

3D Object Pose Estimation Using Chamfer Matching and Flexible CAD File Base

Dewi Mutiara Sari^a, Vina Wahyuni Eka Putranti^b

^aDepartment of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Jl. Raya ITS, Surabaya, 60111, Indonesia
E-mail: dewi.mutiara@pens.ac.id

^bMalaysia-Japan International Institute of Technology – University Teknologi Malaysia, Malaysia
E-mail: vinawep@gmail.com

Abstract—Estimating the object pose is an interesting topic in the industrial robotic vision field. By having an accurate result for detecting object pose, it means the system performs the task as the target in the *bin-picking* technique. The methods which are developed are varies widely. But the challenge for this paper is estimating a 3D object using mono camera accurately. The object which is used in this paper has the symmetric rotational shape, in this case is the sprayer. In this paper, the camera uses a tool from the Blender Software, such that the ground truth is measurable and it will be the reference for calculating the error. The applied algorithms of this paper are Border Line Extraction Algorithm utilized in the template generation step as the reference template, Directional Chamfer Matching for detecting the coarse pose, and Lavenberg-Marquardt Method to optimize the object pose result. The result achieves the average error of the coarse pose for x and y position (translation pose) are 2.05 mm and 0.71 mm. Meanwhile for the optimized pose, the average error for x and y position (translation pose) are 1.82 mm and 0.24 mm. Regarding the rotational pose, the average error of the rotational coarse pose with respect to x and z axis are 0.01 degree and 0.45 degree. Whereas the average error of the rotational optimized pose with respect to x and z axis are 2.88 degree and 0.82 degree.

Keywords— 3D object pose; borderline extraction; directional chamfer matching; optimized pose; robotic vision; industrial robotic.

I. INTRODUCTION

In the industrial robotic field, one of the robot's task is moving the targeted object from the initial place to the destination place. Various methods are developed to satisfy the robot's task for the best result. Another thing, one of the factors which have an essential sub-task of the whole system, is the object detection part. This research presents the object detection topic. Lysenko, et al., analyzes the cost function value by elaborating on the combination of Partial Directed Hausdorff (PDH) and Chamfer Matching methods.

Meanwhile, the reference template used is the silhouette edges, and the surface in 3D and the camera used is stereo camera [1]. Gualtieri et al. developed point cloud data that are produced by RGBD sensor. The success rate achieved 93% [2]. Pas et al. detect the cluttered object pose by developing the point cloud data. But the result shows that the point cloud registration step is the background of the robot couldn't grasp the object (5 from 22 objects failed to be perceived). Besides, the object segmentation is not applied, so the robot could not accurately detect the pose of the specific object [3].

Meanwhile, Hossain *et al.* conducted research on the Deep Believes Neural Network (DBNN) to estimate the

object's location [4]. A device called Kinect was deployed as a motion sensor. However, it was shown that the average error is more than 4 mm. Czajewski *et al.* also deployed the Microsoft Kinect sensor to perform the point cloud matching method [5]. Viewpoint Feature Histogram (VFH) and Camera's Roll Histogram (CRH) descriptors matching continued by Iterative Closest Point (ICP) and Hypotheses Verification (HV) algorithms were used in this approach. Yet, the camera height concerning the object suggested less than 1 m. He *et al.* proposed *another* method using template matching–clustering algorithm,–scoring function related to the template, non-maximum suppression, and combination of point cloud processing. The average F1-Score, which indicates as a weighted average of precision and recall, gained 0.562.

The score was still less than Latent-Class Hough Forest (LCHF) conducted by Tejani *et al.* [6], which achieved 0.633. The drawback of the method was only limited to a simple design. Castro *et al.* used RGBD camera to estimate 6D pose of the object under the light reflection as well as in the harsh and unstructured environment [7]. In this project, two times, estimation should be applied to reduce the error. Otherwise, failure will increase. Abbeloos et al. conduct

another approach to estimate the multiple objects using RGBD camera. The information of the salient geometry's object was maintained to extract critical points before matched them based on the descriptors. However, an additional method was required to improve descriptors and descriptor matching because of the loss of some features [8]. To optimize the object detection, Izatt *et al.* used mixed-integer of point cloud data. But the scaling factor of point clouds is needed in the process—the point clouds given by the experimental data [9]. Sakcak, *et al.* applied Directional Chamfer Matching (DCM) method using the mono industrial camera. But the system could not be applied for the 3D object case [10].

Meanwhile, Liu *et al.* applied a method called Fast Directional Chamfer Matching (FDCM) to estimate the object pose. But the system can be applied less than 11 degrees for the maximum rotation [11]. Apart from the matching step, Zeng *et al.* used the Hidden Line Removal approach for 2D cartoon images. The concept is developed by the updated depth and the visibility test [12]. For the visibility test, Nisha *et al.* used a z-buffer method to detect the visible surface [13]. Li *et al.*, for detecting occlusion, using z-buffer operation with the time operation needed is $O(n^m)$ in which higher than the Sparse Grid method with the time operation $O(n)$ [14]. Eka Samsul Ma'arif *et al.* used the integral projection function to determine the most outer edge. The system applied both for the vertical and horizontal integral projection function [15]. Nevertheless, the system is not robust concerning the light condition.

Overall, the paper presents a 3D object detection method with the optimization method to improve accuracy. The inputs of the system are the combination of both the 3D CAD Model as a reference frame and the query image (captured image). In this case, the query image is created by Blender software, in which the intrinsic camera matrix and the position of the camera with respect to the object can be determined. Then the system is continued by the matching process. After that, the optimization is conducted in this system. The detail process is illustrated in Fig 1.

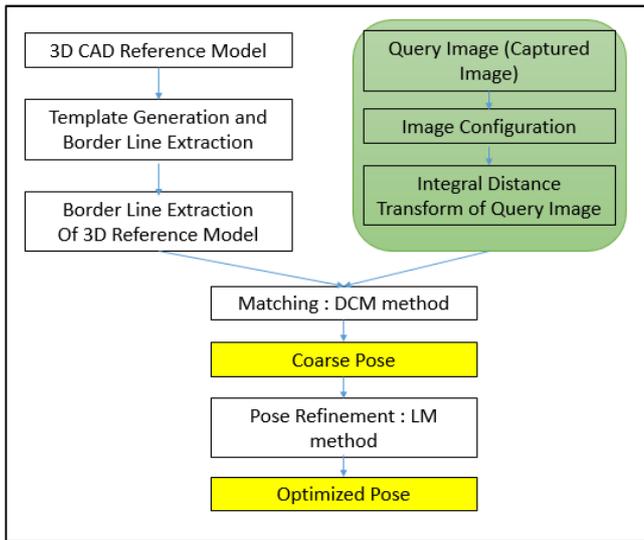


Fig. 1 The system pipeline of estimating 3D object poses

II. MATERIALS AND METHOD

This section discussed three main parts, regarding the process in the template generation, the object detection by estimating the coarse pose of the object, and the optimized pose of the object.

A. Generating Reference Template

The 3D template, which is generated by the CAD file, is projected into 2D by using camera projection matrix. The camera projection matrix is the multiplication of projection and roto-transformation camera matrix. Then the system extracts the borderline of the projected template. The idea is the template can be rotated at any angle concerning any axis. Then the system extracting only the border in any pose of the template.

