest point method integrated with a new developed global optimization method Improve Self Adaptive Differential Evolution (ISADE). Ray-casting based L_2 error calculation method enables the algorithm to find the local minima error while ISADE exploit the searching boundary to find the global minima. The new algorithm is evaluated to show the significant improvement in quality and robustness to state-of-the-art methods.**

***Index Terms*—3D registration, ISADE, hybrid global registration, Ray-casting.**

I. Introduction

The introduction of commercial depth sensing devices such as Microsoft Kinect, Asus Xtion, etc. has shifted robotics, computer vision research areas from 2D based imaging and laser scanning toward 3D based depth scenes of environment processing. As a physical object or scenario cannot be completely captured with a single image, different images from different time and positions need to be aligned into a more completed view of the scenario; the process of alignment is called registration. Registration algorithms estimate the movement of the camera through calculating the transformation that optimally maps two point clouds. Various applications such as 3D object scanning, 3D mapping, and 3D localization use registration algorithms as backbone algorithms. According to how many views or images of the objects are processed at the same time, registration strategies are divided into multi-view registration (for all views case) and pair-wise registration (for two views case). Our paper focuses on the pair-wise registration of constructed range images taken by 3D cameras. As a consequence, starting from two views, i.e., the model and the data, the objective of our registration process is to find the best homogeneous transformation that, when applied to the data, aligns it with the model in a common coordinate system.

Manuscript received April 1, 2016; revised June 5, 2016.

Linh Tao is with Shibaura Institute of Technology, Hasegawa laboratory, Japan (e-mail: taongoclinh@gmail.com).

Iterative Closest Point (ICP) [1] and its variants such as non-linear ICP, generalized ICP and non-rigid ICP have been always indispensable tools in registration tasks. ICP's concept and implementation are easy to understand. ICP uses L_2 errors estimated from pair-wise point-clouds to derive a transformation which draws them closer to each other. Registration process finishes after many iterations of minimizing error and results in a homogeneous transformation.

However, ICP-class algorithms alone cannot solve problems for general registration tasks since they require a further assumption in which an initial near-optimal pose transformation is necessary for right convergences. Otherwise, the registration process would likely converge to local optimal solutions instead of the global optimal or near global optimal one. This result cannot be overcome merely by iteration procedure. In some mesh and point-cloud editor software such as Meshlab [2] registering tool for range data is available. It requires manually data pre-alignment from users before ICP comes into use.

To overcome the shortage of ICP-class methods, in general, registration processes are generally divided into two steps: coarse transformation or initialization and fine transformation. If two point-clouds are close enough, the first step could be omitted. Otherwise, the problem remains a big challenge for researchers. Coarse transformation, pre-alignment estimation or initialization solving has two approaches: local and global. Local methods use local descriptors (or signatures) such as PFH [3] and SIFT [4] which encode local shape variation in neighborhood points. If points with those descriptors appear in both registering point-clouds, initialization movement could be estimated by using sample consensus algorithms such as RANSAC [5]. The problem of local approaches is that those signatures are not always guarantied to appear on both registering point-clouds. On the other hand, global approaches take every point into account such as Go-ICP [6] and SAICP [7] The biggest problem of those methods is computation cost in finding the corresponding points in point-clouds. If there are big numbers of point in point-clouds, the computation cost is going large. However, thanks to new algorithms especially heuristic optimal searching methods as well as the increasing in computer speed especially with parallel computing with multi-core CPU processor and Graphic Computation Unit (GPU) [8] it is possible to find solutions of global approaches of registration problem. After estimating coarse transformation, ICP algorithm is an efficient tool to find the fine transformation.

This paper proposes a new global registration method for

doi: 10.18178/ijmlc.2016.6.3.592

3D constructed images without good initialization. It is called Global Hybrid Registration for 3D Constructed Surface Using Ray-casting and ISADE [9]. As other global registration methods, our method requires no local descriptors on works directly on raw scan surfaces. The method uses ray-casting based method for local minima searching together with ISADE as a search engine to find the global minima without using fine registration. Our method rapidly produces results at high rate convergence of the global optimization solution.

II. Three Dimension Registration

This part summary some approaches for global range image registration task up to date. SVD, PCA [10] are integrated together with ICP as classical methods and global searching algorithms are integrated with ICP as in the most current methods.

