Implementation of Multiple View Approach for Pose Estimation with an Eye-In-Hand Robotic System

Kai Li, Chungang Zhuang, Jianhua Wu, and Zhenhua Xiong(✉)

State Key Laboratory of Mechanical System and Vibration,
School of Mechanical Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China
{leica8244,cgzhuang,wujh,mexiong}@sjtu.edu.cn

Abstract. This paper compares implementation of multiple view approach of pose estimation on an eye-in-hand robotic system. By combining RGB-D frames from multiple view with the eye-in-hand robotic system, geometry information of target objects can be best recovered thus pose estimation performance can get improved. Two primary approaches for pose estimation, namely 3D point cloud registration and 2D image matching are implemented and compared. For the 3D method, we reconstruct target objects by taking advantage of the eye-in-hand system to get an accurate representation of target objects. For the 2D method, we discuss distance metrics and regression for 6DOF pose and apply RANSAC with it to fuse multiple estimation results. State-of-the-art pose estimation algorithms which cover both the 3D and 2D approaches are implemented and compared. Experiments show that the multiple view approach can provide more accurate and reliable pose estimation results when compared with conventional single view approach.

Keywords: Multiple view · Eye-in-hand system · Pose estimation · Object reconstruction · RANSAC

1 Introduction

Accurate localization and pose estimation of 3D objects play an essential role in many robotic applications such as object grasping and random bin picking. When an eye-in-hand system is set up by mounting the camera on the end effector of a robot manipulator, observations can be made from multiple view, which provides additional advantages for pose estimation, as information from different views can be used together to form a more accurate result. Although recent work has started to apply the multiple view approach [1,2], detailed investigations and discussions are still needed on how to perform pose estimation effectively with multiple view approach and the eye-in-hand system.

In computer vision, 3D point cloud registration and 2D image matching are two primary approaches for pose estimation. Reference [1,2] applies the multiple view approach to 3D point cloud based method by reconstructing the scene

Y. Huang et al. (Eds.): ICIRA 2017, Part II, LNAI 10463, pp. 599–610, 2017.
DOI: 10.1007/978-3-319-65292-4_52

with depth data taken from different views. Reference [1] uses 3 mature point cloud registration algorithms to reconstruct the scene and applies superellipsoid-based pose estimation [3] on sweet pepper. Reference [2] takes advantage of robot encoder data to align each frame of depth data and implements principle component analysis (PCA) and iterative closest point (ICP) [4] for pose estimation. However, reference [2] also mentions that due to poor accuracy of eye-hand calibration, the reconstruction process may result in gross errors sometimes. As many implementation details are not given in [2], discussions are still needed to solve potential problems in practice. Except for superellipsoid and PCA, point pair feature (PPF) [5] is another state-of-the-art method for pose estimation on 3D point cloud. Reference [6,7] extend PPF by adding boundary or color information. Although PPF has been proved to be a robust way for pose estimation, more accessible RGB-D camera can't provide high quality depth data like the 3D scanner used in [6]. Therefore when it comes to RGB-D depth data with no color extension, scene object reconstruction with multiple view approach can be a possible solution.

2D image matching is another primary approach for pose estimation. For objects with rich color variation, local image features have shown great success in recovering the 6DOF pose [8], while for objects with little texture, template matching is a frequently used method. Reference [9] applies chamfer matching to pose estimation by using a multi-flash camera to extract depth edges of textureless objects. The state-of-the-art LINEMOD/LINE-2D/LINE-3D [10], is another well-cited template matching method for pose estimation. However, all the above work takes only one frame into account and no attempt has been made to use the multiple view approach.

In this paper we investigate and compare the multiple view approach for pose estimation with a RGB-D camera and the eye-in-hand robotic system. The multiple view approach is applied on both 3D and 2D pose estimation methods. For the 3D method, we revisit the encoder-data based scene object reconstruction and choose to apply PCA and PPF on the reconstructed target objects for pose estimation. Implementation details and problems encountered in practice for both scene reconstruction and pose estimation are discussed. For the 2D method, we perform pose estimation from multiple view and use RANSAC [11] for robust regression on the multiple estimation results. Metrics for 6DOF pose on $SE(3)$ are also discussed for its importance in RANSAC. State-of-the-art 2D method LINE-2D is chosen to apply for its high matching speed. Experiments and comparisons show that our multiple view approach can overcome problems of single view approach and yield more reliable pose estimation results.