1) *Projection Matrix*: The input of the projection matrix is the points of the 3D template. The points of 3D template is multiplied with the camera matrix that is done by camera calibration. The camera matrix that is used is the intrinsic parameter. Then the intrinsic parameter is multiplied with the rotation matrix and the translation matrix. The rotation matrix indicates the pose of the object with respect to the camera in terms of angle during the translation matrix for the position of the object. The pose of the object with respect to the camera is defined by $q = [t_x \ t_y \ t_z \ \theta_x \ \theta_y \ \theta_z]^T$. The subsequent points of the 3D template are defined by $\tilde{U} = \{\tilde{u}_i\}_{i=1}^n$ with the n indicating the setpoint quantity.

The intrinsic matrix (Int) is achieved by using a chessboard in the MATLAB application tools. The intrinsic parameter consists of focal lengths (f_x, f_y) and center points of image (c_x, c_y). The matrix is formulated in (1)

$$Int = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Then the transformation matrix (T_m), with dimension 4 x 4, expressed in (2).

$$T_m(q) = \begin{bmatrix} Rot_m & trans_m \\ 0 & 1 \end{bmatrix} \quad (2)$$

For the rotation matrix (Rot_m) achieved by the multiplication of the Euler angle, explained in (3).

$$Rot_m = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

Meanwhile for the translation matrix ($trans_m$) is expressed in (4).

$$trans_m = [t_x \ t_y \ t_z]' \quad (4)$$

The multiplication between the transformation matrix and the intrinsic matrix is the projection matrix (Proj), which has the 3 x 4 matrix dimension. The formulation is in (5).

$$Proj = [I \ 0] * T_m(q) \quad (5)$$

So, the transformation from 3D points into 2D is achieved by multiplied all the 3D points with the Proj. The formulation is described in (6).

$$u_proj_i = Proj * \tilde{u}_i \quad (6)$$

In the end, u_proj_i has a matrix dimension 2 x n points. The process is the first and second row of the u_proj_i matrix is divided by the last row of the u_proj_i matrix. The first and second row indicates the x and y point; meanwhile, the last row is the scaling factor.

2) *Borderline Extraction:* The purpose of using a border line extraction method in this system is for having a 'flexible' template. The template will be a reference image in the matching step. The word 'flexible' here means that the template (as a reference image) can be used in the various condition of the object position that is captured by the camera.

The template of the object consists of many lines. After the template is projected into 2D, then the system calculates the maximum and minimum nodes of each template's line. The maximum and minimum nodes indicate the maximum and minimum value of the line both in vertical and horizontal directions. The node is illustrated with the circle symbol in Fig 2.

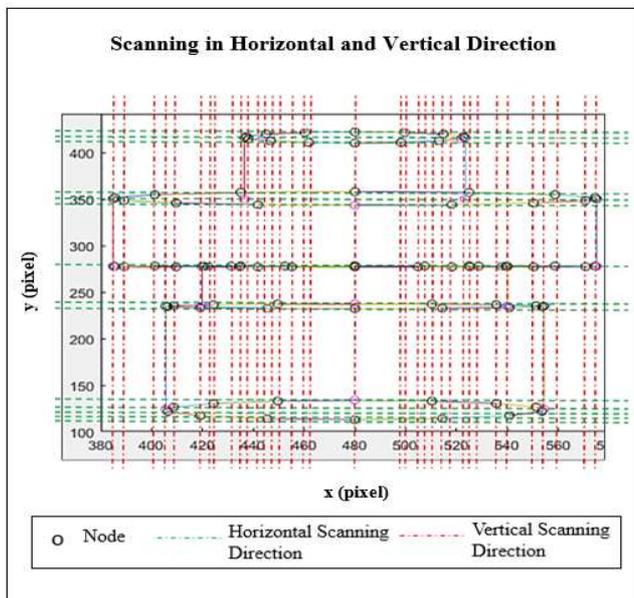


Fig. 2 The scanning process of extracting borderline

Every node has the x and y position. For every coordinate of the node, there is a scanning line that represents both in the vertical and horizontal direction. The vertical scanning line is illustrated with the red dashed line. Meanwhile for the horizontal scanning line is illustrated with the green dashed line.

The function of the scanning line is to determine the most minimum and maximum node that intersects with the scanning line. The value of the node indicating the coordinate of the node. Meanwhile, the scanning line is created by the line equation. The most minimum node in the horizontal direction is located in the most left coordinate that also intersects with the scanning line. Vice versa for the most maximum node in the horizontal direction. While for the most minimum node in the vertical direction is located in the most bottom coordinate that intersects with the scanning line—Vice versa for the most maximum node in the vertical direction.

By applying the algorithm, then the node only existing the one that represents the border of the object. Fig. 3 illustrates the process. The green star indicates the most minimum and maximum value of the scanning line that intersects with the object's node in the vertical direction. Meanwhile, the blue star indicates the most minimum and maximum amount of the scanning line that divides with the object's node in the horizontal direction—the existing nodes after the scanning line process so-called as the node candidate of borderline extraction.

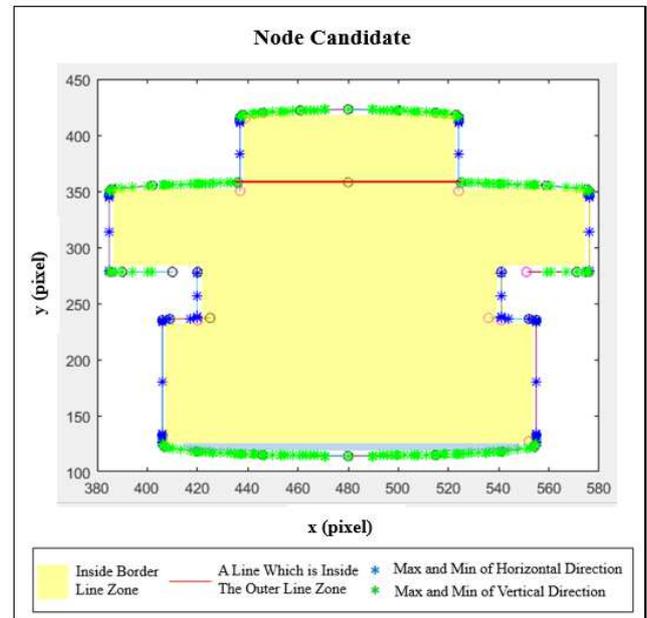


Fig. 3 Node Candidate: Minimum and maximum points of the scanning process

Then the node candidates are re-scanned. The similar step as the previous process. In the end, the result produces borderline extraction by building the line from the existing points. Fig. 4 is a picture of the borderline extraction result.

