Accurate reconstruction of the 3D indoor environment map with a RGB-D camera based on multiple ICP

A N Ruchay[1,2], K A Dorofeev[2], A V Kober[1]

[1]Federal Research Centre of Biological Systems and Agro-technologies of the Russian Academy of Sciences, 9 Yanvarya street 29, Orenburg, Russia, 460000

[2]Department of Mathematics, Chelyabinsk State University, Bratiev Kashirinykh street 129, Chelyabinsk, Russia, 454001

Abstract. In this paper, we propose a new method for 3D map reconstruction using the Kinect sensor based on multiple ICP. The Kinect sensor provides RGB images as well as depth images. Since the depth and RGB color images are captured by one Kinect sensor with multiple views, each depth image should be related to the color image. After matching of the images (registration), point-to-point corresponding between two depth images is found, and they can be combined and represented in the 3D space. In order to obtain a dense 3D map of the 3D indoor environment, we design an algorithm to combine information from multiple views of the Kinect sensor. First, features extracted from color and depth images are used to localize them in a 3D scene. Next, Iterative Closest Point (ICP) algorithm is used to align all frames. As a result, a new frame is added to the dense 3D model. However, the spatial distribution and resolution of depth data affect to the performance of 3D scene reconstruction system based on ICP. In this paper we automatically divide the depth data into sub-clouds with similar resolution, to align them separately, and unify in the entire points cloud. This method is called the multiple ICP. The presented computer simulation results show an improvement in accuracy of 3D map reconstruction using real data.

1. Introduction

Surface reconstruction of real-world objects by using RGB-D sensors such as the Kinect is one of the most important topics in computer vision [1, 2, 3, 4]. These cameras can provide a high resolution RGB color image with depth data of the environment. In contrast to the stereo and monocular cameras, RGB-D sensors are able to obtain depth data directly, even in environments with poor visual textures. A full 3D mapping system that models indoor environment utilizing a joint optimization algorithm combining visual features and shape-based alignment was proposed in [5, 6, 7]. Accurate real-time mapping algorithm for complex indoor scenes in variable lighting conditions was suggected in [8]. It is a frame-to-global method and it maintains the single scene model with a global volumetric, truncated signed distance function representation.

Building dense 3D maps of environments is an important task for mobile robotics, with applications in navigation, manipulation, semantic mapping, face recognition [9, 10, 11, 12, 13, 14, 15, 16, 17], and in augmented reality applications, surveillance systems, medical applications [18, 19, 20, 21, 22].

Typically, a 3D mapping system based on RGB-D cameras starts with detection and analysis of distinctive points of the environment. These points are visual marks helping to create the map. However, due to the diversity of situations and conditions of the problem, such as noise introduced in the capture step, lighting changes, scaling, rotation, and other transformations, the most appropriate detector/descriptor for extraction of visual features has not yet been determined. So, it is important to identify the best visual marks of the environment. It requires a descriptor that provides the transformation invariance of points of interest, and stable detection of the same points along sequences with different variations in images.

Once the key points have been identified and matched with the points in the following frame, the next step is to align the three-dimensional data in consecutive frames, generally using the Iterative Closest Point (ICP) algorithm [23]. ICP based algorithms require an initial guess of the rigid transformation between two consecutive frames. This initial transformation, rotation and translation between two adjacent frames, can be calculated by the random sample consensus (RANSAC) method [23].

We are interested in improving the quality of the 3D map modeling [24, 25, 26, 27, 28, 29]. The following methods are proposed [30, 31, 32, 23, 3] to reduce the number of points to be processed, to select the best points, to improve the accuracy of location of the points, to improve the accuracy of alignment of frames and cloud points, to improve the map quality. In this paper we focus on a strategy to improve the accuracy in the alignment of 3D data by decomposing the data along depth axis and followed by the ICP method. This method is referred as multiple ICP.

A method for improving the accuracy of a global alignment with the ICP was proposed in paper [23]. Because depth data have different resolution, the depth data decomposed into several point regions (sub-clouds) with a similar resolution yields a better 3D scene restoration in terms of the accuracy. Also, one can observe that with a smaller number of points, a better result can be achieved than using all inliers generated by RANSAC. However the problem of automatical dividing of the depth data into sub-clouds with similar resolution has not solved yet. Therefore in this paper we proposed a new algorithm for automatical dividing the depth data into sub-clouds with a similar resolution.

