Localization and Tracking of a 3-D Object Based on Multi-View Image Acquisition

Yao-Cheng Chuang and Huei-Yung Lin
Department of Electrical Engineering
National Chung Cheng University
168 University Rd., Chiayi 621, Taiwan
Email: masatoyao@gmail.com, hylin@ccu.edu.tw

***Abstract*—This paper presents a 3-D object localization and tracking technique based on the CAD model and multi-view image captures of the object. From the given projected 2-D pose model in the image, the matching lines between the model contour and the object's edge feature are used for nonlinear 3-D pose computation. The object location information in the real world is then identified. The method presented in this work has been validated on several experiments with various test objects. The results demonstrate that the proposed approach is robust to the partial occlusion and provides accurate positioning of the real object.**

I. Introduction

Determining the 3-D pose of an object in a scene is an important task in the fields of computer vision, computer graphics, photogrammetry, robotics, and industrial applications, etc. Most of the current tracking techniques can be divided into two main categories: 2-D image based and 3-D pose based tracking. The former approach mainly focuses on tracking the 2-D features such as points, segments, circles, object contours, or regions of interest [1], [2], [3], [4], [5] The latter explicitly uses a 3-D model of the target object, based on the information in the image, to reconstruct a CAD model of the object. The approach takes the CAD model to track the object in the image, and compute a rigid-body transformation. It is necessary to match the features of the 3-D model with part of the visible 2-D image features [6], [7]. Compared with the method based on 2-D features, the model-based tracking approach can track and localize the object more precisely and robustly.

There exist many techniques to track and localize an object. Dementhon [8] proposed the "POSIT" approach for estimating the 3-D rotation and translation of an object from a single 2-D image if an approximate 3-D model of the object is known and the corresponding points in the 2-D image are provided. In the method, the 3-D pose is estimated directly from the feature points of the 3-D model and the 2-D image, and the errors are corrected iteratively until a good estimate is found from the single image. The disadvantages of POSIT are that the 3-D model of the object and the corresponding points in the 2-D image must be given, and it only works on non-coplanar points. (In other words, it does not work for the planar objects.)

If there is no feature correspondence available, how to estimate the 3-D pose of an object using one image is still not well solved. Leng [9] proposed a new contour-based method, which dealt with both the pose estimation and the feature correspondence simultaneously and iteratively. The outer contour of the object is first extracted from the 2-D image. A tentative point correspondence relationship is established between the extracted contour and object's 3-D model. It is used to estimate the pose parameters of the object. Finally, the newly estimated pose parameters are used to update the tentative point correspondences, and the process is iterated until convergence.

If the object is moving, the motion may cause the partial occlusion and let the tracking result incorrect. Azad [10] proposed a tracking approach using particle filter, which can deal with arbitrary shapes or textureless surface. Their method provides a general solution to the rigid object tracking problem, and is able to deal with the partial occlusion. It is practical to use in the context of goal-directed imitation learning involving the observation of object manipulations.

In general, the edge-based tracking method is not feasible in the presence of highly textured environments. Moreover, the texture-based tracking approach is usually not accurate if there is a significant difference between current and reference texture scales. Pressigout [11] proposed a real-time, robust and effective tracking framework for visual servoing applications. Their algorithm is based on the fusion of visual cues and the homography transformation estimation. The transformation parameters are estimated using a non-linear minimization of a uniqueness criterion that integrates the information obtained from the texture and edges of the tracked object.

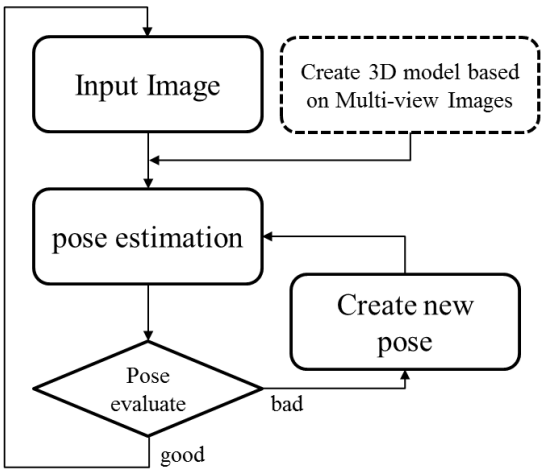

Fig. 1. The flowchart of the proposed 3-D tracking system.