The next process is multiplying the following projection matrix. The goal is converting the 2D of borderline extraction results in 3D. The conversion is needed because it is related to the refinement step that will be explained in the refinement process. The formulation of the back projection is formulated in (7), with w is scaling value.

$$\tilde{u}_i = w[I * Rot_m]^{-1} * u_proj_i - Rot_m^{-1}trans_m \quad (7)$$

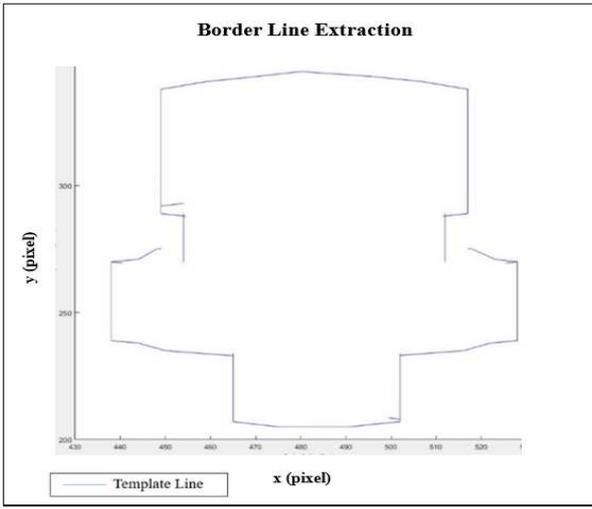


Fig. 4 Borderline extraction result

B. Coarse Pose of Object Detection

The methods which are used to estimate the coarse pose of the object are Line Segment Detector (LSD) and Directional Chamfer Matching (DCM). The inputs are query image (captured image) and the borderline extraction template. The lines are the main component of the method. Besides, the object's part, which has a high possibility of detecting the line is the border. Meanwhile, the output is the cost, which represents the distance between the template and the object captured in the query image.

The template matching method, which represents the case is Directional Chamfer Matching. The main goal is finding the best alignment parameter between the borderline of the template and the borderline of the object in the query image.

In the query image, the line with the direction is detected by the LSD. The LSD is applied in the grayscale with the task is detecting locally straight contour [11].

Defining $Q = \{u_{qj}\}_{j=1}^{|u_{q}|}$ as the points for the template (with the $|u_{q}|$ is the total number of points in the template) and $U_{map} = \{u_{proj_i}\}_{i=1}^u$ as the points of lines in the query image (with u is the total number of points in the query image).

The Distance Transform (DTrans) is finding the most minimum distance (nearest) point between pixel p concerning the pixel in the query image u_{qj} ,

$$DTrans(p) = \min_{u_{qj} \in Q} \|p - u_{qj}\| \quad (8)$$

Meanwhile, to determine the distance value between two points can be stated using chamfer distance. Then for the case of multiple points, the distance can be expressed by finding the average distance value. So, the average distance of the nearest points between the template with respect and the query image can be expressed as,

$$dt_{CM}(U_{map}, Q) = \frac{1}{n} \sum_{u_{proj_i} \in U} \min_{u_{qj} \in Q} \|u_{proj_i} - u_{qj}\| \quad (9)$$

However, the direction of the line detected contributes to the average distance value. So, by considering the direction, the value of Directional Chamfer Matching (DCM) formulated in (10). With $\varphi(u_{qj})$ is the direction for the

query image, $\varphi(u_{qj})$ is the direction for the template, and α is the scaling factor for the direction term.

$$dt_{CM}(U_{map}, Q) = \frac{1}{n} \sum_{u_{proj_i} \in U_{map}} \min_{u_{qj} \in Q} \|u_{proj_i} - u_{qj}\| + \dots \dots + \alpha \|\varphi(u_{proj_i}) - \varphi(u_{qj})\| \quad (10)$$

To calculate direction term is in linear time, the 3D distance transform ($DT3_{V_q}$) is considered. In which the first and second dimensions indicate the image plane's location. Meanwhile, the line's direction defining discretely is in the third dimension. The formulation of 3D distance transform is formulated in (11) for every pixel,

$$DT3_{V_q}(p, \varphi(p)) = \min_{\hat{\varphi}_i \in \Phi} \min_{u_{qj} \in Q} \|\hat{\varphi}_i - u_{qj}\| + \alpha \|\hat{\varphi}(p) - \hat{\varphi}_i\| \quad (11)$$

With $\hat{\varphi}_i$ indicates the cost map's orientation channel where the pixel belongs to and $\hat{\varphi}(p)$ indicates the value of quantized orientation in pixel p . And $\min_{u_{qj} \in Q} \|\hat{\varphi}_i - u_{qj}\|$ means 2D distance transform of the point u_{qj} with the direction $\hat{\varphi}_i$. The system needs forward and backward recursion in pixel x for updating the value after computing the 2D distance transform. The formulation of the backward recursion is stated in (12),

$$DT3_{V_q}(p, \hat{\varphi}_i) = \min\{DT3_{V_q}(p, \hat{\varphi}_i), DT3_{V_q}(p, \hat{\varphi}_{i+1}) + \dots + \lambda \|\hat{\varphi}_{i+1} - \hat{\varphi}_i\|_{\pi}\} \quad (12)$$

Meanwhile, the forward recursion is stated in (13),

$$DT3_{V_q}(p, \hat{\varphi}_i) = \min\{DT3_{V_q}(p, \hat{\varphi}_i), DT3_{V_q}(p, \hat{\varphi}_{i-1}) + \dots + \lambda \|\hat{\varphi}_{i-1} - \hat{\varphi}_i\|_{\pi}\} \quad (13)$$

The purpose of making the forward and backward recursion is to make the tensor entries value is stable. So, the 3D distance transform is updated concerning the orientation cost. Then to produce the directional chamfer matching value for any template U_{map} is formulated in (14)

$$dt_{CM}(U_{map}, Q) = \frac{1}{n} \sum_{u_{proj_i} \in U_{map}} DT3_{V_q}(u_{proj_i}, \hat{\varphi}(u_{proj_i})) \quad (14)$$

As all points in a line have the same direction, so all the pixels which have the same orientation channel computed as Integral Distance Transform (IDT). The formulation of IDT is computed in (15). p_0 is a meeting point between the line segment in the query image and the template line over the p which has direction $\hat{\varphi}_i$.

$$IDT3_{V_q}(p, \hat{\varphi}_i) = \sum_{p_j \in \text{line}_{[p_0, p]}} DT3_{V_q}(p_j, \hat{\varphi}_i) \quad (15)$$

The difference between the summation of Integral Distance Transform endpoint e_j and start point s_j of line l in the template line $Line_{U_{map}}$ is the directional chamfer matching cost in any template U_{map} . The directional chamfer matching cost is calculated over the meeting point and defined in (16),

$$dt_{CM}(U_map, Q) = \frac{1}{n} \sum_{line_{[s_j, e_j]} \in Line_{U_map}} (IDT3_{V_q}(e_j, \hat{\phi}(line_{[s_j, e_j]})) - \dots \dots - IDT3_{V_q}(s_j, \hat{\phi}(line_{[s_j, e_j]}))) \quad (16)$$

C. Refinement Pose of Object Detection

The idea of refinement is to optimize the result by applying the minimum error. The error is minimized by the Lavenberg-Marquardt method. The input of the Lavenberg-Marquardt method is the result of the most minimum DCM cost, which satisfies the threshold. The most minimum DCM cost is attained from the back projection of a coarse pose parameter. The pipeline of the optimized pose step is illustrated in Fig. 5 below.