A. ICP Algorithm

SVD and PCA have been used to find coarse transformation together with ICP as the fine transformation-estimating tool. Original version of ICP algorithm relies on L_2 errors to derive the transformation including rotation and translation. To register two point-clouds $X = \{x_i\}$, $\{i = 1, \ldots, m\}$ (model point-cloud) and $Y = \{y_j\}$, $\{j = 1, \ldots, n\}$ (data point-clouds), where x_i and $x_j \in R^3$ are point coordinates of points in point-cloud. ICP algorithm arms to find rotation $\in SO^3$ and translation $\in R^3$, which minimize L_2 type error as in Equation 1.

$$E(R,t) = \sum_{i=1}^{n} e_i(R,t) = \sum_{i=1}^{n} |R * y_{i^*} + t - x_i| \qquad (1)$$

where R and t are rotation and translation matrix, y_{i^*} is the corresponding point of x_i denoted for its closest point in data point-cloud Y. There are some ICP variants, which rely on different categories to define closest points. Point-to-point and Point-to-plane are two popular examples. Equation 2 is used to search for closest point by Point-to-point category.

$$j^* = \underset{j \in \{1,\ldots,n\}}{argmin} |R * y_j + t - x_i| \qquad (2)$$

The iteration process is as following to archives the final transformation:

- Compute the closest model points for each data point as Equation 2.
- Compute the transformation R and t based on the error from Equation 1.
- Apply R and t to the data point-clouds.
- Repeat step 1, 2, 3 until error as Equation 1 smaller then a set tolerant or the procedure reaches its max iteration.

Step by step, ICP draws the data point-cloud closer to model point-cloud and the process stops at local minima. There are some variants of ICP algorithm based on different methods to calculate the transformation from error $E(R,t)$ and error itself as in LMICP [11] and SICP [12].

B. Global Hybrid Searching Algorithm

ICP algorithms are superior for registering close or pre-aligned point-cloud data; otherwise, it often converges wrongly. Global searching algorithms are solution to solve this problem since they are able to find the global minima instead of local one. To make the task of global searching algorithm less difficult, ICP are often applied to flatten the searching space. Fig. 1 and Fig. 2 show how ICP works as a flattening tool of objective functions. By using ICP, a complex fitness function in black turns into simpler one in red color. And with such a much more flatten fitness function, global searching method find a global minima more effectively.

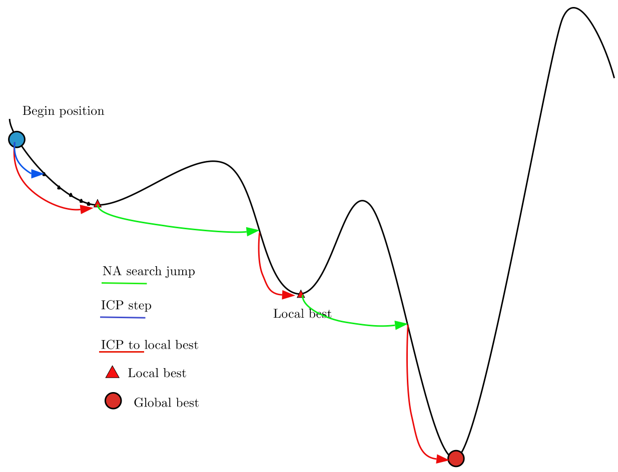

Fig. 1. Global searching algorithm with ICP integrated.

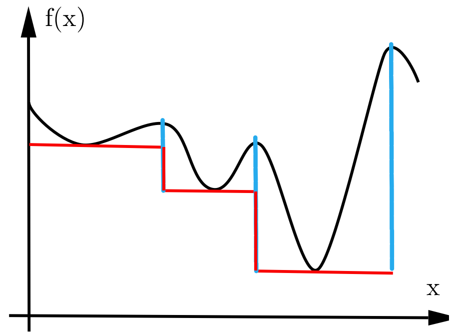

Fig. 2. Example of flatten objective function using ICP.

The integration works well in case of point-cloud data with small point number. For large data case, ICP becomes slow and impossible for applying into real time applications. Our method integrates new global searching algorithm ISADE, which can handle complicated fitness functions without or few flattening process and fast error calculation method based on ray-casting corresponding searching algorithm which accelerates registration procedure.

III. Method Overview

A. Methodology Approach

The biggest disadvantage of ICP based registration methods in calculating cost function is runtime. In KinectFusion [13] a real-time scene reconstruction algorithm, ICP is used as an only method for registering two continuous frames. The method requires a powerful Graphic Card to fasten calculations and reduce runtime. However, in global registration algorithms with thousand times of error function calculation more than ICP through many iterations and populations, to make the algorithm can run real-time, we need a faster error calculation method.