The paper is organized as follows. Section 2 discusses related approaches of improving ICP algorithm. Section 3 presents the proposed algorithm of multiple ICP and the algorithm for automatic dividing of the depth data. Section 4 describes computer simulation results of accuracy of 3D map reconstruction using real data. Finally, Section 5 presents our conclusions.

2. Related work

An improved ICP algorithm that can automatically register multiple 3D data sets from unknown viewpoints is presented in [33]. The sensor projection that represents the mapping of the 3D data into its associated range image and a cross projection are used to determine the overlapping region of two range data sets. A combining ICP algorithm with the sensor projection makes an automatic registration of multiple 3D sets without pre-procedures that are prone to errors and any mechanical positioning device or manual assistance.

An ICP algorithm that uses matching points' center of gravity as reference points and point pair distance constraint to reject false point pairs for point cloud registration was proposed [34]. Authors use the nearest criterion to remove point pairs which contain same points by close point criterion in order to improve speed and accuracy of point cloud registration.

The paper [35] proposed an improved solution to overcome the problem, that the ICP algorithm is to require complicated computation steps. The proposed algorithm selects the best sample points in oder to reduce computation time while still retains accuracy.

An efficient and robust subset-ICP rigid registration method is proposed to reduce the computation complexity and improve the flexibility of ICP algorithm in the paper [36].

The paper [37] presented a more robust ICP algorithm. an extended ICP algorithm solve a problem of the least square registration by an inequality constraint of the rotation angle. A closed-form solution for the rotation is obtained according to the monotonicity of the model with respect to the rotation angle at each iterative step of the algorithm. The proposed approach extends the convergence domain of the ICP algorithm, and it can be used much more widely.

A variant of the ICP registration method by introducing a novel weight-bootstrap scheme was proposed in the paper [38]. The rigid transform between two point sets can be estimated, as long as proper correspondences are given. The accuracy of the estimated transform is mainly determined by the goodness of these matches. The proposed solution parameterizes the pair matching confidence and improves the optimization process to address the challenge, that the confidence of the correspondences is weakened by the observation error.

The paper [39] proposed a new approach for the registration of depth data with color data, which combines epipolar constraints with pointtoplane constraints to improve ICP algorithm and achieve accurate registration.

An improved ICP algorithm which based on k-d tree was presented for tree point cloud data registration [40]. The k-d tree is similar to the general tree structure and it can store, manage and search data efficiently. The proposed algorithm solves the problem, that massive point cloud data bring a great number of troubles to registration method. Also the improved ICP algorithm is presented in the paper, the proposed ICP algorithm is the combination of the original ICP algorithm and k-d tree [41].

3. The proposed algorithm of multiple ICP

In order to reconstruct 3D scene, first we solve the problem that depth data have different resolution [23]. As described above, once the key points have been identified and matched, the RANSAC algorithm is applied to remove outliers, and then the ICP method is applied to the data for aligning. Usually, supposing that transformation is orthogonal and objects are rigid, the ICP method is utilized over all inliers points provided by RANSAC. Note, that the use of a high number of points does not guarantee a better alignment. Moreover, it has been found that the relative motion between frames should be small enough for a good alignment. When the displacement between frames is relatively large, the algorithm takes a lot of time to converge or sometimes one diverges owing to big alignment errors. In this circumstance, the modeling map system yields a bad accuracy. A possible strategy for a lower computation time could be to reduce the number of points to be aligned. However, if the selected points are mainly located on the same deep plane, then the ICP yields wrong results.

On the other hand, it has been shown that the depth data provided by the Kinect sensor have different resolutions. With increasing distance from the sensor, the data resolution becomes worse [42].

With the aim of improving the accuracy in point clouds alignment, one can propose to divide data into regions with similar resolutions. In other words, in this manner we can get several sub-clouds and then align them separately. After, the entire points cloud is reconstructed by unifying of the aligned sub-clouds.

We show that by dividing the 3D data into sub-clouds, aligning them separately, and finally unifying them back, gives us a better 3D scene reconstruction than that if applying a single ICP for all inliers of the scene.

For each experiment presented on this work, the following basic steps carry out:

(i) Detection and matching of global key points in RGB data with SURF algorithm;

(ii) Remove outliers with RANSAC;

(iii) Align 3D data with ICP using the associate 3D points of the inliers;

(iv) Evaluate the quality of alignment;

(v) Construct 3D map.