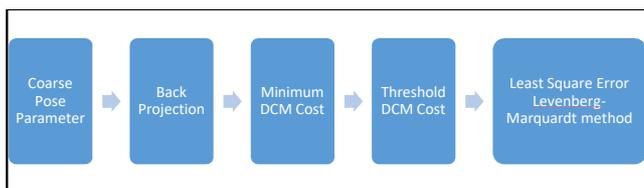


Fig. 5 Pipeline of the optimized pose

The coarse pose parameter of the object is described in the image frame. The coarse pose data which are stored are translation concerning x, y, and z-axis (fix distance between the camera and the object) and the rotation for the z-axis. The Back-Projection formula changes the coarse pose of the object from the image frame into the world frame. The formulation of The Back Projection is described in (7).

w is the scaling value. While for the parameter of u_proj_i , Rot_m , and $trans_m$ are based on the coarse pose parameter, which is attained from the previous step.

The next calculation is achieving the minimum DCM Cost [16], which is processed in the image frame. The inputs are the coarse pose object and the template object. The formulation of the Minimum DCM Cost is illustrated in (17).

$$Min_DCM = \arg \min_{u_proj_i \in U_map} ||u_{q_j} - u_{proj_i}|| + \lambda ||\phi(u_i) - \phi(v_j)|| \quad (17)$$

The u_proj_i indicates the query image point after applying the coarse pose parameter. The U_map is the set points of u_proj_i . Then u_{q_j} is the projected 3D template, as explained previously. The idea of finding the minimum DCM Cost is to find the nearest point between the query image, which applies a coarse pose parameter with respect to the projected 3D template in an image frame.

After the DCM Cost is determined, then the threshold tr value is applied, such that only store the pair points which have the DCM Cost below the threshold tr value. The lower DCM Cost, the closer distance between the query image (applying coarse parameter), and the distance. The threshold value tr is calculated from the median of chamfer matching cost (dt_{CM}) and the certain value (tr_{const}). The formulation is described in (18),

$$tr \begin{cases} tr_{median}(d_{DCM}), & \text{if } tr_{median}(dt_{CM}) > tr_{const} \\ tr_{const}, & \text{if } tr_{median}(dt_{CM}) < tr_{const} \end{cases} \quad (18)$$

The threshold value tr is equal to the median of chamfer matching cost $tr_{median}(d_{DCM})$, if the median of chamfer matching cost more prominent than the determined constant value tr_{const} . So, it is for the vice versa.

III. RESULTS AND DISCUSSION

The result is classified into three main findings, as illustrated in Fig. 6.

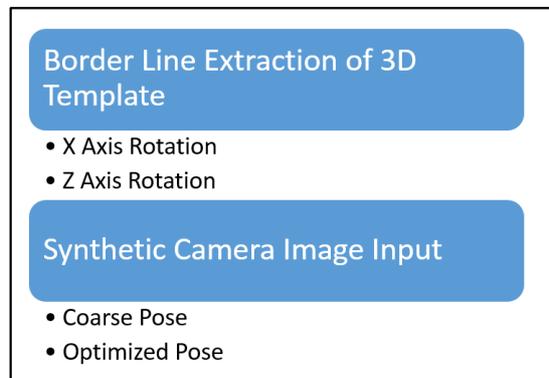


Fig. 6 Experiment classification

The first result is regarding the template generation that classified into two-axis rotations. The rotation concerning the y axis is not conducted as the object shape does not change when the object is rotated for the y-axis. Then the second result is about the object pose of the synthetic camera image. The second experiment is divided into two sub-experiments: Coarse Pose experiment and Optimized Pose experiment.

A. Border Line Extraction of 3D Template

The borderline extraction of the 3D template is specified into two examinations: rotate the template with respect to the x-axis and rotate the template with respect to the z-axis. Moreover, the axis base that is used to rotate the object is described in Fig. 7.

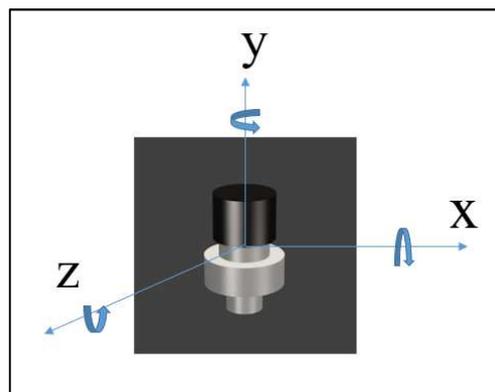


Fig. 7 Axis base of the object

The necessity for this examination is because the borderline extraction result is the reference image for DCM calculation. The high possibility of the detected line in the border of the object is also the reason it is essential for conducting this examination. X-Axis Rotation: Based on

Fig. 7, the template, which rotates for the x-axis, changes the border shape of the template. There are 5 sample degrees for the examination, with * is Degree ($^{\circ}$). The result is explained

presented in Table I and Table III. Table I summarizes the result for the degree 0° up to 90° . For the experiment, more than 90° is summarized in Table II.

TABLE I
BORDER LINE EXTRACTION: ROTATION X AXIS (1)