The outline of this paper is as follows, Sect. 2 introduces the overall workflow of the multiple view approach for both 3D and 2D methods. Section 3 discusses specific pose estimation algorithms and details for their implementations. Section 4 shows the experiment results for both 3D and 2D methods. Finally, Sect. 5 gives conclusion for this paper.

2 Multiple View Approach for Pose Estimation

2.1 3D Point Cloud Method

3D point cloud registration is a primary approach for pose estimation. 6DOF pose of the target object can be recovered by aligning a prior known object model and the target object in the scene. Geometry features of the target objects may not be well-recovered when viewed from one single angle as only partial surfaces can be seen. Moreover, depth data generated from a low-cost RGB-D camera contains large amount of noises and loses many detailed features. This obviously brings negative effects for point cloud registration. The eye-in-hand system can provide a solution to this problem by observing the scene from multiple views and perform scene-reconstruction before pose estimation.

Typical way of 3D reconstruction such as RGB-D SLAM [12] estimates camera poses through image features in multiple frames. When it comes to an eye-in-hand robotic system, camera pose can be easily extracted as the joint encoder data and forward kinematics provide the end effector pose, and eye-hand calibration is done before the pose estimation task. Reference [2] notices this point, yet they also mention that poor accuracy of eye-hand-calibration may result in huge errors for registration. In order to overcome this problem, they teach the robot to observe the scene from several fixed angles and calibrate each camera pose following the standard RGB-D SLAM [12] routine. If the object container is moved or the experiment environment is changed, re-calibration for camera pose is needed, which involves a lot of extra work.

In our approach, we reconstruct the scene object by taking advantage of the robot data and eye-hand calibration result. We teach the robot to perform a pre-scanning motion along a fixed trajectory, during which depth data is taken and aligned. In order to overcome misalignment due to vibrations of the end effector during scanning motion, communication delay between the PC and robot controller, or poor accuracy for eye-hand calibration, we add a registration refinement by ICP.

Although ICP has been extensively used for registration refinement, different implementation procedure may have huge effects on the final registration result. In our approach, target point cloud for ICP is chosen to be the reconstructed scene object and source point cloud the newly-captured depth data. Instead of setting a fixed distance threshold for point pair matches, we set an inlier ratio to make the registration robust to false matches. Statistical outlier removal is applied after each frame is aligned to remove point cloud noises. Voxel grid filtering is also applied to the reconstructed point cloud to make it have an even distribution of points, which avoids ICP local minimum in high density area. All the above procedure overcomes adverse effects from noises, outliers and local minimum and makes best use of ICP for scene reconstruction.

Another important part of the pre-processing is object segmentation. For simple registration method such as PCA, object segmentation is a must as PCA is very vulnerable to outliers and missing data. Although robust registration methods like PPF have been proposed, segmentation still plays an important role for improving performance in terms of both accuracy and run-time speed. While reference [1] uses a reconstruction-then-segmentation approach, we segment the object on each frame and only take object points into account to speed up the registration process. Reference [2] uses a pre-trained fully convolutional network to segment target objects from the scene, which requires large amount of training data and GPU-support. In our approach, we segment target objects from the scene through pixel intensity directly, as the objects and backgrounds have obvious color contrast. Figure 1 shows the overall workflow of the 3D multiple view approach and the experimental setup.

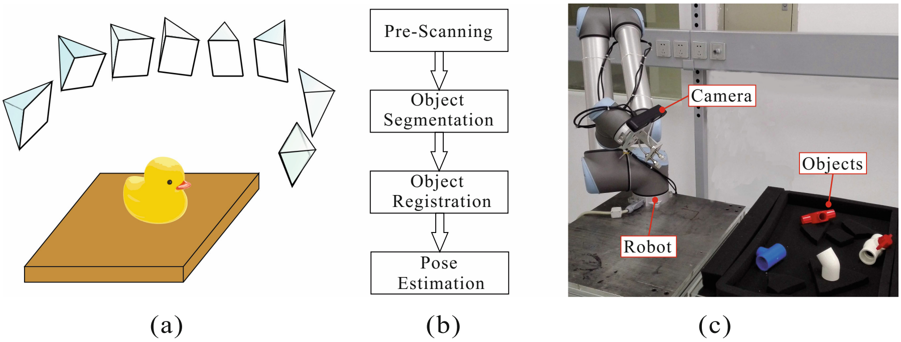


Fig. 1. (a) Multiple view approach. (b) Overall workflow for the 3D method. (c) Experiment setup.