978-1-4799-2398-4/14/$31.00 ©2014 IEEE

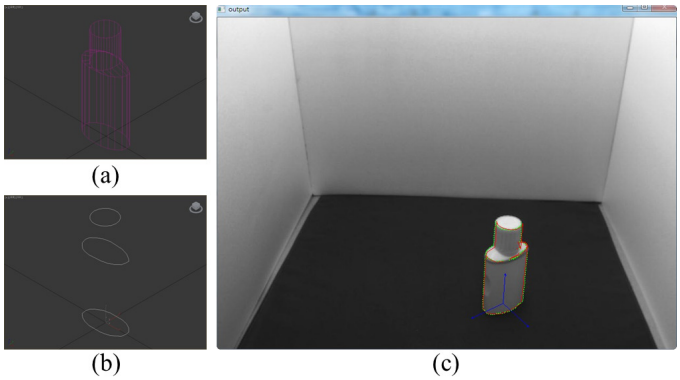

Fig. 2. The 3-D modeling steps of the proposed technique: (a) CAD model, (b) contour model, (c) final model.

In the previous object tracking methods, the transformation of the object pose is computed using the features extracted from the image and compared with a pose of the model. If the object is observed with severe occlusion, it still relies on the features to analyze the object's possible pose. This will make the tracking results incorrect due to the lack of visible object information in the image. One common solution is to wait for the partial occlusion diminished and re-detect the pose. However, if the change of the object's pose is too large, it is not possible to recover its position and orientation.

In this work, we present an object localization and tracking approach with the evaluation of the 3-D pose precision. If the 3-D pose estimation does not achieve a preset threshold, it is discarded and a new 3-D pose is generated without the prior information. Furthermore, multi-view images captured from different cameras are adopted for 3-D pose computation and object tracking. If the partial occlusion occurs on certain viewpoints, it is still possible to recover the correct 3-D pose from other visible camera positions. In the experiments, our approach provides stable tracking results with several test objects.

II. 3-D Tracking Approach

The flowchart of the proposed 3-D object tracking technique is shown in Figure 1. The first step is to create the 3-D model of the object using image-based modeling. From the multi-view image captures, we use 123D Catch[1] to reconstruct the 3-D point cloud of the object. Then the 3-D point cloud model is passed to 3ds Max[2] to build a geometric model of the object manually. The scale of the geometric model is adjusted to obtain a CAD model that is able to fit the object's shape, as illustrated in Figure 2(a). Based on the shape of the CAD model, a contour model as shown in Figure 2(b) is created with the initial pose. The surface patch of the mesh model is identified by comparing the boundary of the visible surface and the edge of the contour model. Finally, a mesh model consistent with the object's surface outline in the image is obtained, as shown in Figure 2(c).

To obtain the 3-D location and orientation information of the object with respect to the camera in the real world, Comport's full scale non-linear optimization [7], Virtual Visual Servoing (VVS), is adopted to solve the pose computation problem. The edge extraction is used to determine the feature point locations in the next imageI^{t+1} with the oriented gradient algorithm. A criterion corresponding to the square root of a log-likelihood ratio ζ_j and the absolute sum of the convolution values are adopted [12]. They are computed at p^t and Q_j respectively in the images I^t and I^{t+1} using a pre-determined mask M_δ of the contour orientation. The new position p^{t+1} is given by:

$$Q_j^* = \arg \max_{j \in [-J,J]} \zeta_j \tag{1}$$

where $\zeta_j = |I^t_{v(p^t)} * M_\delta + I^{t+1}_{v(Q_j)} * M_\delta|$, and $v(\cdot)$ is the neighborhood of the interested point.

In [13], the authors proposed a general framework to compute the interaction matrix L_s. Any kind of geometrical features can be considered in the proposed control law if it is possible to compute the corresponding interaction matrix L_s. The interaction matrix related to a straight line is given by

$$L_{d_l} = L_\rho + \alpha\, L_\theta \tag{2}$$

and we have

$$J_{d_l} = \begin{bmatrix} \lambda_{d_l} \cos\theta \\ \lambda_{d_l} \sin\theta \\ -\lambda_{d_l}\rho \\ (1+\rho^2)\sin\theta\alpha\,\rho\cos\theta \\ -(1+\rho^2)\cos\theta\alpha\,\rho\sin\theta \\ -\alpha \end{bmatrix}^\top \tag{3}$$

where $\lambda_{d_l} = \lambda_\rho + \alpha\lambda_\theta$.