The proposed algorithm takes the advantage of fast error calculating by using ray-casting based corresponding point searching to apply for a new optimization algorithm ISADE with a purpose of getting a faster and global optimal convergence guaranty.

B. Ray-Casting Closest Point Method

ICP-class algorithms often use kd-tree [14] structure to speed up the process of finding j^* in Equation 2. The complexity of kd-tree searching closest algorithm is $O(log(n))$ where n is number of searching point set. Fig. 3 shows an example of corresponding points of the data point-cloud in the model one.

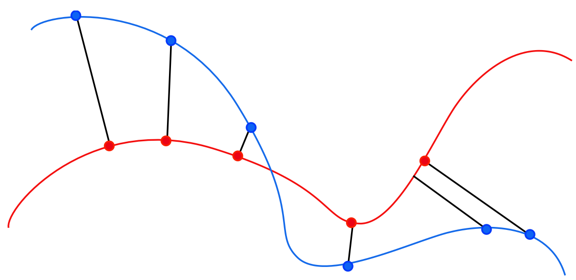

Fig. 3. Kd-tree closest point in original ICP.

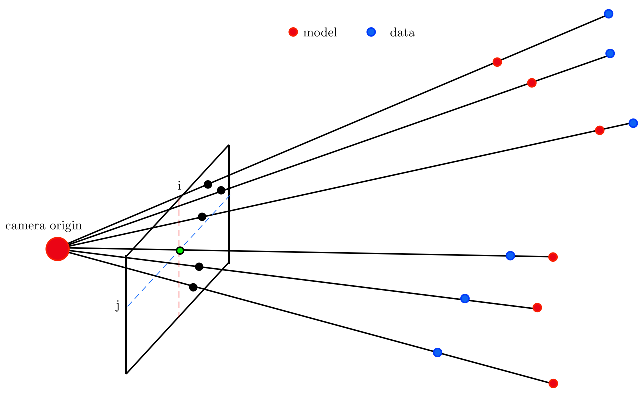

Fig. 4. Ray-casting method for searching corresponding points.

Since depth image or point cloud data are often obtained from 3D range camera in which the data could be consider as an 2D gray image $G$where value of each pixel show the depth of the point.

$$z_{i,j} = G[i,j] \tag{3}$$

where $z_{i,j}$ is depth of image at pixel i, j.

Equations 4 is to convert from depth image and real 3D depth data $\{x, y, z\}$.

$$x_{i,j} = (i - cx)G\,[i,j]/f_x \tag{4a}$$

$$y_{i,j} = (j - cy)G\,[i,j]/f_y \tag{4b}$$

$$z_{i,j} = G[i,j] \tag{4c}$$

where f_x, f_y, cx, cy are intrinsics of the depth camera. In conversion, pixel position and structured expression of a point $\{x, y, z\}$ can be calculated as Equation 5.

$$G[i,j] = z_{i,j} \tag{5a}$$

$$i = round\left(cx + x_{i,j} * f_x/G\,[i,j]\right) \tag{5b}$$

$$j = round\left(cy + y_{i,j} * f_y/G\,[i,j]\right) \tag{5c}$$

Those equations are to calculate i, j of data points which are also i, j of corresponding point in model point-clouds. The idea of the method is showed as Fig. 4, which reminds the ray-casting process in computer vision.

C. Objective Function

The fitness function need to provide an error score that is minimized when the best transformation matrix are applied. The paper uses fitness function as Equation 6.

$$F(R,t) = f(n)\frac{1}{n^2}\sum_{i=1}^{N}\left(R * y_j + t - x_i\right)^2 \tag{6}$$

where $f(n)$ is a function of inlier point number, n. N is the number of points in the data point-cloud.

The error function should be smaller in bigger number of inlier point. Since that, searching algorithm would get rid of the case in which cost function is small for only small inlier points. Function $f(n)$ is calculated as in Equation 7.

$$f(n) = \begin{Bmatrix} \infty & if & \frac{n}{N} < 0.1 \\ 1 - \frac{n}{N} & & otherwise \end{Bmatrix} \tag{7}$$

Instead of using ICP with iteration steps with meeting the condition of maximum iteration steps or error become smaller then a set error to flatten the cost function, the algorithm only do smothering by calculating transformation matrix to minimize error function with one step using SVD method. Equation 6 without $F(R, t)$ can be rewrite as Equation 8 to find the one step better rotation and translation in term of cost from initial transformation matrix.

$$F(\Delta R, \Delta t) = F(R + \Delta R, t + \Delta t) \tag{8}$$

where R, t are initial rotation and translation matrix, ΔR and Δt are smothering or fine matrix.