2.2 2D Image Matching Method

2D image matching is another primary approach for pose estimation. As mentioned in Sect. 1, previous work on 2D pose estimation focuses on how to extract object pose from one single frame. In our approach, we take advantage of the eye-in-hand system and perform 2D pose estimation from multiple view. RANSAC is applied to the multiple estimation results for outlier removal and pose regression.

RANSAC has been well-adopted in computer vision for robust regression. Typical applications of RANSAC such as finding homography between two images uses l_2 norm to measure pixel distances. However, when extended for robust regression on 6DOF poses, l_2 norm would not work as the pose no longer belongs to the vector space $\mathbb{R}^6$. Here we give a discussion about metrics and regression for 6DOF pose on the $SE(3)$ group.

A 6DOF pose on $SE(3)$ contains a rotation part $\mathbf{R} \in SO(3)$ and a translation part $\mathbf{t} \in \mathbb{R}^3$. Distance metric on the l_2 norm sense for two translations $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{R}^3$ can be represented by

$$d_{\mathbf{t}}(\mathbf{t}_1, \mathbf{t}_2) = \|\mathbf{t}_1 - \mathbf{t}_2\|_2 \tag{1}$$

Accordingly, weighted regression on the l_2 norm least square sense for m given translations $\mathbf{t}_1, \ldots, \mathbf{t}_\mathrm{m}$ can be given by

$$\mathbf{t}_\mathrm{a} = \frac{1}{m}\sum_{i=1}^{m} w_\mathrm{i}\mathbf{t}_\mathrm{i} \tag{2}$$

As for the $SO(3)$ rotation part, many different metrics exist and we adopt the Riemannian distance, namely the length of the geodesics connecting $\mathbf{R}_1$ and $\mathbf{R}_2$.

$$d_{\mathbf{R}}(\mathbf{R}_1, \mathbf{R}_2) = \frac{1}{\sqrt{2}}\|\log(\mathbf{R}_1^\mathrm{T}\mathbf{R}_2)\|_\mathrm{F} \tag{3}$$

where

$$\log(\mathbf{R}) = \begin{cases} \mathbf{0} & \mathit{if}\ \theta(\mathbf{R}) = 0 \\ \frac{1}{2}\frac{\theta(\mathbf{R})}{\sin\theta(\mathbf{R})}(\mathbf{R} - \mathbf{R}^\mathrm{T}) & \mathit{otherwise} \end{cases} \tag{4}$$

$$\theta(\mathbf{R}) = \cos^{\text{-}1}(\frac{\mathrm{trace}(\mathbf{R}) - 1}{2}) \tag{5}$$

Accordingly, weighted regression on the Riemannian sense for m given rotations $\mathbf{R}_1, \ldots, \mathbf{R}_\mathrm{m}$ can be given by

$$\mathbf{R}_\mathrm{a} = \underset{\mathbf{R}\in SO(3)}{\arg\min} \sum_{i=1}^{m} w_\mathrm{i}||\log(\mathbf{R}_\mathrm{i}^\mathrm{T}\mathbf{R})||_\mathrm{F}^2 \tag{6}$$

No close form solution for (6) exists and we solve it with an iterative method. First we initialize $\mathbf{R}_\mathrm{a}$ as

$$\mathbf{R}_\mathrm{a} = \sum_{i=1}^{m} w_\mathrm{i}\log(\mathbf{R}_\mathrm{i}) \tag{7}$$

Then we compute the average on the tangent space of $\mathbf{R}_\mathrm{a}$

$$\mathbf{r} = \sum_{i=1}^{m} w_\mathrm{i}\log(\mathbf{R}_\mathrm{a}^\mathrm{T}\mathbf{R}_\mathrm{i}) \tag{8}$$