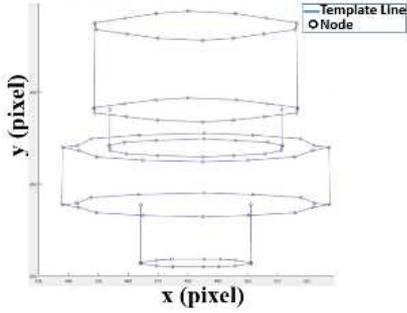
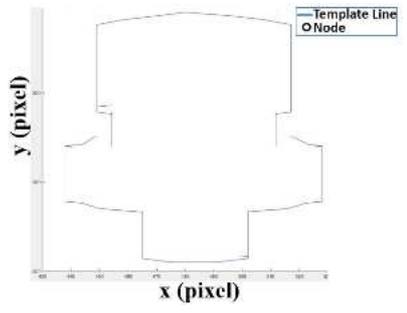
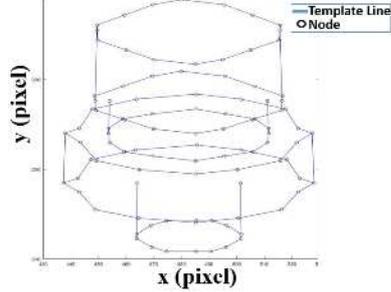
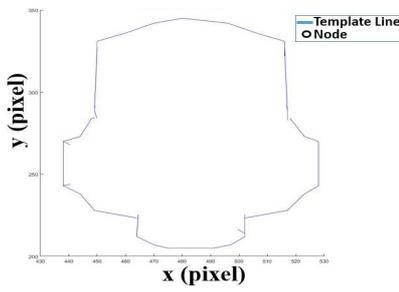
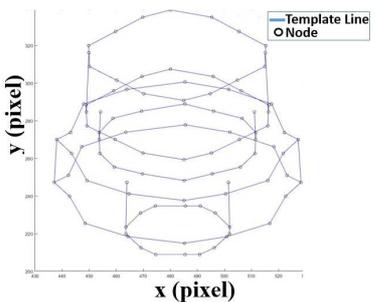
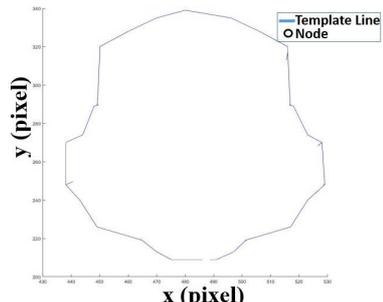
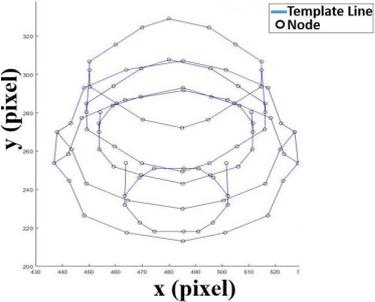
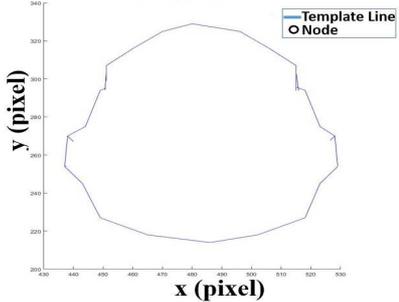
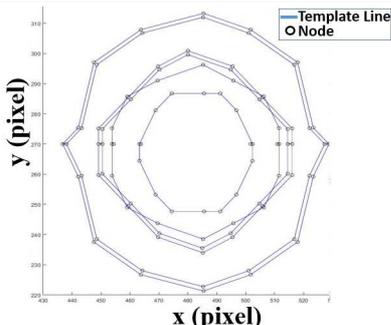
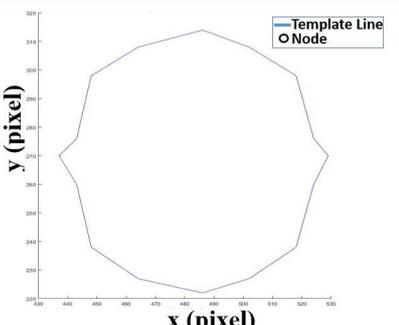
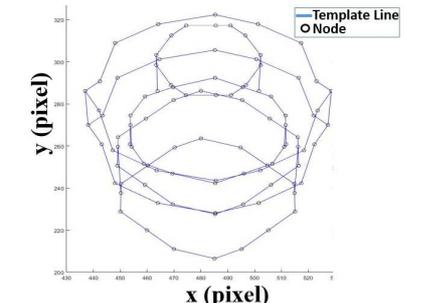
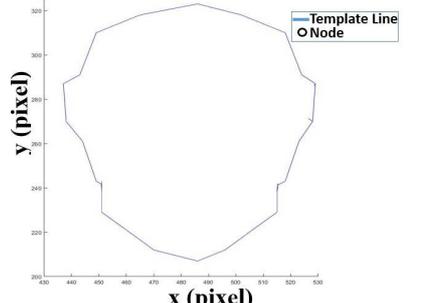
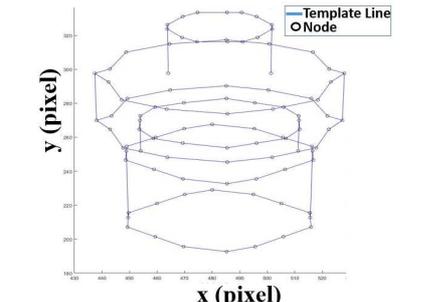
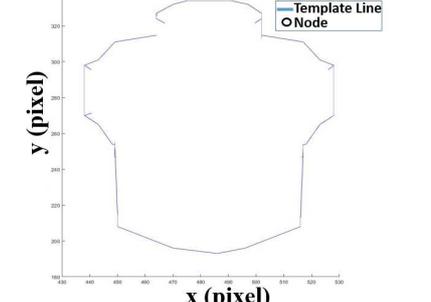
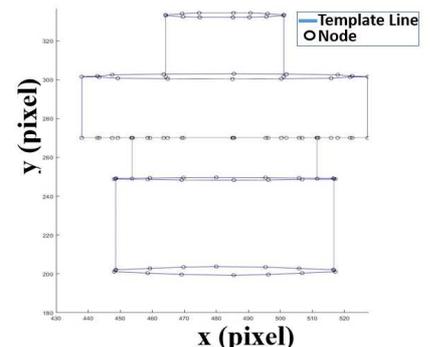
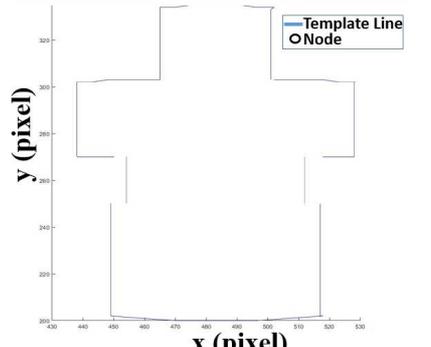
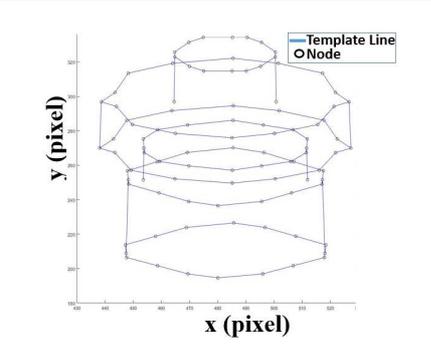
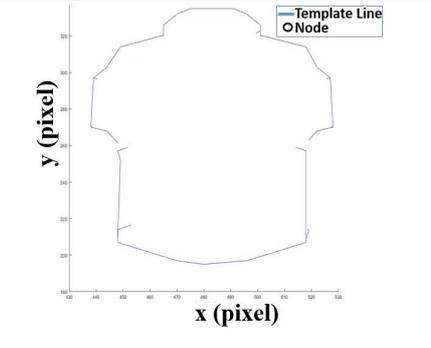
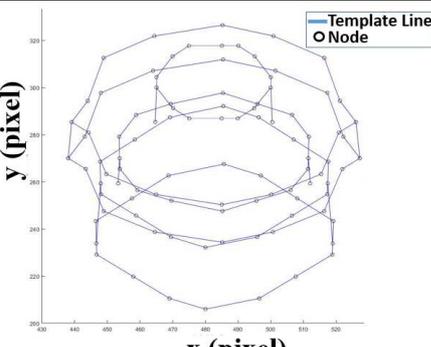
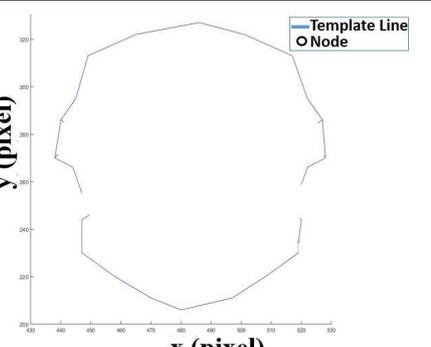
No	*	Projected Template	Border Line Extraction
1	10		
2	30		
3	45		
4	60		
5	90		