D. Translation Computing

We can find the optimal translation by taking derivative of F with respect to Δt and search for its roots.

$$0 = \frac{\partial F}{\partial \Delta t} = \sum_{i=1}^{N} 2\left(\Delta R * y'_{j^*} + \Delta t - x_i\right) = 2t * n + 2R\left(\sum_{i=1}^{N} y'_{j^*}\right) - 2\left(\sum_{i=1}^{N} x_i\right) \tag{9}$$

where y'_{j^*} is new coordinate of y_{j^*} after rough transformation with R and t.

Denote

$$\bar{x} = \frac{\left(\sum_{i=1}^{N} x_i\right)}{n} \quad and \quad \bar{y} = \frac{\left(\sum_{i=1}^{N} y'_{j^*}\right)}{n}$$

The final results for translation:

$$\Delta t = \bar{x} - \Delta R\bar{y} \tag{10}$$

In other words, the translation of first movement draws two pointlouds close to each other so their weighted centroids coincide.

E. Translation Computing

Replacing Δt from Equation 10, $F(\Delta R, \Delta t)$ is calculated as Equation 11.

$$F(\Delta R, \Delta t) = \sum_{i=1}^{N} \left(\Delta R * y'_{j^*} + \Delta t - x_i\right)^2$$
$$= \sum_{i=1}^{N} \left(R * y'_{j^*} + (\bar{x} - R\bar{y}) - x_i\right)^2$$
$$= \sum_{i=1}^{N} \left(R * (y'_{j^*} - \bar{y}) - (x_i - \bar{x})\right)^2$$

Denote $x'_i = x_i - \bar{x}$ and $y'_j = y'_{j^*} - \bar{y}$, rotation matrix is presented as Equation 12.

$$\Delta R = \underset{R}{argmin} \sum_{i=1}^{N} (R * y'_i - x'_i)^2 \tag{12}$$

Using SVD method for least square problem, covariance matrix is calculated as Equation 13.

$$S = XY^T \tag{13}$$

Decomposing S matrix $S = U \sum V^T$, then rotation matrix is calculated as in Equation 14.

$$\Delta R = V \begin{bmatrix} 1 & 0 & & 0 & 0 \\ 0 & \dots & & 0 & 0 \\ 0 & 0 & 1 & & 0 \\ 0 & 0 & 0 & & \det(VU^T) \end{bmatrix} U^T \tag{14}$$

After having rotation matrix, translation matrix is recalculated as Equation 10.

F. ISADE

Differential evolution (DE) is an optimization technique originally proposed by Storn and Price [15]. It is categorized into evolution algorithm group, which is characterized by operators of mutation and crossover. In DE, two important coefficients, which play key rolls to decide the correction and speed of convergence, are scaling factor F and crossover rate C_r. Another important parameter in DE, population size NP remains a user-assigned value to cope with problem complexity. ISADE not only adaptively changes those three coefficients but also integrates different mutation schemes to take advantages of them.

1) Adaptive learning strategies selection

In their paper of ISADE, Tam Bui et al. randomly chose three mutation schemes, which are $DE/best/1/bin$, $DE/best/2/bin$ and $DE/rand\ best/1/bin$. Among DE's schemes, $DE/best/1/bin$ and $DE/best/2/bin$ are known for good convergence property and $DE/rand\ best/1/bin$ is known for good diversity. The probability of applying those strategies are equal equally assigned at with values $p_1 = p_2 = p_3 = 1/3$. Equations 8 show the formula of chosen schemes.

$$DE/best/1: V^G_{i,j} = X^G_{best,j} + F\left(X^G_{r1,j} - X^G_{r2,j}\right) \tag{8a}$$

$$DE/best/2: V^G_{i,j} = X^G_{best,j} + F\left(X^G_{r1,j} - X^G_{r2,j}\right) + F\left(X^G_{r3,j} - X^G_{r4,j}\right) \tag{8b}$$

$$DE/rand\ best/1: V^G_{i,j} = X^G_{best,j} + F\left(X^G_{best,j} - X^G_{r2,j}\right) + F\left(X^G_{r3,j} - X^G_{r4,j}\right) \tag{8c}$$

In APGA/VNC appoach proposed by S. Tooyama and H. Hasegawa [16] scaling factor changes according to iteration as sigmoid function as in Equation 9.