After the pose estimation, the correctness of the pose is evaluated. The evaluation flowchart is shown in Figure 3. If the pose is incorrect, we need to create a new one. The evaluation process for both the singe and multiple cameras is the same. Given an RGB image, the edge detection is carried out to find the object boundary for the segmentation of foreground and background regions. In the edge segmentation block, Canny edge detector [14] is used to find the boundary image $I_c(x, y)$ from the input image $I(x, y)$. In the foreground segmentation block, we use the background subtraction [15] to extract the R, G, B channels of the same pixel position. If the amount of changes in the R, G, B channels is greater than a threshold, the pixel is considered as a foreground point.

The pose evaluation is used to verify the object's perspective projection in the image plane. The model's contour image $I_{mc}(x, y)$ and the overall image $I_{mf}(x, y)$ are compared

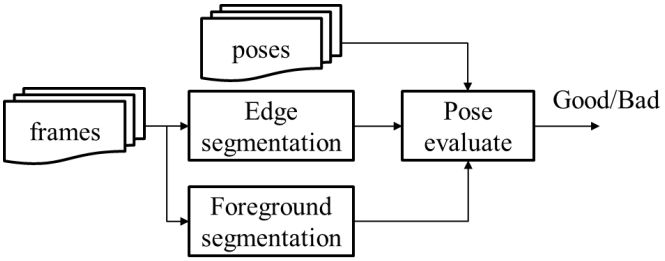

Fig. 3. The pose evaluation flowchart used in this work.

[1]http://www.123dapp.com/catch.
[2]http://www.autodesk.com

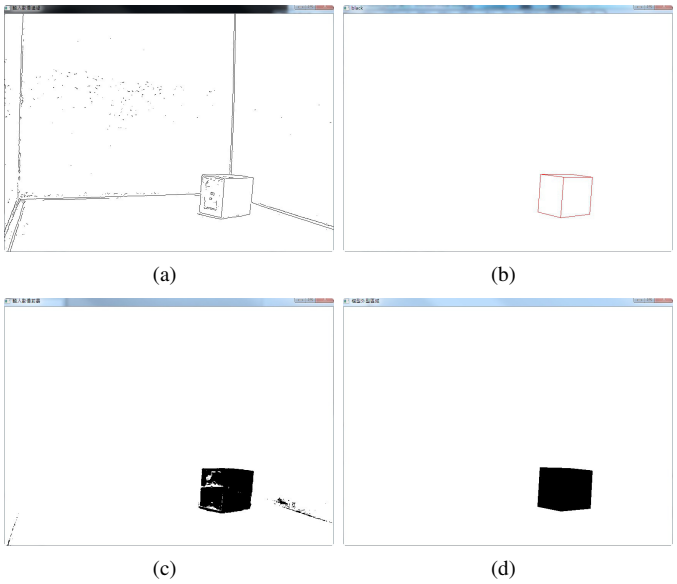

Fig. 4. The images used in the pose evaluation: (a) edge image, (b) boundary image, (c) foreground image, (d) object image.

with the boundary image $I_c(x, y)$ and the foreground image $I_f(d, y)$, respectively. The precision evaluation is given by

$$Precision = \alpha P_{edge} + \beta P_{region} \tag{4}$$

where

$$P_{edge} = I_{mc}(x, y) \cap I_c(x, y) \tag{5}$$
$$P_{region} = I_{mf}(x, y) \cap I_f(x, y) \tag{6}$$

and α and β are the weighting of edge and region. If the pose evaluation result is good, then the next image is processed. If the result is unsatisfactory, then a new pose of the object is created. The images processed in the pose evaluation are shown in Figure 4, which include (a) edge image, (b) boundary image, (c) foreground segmentation, and (d) object image.

One important task for 3-D pose tracking is to make it robust under occlusion. We observe that in most cases the moving object is in linear motion, and there is no sudden involved. Thus, a pose prediction strategy can be adopted to replace the incorrect pose. Figure 5 illustrates the pose prediction process. $\text{Frame}_1 \sim \text{frame}_5$ are consecutive images, and pose_1, pose_2 and pose_3 are the correct poses obtained from estimation. Because the movement of the object is continuously differentiable, we can use the correct $\text{pose}_1 \sim \text{pose}_3$ to calculate the translation and rotation between the poses, and make a prediction to replace the wrong pose in frame_5. We do not use the predicted pose in frame_4 for evaluation because the unclear image feature or serious partial occlusion will affect the evaluation accuracy.