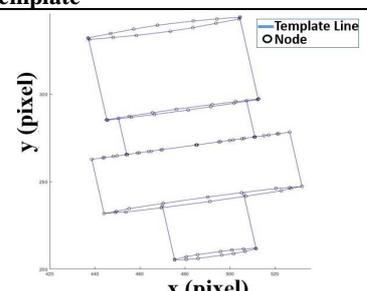
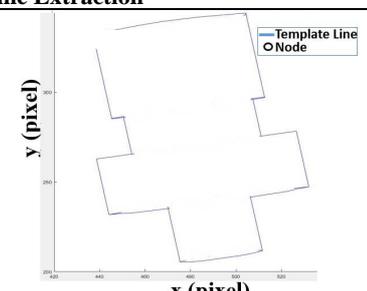
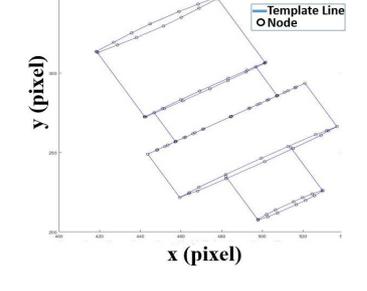
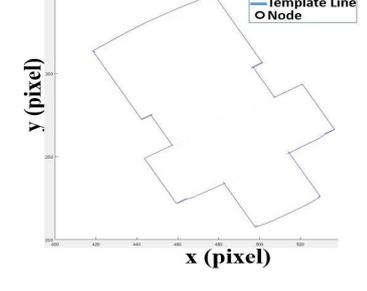
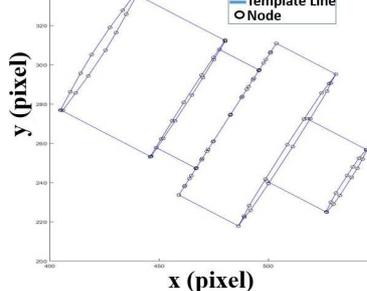
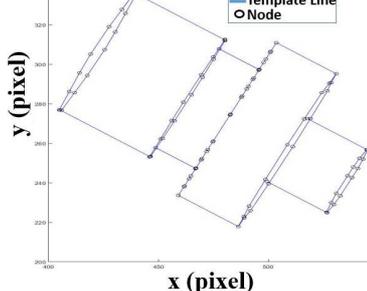
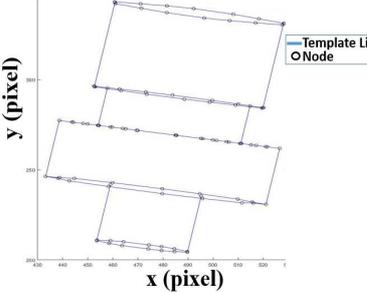
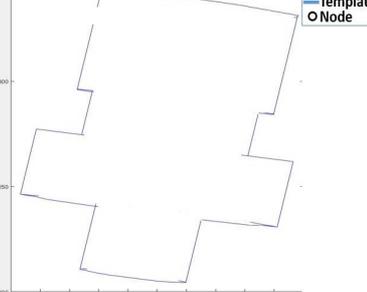
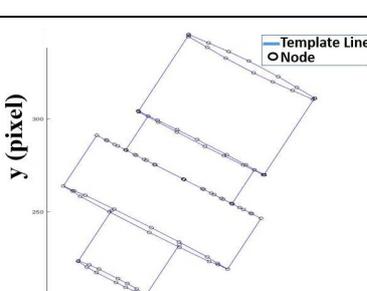
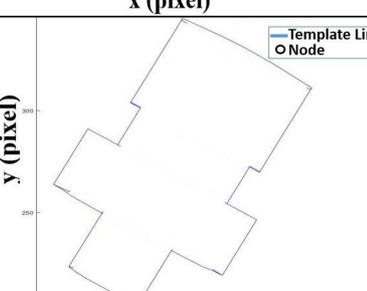
TABLE II
 BORDER LINE EXTRACTION: ROTATION X-AXIS (2)

No	*	Projected Template	Border Line Extraction
1	120		
2	150		
3	180		
4	210		
5	240		

Qualitatively, the system extracts the borderline of the object in any variety of angles. The object shape is changing concerning the camera view.

1) *Z-Axis Rotation*: Refer to Fig. 7, the template is rotated for the z-axis. The result is summarized in Table III,

TABLE III
BORDER LINE EXTRACTION: ROTATION Z-AXIS

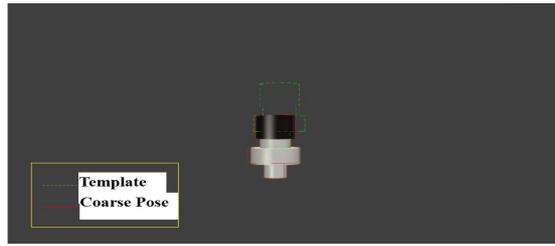
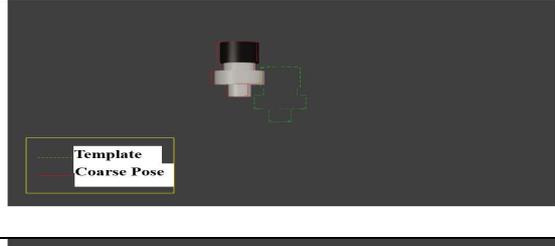
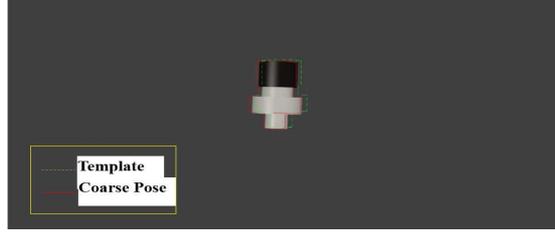
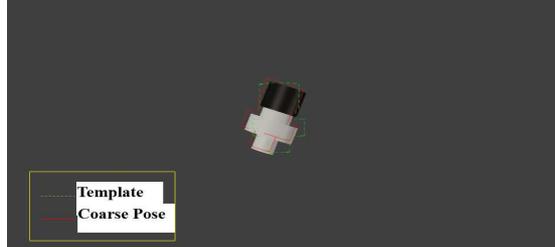
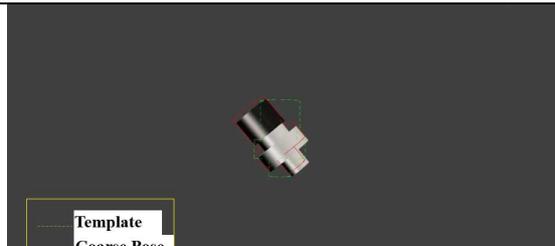
No	*	Projected Template	Border Line Extraction
1	-10		
2	-30		
3	-60		
4	10		
5	30		

The template, which rotates for the z-axis, does not change the border shape of the template. The examination is conducted clockwise and counter-clockwise direction concerning the z-axis.

B. Coarse Pose

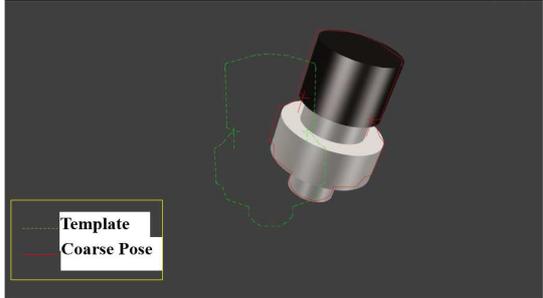
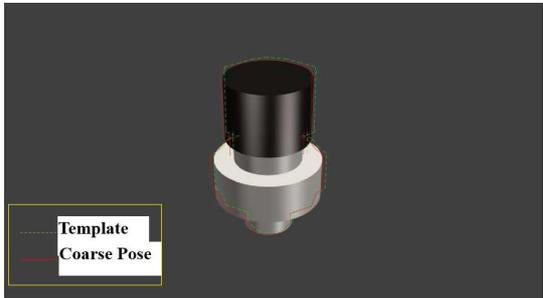
The next examination is determining the coarse pose of the object. The Blender camera acquires the captured image. So, the ground truth can be determined by the parameter setting in the *Blender*. Table IV summarizes the result (samples) for the coarse pose of the object, which is rotated concerning the z-axis.

