Assembly Manipulation Understanding Based on 3D Object Pose Estimation and Human Motion Estimation

Kimitoshi Yamazaki[*1], Taichi Higashide[*2], Daisuke Tanaka[*1], Kotaro Nagahama[*1]

***Abstract*— In this paper, we present a method of assembly manipulation understanding by demonstration. Human demonstrator performs assembly manipulation in front of a 3D range camera system, then the system recognizes each assembly manipulation from two aspects: 3D object poses and hand motion. The pose of assembled parts are calculated by means of LINEMOD that one of template matching methods. Meanwhile, hand motion feature is extracted from hand joints motion. They are extracted by means of OpenPose that is one of human pose estimation method. We combine both features to obtain more robust features and to calculate probability of each action classes. Finally, we confirmed that the proposed method enables to recognize assembly manipulation as highest probability action class.**

I. INTRODUCTION

Industrial robots that are good at repeating work have supported the mass production era. Meanwhile, in recent years, demands for the production of many types of small quantities are increasing. Among them, the assembly process is one of the most problematic process in automation. In assembly, it is necessary to combine various parts by various manipulations. Therefore, the construction of a system necessary for automating one assembly process takes time and labor to implement. It is also difficult to divert the system to other work.

The authors are interested in constructing an automated system that can respond quickly when addition of new parts or change of assembling procedure occurs. There are many technologies necessary for constructing such a system. Among them, we focus on ways to reduce the labor of teaching process. The purpose of this study is to construct an image processing system that can understand assembly work. In this study we target a method of obtaining assembly manipulation skill from observing how a person is operating. Here, important things are how to extract information of the target work and how to transform it into a form suitable for doing assembly work by autonomous robots.

Conventionally, human behavior understanding has been actively studied. In the field of computer vision, many methods using time-series image streams taken by fixed cameras have been proposed [1], [2]. As a basic policy, several action primitives are pre-defined. Therefore in many cases, the purpose is to recognize current behaviors of performers by classifying the behavior into a known action class. Our study also adopts the same approach as conventional methods in this respect. That is, basic actions such as screwing and inserting are known, and the behavior of a performer is distinguished into the action classes. However, only such approach, it is difficult to reproduce work by autonomous robots. It is necessary to clearly describe the movement of assembly parts accompanying the performer's behavior. In addition, it is necessary to grasp the positional relation of parts after assembly. We aim to build a method satisfying mentioned the above.

The contributions of this paper are as follows:

- We propose and verify a method of understanding of an assembly process based on the observation of human demonstration. No artificial marker is used.
- By applying 3D pose estimation to assembly parts, it makes possible to easily obtain data format that is useful for manipulation automation.
- We propose a feature representation based on the visibility of assembly parts. It improves the accuracy of human action classification.

The structure of this paper is as follows. In Section II, we introduce related studies. Section III describes our problem setting and approach. Section IV and Section V describe state recognition of assembly parts based on object pose estimation and human motion recognition, respectively. Section VI explains the results of verification experiments and summarizes them in Section VII.

II. RELATED WORK

Studies that make robot understand work based on human demonstration has been proceeded in the field of intelligent robotics. In the work [3] and [4], a system was proposed in which the robot performs the same work as a human demonstrator. It recognized the simple behavior of the demonstrator using cameras. Zollner et al. [5] used *programming by demonstration* framework, and made the robot to perform daily work that occupies the lid of the bottle. In that study, the human movement was estimated by attaching a data glove with a contact sensor.

Studies that robots understand and reproduce the movement of human beings has continued to develop [7], [8]. Recent studies include assembly of building blocks with humanoid robots [6]. In their study, the robot obtained object orientation using AR markers and point cloud matching, understood the work contents, and preformed the same assembling work. The actions assumed in the work were simple such as lifting and placing, and the problem setting was simplified such that artificial markers were attached so that the object could be easily recognized.