In the single camera tracking system, a newly created pose is used in the pose prediction to replace the incorrect pose. However, there are three possibilities of pose evaluation in the multi-view camera tracking system. The first one is that the pose evaluation is good in all cameras, and the tracking process can directly go to to next image. The second one is that the pose evaluation is satisfactory only in part of the cameras. In this case, we use the coordinate transformation among the cameras to transform the good pose to other views and replace the incorrect results. The third possibility is that the pose evaluation is not correct in all cameras. In this case, we adopt the pose prediction to create a new pose in all cameras, and use the predicted pose to replace the incorrect ones in all viewpoints.

III. Experiments

This section presents the tracking results of various objects using our approach. The cameras used in the experiments are BASLER Gigabit Ethernet aca2500-14gc, and the resolution of the image sequences is 862×646 with a frame rate of 30 fps. We adopt Tsai's calibration technique [16] to obtain the camera parameters. In the experiments, we emphasize three important issues of our localization and tracking approach: (1) the examination of the positioning accuracy, (2) the improvement with multi-view images, (3) the pose prediction under serious occlusion.

To evaluate the correctness of the proposed tracking algorithm, the positioning accuracy of the object pose is calculated. The displacement and rotation are compared with the groundtruth pose, respectively. In the displacement accuracy experiment, 11 images are taken with the object's movement along the Y-axis for every 2 cm. The object location in the first image is set as the coordinate origin. Table I shows the displacement results associated with the object's motion. Similarly, another 11 images are taken with the object's movement along the Z-axis for every 2 cm. The displacement results are shown in Table II. For the rotation accuracy evaluation, 8 images are taken with the object's rotation around the X-axis for every 45 degrees. Table III shows the results of rotation experiment. The results of image frame versus displacement or rotation measurement for these three cases are shown in Figures 6 – 8.

To improve the robustness of the tracking results under occlusion, multiple cameras are used to capture the images from different viewpoints. As shown in Figure 9, the images captured from camera No. 2 contain severe occlusion, which makes the object tracking difficult. In this case, the we transform the object's pose observed from another image to replace the pose in the current viewpoint. The proposed technique is also able to track more complex objects with curved features, such as the bottle illustrated in Figure 10.

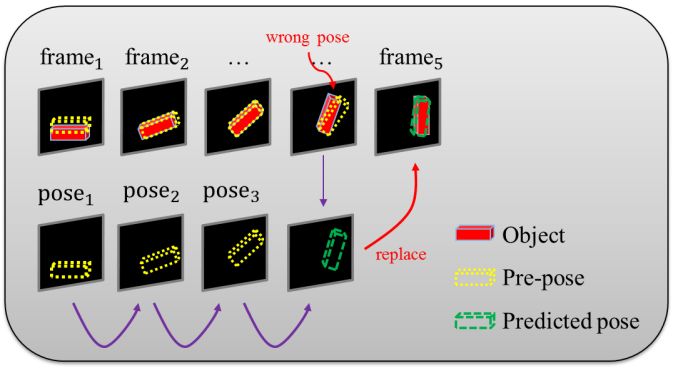

Fig. 5. The pose prediction process illustrated with several input images.

TABLE I. Y-AXIS DISPLACEMENT ACCURACY

Image	distance (cm)	experiment result (cm)
1	(0,2,0)	(0.197 , 2.034 , -0.162)
2	(0,4,0)	(0.342 , 4.094 , -0.344)
3	(0,6,0)	(0.340 , 6.148 , -0.339)
4	(0,8,0)	(0.413 , 8.379 , -0.437)
5	(0,10,0)	(0.364 , 10.644 , -0.392)
6	(0,12,0)	(0.364 , 12.803 , -0.356)
7	(0,14,0)	(0.491 , 14.807 , -0.539)
8	(0,16,0)	(0.493 , 17.003 , -0.550)
9	(0,18,0)	(0.353 , 19.367 , -0.406)
10	(0,20,0)	(0.233 , 21.733 , -0.281)

