Artificial Life and Robotics
https://doi.org/10.1007/s10015-020-00603-9

ORIGINAL ARTICLE

Adaptive evolution strategy sample consensus for 3D reconstruction from two cameras

Yuichiro Toda[1] · Hsu Horng Yz[1] · Takayuki Matsuno[1] · Mamoru Minami[1] · Dalin Zhou[2]

Received: 26 April 2019 / Accepted: 9 March 2020

Abstract
RANdom SAmple Consensus (RANSAC) has been applied to many 3D image processing problems such as homography matrix estimation problems and shape detection from 3D point clouds, and is one of the most popular robust estimator methods. However, RANSAC has a problem related to the trade-off between computational cost and stability of search because RANSAC is based on random sampling. In our previous work, we proposed Adaptive Evolution Strategy SAmple Consensus (A-ESSAC) as a new robust estimator, and we applied ESSAC to the homography matrix estimation for 3D SLAM using RGB-D camera. A-ESSAC is based on Evolution Strategy to maintain the genetic diversity. Furthermore, ESSAC has two heuristic searches. One is a search range control for reducing the computational cost of RANSAC. The other is adaptive/self-adaptive mutation for changing the search strategy of A-ESSAC according to the best fitness value. In this paper, we apply A-ESSAC to 3D reconstruction method using two cameras, and we show an experimental result, and discuss the effectiveness of the proposed method.

Keywords Evolutionary computation · Robust estimator · 3D reconstruction

1 Introduction

Recently, 3D image processing technologies have excelled with the development of 3D distance measurement sensors such as Lidar and RGB-D camera. These kinds of sensors enable to perform the 3D image processing in real time and many kinds of applications related to the 3D image processing have been proposed in the fields of robotics and Intelligent Transport Systems. However, the measurement data of the 3D distance measurement sensor includes many noises according to the environments such as the material of the measurement object and the lightning condition. To estimate a model from the noisy dataset, RANdom SAmple Consensus (RANSAC) proposed by Fischer and Bolles is one of the most popular algorithms in robust estimator [1].

RANSAC has been applied to many 3D image processing problems such as homography matrix estimation problem and shape detection from the 3D point clouds [2–4]. However, one of the problems in RANSAC is a sampling bias in a search since it selects candidate pairs from a data set of pairs randomly. To solve the problem of RANSAC, many researchers have improved RANSAC algorithm. Choi et al. [5] gave a critical survey of RANSAC family algorithms. They synthesized seven research axes that were Partial Evaluation (e.g., Progressive RANSAC), Adaptive Termination (e.g., uMLESAC), Adaptive Evaluation (e.g., pbM-estimator), Local Optimization (e.g., LO-RANSAC), Model Selection (e.g., MAPSAC), Loss Function (e.g., MLESAC), and Guided Sampling (e.g., GASAC). These research axes were discussed from different objectives: being fast, being robust, and being accurate.

In this paper, our objective is to reconstruct 3D point cloud from two cameras in real time from the noisy dataset.Therefore, we focus on evolutionary computation for RANSAC family algorithms because Genetic Algorithm SAmple Consensus (GASAC) proposed by Rodehorst and Hellwich [6] can improve the performance of the search capability by a population-based multi-point search. However, it is sometimes difficult to maintain the genetic

This work was presented in part at the 24th International Symposium on Artificial Life and Robotics (Beppu, Oita, January 23–25, 2019).

✉ Yuichiro Toda
ytoda@okayama-u.ac.jp

[1] 3-1-1 Tsushima-naka, Kita-ku, Okayama, Japan

[2] University of Portsmouth, Portsmouth PO1 3HE, U.K.

Published online: 24 April 2020

diversity in the search if the large size of outliers is included in a data set. To solve this problem, Shojaedini et al. proposed the modified GASAC as Adaptive Genetic Algorithm Sample Consensus (AGASAC) by applying the adaptive mutation, crossover and learning roulette wheel selection to GASAC algorithm [7]. AGASAC can change the dominant operators (mutation and crossover) according to the fitness values in the gene set. Using these new operators, Shojaedini et al. showed AGASAC outperforms GASAC. However, these kinds of evolutionary computation methods require more computing time than any other SAC methods. Therefore, we must deal with the trade-off between computational cost and stability of search.

There are two possible approaches to improve the trade-off. One is to change the strategy of the local search and global search efficiently. This kind of approaches can control the genetic diversity of a population to improve the stability in evolutionary search. The other is to manage the search range in the dataset according to the current search result. This kind of approaches can remove obvious outliers from a dataset. However, the feasible solutions of the model parameters are required to discriminate inliers from outliers. This means that the discrimination of inliers and outliers requires model parameters, while the estimation of model parameters requires a set of inliers. This is a nesting structure each other. In our previous work, we proposed Adaptive Evolution Strategy SAmple Consensus (A-ESSAC) as a new robust estimator method to improve the trade-off between computational cost and stability of search in RANSAC. Furthermore, we applied A-ESSAC to 3D map building method using RGB-D camera for realizing the real-time 3D SLAM [8]. In this paper, we apply A-ESSAC to 3D reconstruction method using two cameras for verifying the effectiveness and possible application of A-ESSAC. This paper is organized as follows. Section 2 explains our 3D reconstruction method. Section 3 explains A-ESSAC. Section 4 shows experimental results of the proposed method.