Finally we update $\mathbf{R}_\mathrm{a}$ by moving it towards $\mathbf{r}$.

$$\mathbf{R}_\mathrm{a} \longleftarrow \mathbf{R}_\mathrm{a}e^{\mathbf{r}} \tag{9}$$

By iteratively solving (8) and (9), $\mathbf{R}_\mathrm{a}$ gets converged and result of (6) can be extracted. It is noteworthy that in order to make it easy to implement, approximation for (6) can be represented on the Euclidean sense by orthogonal projection of $\overline{\mathbf{R}_\mathrm{t}} = \frac{1}{m}\sum_{i=1}^{m} w_\mathrm{i}\mathbf{R}_\mathrm{i}^\mathrm{T} = \mathbf{U}\mathit{\Sigma}\mathbf{V}^\mathrm{T}$ onto $SO(3)$

$$\mathbf{R}_\mathrm{a} = \begin{cases} \mathbf{VU}^\mathrm{T} & \mathit{if}\,\det(\overline{\mathbf{R}_\mathrm{t}}) > 0 \\ \mathbf{VHU}^\mathrm{T} & \mathit{otherwise} \end{cases} \tag{10}$$

$$\mathit{where}\ \mathbf{H} = \mathit{diag}[1, 1, -1] \tag{11}$$

Detailed description about the mathematical background of motion groups can be found in [13].

3 Implementations of Pose Estimation

The previous section introduces overall workflow for the multiple view pose estimation approach for 3D and 2D methods. In this section we discuss implementations for specific pose estimation algorithms which covers both the 3D and 2D methods (Fig. 2).

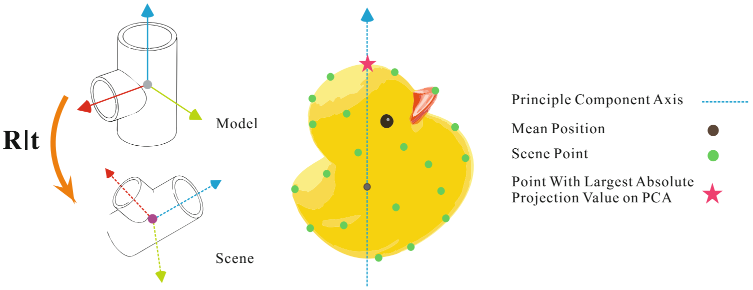


Fig. 2. Coarse registration can be achieved by aligning each corresponding principle component and mean position. Principle component sign uncertainty can be avoided by using a projection selection method.

For the 3D method, we first choose to implement PCA on the segmented scene object for coarse registration and add ICP as refinement. Despite being vulnerable to outliers, PCA is an effective and straightforward way for coarse registration. Although similar idea has been used in [2], a hidden problem which is not discussed is that PCA can't determine the sign of each principle component, therefore the result orientation might rotate 180° along certain axis. In [2]'s experiment environment, most target objects are symmetric along certain axis thus the sign ambiguity is not a big issue. However a general solution is needed to make sure that PCA can be applied to objects of all kinds of shapes. Here we provide a solution as follows. After PCA is applied for a given point cloud set, we traverse each mean-normalized point and calculate its projection on each principle component axis. For all projections on one axis, we find the point with the largest projection absolute value and make the axis direction pointing towards it. Workflow of this solution is shown in Algorithm 1. Note that when Algorithm 1 returns -1, it means the point set has equal projections on both directions of the input principle component. Result of Algorithm 1 and the corresponding rotation part of PCA pose estimation are shown in Table 1.