[*1] Faculty of Engineering, Shinshu University, 4-17-1 Wakasato, Nagano, Nagano, 380-8553, Japan. [*2] The University of Tokyo, 7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan. {kyamazaki}@shinshu-u.ac.jp

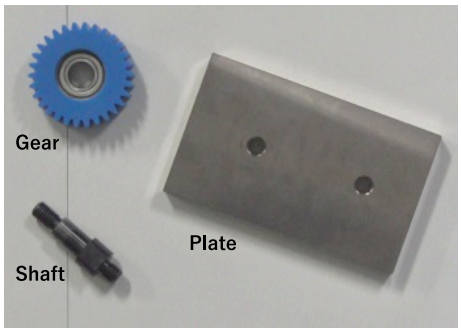

Fig. 1. Parts for assembly operation: Gear, shaft and base plate.

Fig. 2. A snapshot of assembly work.

Besides assembly work, there are studies on understanding work such as simple cooking and tidying etc. [9], [10]. Many of them use time-series image streams and aim to classify the movement of a performer into predefined action classes. Basically, it is not considered the manipulation automation using the result. On the other hand, there are also studies which intend to use such classification result to generate the behavior of robots [11]. In their study, action classification was done using decision trees. The result was used to reproduce the motion by a dual-arm robot, but there were no deep consideration to recognize the state of operated objects.

III. Problem Setting and Approach

A. Problem setting

Fig. 2 shows a demonstration of an assembly work. In this study, as an element of manufacturing processes, we target a work for assembling a shaft and a gear to a base plate. The parts [18] to be used are shown in Fig. 1. These are parts which are used for some evaluation: e.g. Robotic Grasping and Manipulation Competition [17].

In this problem setting, the action to be performed for gear unit assembly is as follows. First, screw the shaft to the base plate. Second, insert the gear into the shaft. Therefore, we define three action classes, which are *Screw*, *Insert* and *Other*, and classify performer action into these three. The initial placement of parts shall be changed variously. Also assume that the shape of the parts is known, and its 3D shape model is given in advance.

In order to understand the work procedure performed by humans and convert the result to the motion of robots, it is necessary to detect the assembly parts before the manipulation and to recognize its posture. Also, it is necessary to estimate what kind of manipulation is necessary during work from human movement. Therefore, we combine the result of estimating the pose of the assembly parts and the result of estimating the action class from the motion of the human demonstrator. That is, we make it possible to record explicitly what kind of manipulation is applied to the parts and where the parts are installed.

The vision system constructed in this study equips a three-dimensional range image sensor. That is, we use color images and depth images to make recognition necessary for understanding assembly work. From the next subsection, we will explain each of the above two methods: estimation of posture of assembly parts and action classification of hand motion.

B. 3D object pose estimation

In order to clarify which assembly parts were assembled to where, 3D pose estimation is performed. As set in Section III-A, the 3D shape model is given in advance. The state of the assembly work can be measured as color images and depth images. We estimate the 3D pose parameters of the assemble parts by using pose recognition method based on template matching. In our aim, it is difficult to use texture-dependent image feature points (e.g. SIFT [19]) because assembly parts are basically homochrome object and some of them have glossy parts. Therefore, it is necessary to adopt means that can be applied to textureless objects.

In the assembly work, the visibility of parts varies depending on whether the operator holds the part in his/her hand or not. For example, since assembly parts prior to work are in a stationary state, they can often easily be recognized. On the other hand, during manipulation, there is a high possibility that it will be hidden by the hands of the demonstrator and disappear. Because these differences in status can be used as a clue to robust recognition, we describe the state of manipulation taking this into consideration.

C. Hand action classification

In understanding the assembly work, how the demonstrator moves is important information. In this study, we take an approach to discriminate the type of action from the time series change of the posture of the hand. For this purpose, it is necessary to estimate the posture of the hand photographed in each image. However, the hand has many joints. Besides, many self-occlusions occur during manipulation. Due to the development of deep learning technology in recent years, pose estimation of articulated objects from image is dramatically advanced, but it is yet difficult to estimate the hand posture perfectly in assembly work mentioned the above.

In this study, we aim at action classification robust against the errors of estimation propriety and estimated joint angles. Therefore, the posture estimation of the hand is performed on each image, and the shape of the hand is used for generating a feature vector. By using at the time-series change of this feature vector, the classification is performed.