$$F_i = \frac{1}{1 + exp\left(\alpha \frac{i - \frac{NP}{2}}{NP}\right)} \tag{9}$$

ISADE give addition scaling F_i^{mean} as in Equation 10.

$$F_i^{mean} = F_{min} + (F_{max} - F_{min}) \left(\frac{i_{max} - i}{i_{max}}\right)^{n_{iter}} \tag{10}$$

where

$$n_{iter} = n_{min} + (n_{max} - n_{min}) \frac{i}{i_{max}} \tag{11}$$

F_i in Equation 9 is modified as in Equation 11.

$$F_i = \frac{F_i + F_i^{mean}}{2} \tag{12}$$

Now scaling factor is set to be high in first iterations and after certain generations it become smaller for proper exploitation.

2) Crossover control parameter

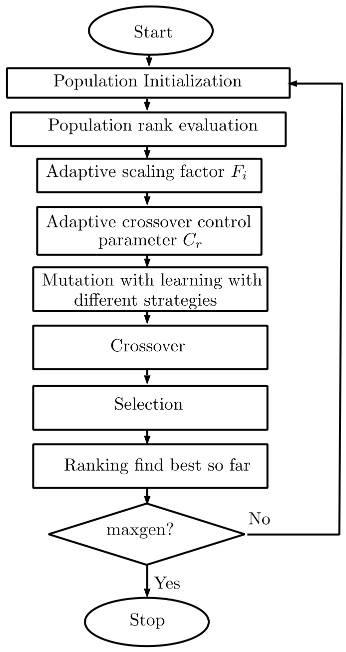

Fig. 5. ISADE implementation flowchart.

ISADE algorithm is able to detect whether high values of C_r are useful and if a rotationally invariant crossover is required. A minimum base for C_r around its median value is incorporated to avoid stagnation around a single value. The control parameter C_r is assigned as Equation 13.

$$C_r^{i+1} = \begin{Bmatrix} rand_2 & if \quad rand_1 < \tau \\ C_r^i & otherwise \end{Bmatrix} \tag{13}$$

where $rand_1$ and $rand_2$ are random values $\in [0,1]$, τ presents probability to adjust C_r. C_r is adjusted as in Equation 14.

$$C_r^{i+1} = \begin{Bmatrix} C_{r_{min}} & if \quad C_{r_{min}} \le C_r^{i+1} \le C_{r_{medium}} \\ C_{r_{max}} & if \quad C_{r_{medium}} \le C_r^{i+1} \le C_{r_{max}} \end{Bmatrix} \tag{14}$$

where $C_{r_{min}}$, $C_{r_{medium}}$, $C_{r_{max}}$ denote low value, median value and high value of crossover parameter respectively. As in [12], we take $\tau = 0.1$, $C_{r_{min}} = 0.05$, $C_{r_{medium}} = 0.5$, $C_{r_{max}} = 0.95$.

All above ideas and theories are implemented as in flowchart in Fig. 5.

G. A New Combination

From initial position matrix, using one ICP iteration to gain a slightly better rotation and translation matrix, the algorithm recalculates the error as in Equation 6 and uses it in ISADE searching algorithm. Flowchart in Fig. 6 shows implementation of the whole algorithm.

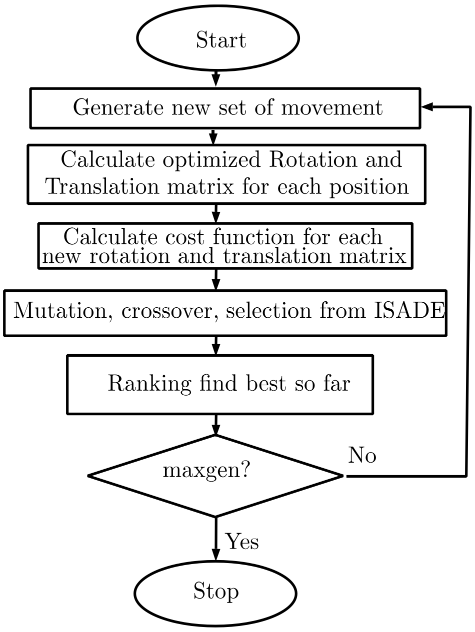

Fig. 6. Hybrid registration method with ISADE and ray-casting.

IV. EXPERIMENT AND RESULTS

This section arms at presenting a number of experimental results to study how robust and accurate of ISADE results in comparison to other Global searching algorithm in using the same ray-casting based error function as well as comparison of result from new algorithm to KinecFusion in term of accuracy.