TABLE II. Z-AXIS DISPLACEMENT ACCURACY

Image	distance (cm)	experiment result (cm)
1	(0,0,2)	(0.148 , -0.030 , 2.038)
2	(0,0,4)	(0.275 , -0.057 , 4.182)
3	(0,0,6)	(0.506 , -0.206 , 6.048)
4	(0,0,8)	(0.514 , -0.083 , 8.250)
5	(0,0,10)	(0.471 , 0.030 , 10.561)
6	(0,0,12)	(0.444 , 0.146 , 12.860)
7	(0,0,14)	(0.612 , 0.084 , 14.834)
8	(0,0,16)	(0.799 , 0.130 , 16.888)
9	(0,0,18)	(0.503 , 0.380 , 19.460)
10	(0,0,20)	(0.394 , 0.611 , 21.845)

TABLE III. ROTATION ACCURACY

Image	angle (°)	experiment result (°)
1	(45,0,0)	(43.638 , 1.307 , 0.732)
2	(90,0,0)	(87.824 , -0.542 , -0.622)
3	(135,0,0)	(133.420 , 0.186 , -0.354)
4	(180,0,0)	(179.480 , 0.362 , -0.890)
5	(225,0,0)	(223.531 , 0.155 , -1.528)
6	(270,0,0)	(267.608 , 0.445 , -0.708)
7	(315,0,0)	(313.296 , -0.826 , -0.334)

In the last experiment, we let the object's occlusion is too severe to perform the visual tracking using a single camera. The proposed technique deals with this situation by generating a new pose to continue the tracking process, as illustrated in Figure 11. When the pose evaluation result is correct, the red contour is used to represent the object pose. However, if the pose evaluation result is incorrect, we generate a new pose by pose prediction method and used the cyan contour to represent the pose.

IV. CONCLUSION

In this paper we present a model-based tracking system using multiple cameras and the CAD model to identify the object's 3-D pose. If the occlusion in the images is too severe and leads to incorrect object pose estimation, a pose prediction approach is adopted for the single camera tracking. Furthermore, the coordinate transformation is carried out to replace the incorrect pose using the information obtained from other viewpoints in the multi-camera tracking setting. The proposed technique has been tested using various objects and the experimental results has demonstrated its feasibility.

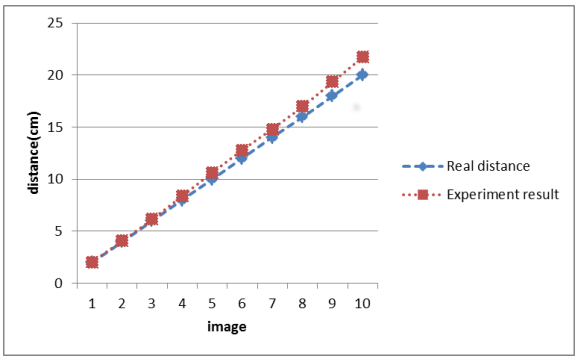

Fig. 6. Y-axis displacement accuracy.

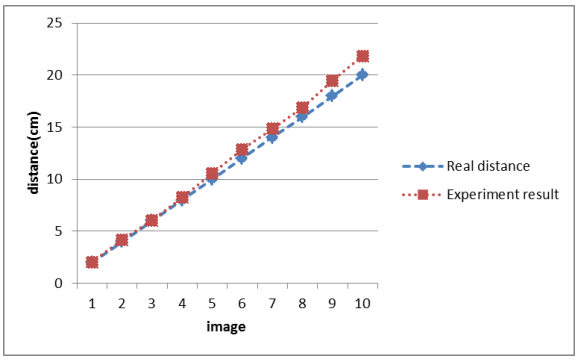

Fig. 7. Z-axis displacement accuracy.

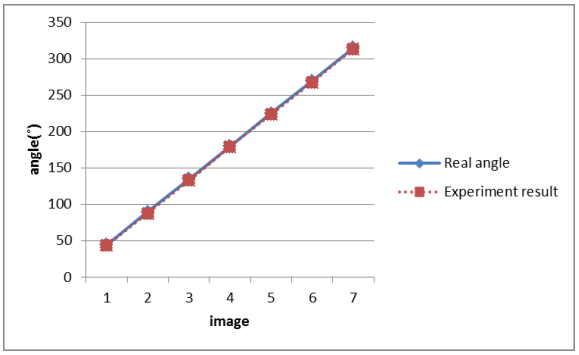

Fig. 8. Rotation accuracy in our experiments.