D. Robustness improvement by complementary estimation

In assembly work understanding using actual images, there might be cases that are difficult to deal with only by methods mentioned in III-B and III-C, respectively. For example, although the hand moves almost in the same way as the insert manipulation, in actual that is a motion of the hand reaching out to take parts. On the other hand, when identifying the action class based on the appearance state of the assemble parts, recognition might be erroneous depending on unforeseen occlusion. Therefore, by combining each estimation method, we compensate their disadvantages and perform more robust action classification. we formulate according to probabilistic theory and obtain final classification result based on the reliability of each feature.

The following equation is used:

$$p(c|\mathbf{x},\mathbf{y}) = \frac{p(\mathbf{x},\mathbf{y},c)}{p(\mathbf{x},\mathbf{y})} = \frac{p(c)p(\mathbf{x}|c)p(\mathbf{y}|c)}{\Sigma_c p(c)p(\mathbf{x}|c)p(\mathbf{y}|c)}, \quad (1)$$

where c is the class of action. $p(c)$ is the probability of the class c. We assume a uniform distribution in this study. $\mathbf{x}$ is the feature vector of object pose estimation and $\mathbf{y}$ is the feature vector of hand action. We assume that the probability density from these variables follows a normal distribution. Also, $p(\mathbf{x}|c)$ and $p(\mathbf{y}|c)$ are the conditional probability of classification based on the pose estimation of the assemble parts and the hand posture estimation. The method of calculating each probability value will be explained in detail from the next section.

IV. Object pose estimation

A. LINEMOD[12]

For 3D pose estimation, we adopt LINEMOD. LINEMOD is one of template matching methods. Based on image contour and surface normal, a template image is collated in the input image. The characteristic of LINEMOD is that it can perform collation processing faster and more robust than conventional template matching.

Conventionally, in the template matching method using contour information, there is a method to generate an edge image from a grayscale image and calculate similarity with template images [13]. Another method uses brightness gradient [14]. However, in these methods, the performance might be degraded due to illumination fluctuation and so on. Also, if the number of templates increases, processing time also increases. LINEMOD improves the robustness of matching by devising the calculation method of feature points and performing collation processing invariant to minute variation. Moreover, by improving the efficiency of memory access, even if there are many templates, it can suppress the increase of calculation.

Similarity measure of LINEMOD is calculated as follows:

$$\epsilon(I,O,c) = \Sigma_{r\in R(c+r)}[\max_{t\in R(c+r)} |cos\{ori(O,r) - ori(I,t)\}|], \quad (2)$$

where I is an input image, O is a template image, c is shift amount, r is position coordinate. ori is an edge gradient image and is calculated as follows. Find the gradient orientation map $I_g(x)$ at position x by Eq.(3) on the input color image. Then, the position with the greatest edge intensity is output. That is,

$$I_g(x) = ori(\hat{C}), \quad (3)$$

where

$$\hat{C}(x) = arg \max_{\in R,G,B} \|\frac{\partial C}{\partial x}\|. \quad (4)$$

However, it is a heavy load to calculate Eq. (2) every time. Therefore, in LINEMOD, the intensity value is quantized. Maximum value in the neighboring region is calculated for template image T in advance, then a response map S is prepared.

$$\epsilon(I,O,c) = \Sigma_{r\in P} S_{ori(O,r)}(c+r) \quad (5)$$

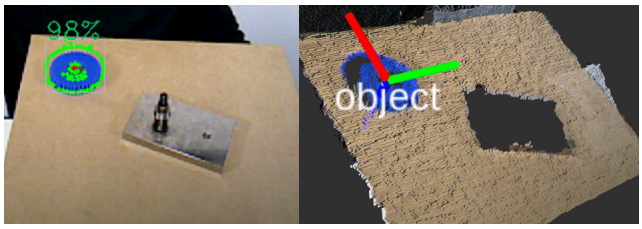

Fig. 3. An example of pose estimation. Left: A result of linemod, right: 3D pose estimation combining the linemod result and depth image.

Figure 3 is an example of posture recognition using LINEMOD. First, it detects the gear on the image, then calculated three dimensional pose by combining depth value. There is an assumption that the assemble parts are placed on a horizontal table.