Except for PCA, PPF is another state-of-the-art pose estimation method focusing on 3D point cloud. Although reference [5,6] have demonstrated PPF's capability of dealing with heavily clustered environment, in practice we find that PPF can't give reliable estimation results using one single frame of RGB-D depth image. One obvious reason is that single depth image captured from a RGB-D camera can't provide enough geometry information for target objects, which can be improved by the multiple view scene-object-reconstruction approach. Another

Algorithm 1. Determine Sign of PCA

```
Input: $\mathcal{P} = \{\mathbf{p_1}, \mathbf{p_2}, \mathbf{p_3}, ...\mathbf{p_m}\}, \mathbf{r}$
Output: $\mathbf{r}$
function GetSignPCA($\mathbf{r}$,$\mathcal{P}$)
    $\mathcal{H} \leftarrow \{\phi\}$
    for all $\mathbf{p_i} \in \mathcal{P}$ do
        $\mathcal{H} \leftarrow \mathbf{p_i} \cdot \mathbf{r}$
    end for
    $projMax \leftarrow$ FindMax($\mathcal{H}$)
    $projMin \leftarrow$ FindMin($\mathcal{H}$)
    if $projMax < (-1) * projMin$ then
        $\mathbf{r} \leftarrow \ \mathbf{r} * (-1)$
    else
        if $projMax = (-1) * projMin$ then
            return $-1$
        end if
    end if
    return $\mathbf{r}$
end function
```

Table 1. Rotation matrix result of PCA

GetSignPCA($\mathrm{r_x}, \mathcal{P}$)	GetSignPCA($\mathrm{r_y}, \mathcal{P}$)	GetSignPCA($\mathrm{r_z}, \mathcal{P}$)	R
else	else	else	$(\mathbf{r_x}, \mathbf{r_y}, \mathbf{r_x} \times \mathbf{r_y})$
-1	else	else	$(\mathbf{r_y} \times \mathbf{r_z}, \mathbf{r_y}, \mathbf{r_z})$
else	-1	else	$(\mathbf{r_x}, \mathbf{r_z} \times \mathbf{r_x}, \mathbf{r_z})$
else	else	-1	$(\mathbf{r_x}, \mathbf{r_y}, \mathbf{r_x} \times \mathbf{r_y})$
else	-1	-1	$(\mathbf{r_x}, \mathbf{r_y}, \mathbf{r_x} \times \mathbf{r_y})$
-1	else	-1	$(\mathbf{r_x}, \mathbf{r_y}, \mathbf{r_x} \times \mathbf{r_y})$
-1	-1	else	$(\mathbf{r_y} \times \mathbf{r_z}, \mathbf{r_y}, \mathbf{r_z})$
-1	-1	-1	$(\mathbf{r_x}, \mathbf{r_y}, \mathbf{r_x} \times \mathbf{r_y})$

reason is due to the uncertainty of surface normal sign. Surface normal of the point cloud serves as one dimension of PPF. The typical way of determining the sign of the normal vector for a given point $\mathbf{p_i}$ is by setting a vector $\mathbf{r} = \mathbf{p_i} - \mathbf{p_r}$ pointing from a given reference point $\mathbf{p_r}$ to $\mathbf{p_i}$. Then the sign of the normal vector can be determined in the way that its projection on $\mathbf{r}$ is positive. This can't ensure that each corresponding point on the object model and the scene have the same normal sign, as the reference point is randomly selected. In our approach, the reference point is set at the mean position for both the prior known model and the segmented scene object, which helps to overcome the problem of normal sign uncertainty. Also, pre-segmentation reduces the number of points needed to be matched thus speeds up the pose estimation process (Fig. 3).

For 2D image matching method, we choose to implement LINE-2D, which is the 2D version of LINEMOD [10] and has been extensively used in single frame pose estimation. As the matching process of LINE-2D is smartly designed, high

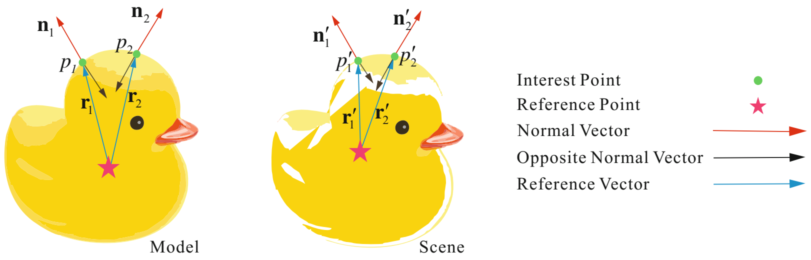


Fig. 3. Reference points for both the scene object and the model object are set at the mean position. Although positions of the two reference points are slightly different due to scene point cloud corruptions, normal sign remains consistent, both pointing from the inner to the outer.

processing speed can be achieved, making LINE-2D particularly suitable for real-time multiple view pose estimation. LINE-2D uses color gradient as features for matching. When viewed from certain angles, the object may contain very few features, resulting in unreliable estimation result. Multiple view approach for LINE-2D is as follows. Like the previous process, the robot first takes a scanning motion along a predefined route. Meanwhile, matching by LINE-2D is constantly running and a series of estimation results is extracted from frames with high matching score. Finally, all results with high confidence are fused with RANSAC to form an accurate final result (Fig. 4).