- De Falco *et al*.'s proposal (DE), Differential Evolution as a viable tool for satellite image registration [17].
- Valsecchi et al.'s proposal (GA), An Image Registration Approach using Genetic Algorithms [18].
- Talbi *et al*.'s proposal (PSO), Particle Swarm Optimization for Image Processing [19].
- Luck *et al*.'s proposal (SA), registration of range data using a hybrid simulated annealing and iterative closest point algorithm [20].

The proposed algorithm is implemented in C++ and compiled with GNU/g++ tool.

In order to perform a fair comparison between different optimization tools, in all methods, maximum iteration is set to 100 with population of 25 each generation. As SAICP is not a multi-agent method, its maximum iteration is set to 2500.

A. Range Image Datasets

Our experiments carried out number of pair-wise registration task using well-known Depth data taken from Kinect Microsoft Camera downloaded from website of Microsoft Research http://research.microsoft.com/en-us/projects/7-scenes/. Specifically, Fig. 7 shows all scenes: Chess, Fire, Heads, Office, Pumpkin, RedKitchen, and Stairs.

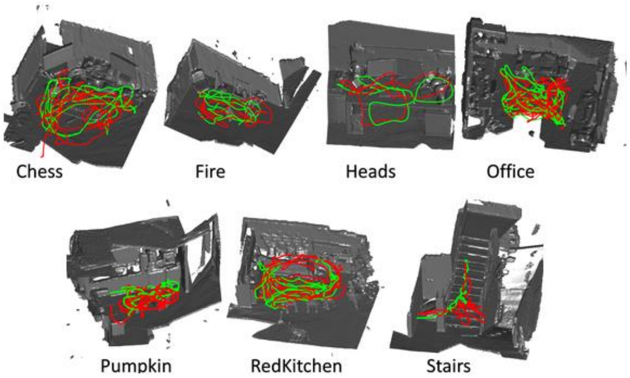

Fig. 5. RGB-D 7 scenes datasets.

Those .PNG format depth images were sub-sampled into smaller solution of 128×96, which is 5 times smaller than original solution of 640×480 in each dimension. The reason for using smaller number of point dataset is to archive considerable suitable runtime while accuracy remains unchanged.

B. KinectFusion Error from Data Transpose

Accompany with depth datasets, 7 scenes database give us camera homogeneous transposes at each frame calculated from Kinect-Fusion algorithm. Using those transpose, we could calculate transformation matrix between two scenes as Equations 15.

$$T_j^i = T_i^{-1} * T_j \quad (15a)$$

$$T_j^i = \begin{bmatrix} & R_j^i & & t_j^i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (15b)$$

where T_j^i is transformation matrix to move frame j to align with frame i , T_i and T_j are homogeneous transpose matrix for camera at frame i and j respectively, R_j^i ,t_j^i are rotation and translation matrix of T_j^i.

R_j^i , t_j^i are applied into ray-casting error calculation methods for two frames as in Equation 6 to draw errors of KinectFusion algorithm for the next comparison step.

C. Parameter Setting

In each methods 30 runs were executed with two registration depth images are at distance of 20 frames in the sequence. The searching space is set so rotation and translation limitation at $[-2\pi/10, 2\pi/10]$ and $[-0.3,0.3]$separately. All methods are run on a PC of Intel core I7-4790 CPU 3.60 GHz ×8 processor and 8 GB of RAM memory.

D. Results Comparison between Algorithms

ISADE searching algorithm results are compared with other algorithms' results in three categories including convergent rate, mean and standard deviation, which are shown in Table I.