B. Template image generation

LINEMOD is a method of detecting the region that best matches the template image from the input image. In other words, target objects are photographed in advance from various viewpoints, and a template image with the highest matching similarity are selected from them. Then, the posture of the object when photographing the template is output as the estimation result.

In this study, we assume that the posture of assemble parts changes three-dimensionally. Therefore, it is necessary to prepare templates in regard to postures with 6 degrees of freedom. To generate a template, there is a method using a turntable, or a method of estimating the attitude of a camera with a checker board etc. However, they require special machines and expert knowledge. Therefore, we adopt a method of creating templates using simulator.

The procedure is as follows. First, prepare the 3D-CAD model of a target object and the camera model, and then load them by a simulator. Next, define a spherical surface centered on the object, and define points at substantially equal intervals on the spherical surface. These are viewpoint of the camera. Then virtually photograph the object from each viewpoint and obtain an image of the object. Fig. 4 shows the viewpoints on the spherical surface, here we use the method of recursively dividing regular icosahedrons [15].

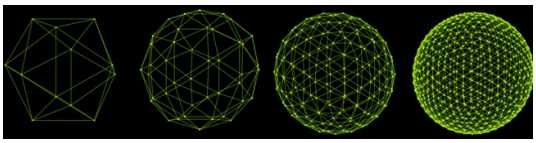

Fig. 4. Sphere approximation by icosahedron

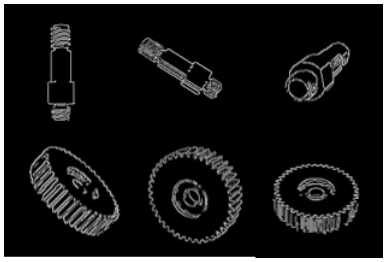

Fig. 5. Examples of the generated templates

TABLE I
OBEJECT RECOGNITION STATE

	Screw	Insert	Other
Gear	1	0	1
Shaft	0	0	1
Base plate	0	0	1

In this method, the number of templates can be adjusted by changing the number of the divisions.

Figure 5 shows examples of a template image that was automatically generated in the simulator. Target objects in this study are parts used for assembly. Such parts have less texture on the surface. In addition, its appearance tends to change depending on lighting conditions because the surface is glossy. Therefore, we use a color edge image as template image. This enables for matching process to use information on geometric shapes and patterns existing inside the object. Also, it is possible to suppress the influence of the apparent difference between the real environment and the virtual environment.

C. Motion estimation based on object pose estimation

There are two kinds of roles for parts: one is parts to be assembled and the other is basing parts. For example, when assembling a shaft with a base plate, the state of the occlusion of these assembly parts changes before and after the assembly is started. Therefore, this appearance state enables one clue of motion estimation. A case where one assembly part is visible is defined as 1, and a case where it is not seen is defined as 0. Then a feature vector is defined as a list in which the appearance of parts in each operation is listed. In this study, there are three types of parts and three kinds of states. Based on whether or not the recognition result based on LINEMOD is possible, a correct feature vector is determined as shown in TABLE I. That is, if the visibility pattern obtained in a certain image is close to one of the three types, it is judged that the action is the current action.

The method of calculating the feature vector is as follows. The visibility of each object is quantified using the similarity ϵ output by LINEMOD. To express the degree of recognition stochastically, the similarity ϵ is input to the sigmoid function as follows:

$$x_{part} = \frac{1}{1 - exp(-a(\epsilon - b))}, \tag{6}$$

where *part* corresponds to the one of *Screw*, *Insert* and *Other*. By means of appropriately setting of the gain a and the threshold b in Eq. (5), the output is divided into a value close to 1 or close to 0.

Finally, this result is used to specify an action class. The likelihood in each class is calculated by the following formula. Then the class showing the largest magnitude is selected.

$$p(x_{part}|c) = \max_{\mu_{\mathbf{c}}, \mathbf{\Sigma_c}} N(x_{part}|\mu_{\mathbf{c}}, \mathbf{\Sigma_c}) \tag{7}$$

The average value $\mu_{\mathbf{c}}$ and covariance matrix Σ_c are manually given in advance.