2 3D reconstruction using two cameras

2.1 Algorithm of 3D reconstruction

In this paper, we focus on 3D reconstruction from two cameras. Our 3D reconstruction method uses a local feature extraction method for reconstructing the sparse point cloud data. For the local feature-based 3D reconstruction method, the algorithm can mainly divide into two steps. One step is the image processing step that is for searching correct corresponding points between two camera data. The other is the matching step of 3D point clouds using the relation between two data. In these algorithms, a homography estimation method such as RANSAC is required for extracting the corresponding points. Therefore, we deal with the 3D modeling method as an application of our method. Figure 1 shows the flowchart of this algorithm. In Fig. 1, the tth measurement data from left and right camera are expressed by $D_L(t)$ and $D_R(t)$, respectively. Specifically, the possible pairs of corresponding points between $D_L(t)$ and $D_R(t)$ are generated by extracting any features of camera image. However, the pairs include many mismatched pairs when the pairs are generated. Therefore, the homography matrix is estimated using A-ESSAC (our proposed method) to remove the mismatched pairs. Finally, the coordinate transform matrix is estimated using 3D distance information of the corresponding points to update 3D environmental map.

2.2 Feature extraction

This subsection explains the detail of feature extraction and matching method. Recently, various types of local features have been proposed for the object recognition and extracting the corresponding points from multiple images. In local features, Scale-Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) are the most commonly used to extract the corresponding points from the multiple images [9, 10]. These local features are robust to the change of illumination and local affine distortion of images. Especially, SURF proposed by Bay et al. is based on 2D Haar wavelet responses as a robust local feature detector inspired by SIFT. The standard SURF is several times faster than SIFT. Furthermore, we must reduce computational time as much as possible in real-time image processing. General-purpose graphics processing unit (GPGPU) has been applied to calculate and extract features in real-time. In this paper, we use SURF implemented on GPGPU to describe features for pattern matching between the left and right camera images. After extracting SURF from the two images, each SURF feature is described by a vector

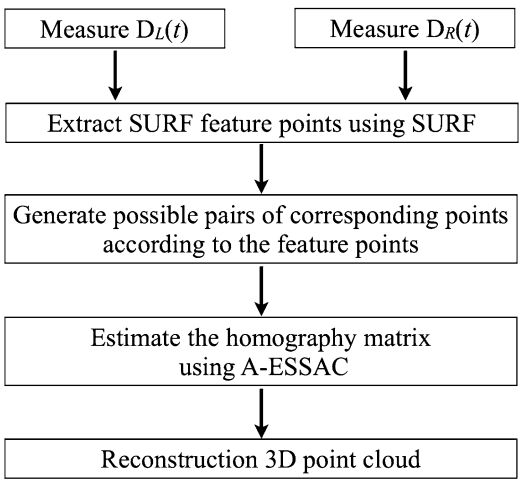


Fig. 1 Flowchart of proposed 3D modeling method

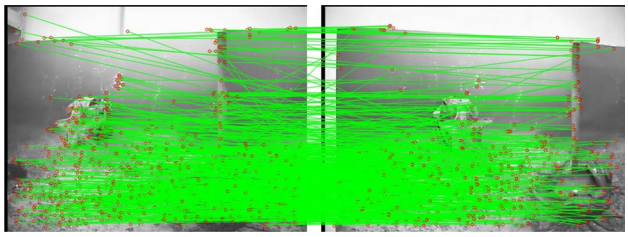


Fig. 2 A result of the feature extraction and matching using SURF. Red circles and green lines indicate feature points and matching results, respectively

containing 64 or 128 elements. An initial set for estimating homography matrix (possible pairs of corresponding points) is obtained by selecting the pairs with the minimum Euclidian distance of the feature vector between the left and right camera images (Fig. 2).

2.3 Homography estimation

After the feature extraction and matching, we should extract the correct pairs of corresponding points from the dataset of possible pairs. In many researches, homography matrix is estimated for extracting the correct pairs [11, 12]. In this way, the homography matrix estimation problem is one of the most important problems not only in 2D image processing but also in 3D image processing because the matrix is required in various types of 3D image processing such as stereovision [12] and 3D environment map building. The homography matrix H that has 9 elements is the matrix that describes the relation between two images. Figure 3 displays the concept image of homography matrix between two images. The set of homogeneous image points $\{\mathbf{x}_i\}(i = 1, \ldots, n)$ as viewed in the first image is transformed into the set $\mathbf{x}'_i$ in the second image, with the positions related by

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i, \tag{1}$$

where $\mathbf{x}_i$ and $\mathbf{x}'_i$ are homogeneous three vectors $\mathbf{x} = (x, y, 1)^T$, $\mathbf{x}' = (x', y', 1)^T$ because the matrix $\mathbf{H}$ is composed of 3×3 matrix. In addition, Eq. (1) can be also defined by the following equation:

$$x' = \frac{a_1 x + a_2 y + a_3}{a_7 x + a_8 y + 1} \tag{2}$$

$$y' = \frac{a_4 x + a_5 y + a_6}{a_7 x + a_8 y + 1}, \tag{3}$$

where a_1–a_8 are the parameters. Therefore, the homography matrix is estimated by estimating these eight parameters. Estimating the homography matrix between two images enables to extract the correct pairs of corresponding points.

