

Adaptive Evolution Strategy Sample Consensus for 3D Reconstruction from Two Cameras

Yuichiro Toda¹, Hsu Horng Yz¹, Takayuki Matsuno¹, and Mamoru Minami¹

¹Okayama University, Japan
(Tel: 81-97-594-0181, Fax: 81-97-547-9242)
¹ytoda@okayama-u.ac.jp

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 in order 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: Robust estimator, Evolutionary computation, 3D reconstruction.

1 INTRODUCTION

Recently, 3D image processing technologies have expected as a development of Robotics and Intelligent Transport Systems. In 3D image processing technologies, RANdom SAMple Consensus (RANSAC) proposed by Fischer and Bolles in order to estimate a model from a data set, is one of the most popular algorithms in robust estimator fields [1]. RANSAC has been applied to many 3D image processing problems such as homography matrix estimation problem and shape detection from 3D point clouds [2-4]. However, one of problems in RANSAC is a sampling bias in a search, since it selects candidate pairs from a data set of pairs randomly. In order 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, being accurate. In this paper, our objective is proposing new robust estimator in order to improve the trade-off between computational cost and search stability. Furthermore, Choi et al said that “the new tool can stimulate this field such as genetic algorithm of GASAC”. In this paper, therefore, we focus on evolutionary computation for RANSAC family algorithms. V. Rodehorst and O. Hellwich proposed Genetic Algorithm SAMple Consensus (GASAC) based on a population-based multi-point search to improve RANSAC [6]. GASAC can

improve the performance of search, however it is sometimes difficult to maintain the genetic diversity in the search if the large size of outliers is included in a data set. Furthermore, GASAC 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 control the rate of local search and global search. We can control the genetic diversity of a population to improve the stability in evolutionary search. The other is to manage the search range in a search space. We can remove obvious outliers from a data set. 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 in order to improve the trade-off between computational cost and stability of search in RANSAC. Furthermore, we applied to A-ESSAC to 3D map building method using RGB-D camera for realizing the real time 3D SLAM [7]. 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 explain 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 t -th 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 in order to remove the mismatched pairs. Finally, the coordinate transform matrix is estimated by using 3D distance information of the corresponding points in order to update 3D environmental map.

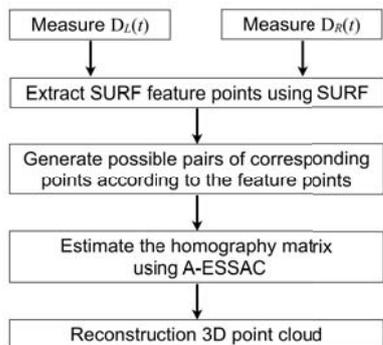


Figure 1. Flowchart of proposed 3D modeling method.

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 in order to extract the corresponding points from the multiple images [8, 9]. 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. For example, OpenCV provides the library that includes SURF implemented on GPGPU [10]. 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 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 \mathbf{H} that has 9 elements is the matrix that describes the relation between two images. Fig. 3 displays the concept image of homography matrix between two images. The set of homogeneous image points $\{\mathbf{x}_i\}$ ($i = 1, \dots, n$) as viewed in the first image is transformed into the set $\{\mathbf{x}'_i\}$ in the second image, with the positions related by

$$\mathbf{x}'_i = \mathbf{H}\mathbf{x}_i, \quad (1)$$

where \mathbf{x} and \mathbf{x}' 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, equation (1) can be also defined by the following equation:

$$x' = \frac{a_1x + a_2y + a_3}{a_7x + a_8y + 1} \quad (2)$$

$$y' = \frac{a_4x + a_5y + a_6}{a_7x + a_8y + 1} \quad (3)$$

where a_1 - a_8 are the parameters. Therefore, the homography matrix is estimated by estimating these 8 parameters. Estimating the homography matrix between two images enables to extract the correct pairs of corresponding points. However, the set of possible pairs includes many mismatch pairs. Therefore, robust estimators are applied to this problem [13, 14].

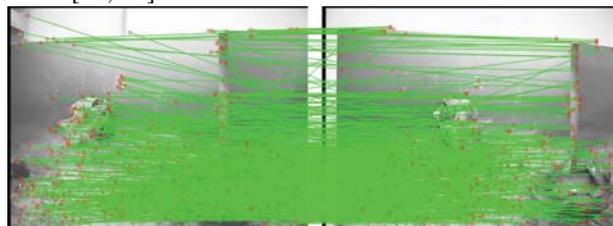


Figure 2. A result of the feature extraction and matching by using SURF.