V. ACTION CLASSIFICATION FROM HUMAN MOTION

A. Hand motion extraction using OpenPose[16]

In estimating the movement of a demonstrator, the motion of the hand is important information. In this study, OpenPose which is a well-known method for human pose estimation is used. OpenPose uses a color image as an input and outputs the position of the joints and limbs in two-dimensional coordinates. OpenPose also supports hand posture estimation. It outputs 21 joints of a hand by two-dimensional coordinates and also estimate reliability in each. We use only the hand posture estimation and combine the result with the distance information obtained from a depth image, thereby calculating the three-dimensional position of the joint angle of the hand. By using the time series change of them, we extract the characteristics of the hand motion.

In motion recognition of parts assembling, it is necessary to quantify which parts move with respect to which parts. Therefore, we take an approach to first estimate the posture of the base parts using LINEMOD, and then to transform the coordinate of the hand joints with reference to the coordinate system defined on the base. Thus, the motion of the joints is represented on the coordinate system of the part. The reason is that almost assembling work is done by assembling other parts to the base. This also enables to make the hand motion invariant to the direction of assembly parts.

B. Characterization of hand motion

In this study, insertion and screwing should be recognized. Motion trajectory of them is simple or periodic. Therefore, we generate a feature that makes it easy to extract such motions. Since the transition of the joint position of the finger can be extracted by the process explained in Section V-A, the feature description is generated with the following policy.

1) As shown on the left panel of the Fig. 6, joint positions for a certain period are collected and chunks of one action are made as shown in a red circle. Then principal component analysis is applied to the coordinates to

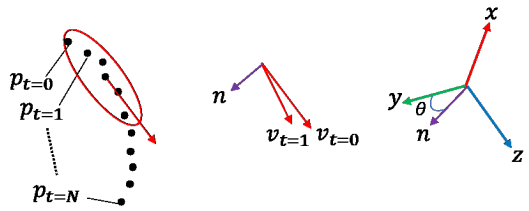

Fig. 6. Method for hand motion feature extraction. **Left:**Principal component vector v(t) is calculated from motion chunk, **Middle:**Normal composed of each v(t) **Right:**The angle of normal and orientation of reference coordinate system

obtain the principal component vector as shown in red arrow.

2) As shown in the center panel of Fig. 6, two principal component vectors calculated at different times are selected, and the normal vector is obtained as the outer product of them.
3) The angle θ is calculated between an axis in the reference coordinate system (y axis in Fig. 6) and the normal vector calculated at item 2.

The above procedure is performed several tens of times while shifting by a predetermined period. After that, a frequency histogram in regard to θ is calculated.

The action classification is done by comparing histograms. Where the input histogram is $\mathbf{y}$, the likelihood of the action class c is obtained by the following equation.

$$p(\mathbf{y}|c) = N(\mathbf{y}|\mu_{\mathbf{c}}, \mathbf{\Sigma_c}) \tag{8}$$

The average value $\mu_{\mathbf{c}}$ and the covariance $\mathbf{\Sigma_c}$ of each histogram are manually given by preliminary experiments.

VI. Experiments

A. Settings

Assembly parts were placed on the table and a performer sat down before that. A three-dimensional range image sensor (realsense D435, manufactured by Intel Corporation) was installed at a position where both the parts and the performer can be observed. A color image and a depth image of 1280 $\times$ 720 [pixel] size were acquired by the sensor. Evaluation was carried out by comparing the classification results with the result of manually annotated data.

Three kinds of parts shown in Fig. 1 were used. The base plate had asymmetrical fitting holes, so it was necessary to accurately estimate the orientation of the place. In this experiment, the manipulation procedure was to screw the shaft into one of the hole, and then to insert the gear into the shaft. Using the proposed method, the posture of each part and the movement of the hand are estimated respectively, and the action was classified.