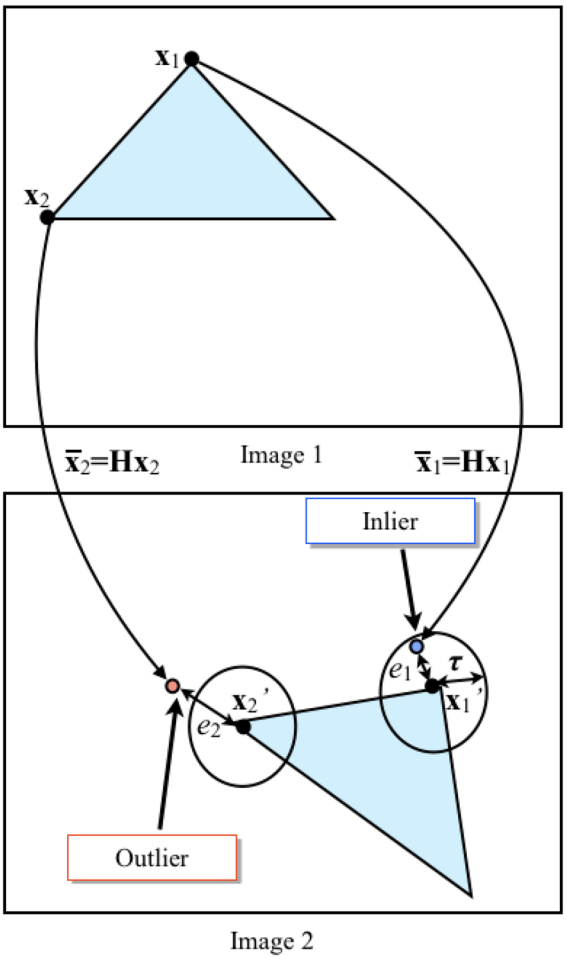


Fig. 3 Concept image of 2D homography, where $\overline{\mathbf{x}_\mathbf{i}}$ indicates an estimated point of the *i*th point using the homography matrix **H**

However, the set of possible pairs includes many mismatch pairs. Therefore, robust estimators are applied to this problem [13, 14].

2.4 Update of 3D model

The update of the 3D model is to obtain the position $\mathbf{x}'_{ri} = (x'_{ri}, y'_{ri}, z'_{ri})$ of a pixel in the 3D space based on the position $\mathbf{x}_{ri} = (x_{ri}, y_{ri}, z_{ri})$ according to the relationship between (x'_i, y'_i) and (x_i, y_i). An interactive closest point (ICP) algorithm is one of the most widely used methods of matching a set (X^r) of points with point clouds (X'_r) in 3D space [15]. The error function to be minimized is defined as

$$E(\mathbf{R}, \mathbf{t}) = \frac{1}{N_c} \left\| \sum_{i=1}^{N_c} \mathbf{R}\mathbf{x}'_{ri} + \mathbf{t} - \mathbf{x}_{\mathbf{ri}} \right\|, \tag{4}$$

where $\mathbf{R}$ is the rotation matrix; $\mathbf{t}$ is the translation vector; We apply the unit quaternion proposed by Horn [16]. The

quaternion is defined as $\hat{q} = (q_0, q_1, q_2, q_3)$. First, the center of gravity (COG) of each point cloud is calculated by the following:

$$\mathbf{x}_r^g = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{x}_{ri} \tag{5}$$

$$\mathbf{x}_r'^g = \frac{1}{N_c}\sum_{i=1}^{N_c}\mathbf{x}'_{ri}, \tag{6}$$

where N_c is the number of points in each point cloud. Next, the relative position from the COG is calculated in the following;

$$\mathbf{x_i^a} = \mathbf{x_{ri}} - \mathbf{x_r^g} \tag{7}$$

$$\mathbf{x}_i'^{\mathbf{b}} = \mathbf{x'_{ri}} - \mathbf{x}_\mathbf{r}'^{\mathbf{g}}. \tag{8}$$

Next, S_{ab} is defined as

$$S_{ab} = \sum_{i=1}^{N_c}\mathbf{x_i^a}\mathbf{x}_\mathbf{i}'^{\mathbf{b}}. \tag{9}$$

According to S_{ab}, a matrix $\mathbf{P}$ is defined as

$$\mathbf{P} = \begin{pmatrix} S_{xx}+S_{yy}+S_{zz} & S_{yz}-S_{zy} & S_{zx}-S_{xz} & S_{xy}-S_{yx} \\ S_{yz}-S_{zy} & S_{xx}-S_{yy}-S_{zz} & S_{xy}+S_{yx} & S_{zx}+S_{xz} \\ S_{zx}-S_{xz} & S_{xy}+S_{yx} & S_{yy}-S_{xx}-S_{zz} & S_{yz}+S_{zy} \\ S_{xy}-S_{yx} & S_{zx}+S_{xz} & S_{yz}+S_{zy} & S_{zz}-S_{xx}-S_{yy}. \end{pmatrix} \tag{10}$$

Here the eigenvector corresponding to the maximum positive eigenvalue of $\mathbf{P}$ is quaternion ($\hat{q}$). The rotation matrix is obtained by $\hat{q}$ in the following:

$$\mathbf{R} = \begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_2q_1+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_3q_1-q_0q_2) & 2(q_3q_2+q_0q_1) & q_0^2-q_1^2+q_2^2+q_3^2. \end{pmatrix} \tag{11}$$

Furthermore, the translation vector is also obtained by $\mathbf{R}$ in the following:

$$\mathbf{t} = \mathbf{x_r^g} - \mathbf{R}\mathbf{x}_\mathbf{ri}'^{\mathbf{g}}. \tag{12}$$

3 Adaptive evolution strategy sample consensus (A-ESSAC)

3.1 Total algorithm of A-ESSAC

In ESSAC, the genotype is composed of k candidate data needed to calculate the model parameters and the combination of candidate data is optimized by global search and hill-climbing search using genetic operators. The fitness value fit_i is calculated by following fitness function:

$$\begin{aligned} fit_i &= fitnessFunction(g(i)) = \sum_j \rho(e_j) \\ \rho(e_j) &= \begin{cases} 1 & (\text{if } e_j > \tau) \\ 0 & (\text{otherwise}), \end{cases} \end{aligned} \tag{13}$$

where e_j indicates the jth error calculated by the following equation:

$$e_j = \|\mathbf{x}_\mathbf{j}' - \mathbf{x_j}\|, \tag{14}$$

where $\mathbf{x_j}$ and $\mathbf{x}_\mathbf{j}'$ are homogeneous three vectors defined in Eq. (1).