TABLE I: RESULTS FROM DIFFERENT SEARCHING ALGORITHM ON 7-SCENE DATA

Scene name	Algorithm	CvR(%)	Mean	St.dev
Chess	ISADE	100	0.0695	0.0107
ref: 0.2483	DE	100	0.0752	0.0144
	GA	0	1.8018	0.6643
	PSO	0	0.6753	0.4502
	SA	6.6667	0.9413	0.7171
Fire	ISADE	100	0.0230	8.85e-04
ref: 0.2431	DE	100	0.0290	2.55e-04
	GA	0	0.7740	0.2300
	PSO	20	0.3497	0.2826
	SA	6.6667	0.3306	0.2679
Heads	ISADE	100	0.0024	3.59e-05
ref: 2.9907	DE	100	0.0027	0.0048
	GA	100	0.3080	0.1349
	PSO	100	0.0824	0.0836
	SA	73.3333	0.4494	0.3385
Office	ISADE	100	0.0358	8.47e-05
ref: 0.6294	DE	100	0.0371	8.24e-04
	GA	100	0.8577	0.3445
	PSO	100	0.2819	0.3702
	SA	33.3333	0.5526	0.5851
Pumpkins	ISADE	100	0.0407	0.0071
ref: 0.111361	DE	100	0.0489	0.0127
	GA	0	1.1097	0.4057
	PSO	6.6667	0.3779	0.3330
	SA	0	6.6667	0.6984
RedKitchen	ISADE	100	0.0315	0.0049
ref: 0.0984	DE	93.3333	0.0473	0.0239
	GA	0	1.4215	0.6508
	PSO	0	0.4863	0.3829
	SA	10	0.3021	0.2898
Stairs	ISADE	100	0.0056	5.63e-06
ref: 0.0156	DE	100	0.0062	0.0014
	GA	0	0.9413	0.3373
	PSO	0	0.2441	0.2435
	SA	3.3333	0.4808	0.6281

KinectFusion error or reference value is considered as correct convergence. In Table I, convergence rate (CvR) means percentage of algorithms results smaller than reference value.

Proposed algorithm and DE are superior to other methods in every category. ISADE are better than DE in almost cases only in the Fire scene standard deviation of ISADE method larger than DE method's.

The proposed method are qualified in all tested scenes with convergence value are always smaller than reference value. This can be explained by accumulating error by using ICP algorithm from frame to frame. As using ICP continuously from frame to frame in Kinect Fusion algorithm error would be accumulated and become large. The final transformation matrix becomes less accurate than which gained from direct registration method using only two frames.

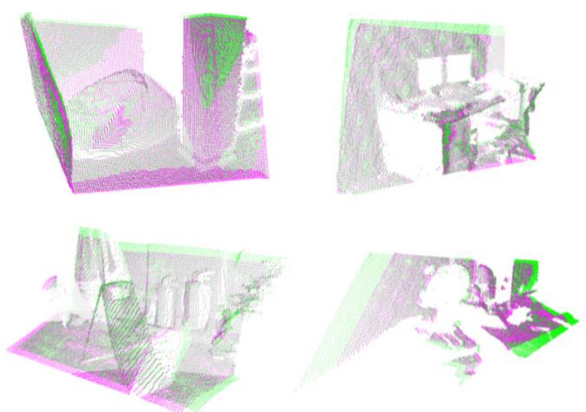

Fig. 8. Registration output examples.

Fig. 8 shows four scenes registration results using ISADE integrated algorithms including: Fire, Head, Office, Stairs. Model pointcloud are in pink and data pointcloud are in green color.

E. Runtime

For the data of 128×96 resolution, average running time for the proposed method are shown in Table II.

TABLE II: AVERAGE RUNTIME WITH DIFFERENT SCENES IN SECOND

Chess	Fire	Heads	Office	Pumpkin	Redkitchen	Stairs
7.5053	5.8596	8.0114	7.4527	5.9005	6.0466	7.8627

The results show the average time for registration at around 8 seconds. Two registering frames are at distance of 20 frames. That means the rate of registering equivalence at rate of 2.5 fps (frames per second). To make algorithm run at real-time rate of 20fps, the speed need to be increased by 8 times. If we exploit all core of 8-core-processors or GPU multi-core processors, this target could be archived.

V. DISCUSSION AND CONCLUSION

Image registration has been a very active research area. Recently, the approach of using evolutionary algorithms (EAs), especially new methods, proved their potential of tackling image registration problem based on their robustness and accuracy on searching for global optimal. With EAs algorithm as searching tools, it is not necessary to have good initials to avoid local minima and converge to near-global minima solutions. To do that, EAs algorithms need tuning carefully to gain best results.

We proposed the new registration algorithm by integrating a new self-adaptive optimization algorithm (ISADE) into a fast closest point searching method to tackle well-known challenging task of computer vision area. In the experiments, the results show that ISADE is able to find a robust and accurate transformation matrix of camera movement.

What is more important, accuracy and robustness results have been obtained in comparison with other state-of-the-art evolution based algorithms. ISADE shows its superior than GA, PSO, SA in searching for global minima solution. In comparison with DE, ISADE also show its much better in almost tested scenes. The robustness and accuracy is tested and proved in real 3D scenes captured by Microsoft Kinect camera.