In order to compare the accuracy of the alignment, ICP uses the root mean square error (RMSE) between two point clouds after each iteration,

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} |s_i - t_i|^2}, \tag{1}$$

where s_i and t_i are the corresponding pair points in two point clouds, and n the total number of valid pair points. The correspondence is determined by using the k-d tree search algorithm.

The proposed algorithm for automatic dividing the depth data into sub-clouds with similar resolution consists of several simple procedures written in Matlab.

(i) `AutoLength` divides the point cloud to n point sub-clouds evenly according to length of intervals. Generally, the point cloud contains points with depth values from 0 to 10 meters. So, a cloud can be divided into n equal sub-clouds with intervals $[0, 10/n), [10/n, 2 \cdot 10/n), \ldots, [9 \cdot 9/n, 10]$.

(ii) `AutoLengthIncrease` divides the point cloud into n point sub-clouds according to quadratic increase of $x, x^2, x^4, x^8, \ldots, x^{2^{n-1}}$ length of intervals of sub-clouds, respectively. We calculate x such a manner that $x + x^2 + x^4 + ... + x^{2^{n-1}} = 10$ (full length of the interval of interest from 0 to 10 meters).

(iii) `AutoLengthDecrease` divides the point cloud into n point sub-clouds according to quadratic decrease of $x^{2^{n-1}}, x^{2^{n-2}}, \ldots, x^4, x^2, x$ length of intervals of sub-clouds, respectively. We calculate x such a manner that $x + x^2 + x^4 + ... + x^{2^{n-1}} = 10$ (full length of the interval of interest from 0 to 10 meters).

(iv) Previous procedures do not take into account the number of points in each automatical divided sub-clouds. Sometimes it leads to serious problems because ICP algorithm does not owing to a sub-cloud contains a small number of points. A new procedure `AutoCountAllPoints` divides the point cloud into n sub-clouds with the same number of points. Let a point cloud contain points with depth values from 0 to 10 meters with the total number P of points. The point cloud can be automatically divided into n intervals with the number P/n of points based on a heuristic method which consistently increases the current interval of 0.01 meter. At each iteration we count the current number of points containing into the interval untill the number of point in sub-clouds reaches P/n.

(v) To divide the point cloud into n sub-clouds with the same number of inliers points from SURFF algorithm. Let a point cloud contain I inliers points (the total number of points is much larger). `AutoCountInlierPoints` divides the point cloud automatically into n intervals with the number I/n of points based on a heuristic method which consistently increases the current interval of 0.01 meter. At each iteration we count the current number of points containing into the interval untill the number of point in sub-clouds reaches I/n.

Note that by applying the ICP algorithm to sub-clouds a set of transformation matrices are obtained. The transformation matrices could be used in different ways for reconstruction of the 3D dense map of the environment: each matrix `tforms_i` from ICP algorithm to each i sub-cloud, or `tform_1` for all sub-clouds.

4. Computer simulation

In this section, computer simulation results of accuracy of 3D map reconstruction using real data are presented and discussed.

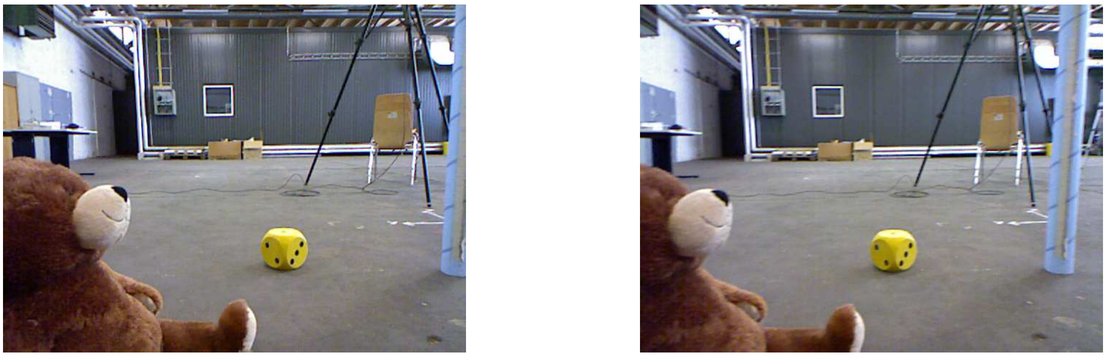

Figure 1. Pair of frames.