In ESSAC, the fitness function usually uses the number of inliers. Therefore, this problem is a maximization problem. Furthermore, ESSAC has a search range control method to reduce computational cost and improve the stability of search simultaneously.

3.2 Evolution strategy

Basically, the random sampling required to estimate parameters of a mathematical model in the generation of hypothesis is one of combinational optimization problems, but we can incorporate local search or heuristics to reduce computational cost. Evolutionary computation (EC) is used to solve optimization problem by simulating evolution on a computer. From the historical point of view, EC can be divided into genetic algorithm (GA), evolutionary programming (EP), and evolution strategy (ES). These methods are fundamentally iterative generation and alternation processes operating on a set of candidate solutions called a population. All the population evolves toward better candidate solutions by selection operation and genetic operators (crossover and mutation). The selection decides candidate solutions evolving into the next generation, which limits the search space spanned by the candidate solutions. The crossover and mutation generate new solution candidates. However, genetic operators used for generating new solution candidates are a little different among GA, EP, and ES from the historical point of view [17]. The important feature of ES is the self-adaptation which can self-tune the diversity of mutation parameters according to the success records. Rechenberg suggested that the ratio of successful mutations to all mutations should be 1/5 [18]. If this ratio is greater than 1/5, increase the variance; if it is less, decrease the variance. This ratio has often been discussed in the previous studies, but the self-adaptive mutation can change the variance of the normal random value according to the success ratio based on the landscape of a fitness function. While a self-adaptive

mutation refers to its own fitness record, an adaptive mutation refers to the average, maximum, and minimum of fitness values of the candidate solutions in the population, i.e., the adaptive mutation relatively changes the distribution of genotype in a population according to the fitness values of the candidate solutions. ES was proposed by Rechenberg, and extended further by Schwefel. Basically, ES is classified into $(\mu + \lambda)$-ES and (μ, λ)-ES. First, Algorithm 1 presents the procedure of a standard $(\mu + \lambda)$-ES. Initialization randomly generates an initial population of individuals. Creation (λ) generates λ children from μ parents by genetic operators in a single generation. As a result, the $(\mu + \lambda)$-ES has the intermediate population of $(\mu + \lambda)$ individuals. Selection (μ) deterministically selects the best μ individuals from the intermediate population. On the other hand, in (μ, λ)-ES, Selection (μ) selects the best μ individuals only from the created λ children $(\mu < \lambda)$. Therefore, $(\mu + \lambda)$-ES is considered as a continuous model of generation, while the (μ, λ)-ES is considered as a discrete model of generation. Especially, as the special cases of ES, (1,1)-ES is a random search, (1+1)-ES is an iterative improvement method, $(1, \lambda)$-ES or $(1 + \lambda)$-ES is a multi-point neighboring search, and $(\mu + 1)$-ES is a local hill-climbing search. In our proposed method, the search method is mainly based on mutation operators and uses self-adaptive mutation since we assume that the dataset includes a huge number of noises. The mutation operator is very important to extract the correct pairs from the dataset effectively. Therefore, we use ES in this study. In ESSAC, we use uniform crossover as a recombination and simple mutation that changes genes randomly according to the mutation rate.

Algorithm 1 Standard ES

```
Initialization
while until termination condition is True do
    Creation (λ)
    Evaluation
    Selection (μ)
end while
```

3.3 Search range control

ESSAC performs a search range control to reduce computational cost and improve the stability of search simultaneously. This subsection proposes the search range control method that is the feature of ESSAC. The step that requires computational cost in SAC algorithm is the hypothesis evaluation step. In the hypothesis evaluation, the generated model parameters are evaluated using all data in a data set S of the possible pairs. Therefore, if the data set has a huge mount of data such as 3D image processing, the computational cost is very expensive. Furthermore, it is difficult to optimize the combination of candidate data if the outlier rate in the set S is very high. Therefore, in the search range control method of ESSAC, if an individual satisfied with starting condition is generated, the search space is reduced by removing obvious outliers from the data set S. Figure 4 shows the concept image of search range control method of ESSAC. Specifically, using the model parameters of the best individual and predefined threshold $\tau_a(\tau_a > \tau)$, the number of removal data N_a and the set S_a are calculated as follows:

$$N_a = \sum_{i=1}^{N} \rho_a(e_i) \tag{15}$$

$$S_a \leftarrow S_a + \{i\} \tag{16}$$

$$\rho_a(e_i) = \begin{cases} 1 & (\text{if } e_i > \tau_a) \\ 0 & (\text{otherwise}) \end{cases}. \tag{17}$$