TABLE IV
QUALITATIVELY COARSE POSE RESULT (1)

No	Coarse Pose
1	
2	
3	
4	
5	

The dashed green line shows the template line, which is projected for the first time at the image plane. Then by using the algorithm to find the coarse pose object, the red line shows the result of the coarse pose. Table V is the result of the coarse pose object when the object is rotated some degrees concerning the x-axis.

TABLE V
QUALITATIVELY COARSE POSE RESULT (2)

No	Coarse Pose
1	
2	
3	

Based on the result of Table IV and Table V, qualitatively, all the coarse poses result in almost the same as the object pose. Quantitatively, the result of the coarse pose accuracy for the 50 poses of the object (various poses) is explained in Table VI

TABLE VI
QUANTITATIVELY COARSE POSE RESULT

Parameter	Value	Unit
Translation of x axis (tx)	2.05	mm
Translation of y axis (ty)	0.71	mm
Translation of z-axis (tz)	-	
Rotation of x axis (Rx)	0.01	degree
Rotation of y-axis (Ry)	-	
Rotation of z-axis (Rz)	0.45	degree

The translation of the z-axis does not have the value as the distance between the camera and the object is constant or fixed. The rotation of the y-axis also does not have the value because the object shape is not changing when it is rotated to the y-axis. Fig. 8 shows the data of coarse pose examination results when it is represented as a graph.

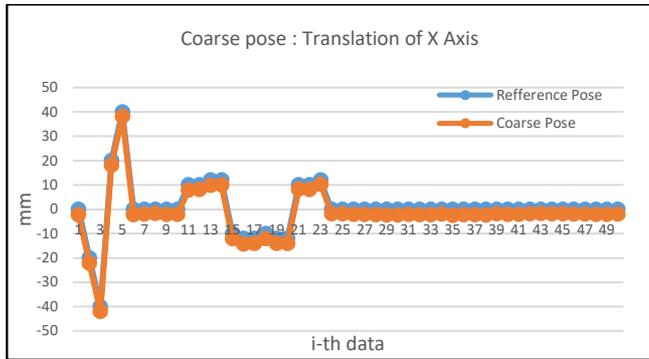


Fig. 8 Coarse poses: Translation of x-axis data

Fig. 9 represents the coarse pose result for the translation for the y-axis.

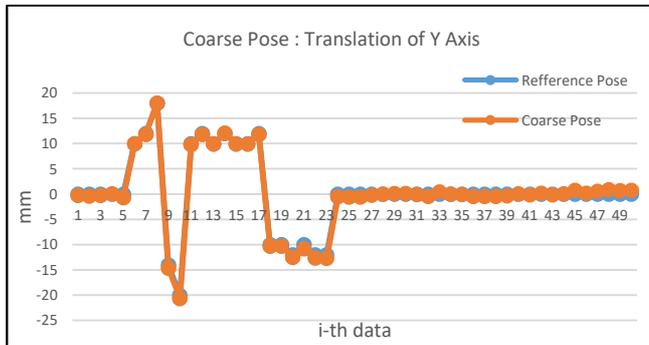


Fig. 9 Coarse poses: Translation of y-axis data

Fig. 10 represents the comparison between reference pose and coarse pose of the object for rotating for the z-axis.

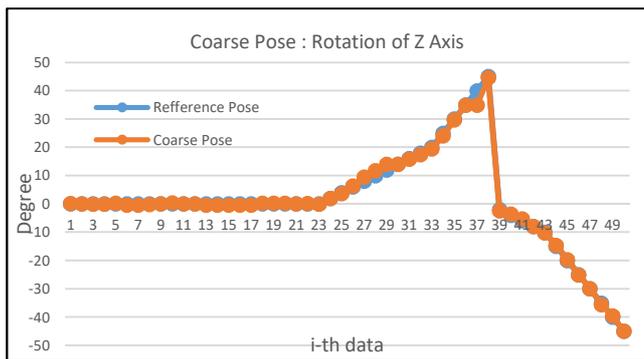


Fig. 10 Coarse poses: Rotation of z-axis data

Quantitatively, the average error of the coarse pose for the translation is 1.38 mm, and the rotation is 0.23°.

C. Optimized Pose

The following result is regarding the optimized pose. By applying the formulation (17) and (18), the result is summarized in Table VII. The same as the coarse pose

examination, the captured image also attained by using *Blender* software.

TABLE VII
QUALITATIVELY OPTIMIZED POSE RESULT

No	Optimized Pose
1	
2	
3	
4	
5	

Quantitatively, the result of the optimized pose is shown in Table VIII.

TABLE VIII
QUANTITATIVELY OPTIMIZED POSE RESULT

Parameter	Value	Unit
Translation of x axis (tx)	1.82	mm
Translation of y axis (ty)	0.24	mm
Translation of z-axis (tz)	-	
Rotation of x axis (Rx)	2.88	degree
Rotation of y-axis (Ry)	-	
Rotation of z-axis (Rz)	0.82	degree

If the result is presented in a graph, the comparison between the reference pose and the optimized pose for the translation for the x-axis is presented in Fig. 11.

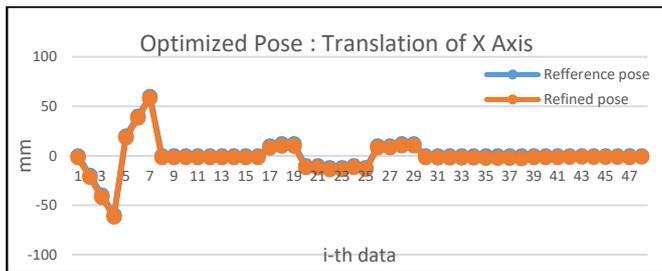


Fig. 11 Optimized pose: Translation of x-axis data

Fig. 12 represents the optimized pose versus the reference pose for the translation for the y-axis.

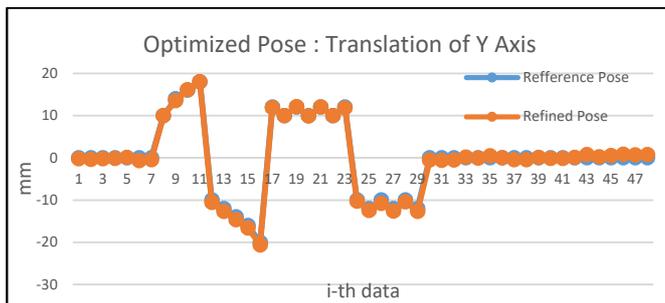


Fig. 12 Optimized pose: Translation of y-axis data

Fig. 13 illustrates the optimized pose of the object and the reference pose of the object in terms of rotation with respect to the x-axis.