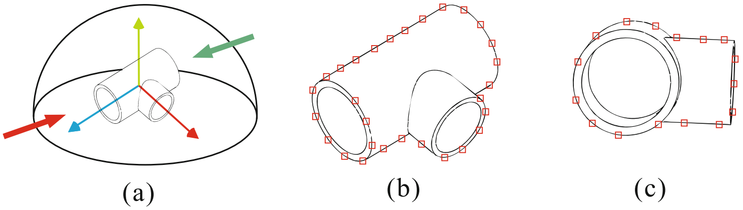


Fig. 4. (a) View angles of the target object. (b) View angle with rich selected features. Red rectangle represents color gradient features. (c) View angle with few selected features. (Color figure online)

4 Experiments

In this section, we test the above approach on four kinds of objects and evaluate pose estimation performance for the multiple view approach. Point cloud models of all objects are generated from the open source software Blender (Fig. 5).

We first test the overall performance of the proposed multiple view approach for PCA, PPF and LINE-2D. For PCA and PPF, 15 frames of color and depth images are recorded during the scanning motion to reconstruct scene objects. For LINE-2D, matching similarity threshold is set to 80% and 140 frames of estimation results are fused together for RANSAC.

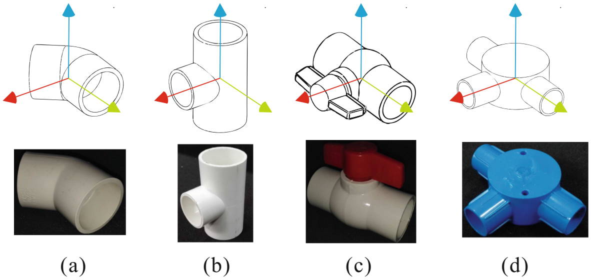
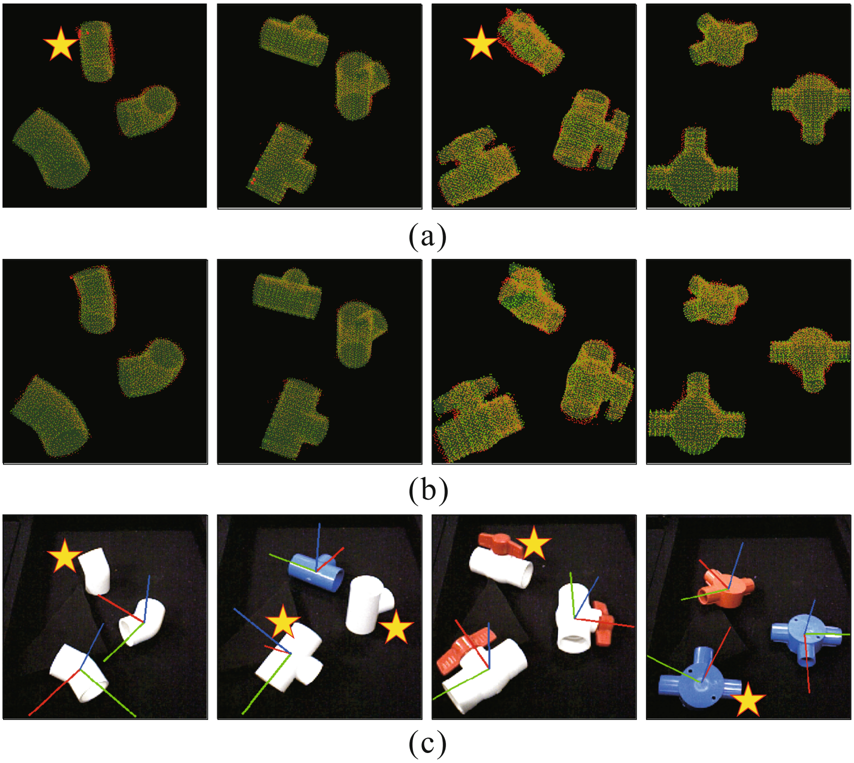