Furthermore, using N_a and S_a, the number of data N and the set S are updated as follows:

$$N \leftarrow N - N_a \tag{18}$$

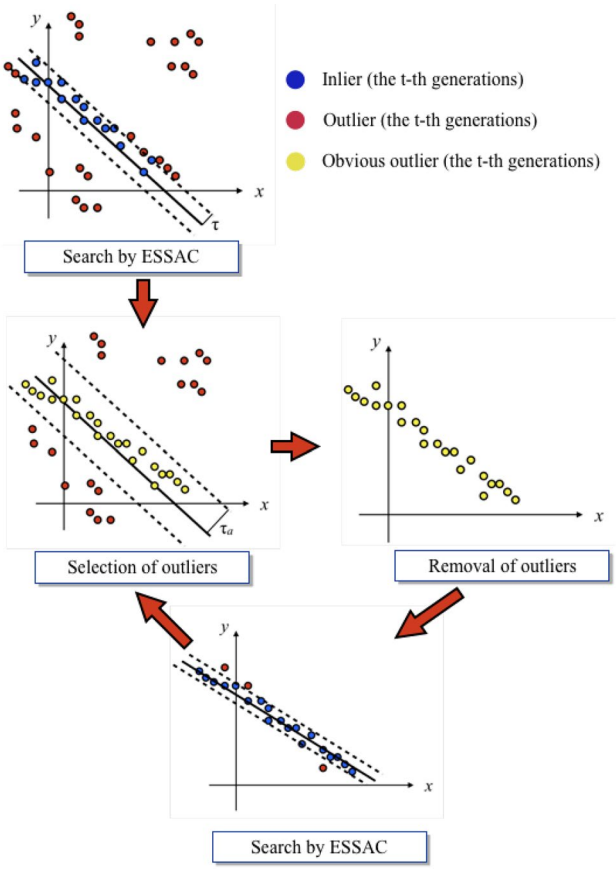


Fig. 4 Concept image of search range control method in ESSAC

$$S \leftarrow S - S_a. \tag{19}$$

By removing the obvious outliers from the dataset S, ESSAC reduces the computational cost by the number of obvious outliers N_a in the evaluation step as compared with RANSAC algorithm. Therefore, the computational cost of ESSAC depends on the rate of the outliers in the dataset is significantly reduced as the rate is increased. The model parameters are estimated from the updated data set S. In addition, the starting condition in homography matrix estimation is defined by

$$\frac{fit_{best}}{N} > \alpha, \tag{20}$$

where α is a threshold value. The condition means the ratio of the fitness value of the best individual to the number of data in the data set. After the search range control method is once performed, the obvious outliers are removed from the set S when the best individual is improved.

3.4 Adaptive mutation

This subsection explains the adaptive mutation rate that enables to change the mutation rate according to the fitness value of the best individual to improve the stability and accuracy for ESSAC. Because if the outlier rate is high, the good combination is not generated even in the later phase of the search in some cases. Specifically, the adaptive mutation rate is determined by the following equation:

$$P_{m1} = 1 - \exp\left(-\frac{fit_{best}}{T_m}\right), \tag{21}$$

where fit_{best} is the fitness value of the best individual and T_m is coefficient. In this paper, we use $T_m = 0.2 \cdot N$. This value is empirically determined. On the other hand, the search is based on the recombination operator in Eq. (20) when the fitness value of the best individual is low during an early stage. In Eq. (20), the recombination randomly selects two host individuals to maintain the genetic diversity.

3.5 Self-adaptive mutation

To start search range control method efficiently, the search capability of ES is very important since we must search the feasible solution from the dataset including the large number of outliers quickly. Therefore, we need to control a ratio of exploration and exploitation. To control the ratio, ESSAC uses a self-adaptive mutation. In the adaptive mutation, if the search fails for m times in a row, the mutation rate is calculated by following equation:

$$P_{m2} = 1 - P_{m1}, \tag{22}$$

where P_{m1} is calculated by Eq. (20). In Eq. (21), the search is based on the mutation operator when the fitness value of the best individual is low and the mutation rate decreases when an individual having good genetic information is generated. Therefore, ESSAC can control the ratio using adaptive and self-adaptive mutation. Algorithm 2 shows the procedure of adaptive ESSAC.

Algorithm 2 A-EESAC

```
for i = 1 to μ parents do
  g(i) =Random sample k data from set S
  fit_i = fitnessFunction(g(i))
end for
while until termination condition is True do
  for i = 1 to λ children do
    if starting condition do then
      Select best individual and another individual in relation to their fitness
    else
      Select two host individuals randomly
    end if
    Apply recombination operator with probability P_c
    if search fails for m times in a row do then
      Apply mutation operator with probability P_m1
    else
      Apply mutation operator with probability P_m2
    end if
    fit_i = fitnessFunction(g(i))
  end for
  if starting condition of search range control method do then
    Remove obvious outliers from data set S by using threshold τ_a
  end if
  Select best λ individuals as the next generation
end while
```

4 Experimental result

We conducted an experiment on 3D reconstruction to verifying the effectiveness of A-ESSAC in a moving image. Left and right images were acquired using a robot arm equipped with two cameras and the number of frames is 670 (Fig. 5). Figure 6 shows examples of camera images from the left camera. Table 1 shows the experimental parameters of A-ESSAC, and these parameters were determined empirically (Table 2).