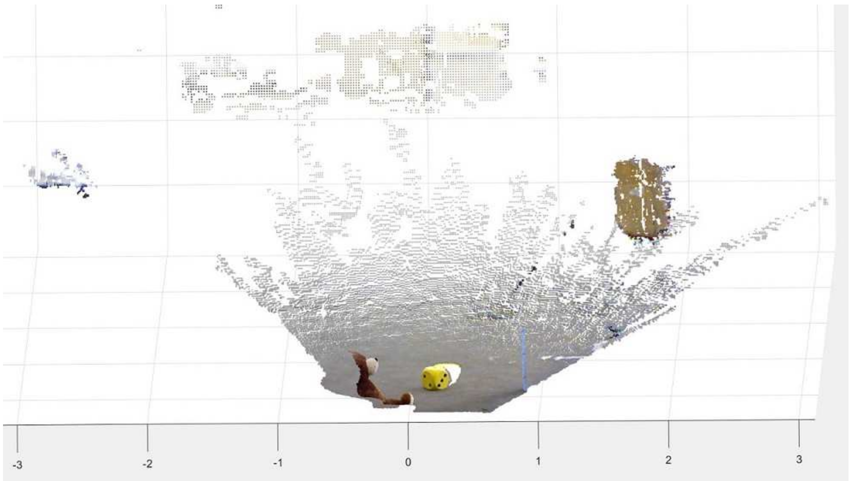

Figure 2. Point cloud from the first frame.

The tests were carried out with frames of ”freiburg/pioneer slam”, ”freiburg2/pioneer slam”, ”freiburg3/pioneer slam” datasets of TUM RGB-D benchmark [4]. We calculate RMSE errors using real data for the proposed method of 3D map reconstruction based on multiple ICP and automatical dividing the depth data into sub-clouds. Fig. 1 shows the pair of frames to be aligned. If the point cloud is aligned, the data are transformed to the coordinate system of the ground truth for further evaluation using the RMSE. Fig. 2 shows the corresponding point cloud of the first frame.

Next we divide points clouds into sub-clouds by our proposed procedures. Fig. 3 shows results of automatical dividing into two point sub-clouds according to quadratic increase of the length and the distribution of all inlier valid SURFF points in the first frame after the RANSAC algorithm. The distribution of 283 and 153 inlier points for the sub-clouds is shown in Figs. 4. The procedure of automatical dividing into two point sub-clouds according to quadratic increase of the length yields the minimal RMSE errors comparing with other procedures of automatical dividing. We define a global alignment when the ICP uses all inliers for alignment of two point clouds. For the test frames in Fig. 1, we use 992 inlier points, which are distributed as shown in Fig. 3.

The transformation matrix computed with the help of the ICP algorithm for two frame is

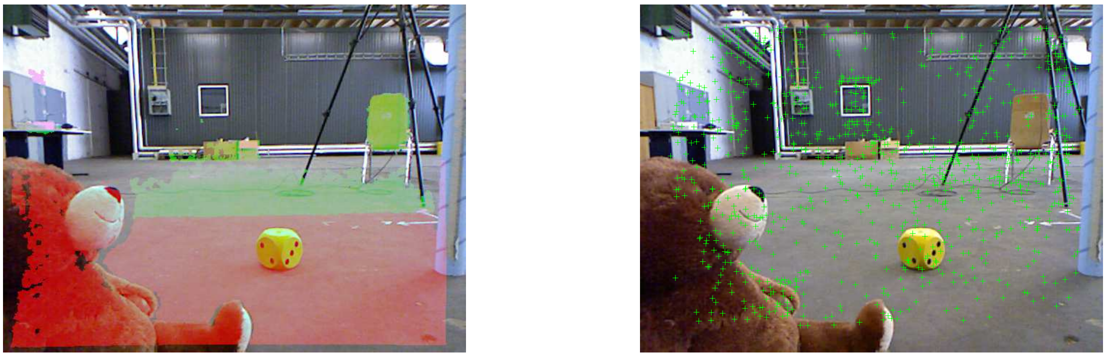

Figure 3. Left picture shows sub-clouds, and right picture shows all inlier points.