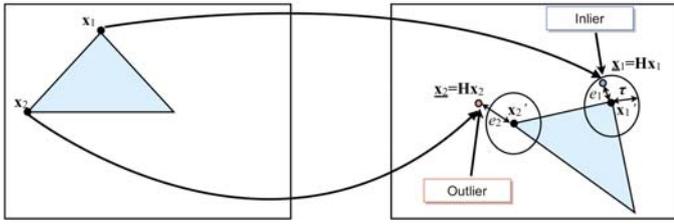


Figure 3. Concept image of 2D homography

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

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

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 in the following;

$$\mathbf{x}_r^g = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{x}_{rj}, \quad (5)$$

$$\mathbf{x}'_r{}^g = \frac{1}{N_c} \sum_{i=1}^{N_c} \mathbf{x}'_{rj}, \quad (6)$$

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

$$\mathbf{x}_i^a = \mathbf{x}_{ri} - \mathbf{x}_r^g, \quad (7)$$

$$\mathbf{x}_i^b = \mathbf{x}'_{ri} - \mathbf{x}'_r{}^g. \quad (8)$$

Next, S_{ab} is defined as;

$$S_{ab} = \sum_{i=1}^{N_c} \mathbf{x}_i^a \mathbf{x}_i^b. \quad (9)$$

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

$$\mathbf{P} = \begin{bmatrix} S_{xx} + S_{yy} + S_{zz} & S_{xc} - S_{cy} & S_{xc} - S_{cz} & S_{cy} - S_{cz} \\ S_{xc} - S_{cy} & S_{xx} - S_{yy} - S_{zz} & S_{yx} + S_{xz} & S_{cx} + S_{cz} \\ S_{xc} - S_{cz} & S_{yx} + S_{xz} & S_{yy} - S_{xx} - S_{zz} & S_{yc} + S_{cy} \\ S_{cy} - S_{cz} & S_{cx} + S_{cz} & S_{yc} + S_{cy} & S_{zz} - S_{xx} - S_{yy} \end{bmatrix}. \quad (10)$$

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;

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

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

$$\mathbf{t} = \mathbf{x}_r^g - \mathbf{R} \mathbf{x}'_r{}^g. \quad (12)$$

3 Adaptive Evolution Strategy Sample Consensus (A-ESSAC)

3.1 Total algorithm of A-ESSAC

At first, Fig. 6 shows the procedure and flowchart of ESSAC, respectively. 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

$$fit_i = fitnessFunction(g(i)) = \sum_j \rho(e_j). \quad (13)$$

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 in order 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 combinatorial 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 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, Algorithm3 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, λ)-ES or (1+ λ)-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.

<p>Algorithm 3 Standard ES:</p> <ol style="list-style-type: none"> 1: Initialization 2: while (until termination_condition is True) 3: Creation (λ) 4: Evaluation 5: Selection (μ) 6: end while
--

3.3. Search range control

ESSAC performs a search range control in order 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 by using all data in a data set S of the possible pairs. Therefore, if the data set has a huge amount 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 . Fig. 5 shows the concept image of search range control method of ESSAC. Specifically, by using the model parameters of the best individual and predefined threshold

τ_a ($\tau_a > \tau$), the number of removal data N_a and the set S_a are calculated as follows,

$$N_a = \sum_{i=1}^N \rho_i(e_i), \tag{14}$$

$$S_a \leftarrow S_a + \{i\} \quad \text{if } \rho_i(i) = 1, \tag{15}$$

$$\rho_i(e_i) = \begin{cases} 1 & \text{if } e_i > \tau_a \\ 0 & \text{otherwise} \end{cases}. \tag{16}$$

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

$$N \leftarrow N - N_a, \tag{17}$$

$$S \leftarrow S - S_a. \tag{18}$$

The model parameters are estimated from the updated data set S . In addition, the starting condition in homography matrix estimation are defined by

$$\frac{fit_{best}}{N} > \alpha, \tag{19}$$

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 in order to improve the stability and accuracy for ESSAC. Because if the outlier rate is high, the good combination is not generated even in later phase of the search in some cases.

Specifically, the adaptive mutation rate is determined by the following equation

$$P_{m1} = 1 - \exp\left(-\frac{fit_{best}}{T_m}\right) \tag{20}$$

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

3.4 Self-adaptive mutation

In order 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. In order 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,

$$P_{m2} = 1 - P_{m1}. \tag{21}$$

In equation (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 by using adaptive and self-adaptive mutation. Algorithm 4 shows the procedure of adaptive ESSAC.