Figure 7 shows the experimental result of homography matrix estimation between $D_L(t)$ and $D_R(t)$ and Table 3 shows the results of a comparison between A-ESSAC, RANSAC and GASAC with Simulated Annealing (GASAC-SA) [19] (The number of trials of each dataset is set to 1000). A-ESSAC and GASAC-SA outperforms RANSAC in

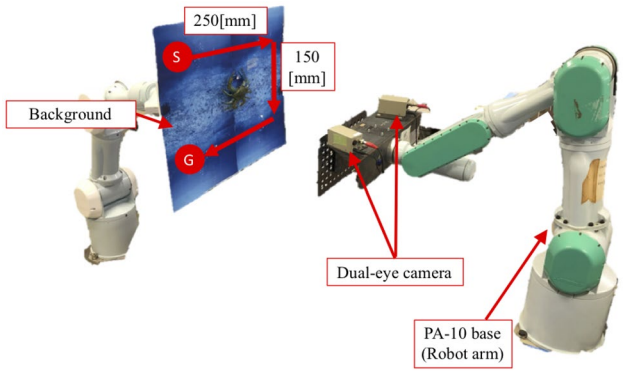


Fig. 5 Experimental environment (*S* and *G* indicate Start and Goal, respectively)

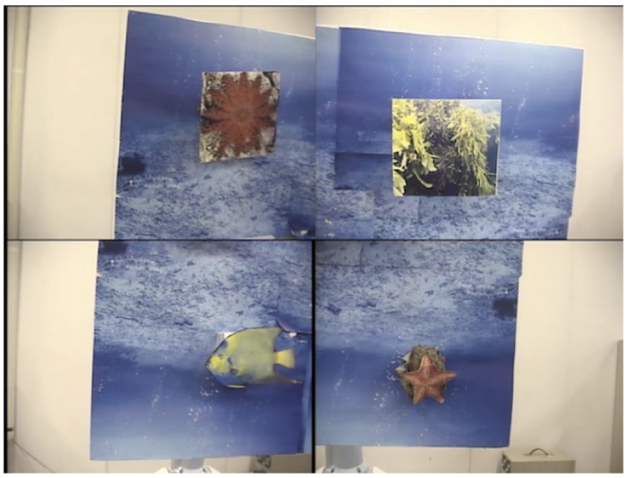


Fig. 6 Examples of camera images from the left camera

Table 1 Experimental condition and parameter setting

The number of evaluations	1000
The number of trials	10
The number of parents μ	100
The number of offspring λ	10
Threshold of Inlier/Outlier τ	3
Threshold of search range control τ_a	10
Initial condition of search range control α	0.01

all of the datasets from the viewpoint of the average fitness value since the search capability of the genetic operators. In addition, A-ESSAC performs the search range control for searching better feasible solution by removing the obvious outliers from the dataset. Therefore, the result of A-ESSAC is slightly better than that of GASAC-SA. Furthermore, the computational time of A-ESSAC is less than the other methods because of the search range control. From the result,

Table 2 Results of a comparison experiment between A-ESSAC, RANSAC and GASAC-SA

Step(t)	A-ESSAC	RANSAC	GASAC-SA
Average fitness value (variance)			
1	168.7 (1243)	99.2 (2624)	150.8 (664)
230	147.2 (900)	58.7 (1619)	143.4 (316)
380	143.7 (41)	113.5 (660)	139.0 (286)
670	239.6 (50)	196.7 (1199)	218.6 (234)
Average computational time (variance)			
1	25.8 (47)	52.4 (2)	76.2 (11)
230	32.6 (197)	50.8 (2)	62.9 (12)
380	17.8 (3)	30.2 (1)	45.5 (4)
670	26.1 (5)	44.5 (3)	48.9.5 (5)

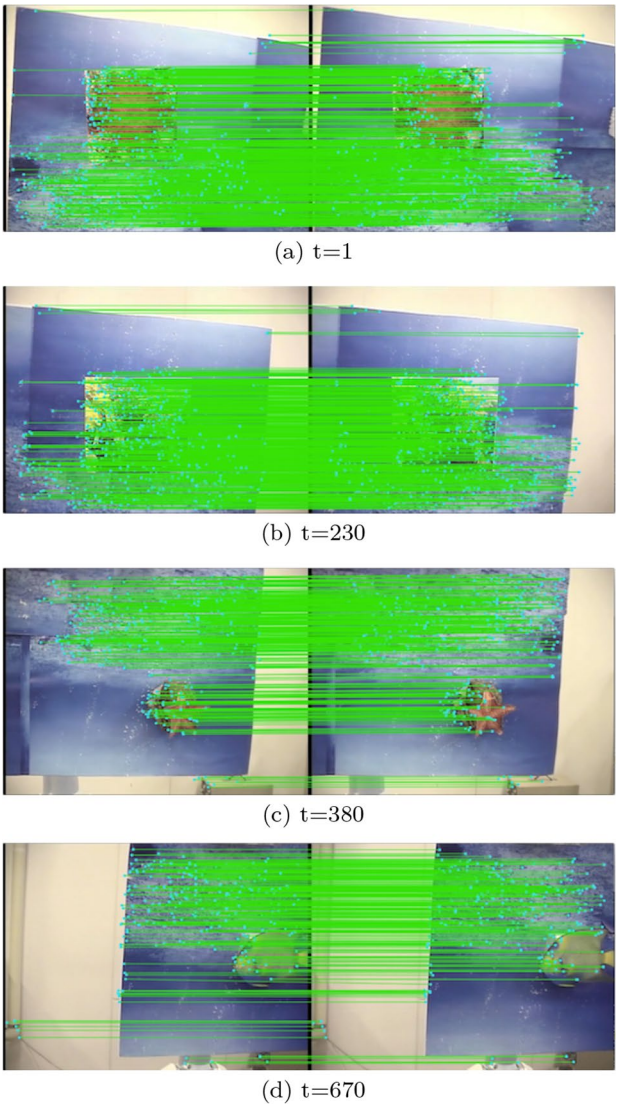


Fig. 7 Experimental results of a homography matrix estimation (left and right images are the measurement data from the left and right cameras, respectively)