Fig. 5. Four types of objects used in our experiment. (a) 2-hole PVC pipe. (b) 3-hole PVC pipe. (c) PVC valve. (d) 3-hole PVC box

Figure 6 shows a typical experiment result on the 4 objects. As the ground truth pose of the target object is unknown, we follow the error evaluation method in [6] by repeating pose estimation process and choosing absolute deviation from the results median as the error metric. We first run our multiple view approach for 100 times for each tested object and get 100 estimation results. Next we calculate each result's deviation from the results median. Pose deviation metric is defined as discussed in Sect. 2. Histograms of pose estimation deviation are shown in Fig. 7. Note that the position metric is in meter and the rotation metric is in arbitary unit. Average deviation values are shown in Table 2.

From the histograms we can see that deviations from the estimation result median is generally reduced by the multiple view approach. For PCA, single view estimation result shows large deviation and the multiple view approach greatly helps to improve estimation accuracy. For PPF, which is robust to noise and missing data, performance on rotation is generally the same for both approaches, and the multiple view approach help to improve position estimation accuracy to some extent. For LINE-2D, the multiple view approach helps to narrow the deviation from result median for both rotation and position. Numeral comparison among the 3 algorithms shows that repeated experimental results of LINE-2D have the least variance among the 3 tested algorithms, which means that the LINE-2D's performance is the most stable one in terms of repeated accuracy. However, graphical display for the estimation results in Fig. 6 shows that LINE-2D also has the most failure cases. This is due to the fact that template library for LINE-2D can hardly cover all possible view angles and objects with extreme poses may be out of the matching scope. On the other hand, 3D methods are free of this problem and able to handle any object poses. For PCA which is vulnerable to outliers, misalignment case with opposite orientation still exists, as noise points may disturb the projection value on each principle axis and affect the final estimation result as a whole. PPF shows more stable and robust performance with no failure cases when compared with PCA. In terms of robotic picking task, PPF is the best choice for its reliability and robustness.

The experiment is implemented on a PC with Intel Core i5-4570 CPU and 12 GB RAM. Open source library OpenCV and PCL are used for image and

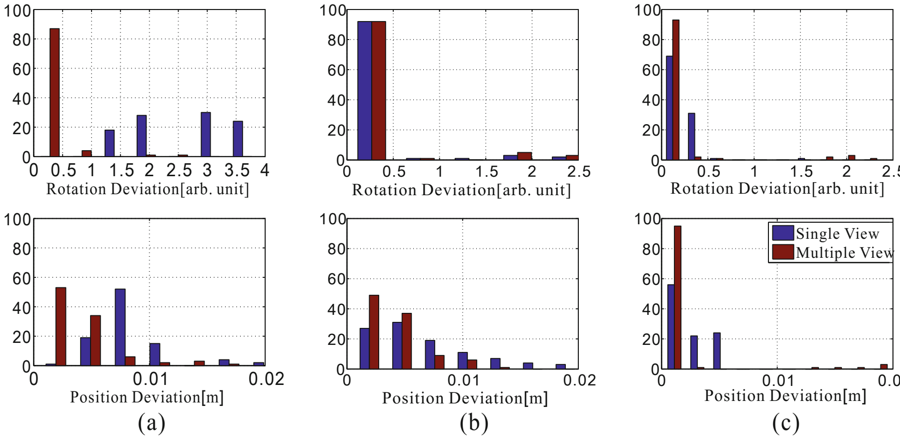


(a)

(b)

(c)

Fig. 6. Experiment result of the propsed approach. (a) PCA. (b) PPF. (c) LINE-2D. Failure cases with gross error or invalid matches are marked with yellow stars. (Color figure online)



(a)

(b)

(c)

Fig. 7. Histograms of deviation from estimation results median. (a) PCA. (b) PPF. (c) LINE-2D.

point cloud processing and visualizing. We use open source code of PPF and LINE-2D provided by OpenCV. The experiment setup consists of a 6-DOF robot manipulator from Universal Robots and an ASUS Xtion Pro RGB-D camera. The runtime speeds for single frame pose estimation are as follows: 50 ms for PCA+ICP, 1500 ms for PPF+ICP and 100 ms for LINE-2D. Considering the pre-scanning motion, it takes 15 to 20 s for the whole process.