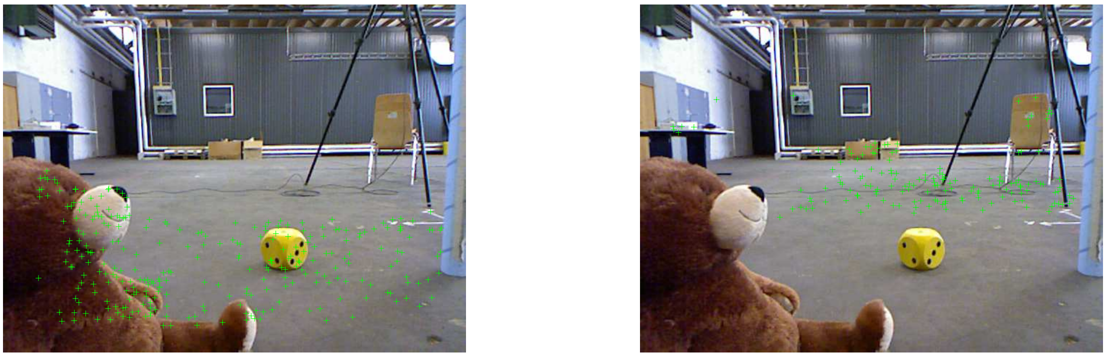

Figure 4. Inlier points for sub-clouds.

given as

$$\begin{pmatrix} 0.999381 & -0.012965 & -0.032708 & 0.000000 \\ 0.012638 & 0.999868 & -0.010166 & 0.000000 \\ 0.032836 & 0.009746 & 0.999413 & 0.000000 \\ -0.049280 & -0.024187 & 0.010850 & 1.00000 \end{pmatrix}$$

The multiple ICP algorithm returns the following transformation matrices for sub-clouds. Transformation matrix for the first sub-cloud is given as

$$\begin{pmatrix} 0.999920 & 0.000987 & -0.012579 & 0.000000 \\ -0.000981 & 0.999999 & 0.000433 & 0.000000 \\ 0.012579 & -0.000420 & 0.999921 & 0.000000 \\ -0.002609 & -0.000339 & 0.002048 & 1.000000 \end{pmatrix}$$

Transformation matrix for the second sub-cloud is given as

$$\begin{pmatrix} 0.999942 & -0.000517 & -0.010775 & 0.000000 \\ 0.000472 & 0.999991 & -0.004132 & 0.000000 \\ 0.010777 & 0.004126 & 0.999933 & 0.000000 \\ 0.002941 & -0.017613 & 0.002517 & 1.000000 \end{pmatrix}$$

The computed transformation is applied to the entire sub-clouds and compared with the ground truth. Finally, the reconstructed 3D model of the object is shown in Fig. 5.

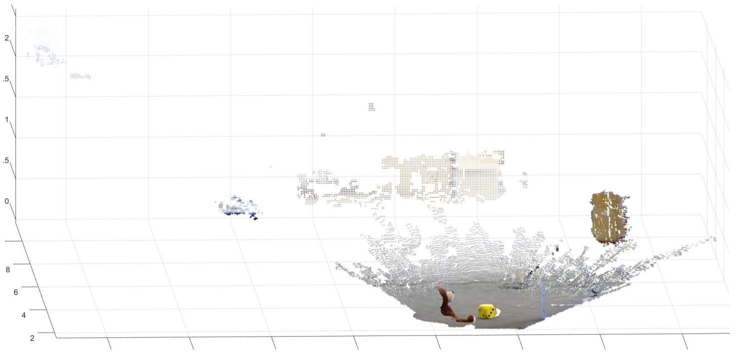

Figure 5. View of 3D scene reconstruction.

Table 1 shows RMSE errors for the proposed method of 3D map reconstruction based on multiple ICP and the various procedures of automatical dividing the depth data into sub-clouds with the real BEAR1 dataset [4].

Results of the proposed methods of automatical dividing into $n = 4$ and more sub-cloud were unsatisfactory with huge RMSE errors. The procedure `AutoLengthIncrease` with dividing the point cloud into $n = 2$ point sub-clouds according to quadratic increase of the length yields the best result in terms of RMSE. Note that the procedure can not be used for all datasets because of empty sub-cloud of inlier points. For other datasets the procedure `AutoCountInlierPoints` dividing the point cloud automatically into n intervals with the number I/n of points yields the best results. Finally, we can conclude that the accuracy of 3D reconstruction of a scene depends on the following reasons:

(i) Detection and matching of global points by feature detection algorithms. We evaluated SURFF, FAST, BRISK, Harris, MinEigen algorithms in terms of RMSE errors and the best results were obtained using SURFF algorithm, other algorithms return too few matched points;

(ii) Distribution of points in frame. If most points are close to the sensor position then no sense to divide point cloud into sub-clouds, a simple ICP algorithm provides good results.