Table 3 A result of a comparison experiment between A-ESSAC (using Eqs. 20 and 21), A-ESSAC (using only Eq. 20) and fixed value in [6]

	Average fitness value (variance)		
Step(t)	A-ESSAC	Eq. (20)	[6]
35	115.3 (1497)	106.2 (2050)	96.3 (2765)

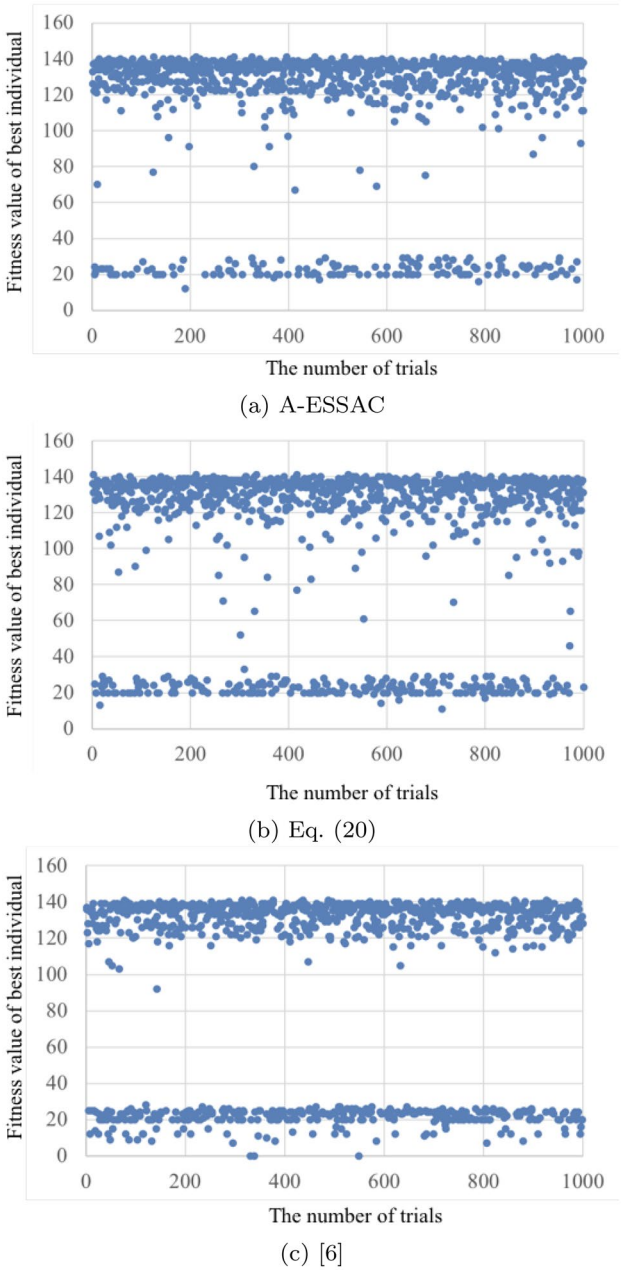


Fig. 8 1000 Trials result of the comparison experiment shown in Table 3

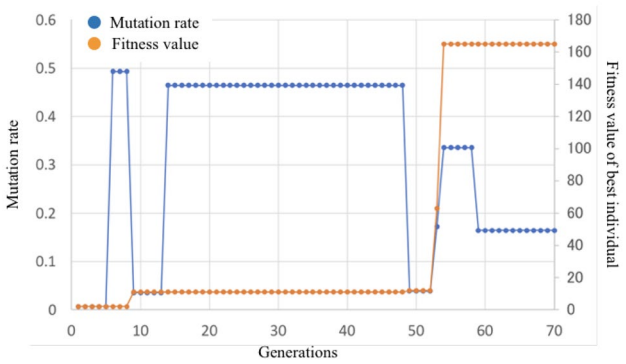
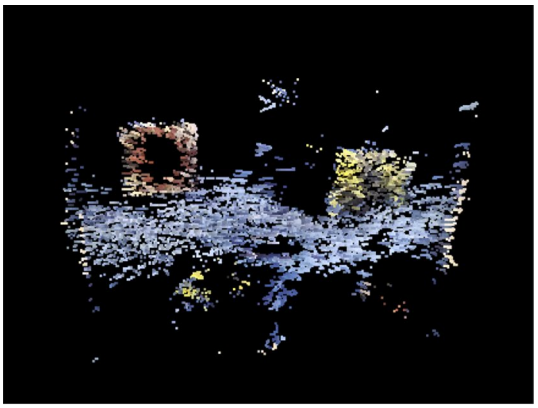


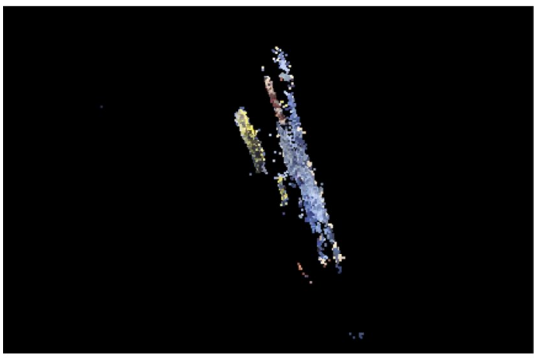
Fig. 9 An example of mutation rate transition ($t = 1$). Blue and Orange dots indicate the result of the mutation rate and fitness value of best individual, respectively

A-ESSAC can improve the trade-off between computational cost and stability of search.