ACKNOWLEDGMENT

The support of this work in part by the National Science Council of Taiwan, R.O.C, under Grant NSC-99-2221-E-194-005-MY3 is gratefully acknowledged.

REFERENCES

[1] S. Boukir, P. Bouthemy, F. Chaumette, and D. Juvin, "A local method for contour matching and its parallel implementation," *Machine Vision and Applications*, vol. 10, no. 5-6, pp. 321–330, 1998.

[2] M. Vincze, "Robust tracking of ellipses at frame rate," *Pattern Recognition*, vol. 34, no. 2, pp. 487–498, 2001.

[3] G. D. Hager and K. Toyama, "X vision: A portable substrate for real-time vision applications," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 23–37, 1998.

[4] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *IJCAI*, vol. 81, 1981, pp. 674–679.

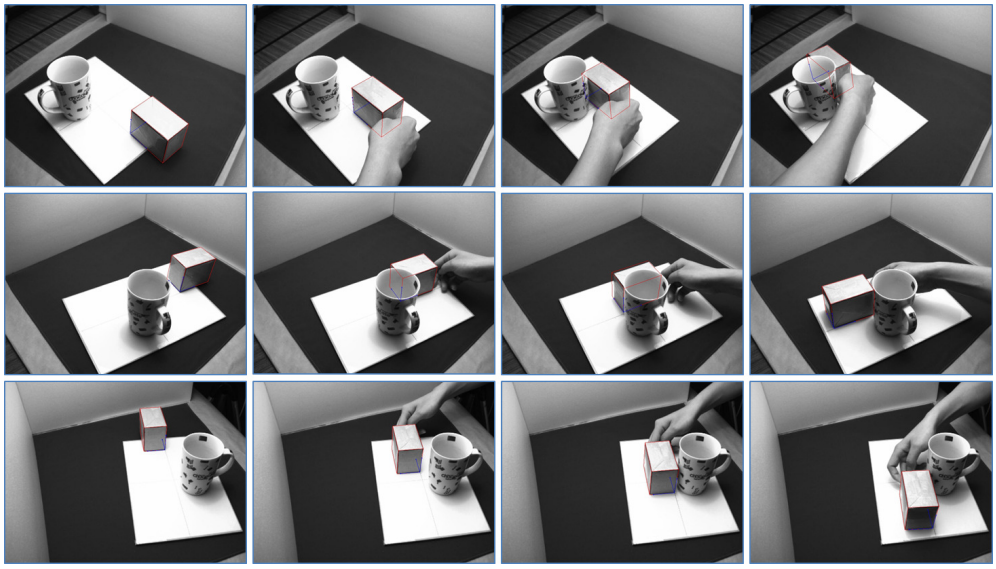

Fig. 9. Multi-view tracking with partial occlusion. Top, middle and bottom rows are the images captured by camera No. 1, No. 2 and No. 3, respectively.