4 Experimental result

We conducted an experiment on 3D reconstruction in order to verifying the effectiveness of A-ESSAC in a moving image. Left and right images were acquired by using a robot arm equipped with two cameras and the number of frames is 670. Fig. 6 shows examples of camera images from the left camera. Fig. 7 shows the experimental result of homography matrix estimation between $D_L(t)$ and $D_R(t)$. In addition, Table 1 shows the results of a comparison between A-ESSAC and RANSAC (The number of trials of each dataset is set to 1000). A-ESSAC outperforms RANSAC in all of the datasets from the viewpoint of the average fitness value and computational cost because the search capability of ES and the search range control can improve the trade-off between computational cost and stability of search. However, the variance results ($t=1$ and 230) of the fitness value are slightly large because A-ESSAC gets stuck in a local optimum in some trials. However, 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 by using A-ESSAC (Fig.8).

4 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 Evolution Strategy in order 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, Communications of the ACM, 24(6), pp. 381–395
 [2] Tam G. K.L, et al. (2013), Registration of 3D Point Clouds and Meshes: A Survey From Rigid to Non-Rigid, IEEE Trans on Visualization and Computer Graphics, 19(7), pp.1199-1217

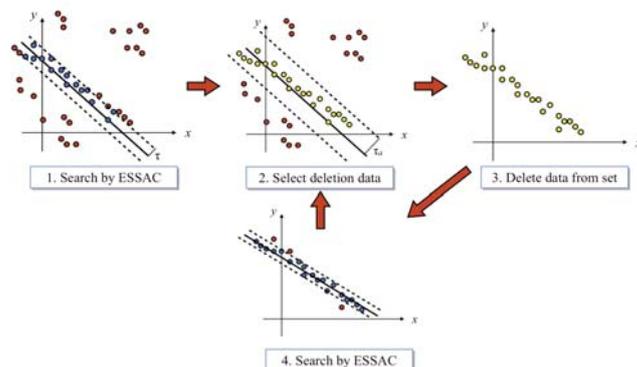


Figure 5. Concept image of search range control method in ESSAC

```

Algorithm 4 Adaptive ESSAC:
1: for  $i=1$  to  $m$  parents do
2:    $g(i)$ = Random sample  $k$  data from set  $S$ 
3:    $fit_i$ =fitnessFunction( $g(i)$ )
4: end for
5: while not Stopping criteria do
6:   for  $i=1$  to  $\lambda$  children do
7:     if starting condition do
8:       Select best individual and another individual in
         relation to their fitness
9:     else
10:      Select two host individuals randomly
11:    end if
12:    Apply recombination operator with probability  $P_c$ 
13:    if search fails for  $\mu$  times in a row do
14:      Apply mutation operator with probability  $P_{m1}$ 
        (Adaptive mutation)
15:    else
16:      Apply mutation operator with probability  $P_{m2}$ 
        (Adaptive mutation)
17:    end if
18:     $fit_i$ =fitnessFunction( $g(i)$ )
19:  end for
20:  if starting condition do
21:    Remove obvious outliers from data set  $S$ 
        by using threshold  $\tau_a$ 
22:  end if
23:  Select best  $\mu$  individuals as the next generation
24: end while
    
```



Figure 6. Examples of camera images from the left camera.