B. Experimental results

Figure 7 is an example of a histogram calculated from one of the joint of a hand. The left side shows the one when inserting and the right side shows it when screwing. It turns out that there is a clear difference between the two. Such

TABLE II
Experimental Condition

	frame	motion
1	100-115	other
2	115-235	screw with fake object recognition state
3	235-260	other with fake hand motion
4	260-290	other
5	290-310	other with fake hand motion
6	310-330	other
7	330-355	insert
8	355-400	other

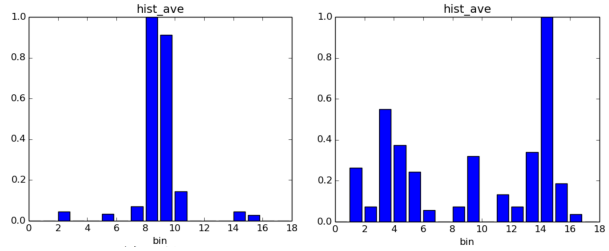

Fig. 7. The histograms of hand motion feature. **Left:**insert motion, **Right:**screw motion

data were collected beforehand for each action class, and the average histogram was used as a reference vector. In addition, since the histogram of *other* action is diverse, a feature vector with all bin values set to 0.5 was generated and used as a reference vector.

For evaluation, multiple patterns of movies recording of the state of assembly work were prepared. Table II is the result of manually annotating one of them. In addition, Fig. 8 shows the estimation results by three methods, respectively. The vertical axis shows the likelihood value, with red for *screw*, blue for *insert*, yellow for *other* actions. The graph shows the estimation results. From the top, by the combination of hand motion and parts state, by only hand motion, by only parts state.

As shown in Table II, the action class at the 1st section is *other*, and the 2nd section is *screw*. This is because the gear was disappeared by the hand of demonstrator when screwing the shaft. In such a situation, the estimation result by object recognition becomes *insert*. On the other hand, the estimation result using hand motion (center of Fig. 8) could correctly estimate the screwing action. In the 3rd and 5th section, there was a hand motion similar to the *insert* manipulation. In that case, the estimation using hand motion had a high likelihood of being *insert*. However, the classification result by object recognition could correctly estimate as *other*. At 7th section, It was difficult to estimate as *insert* by only hand motion, but by combining with object recognition, it was possible to increase the likelihood of *insert*.

As described the above, according to the proposed method, the type of assembly manipulation and its appearance timing can be grasped. In addition, it was possible to obtain the 3D posture of assemble parts before and after assembly. Each posture is expressed in a coordinate system defined on the basing parts. Using these information, end-effector poses and

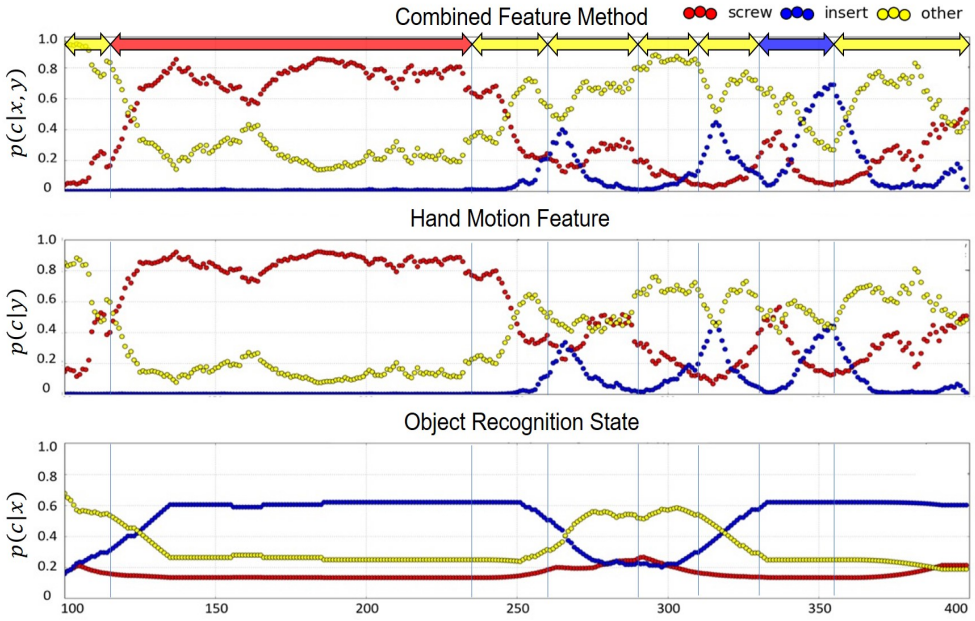

Fig. 8. Time series data of motion probability

action types can be explicitly given to the robot. Of course, even if the placement of parts is changed, it was able to deal with. From the above, it can be said that the information necessary for working with the robot could be extracted.