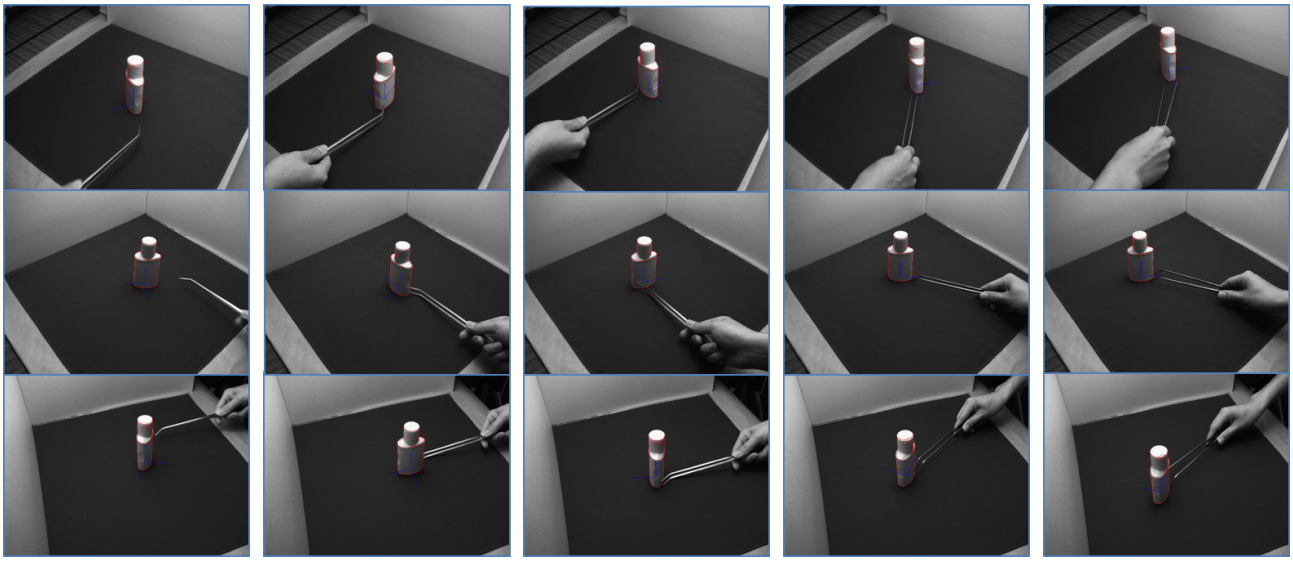

Fig. 10. Multi-view tracking with partial occlusion. Top, middle and bottom rows are the images captured by camera No. 1, No. 2 and No. 3, respectively.

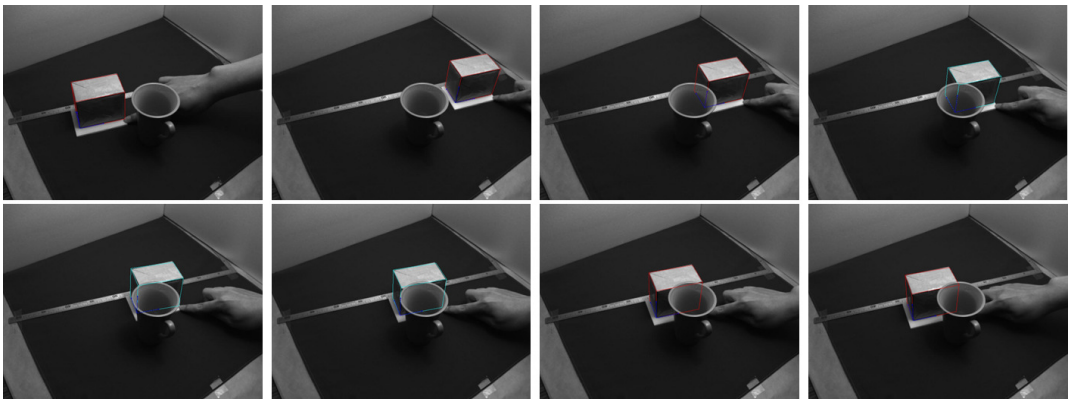

Fig. 11. A new pose is generated if severe occlusion occurs. The red pose obtained from the original tracking method, and the cyan pose is obtained using pose prediction method.

[5] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," in *Computer VisionECCV'96*. Springer, 1996, pp. 343–356.

[6] K. B. Yesin and B. J. Nelson, "Robust cad model based visual tracking for 3d microassembly using image space potentials," in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 2. IEEE, 2004, pp. 1868–1873.

[7] A. I. Comport, É. Marchand, and F. Chaumette, "Robust model-based tracking for robot vision," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2004, pp. 692–697.

[8] D. F. Dementhon and L. S. Davis, "Model-based object pose in 25 lines of code," *International journal of computer vision*, vol. 15, no. 1-2, pp. 123–141, 1995.

[9] D. Leng and W. Sun, "Contour-based iterative pose estimation of 3d rigid object," *IET computer vision*, vol. 5, no. 5, pp. 291–300, 2011.

[10] P. Azad, D. Munch, T. Asfour, and R. Dillmann, "6-dof model-based tracking of arbitrarily shaped 3d objects," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 5204–5209.

[11] M. Pressigout and E. Marchand, "Real-time hybrid tracking using edge and texture information," *The International Journal of Robotics Research*, vol. 26, no. 7, pp. 689–713, 2007.

[12] P. Bouthemy, "A maximum likelihood framework for determining moving edges," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 11, no. 5, pp. 499–511, 1989.

[13] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 3, pp. 313–326, 1992.

[14] J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.

[15] M. Piccardi, "Background subtraction techniques: a review," in *Systems, man and cybernetics, 2004 IEEE international conference on*, vol. 4. IEEE, 2004, pp. 3099–3104.

[16] R. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *Robotics and Automation, IEEE Journal of*, vol. 3, no. 4, pp. 323–344, 1987.