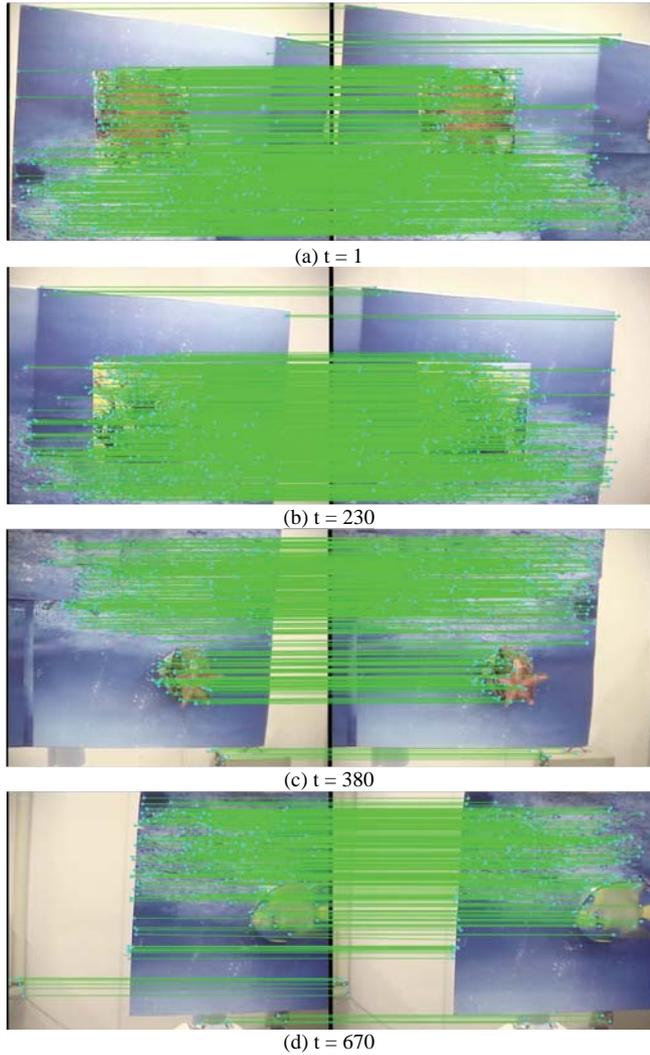


Figure 7. Experimental results of a homography matrix estimation (Left and Right images are the measurement data from the left and right camera, respectively)

Table 1 Results of a comparison experiment between A-ESSAC and RANSAC

Step(t)	Average of fitness value (Variance)		Average of computational time (Variance)	
	A-ESSAC	RANSAC	A-ESSAC	RANSAC
1	168.7(1243)	99.2(2624)	25.8(47)	52.4(2)
230	147.2(900)	58.7(1619)	32.6(197)	50.8(2)
380	143.7(41)	113.5 (660)	17.8(3)	30.2(1)
670	239.6(50)	196.7(1199)	26.1(5)	44.5(3)

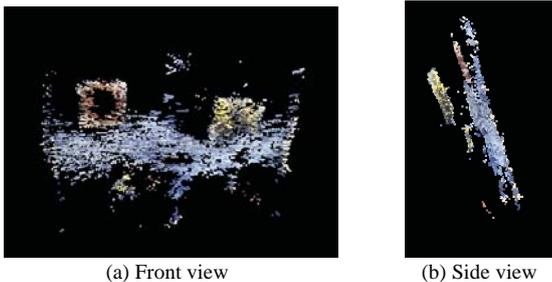


Figure 8. Experimental result of 3D reconstruction using two camera images

[3] Han J, Shao L, Xu D, Shotton J (2013), Enhanced Computer Vision With Microsoft Kinect Sensor: A Review, *Cybernetics, IEEE Transactions on*, 43(5), pp.1318-1334

[4] Qian X, Ye C (2014), NCC-RANSAC: A Fast Plane Extraction Method for 3-D Range Data Segmentation, *Cybernetics, IEEE Transactions on*, 44(12), pp.2771-2783

[5] Choi S, Taemin K, Wonpil Y (2009), Performance evaluation of RANSAC family, *Journal of Computer Vision*, .24(3) pp.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] Toda Y, Kubota N (2016), Evolution Strategy Sampling Consensus for Robust Estimator, *J. Adv. Comput. Intell. Intell. Inform.*, 20(5), pp. 788-802

[8] Lowe D. G (1999), Object recognition from local scaleinvariant features, *Proc. of IEEE International Conference on Computer Vision*, pp.1150-1157

[9] Bay H, Tuytelaars T, Gool L. V (2006), Surf: Speeded up robust features, in *European Conference on Computer Vision*, pp. 404-417

[10] OpenCV, <http://opencv.org/>.

[11] Cornelis N, Leibe B, Cornelis K, L. Gool V (2008), 3D Urban Scene Modeling Integrating Recognition and Reconstruction, *International Journal of Computer Vision*, 78 (2-3), pp. 121-141

[12] Cyganek B, Siebert J. P (2009), Introduction to 3D Computer Vision Techniques and Algorithms, Wiley, John & Sons, Incorporated

[13] Torr P.H.S (2002), Bayesian model estimation and selection for epipolar geometry and generic manifold fitting, *International Journal of Computer Vision*, 50(1), pp. 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 Transactions on Pattern Analysis and Machine Intelligence*, 24(1), pp.90-102

[16] Horn B (1987), Closed-Form Solution of Absolute Orientation Using Unit Quaternions, *Journal of the Optical Society of America A*, 4(4), pp.629-642

[17] Fogel D.B (1995), "Evolutionary Computation", IEEE Press

[18] Rechenberg I (1973), "Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution", Stuttgart: FrommannHolzboog Verlag