5. Conclusion

We proposed a method of 3D map reconstruction based on multiple ICP and introduced various procedures for automatical dividing the depth data into sub-clouds. We evaluated RMSE errors using real data for the proposed method and procedures. The procedure `AutoLengthIncrease` which divides the point cloud into $n = 2$ point sub-clouds according to quadratic increase of length of intervals of sub-clouds yields the best result in terms of RMSE. In future, different datasets, especially with built accurate models will be studied and tested. Also we want to verify the performance of different descriptors for 3D scene restoration. Finally, the developed method will be incorporated into a complete SLAM system.

6. References

[1] Wang K, Zhang G and Bao H 2014 *IEEE Transactions on Image Processing* **23** 4893-4906

[2] Lee K R and Nguyen T 2016 *Mach. Vision Appl.* **27** 377-385

[3] Li H, Liu H, Cao N, Peng Y, Xie S, Luo J and Sun Y 2017 *International Journal of Advanced Robotic Systems* **14** 1729881417695560

[4] Cai Z, Han J, Liu L and Shao L 2017 *Multimedia Tools and Applications* **76** 4313-4355

[5] Henry P, Krainin M, Herbst E, Ren X and Fox D 2014 *RGB-D Mapping: Using Depth Cameras for Dense 3D Modeling of Indoor Environments* (Berlin, Heidelberg: Springer Berlin Heidelberg) 477-491

Table 1. RMSE of 3D map reconstruction for the tested algorithms using the BEAR1.

Algorithm	RMSE
SimpleICP	0.0397
n = 2 intervals	
AutoLen tforms_i	0.0172
AutoLen tform_1	0.0190
AutoLenIncrease tform_i	0.0082
AutoLenIncrease tform_1	**0.0074**
AutoLenDecrease tform_i	0.0184
AutoLenDecrease tform_1	0.0167
AutoCountAllPoints tform_i	0.0162
AutoCountAllPoints tform_1	0.0128
AutoCountInlierPoints tform_i	0.0111
AutoCountInlierPoints tform_1	0.0108
n = 3 intervals	
AutoLen tform_i	0.0177
AutoLen tform_1	0.0117
AutoLenIncrease tform_i	0.0122
AutoLenIncrease tform_1	0.0129
AutoLenDecrease tform_i	-
AutoLenDecrease tform_1	-
AutoCountAllPoints tform_i	0.0210
AutoCountAllPoints tform_1	0.0392
AutoCountInlierPoints tform_i	0.0101
AutoCountInlierPoints tform_1	0.0136
n = 4 intervals	
AutoLen tform_i	0.0144
AutoLen tform_1	0.0081
AutoLenIncrease tform_i	0.0130
AutoLenIncrease tform_1	0.0188
AutoLenDecrease tform_i	-
AutoLenDecrease tform_1	-
AutoCountAllPoints tform_i	0.0154
AutoCountAllPoints tform_1	0.0287
AutoCountInlierPoints tform_i	0.0180
AutoCountInlierPoints tform_1	0.0299

[6] Wang H, Wang J and Wang L 2016 *IEEE Conference on Computer Vision and Pattern Recognition* 3271-3279

[7] Li Y, Wang Y and Wang D 2017 *Computers & Electrical Engineering*

[8] Newcombe R A, Izadi S, Hilliges O, Molyneaux D, Kim D, Davison A J, Kohi P, Shotton J, Hodges S and Fitzgibbon A 2011 *10th IEEE International Symposium on Mixed and Augmented Reality* 127-136

[9] Echeagaray-Patron B A, Miramontes-Jaramillo D and Kober V 2015 *International Conference on Computational Science and Computational Intelligence (CSCI)* 843-844

[10] Echeagaray-Patron B A and Kober V 2015 *3D face recognition based on matching of facial surfaces* **9598** 95980V-8

[11] Sochenkov I and Vokhmintsev A 2015 Visual Duplicates Image Search for a Non-cooperative Person Recognition at a Distance **129** 440-445