In term of running time, by using fast searching closest point methods, proposed algorithms are considered fast in our sense. It shows potential of applying in real-time application if using parallel programing technique with multi-core processors.

In future work, ISADE algorithm can be implemented in parallel in GPU (Graphic Processor Unit) which can help algorithm reduces runtime to prove real-time implement possibility in 3D reconstruction, 3D mapping and 3D localization.

REFERENCES

[1] 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 1992J. U. Duncombe, "Infrared navigation — Part I: An assessment of feasibility," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34-39, Jan. 1959.

[2] Meshlab opensource software. [Online]. Available: http://meshlab.sourceforge.net/
[3] R. B. Rusu *et al*., "Persistent point feature histograms for 3D point clouds," in *Proc. the 10th International Conference on Intelligent Autonomous Systems*, 2008.
[4] A. Sehgal, D. Cernea, and M. Makaveeva, *Real-Time Scale Invariant 3D Range Point Cloud Registration*.
[5] 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, 2007.
[6] J. L. Yang, H. D. Li, and Y. D. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *Proc. 2013 IEEE International Conference on Computer Vision*, pp. 1457-1464, 1-8 Dec., 2013.
[7] 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*, vol. 4, pp. 3739-3744, 2000.
[8] D. Neumann *et al*., "Real-time RGB-D mapping and 3-D modeling on the GPU using the random ball cover data structure," in *Proc. 2011 IEEE International Conference on Computer Vision Workshops*, pp. 1161-1167, 6-13 Nov., 2011.
[9] 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, pp. 59-74.
[10] S. Marden and J. Guivant, "Improving the performance of ICP for real-time applications using an approximate nearest neighbour search," in *Proc. Australasian Conference on Robotics and Automation*, Victoria University of Wellington, New Zealand, 3-5 Dec, 2012.
[11] K.-L. Low, "Linear least-squares optimization for point-toplane ICP surface registration," Technical Report TR04-004, Department of Computer Science, University of North Carolina at Chapel Hill, February 2004.
[12] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," *Journal Computer Graphics Forum*, vol. 32, no. 5, pp. 1-11, 2013.
[13] S. Izadi, D. Kim, and O. Hilliges *et al*., "KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. ACM Symposium on User Interface Software and Technology*, October 2011.
[14] C. Sharat, *Introduction to Kd-Trees*, University of Maryland, Department of Computer Science.
[15] A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.
[16] S. Tooyama and H. Hasegawa, "Adaptive plan system with genetic algorithm using the variable neighborhood range control," in *Proc. IEEE Congress on Evolutionary Computation*, 2009.
[17] I. D. Falco, A. D. Cioppa, D. Maisto, and E. Tarantino, "Differential Evolution as a viable tool for satellite image registration," *Applied Soft Computing*, vol. 8, no. 4, pp. 1453-1462, 2008.
[18] A. Valsecchi, S. Damas, and J. Santamaria, "An image registration approach using genetic algorithms," in *Proc. IEEE Congress on Evolutionary Computation*, 10-15 June, 2012, pp. 1-8.
[19] H. Talbi and M. C. Batouche, "Particle swam optimization for image registration," in *Proc. International Conference on Information and Communication Technologies: From Theory to Applications*, 2004.
[20] L. Ingber, "Simulated annealing: Practice versus theory," *Math. Comput. Modelling*, vol. 18, pp. 29-57, 1993.

Tao Ngoc Linh received his B.E. in 2010 at Hanoi University of Science and Technology, Vietnam and M.E. in 2013 from Taiwan University of Science and Technology, Taiwan. He is pursuing Dr. Eng. degree at the Department of Functional Control Systems, Graduate School of Engineering and Science, Shibaura Institute of Technology, Japan. His research interests include 3D computer vision and intelligent algorithms.

Tinh Nguyen received the B.E. degrees in 2012 at Hanoi University of Science and Technology. Currently, he is a second year Master student at Graduate School of Engineering, Shibaura Institute of Technology, Japan. His research interests include optimization system design, multi-body system, humanoid robot, and evolutionary algorithm.

Hiroshi Hasegawa received his B.E. in 1992 and M.E. in 1994 from Shibaura Institute of Technology, Japan. He received Dr. Eng. in 1998 in mechanical engineering from Tokyo Institute of Technology, Japan. He has been working at Shibaura Institute of Technology, and currently is a Professor at the Department of Machinery and Control System, College of Systems Engineering and Science. He is a member of JSEE, JSME, ASME, JSCES, JSST and KES. His research interests include computer-aided exploration, creativity of design and systems engineering.