Table 2. Average deviation from the result median

DOF type	PCA	PPF	LINE-2D
Position[a](Multiple/Single)	**0.0042**/0.0100	**0.0044**/0.0069	**0.0017**/0.0024
Rotation[b](Multiple/Single)	**0.4192**/2.5421	**0.3305**/0.3106	**0.1735**/0.1857

[a] Metric follows (1), in meter.

[b] Metric follows (3), arbitrary unit.

5 Conclusions

In this paper, we discuss implementations of multiple view approach for pose estimation with an eye-in-hand robotic system. We apply the multiple view approach on both the 3D and 2D pose estimation methods. For the 3D method, we revisit the reconstruction-then-estimation scheme and take advantage of the eye-in-hand robotic system for camera pose extraction. For the 2D method, we adopt RANSAC for outlier removal and results regression. Metrics and regression are discussed in detail for the 6DOF pose. We choose to implement PCA, PPF and LINE-2D for pose estimation, which covers both the 3D and 2D approach. Experiment results show that the multiple view approach can improve pose estimation performance for both the 3D and 2D methods. Comparison among the 3 implemented methods also show that PPF has the most reliable performance.

Acknowledgements. This research was supported in part by National Natural Science Foundation of China under Grant 51675325 and National Science and Technology Support Program under Grant 2015BAF01B02.

References

1. Lehnert, C., Sa, I., McCool, C., Upcroft, B., Perez, T.: Sweet pepper pose detection and grasping for automated crop harvesting. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 2428–2434. IEEE (2016)
2. Zeng, A., Yu, K.T., Song, S., Suo, D., Walker Jr., E., Rodriguez, A., Xiao, J.: Multi-view self-supervised deep learning for 6D pose estimation in the Amazon picking challenge. arXiv preprint arXiv:1609.09475 (2016)
3. Duncan, K., Sarkar, S., Alqasemi, R., Dubey, R.: Multi-scale superquadric fitting for efficient shape and pose recovery of unknown objects. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 4238–4243. IEEE (2013)

4. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: Robotics-DL Tentative, pp. 586–606. International Society for Optics and Photonics (1992)
5. Drost, B., Ulrich, M., Navab, N., Ilic, S.: Model globally, match locally: efficient and robust 3D object recognition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 998–1005. IEEE (2010)
6. Choi, C., Taguchi, Y., Tuzel, O., Liu, M.Y., Ramalingam, S.: Voting-based pose estimation for robotic assembly using a 3D sensor. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 1724–1731. IEEE (2012)
7. Choi, C., Christensen, H.I.: RGB-D object pose estimation in unstructured environments. Robot. Autonom. Syst. **75**, 595–613 (2016)
8. Collet, A., Berenson, D., Srinivasa, S.S., Ferguson, D.: Object recognition and full pose registration from a single image for robotic manipulation. In: 2009 IEEE International Conference on Robotics and Automation (ICRA), pp. 48–55. IEEE (2009)
9. Liu, M.Y., Tuzel, O., Veeraraghavan, A., Taguchi, Y., Marks, T.K., Chellappa, R.: Fast object localization and pose estimation in heavy clutter for robotic bin picking. Int. J. Robot. Res. **31**(8), 951–973 (2012)
10. Hinterstoisser, S., Cagniart, C., Ilic, S., Sturm, P., Navab, N., Fua, P., Lepetit, V.: Gradient response maps for real-time detection of textureless objects. IEEE Trans. Pattern Anal. Mach. Intell. **34**(5), 876–888 (2012)
11. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM **24**(6), 381–395 (1981)
12. Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., Burgard, W.: An evaluation of the RGB-D slam system. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 1691–1696. IEEE (2012)
13. Wang, Y., Chirikjian, G.S.: Nonparametric second-order theory of error propagation on motion groups. Int. J. Robot. Res. **27**(11–12), 1258–1273 (2008)