Next, Table 3 and Fig. 8 show a result of a comparison between A-ESSAC using the adaptive and self-adaptive mutations (Eqs.20, 21), A-ESSAC using only the adaptive mutation (Eq. 20) and ESSAC using the fixed mutation rate in [6] (the mutation rate $P_m = 0.125$) for verifying the effectiveness of our proposed mutation strategy. In this experiment, the number of trials is set to 1000. From the result, our proposed mutation strategy outperforms the other mutation strategies in the viewpoint of the average fitness value and variance. Furthermore, Fig. 9 shows a result of the mutation rate transition for considering the result of Table 3. In Fig. 9, the self-adaptive mutation was performed in the 5th and 13th generations since A-ESSAC could not improve the fitness value for 5 times in a row. After these self-adaptations, the search strategy is similar to the random search method since the mutation rate was over 0.4 for increasing the genetic diversity. In the 47th generation, the best fitness value was improved, and the mutation rate was decreased for searching the better combination in the current gene set. In this situation, the crossover operation is dominant compared with the mutation operator since the mutation rate is less than 0.1. From these results, by changing the search strategy according to the best fitness value, A-ESSAC could search the feasible combination in this result (Fig. 9). Although A-ESSAC sometimes gets stuck in a local optimum in some trials (e.g. the variance results (t=1 and 230) of the fitness value are slightly large), A-ESSAC can recover from such a situation by extracting the correct pairs between the next images. Therefore, the result of 3D reconstruction could be stably performed using A-ESSAC (Fig. 10).



(a) Front view



(b) Side view

Fig. 10 Experimental result of 3D reconstruction using two camera images

5 Conclusion

In this paper, we applied A-ESSAC to 3D reconstruction method using two cameras. At first, we explained the 3D reconstruction method from two cameras and defined the homography matrix estimation problem. Next, we explained A-ESSAC whose search strategy is based on an evolution strategy to maintain the genetic diversity. In the experiments, we showed that A-ESSAC outperforms RANSAC in the average fitness and computational time and our proposed method could reconstruct the 3D model from two cameras. However, our proposed method has the problem of the accuracy of 3D reconstruction because of accumulated errors in each frame. Therefore, we will apply a closed-loop method to our proposed method for improving the accuracy of the 3D model.

References

1. Fischler M, Bolles R (1981) Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun ACM 24(6):381–395
2. Adam A, Chatzilari E, Nikolopoulos S, Kompatsiaris I (2018) H-RANSAC: a hybrid point cloud segmentation combining 2D and 3D data. ISPRS Ann photogramm Remote Sens Spat Inf Sci 4(2):1–8
3. Narksri P, Takeuchi E, Ninomiya Y, Morales Y, Akai N, Kawaguchi N (2018) A slope-robust cascaded ground segmentation in 3D point cloud for autonomous vehicles. In: 2018 21st International Conference on intelligent transportation systems (ITSC), pp 497–504
4. Qian X, Ye C (2014) NCC-RANSAC: a fast plane extraction method for 3-D range data segmentation. Cybern IEEE Trans 44(12):2771–2783
5. Choi S, Taemin K, Wonpil Y (2009) Performance evaluation of RANSAC family. J Comput Vis 24(3):1–12
6. Rodehorst V, Hellwich O (2006) Genetic algorithm SAmple consensus (GASAC)—a parallel strategy for robust parameter estimation. In: Proceedings of the IEEE Conference on computer vision and pattern recognition workshop, pp 103–110
7. Shojaedini E, Majd M, Safabakhsh R (2019) Novel adaptive genetic algorithm sample consensus. Appl Soft Comput 77:635–642
8. Toda Y, Kubota N (2016) Evolution strategy sampling consensus for robust estimator. J Adv Comput Intell Intell Inform 20(5):788–802
9. Lowe DG (1999) Object recognition from local scaleinvariant features. Proc. of IEEE International Conference on computer vision, pp 1150–1157
10. Bay H, Tuytelaars T, Gool LV (2006) Surf: speeded up robust features. In: European Conference on computer vision, pp 404–417
11. Cornelis N, Leibe B, Cornelis K, Gool LV (2008) 3D Urban scene modeling integrating recognition and reconstruction. Int J Comput Vis 78(2–3):121–141
12. Cyganek B, Siebert JP (2009) Introduction to 3D computer vision techniques and algorithms, Wiley. John & Sons, Incorporated
13. Torr PHS (2002) Bayesian model estimation and selection for epipolar geometry and generic manifold fitting. Int J Comput Vis 50(1):35–61
14. Chum O, Matas J (2005) Matching with PROSAC—Progressive Sample Consensus. In: Conference on computer vision and pattern recognition, pp 220–226
15. Sharp G, Lee S, Wehe D (2002) ICP registration using invariant features. IEEE Trans Pattern Anal Mach Intell 24(1):90–102
16. Horn B (1987) Closed-form solution of absolute orientation using unit quaternions. J Opt Soc Am A 4(4):629–642
17. Fogel DB (1995) Evolutionary computation. IEEE Press, New York
18. Rechenberg I (1973) Evolutionsstrategie: optimierung technischer systeme nach prinzipien der biologischen evolution. FrommannHolzboog Verlag, Stuttgart
19. Vasconcelos F, Henggeler C, Barreto JP (2011) Adaptive and Hybrid Genetic Approaches for Estimating the Camera Motion from Image Point Correspondences. ACM Conf. in Genetic and Evolutionary Computing Conference, pp 949–956