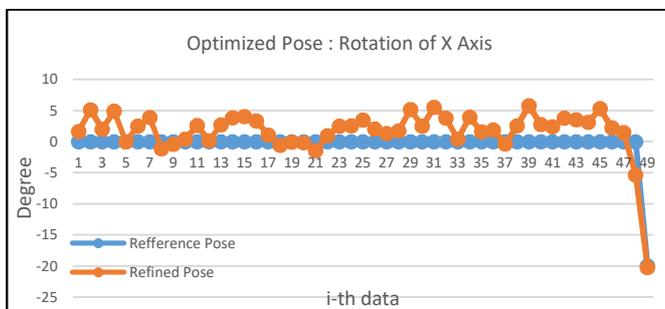


Fig. 13 Optimized pose: Rotation of x-axis data

Fig. 14 explains the object pose, which is as a reference and as an optimized pose for the rotation for the z-axis.

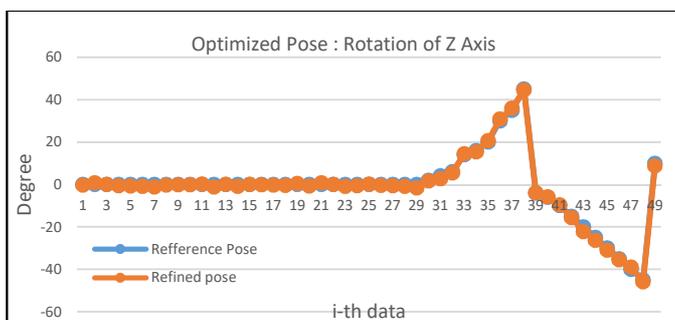


Fig. 14 Optimized pose: Rotation of z-axis data

Quantitatively the average error of rotation using optimized pose formula achieved 1.85° . Meanwhile, the average error of the translation is 1.03 mm.

IV. CONCLUSIONS

The template generation applied Border Line Extraction Method. The template is the reference template that is calculated for the Directional Chamfer Matching Method. Directional Chamfer Matching Method to determine the coarse pose. The optimized object pose algorithm has a better result for the translation pose (x, y, z) of the object. The average error for the translation (x, y, z) of optimized pose achieved 1.03 mm, meanwhile for the coarse pose is 1.38 mm. The optimization method applied the Least Square Error Lavenberg-Marquardt method. The average error of rotation pose (θ_x and θ_z) performs better results in the coarse pose mode. The average error of rotation pose for the coarse pose mode is 0.23° ; meanwhile, the optimized mode is 1.85° .

ACKNOWLEDGMENT

The research is sponsored by MERLIN Lab and LPDP Indonesia.

REFERENCES

- [1] Lysenkov I, Eruhmov V. Pose Refinement of Transparent Rigid Objects with a Stereo Camera. Lecture Notes in Computer Science [Internet]. Springer Berlin Heidelberg; 2013;143–57. Available from: http://dx.doi.org/10.1007/978-3-642-39759-2_11.
- [2] Gualtieri M, ten Pas A, Saenko K, Platt R. High precision grasp pose detection in dense clutter. 2016 IEEE/RSSJ International Conference on Intelligent Robots and Systems (IROS) [Internet]. IEEE; 2016 Oct; Available from: <http://dx.doi.org/10.1109/iros.2016.7759114>.
- [3] Ten Pas A, Gualtieri M, Saenko K, Platt R. Grasp Pose Detection in Point Clouds. The International Journal of Robotics Research [Internet]. SAGE Publications; 2017 Oct 30;36(13-14):1455–73. Available from: <http://dx.doi.org/10.1177/0278364917735594>.
- [4] Hossain D, Capi G, Jindai M, Kaneko S. Pick-place of dynamic objects by robot manipulator based on deep learning and easy user interface teaching systems. Industrial Robot: An International Journal [Internet]. Emerald; 2017 Jan 16;44(1):11–20. Available from: <http://dx.doi.org/10.1108/ir-05-2016-0140>.
- [5] Czajewski W, Kołomyjec K. 3D Object Detection and Recognition for Robotic Grasping Based on RGB-D Images and Global Features. Foundations of Computing and Decision Sciences [Internet]. Walter de Gruyter GmbH; 2017 Sep 1;42(3):219–37. Available from: <http://dx.doi.org/10.1515/fcds-2017-0011>.
- [6] Tejani A, Tang D, Kouskouridas R, Kim T-K. Latent-Class Hough Forests for 3D Object Detection and Pose Estimation. Lecture Notes in Computer Science [Internet]. Springer International Publishing; 2014;462–77. Available from: http://dx.doi.org/10.1007/978-3-319-10599-4_30.
- [7] Di Castro M, Vera JC, Masi A, Pérez MF. Novel Pose Estimation System for Precise Robotic Manipulation in Unstructured Environment. Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics [Internet]. SCITEPRESS - Science and Technology Publications; 2017; Available from: <http://dx.doi.org/10.5220/0006426700500055>.
- [8] Abbeloos W, Ataer-Cansizoglu E, Caccamo S, Taguchi Y, Domae Y. 3D Object Discovery and Modeling Using Single RGB-D Images Containing Multiple Object Instances. 2017 International Conference on 3D Vision (3DV) [Internet]. IEEE; 2017 Oct; Available from: <http://dx.doi.org/10.1109/3dv.2017.00056>.
- [9] Izatt G, Dai H, Tedrake R. Globally Optimal Object Pose Estimation in Point Clouds with Mixed-Integer Programming. Robotics Research [Internet]. Springer International Publishing; 2019 Nov 28;695–710. Available from: http://dx.doi.org/10.1007/978-3-030-28619-4_49.

- [10] Sakcak B, Bascetta L, Ferretti G. Model based Detection and 3D Localization of Planar Objects for Industrial Setups. Proceedings of the 13th International Conference on Informatics in Control, Automation and Robotics [Internet]. SCITEPRESS - Science and Technology Publications; 2016; Available from: <http://dx.doi.org/10.5220/0005982503600367>.
- [11] Liu M-Y, Tuzel O, Veeraraghavan A, Taguchi Y, Marks TK, Chellappa R. Fast object localization and pose estimation in heavy clutter for robotic bin picking. The International Journal of Robotics Research [Internet]. SAGE Publications; 2012 May 8;31(8):951–73. Available from: <http://dx.doi.org/10.1177/0278364911436018>.
- [12] Zeng Z, Yan H. Hidden Line Removal for 2D Cartoon Images. In: Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing - Volume 11. AUS: Australian Computer Society, Inc.; 2001. p. 89–92. (VIP '01).
- [13] Nisha N. Visible Surface Detection Algorithms: A Review. International Journal of Advanced Engineering Research and Science [Internet]. AI Publications; 2017;4(2):147–9. Available from: <http://dx.doi.org/10.22161/ijaers.4.2.29>.
- [14] Li M, Guo B, Zhang W. An Occlusion Detection Algorithm for 3D Texture Reconstruction of multi-View Images. International Journal of Machine Learning and Computing [Internet]. EJournal Publishing; 2017 Oct;7(5):152–5. Available from: <http://dx.doi.org/10.18178/ijmlc.2017.7.5.638>.
- [15] Maarif ES, Pitowarno E, Widodo RT. A Trajectory Generation Method Based on Edge Detection for Auto-Sealant Cartesian Robot. Journal of Mechatronics, Electrical Power, and Vehicular Technology [Internet]. Indonesian Institute of Sciences; 2014 Jul 23;5(1):27. Available from: <http://dx.doi.org/10.14203/j.mev.2014.v5.27-36>.