VII. Conclusions

In this paper, we proposed a method of assembly manipulation understanding by demonstration. Human demonstrator performs assembly operation in front of a 3-D range camera system, then the system recognizes each assembly manipulation from two aspects: 3D object poses and hand motion.

We used LINEMOD to estimate the 3D posture of assembly parts. It enables to clarify the 3D coordinates of the parts. This is important for motion planning of robots. Meanwhile, we also proposed a method of action classification. We designed hand motion feature that is calculated from time-series results of OpenPose. Then, we combined both results and obtained assembly motion class by means of probabilistic approach. Experimental results showed the effectiveness of the proposed method.

Future work includes automatic parameter estimation for Eq.(1) and experiments with more complex assembly task.

References

[1] M. Raptis and L. Sigal: "Poselet Key-Framing: A Model for Human Activity Recognition," IEEE Conference on Computer Vision and Pattern Recognition, pp. 2650-2657, 2013.

[2] E. Mavroudi, L. Tao and R. Vidal, "Deep Moving Poselets for Video Based Action Recognition," 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Santa Rosa, CA, 2017, pp. 111-120.

[3] Y. Kuniyoshi, M. Inaba, H. Inoue: "Learning by watching: Extracting reusable task knowledge from visual observation of human performance," IEEE Transactions on Robotics and Automation, Vol.10, No.6, pp. 799-822, 1994.

[4] K Ikeuchi, T Suehiro: "Toward an assembly plan from observation: Task recognition with polyhedral objects," IEEE Transactions on Robotics and Automation, Vol.10, No.3, pp.368-385, 1991.

[5] R. Zollner, T. Asfour and R. Dillmann, "Programming by demonstration: dual-arm manipulation tasks for humanoid robots," in Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems, Vol. 1, pp. 479-484, 2004.

[6] WAN, Weiwei, et al. Teaching robots to do object assembly using multi-modal 3d vision. Neurocomputing, 2017, 259: 85-93.

[7] R. Slama, H. Wannous, M. Daoudi, A. Srivastava: "Accurate 3D action recognition using learning on the grassmann manifold," Pattern Recogn. Vol. 48, No.2, pp.556567, 2014.

[8] Jiang, Qiannan et al.:"Human Motion Segmentation and Recognition Using Machine Vision for Mechanical Assembly Operation," SpringerPlus, Vol.5, No.1: 1629, 2016.

[9] S. Bianco et al.: "Cooking Action Recognition with iVAT: An Interactive Video Annotation Tool," in Proc. of Int'l Conf. on Image Analysis and Processing, pp. 631-641. 10.1007/978-3-642-41184-7_64, 2013

[10] J. Monteiro, R. Granada, R. C. Barros and F. Meneguzzi: "Deep neural networks for kitchen activity recognition," International Joint Conference on Neural Networks, pp. 2048-2055, 2017.

[11] RAMIREZ-AMARO, Karinne; BEETZ, Michael; CHENG, Gordon. Automatic segmentation and recognition of human activities from observation based on semantic reasoning. In: Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014. p. 5043-5048.

[12] HINTERSTOISSER, Stefan, et al. Gradient response maps for real-time detection of textureless objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34.5: 876-888.

[13] GAVRILA, Dariu M.; PHILOMIN, Vasanth. Real-time object detection for" smart" vehicles. In: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on. IEEE, 1999. p. 87-93.

[14] STEGER, Carsten. Occlusion, clutter, and illumination invariant object recognition. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 2002, 34.3/A: 345-350.

[15] HINTERSTOISSER, Stefan, et al. Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes. In: Asian conference on computer vision. Springer, Berlin, Heidelberg, 2012. p. 548-562.

[16] https://github.com/CMU-Perceptual-Computing-Lab/openpose

[17] http://www.rhgm.org/activities/competition_iros2017/

[18] https://www.nist.gov/el/intelligent-systems-division-73500/robotic-grasping-and-manipulation-competition-manufacturing

[19] D. G. Lowe: "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, Vol. 60, No.2, pp. 91-110, 2004.