[12] Vokhmintsev A, Makovetskii A, Kober V, Sochenkov I and Kuznetsov V 2015 A fusion algorithm for building three-dimensional maps **9599** 959929-7

[13] Tihonkih D, Makovetskii A and Kuznetsov V 2016 *A modified iterative closest point algorithm f or shape registration* 9971 99712D-8

[14] Sochenkov I, Sochenkova A, Vokhmintsev A, Makovetskii A and Melnikov A 2016 *Effective indexing for face recognition* 9971 997124-9

[15] Picos K, Diaz-Ramirez V, Kober V, Montemayor A and Pantrigo J 2016 *Optical Engineering* **55** 55-55-11

[16] Echeagaray-Patron V K 2016 *Face recognition based on matching of local features on 3D dynamic range sequences* 9971 9971-6

[17] Echeagaray-Patrón B A, Kober V I, Karnaukhov V N and Kuznetsov V V 2017 *Journal of Communications Technology and Electronics* **62** 648-652

[18] Zou R, Ge X and Wang G 2016 *IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)* 375-378

[19] Jun C, Kang J, Yeon S, Choi H, Chung T Y and Doh N L 2017 *Intelligent Automation & Soft Computing* **23** 207-218

[20] Guo K, Xu F, Yu T, Liu X, Dai Q and Liu Y 2017 *ACM Trans. Graph.* **36** 32:1-32:13

[21] Cheng S C, Su J Y, Chen J M and Hsieh J W 2017 *Model-Based 3D Scene Reconstruction Using a Moving RGB-D Camera* (Cham: Springer International Publishing) 214-225

[22] Holz D and Behnke S 2015 *European Conference on Mobile Robots (ECMR)* 1-8

[23] Gonzalez-Fraga J A, Kober V, Diaz-Ramirez V H, Gutierrez E and Alvarez-Xochihua O 2017 *Proc. SPIE* 10396 10396-7

[24] Aguilar-Gonzalez P M and Kober V 2011 *Optical Engineering* **50** 50-59

[25] Aguilar-Gonzalez P M and Kober V 2012 *Optics Communications* **285** 574-583

[26] Ruchay A and Kober V 2016 *Clustered impulse noise removal from color images with spatially connected rank filtering* **9971** 99712Y-99712Y-10

[27] Ruchay A 2017 *Removal of impulse noise clusters from color images with local order statistics* **10396** 10396-10

[28] Ruchay A 2017 *Impulsive noise removal f rom color video with morphological filtering* **10396** 10396-9

[29] Ruchay A and Kober V 2018 *Analysis of Images, Social Networks and Texts* (Cham: Springer International Publishing) 280+291

[30] Takimoto R Y, de Sales Guerra Tsuzuki M, Vogelaar R, de Castro Martins T, Sato A K, Iwao Y, Gotoh T and Kagei S 2016 *Mechatronics* **35** 11+22

[31] Kim D, Choi J, Leksut J T and Medioni G 2016 *IEEE International Conference on Image Processing* 3011+3015

[32] Song X, Zheng J, Zhong F and Qin X 2017 *Multimedia Tools and Applications* 1+07

[33] Kim S H, Hwang Y H, Hong H K and Choi M H 2004 *MICAI: Advances in Artificial Intelligence* 642-651

[34] Xin W and Pu J 2010 *International Conference on Computational and Information Sciences* 565-568

[35] Tran M D, Kang H J and Ro Y S 2010 *Improved ICP control algorithm in robot surgery application* 66-69

[36] Chen J and Belaton B 2014 *Machine Learning and Cybernetics* 55-263

[37] Zhang C, Du S, Xue J and Qi X 2014 *Foundations and Practical Applications of Cognitive Systems and Information Processing* 523-530

[38] Guo F, He Y and Guan L 2015 *An improved ICP registration algorithm with a weight-bootstrap scheme*

[39] Ye Q, Yao Y, Gui P and Lin Y 2016 *12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* 2109-2114

[40] Li S, Wang J, Liang Z and Su L 2016 *Tree point clouds registration using an improved ICP algorithm based on kd-tree* 4545-4548

[41] Yang H, Jiang J, Zhao G and Zhao J 2015 *World Journal of Engineering and Technology* **03** 302-308

[42] Khoshelham K and Elberink S O 2012 *Sensors* **12** 1437-1454

Acknowledgments

This work was supported by the Russian Science Foundation, grant no. 17-76-20045.