Sensors and Materials, Vol. 32, No. 4 (2020) 1171–1183
MYU Tokyo

S & M 2169

Manipulated-object Detection and Pose Estimation Based on Multimodal Feature Points and Neighboring Patches

Xunwei Tong, Ruifeng Li,* Lijun Zhao, Lianzheng Ge, and Ke Wang

State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, Harbin 150001, China

(Received July 31, 2019; accepted February 7, 2020)

Keywords: object detection, pose estimation, multimodal feature point, local patch, manipulated object

The detection and pose estimation of objects in human demonstrations remain challenging yet crucial tasks. The increasing availability of red-green-blue and depth sensors makes it possible to synthetize local features of color and three-dimensional (3D) geometry, which are useful for processing a wider range of objects. However, existing methods fail to combine the inherent advantages of these two features. Moreover, pose refinement methods based on whole point clouds are often affected by occlusion and background noise. In this paper, feature points of the speeded-up robust feature and the fast point feature histogram were transformed into the same 3D space. After matching them separately, multimodal feature points were jointly used to estimate a coarse pose. Subsequently, the coarse pose was refined by aligning point clouds composed of feature points' neighboring patches. During the iterative closest point process, we selected corresponding points in matched local patches. In our first and second comparative experiments, F1 scores were respectively increased by 0.1349 and 0.1633, which verified the validity of our method. Finally, the third qualitative experiment showed that the proposed method is applicable to manipulated-object detection and pose estimation.

1. Introduction

As robotic tasks become increasingly flexible and diverse, robots are required to learn new tasks more easily and quickly. To learn manipulation tasks from observing the motion executed by human demonstrators, methods of learning from demonstration (LfD) have been widely used.[1–3] Since manipulated objects are key components of daily tasks, object detection and pose estimation are hot issues in understanding human manipulation.

In recent years, methods based on convolutional neural networks (CNNs) have shown significant advantages in tasks of object detection and pose estimation, which benefit from their ability to automatically learn features from raw images.[4–6] However, for new user-defined objects, hundreds of new samples together with their ground-truth poses are needed to retrain CNNs. This is usually very inconvenient in practical application scenarios, where various and changing objects are commonly involved.

*Corresponding author: e-mail: lrf100@hit.edu.cn
https://doi.org/10.18494/SAM.2020.2539

ISSN 0914-4935
https://myukk.org/

Other methods of object detection and pose estimation are mainly based on global templates or local feature points. In global-template-based methods, holistic templates of objects are sampled from many viewpoints and matched against each sliding window in the testing scene according to their global features.[7–9] However, objects in manipulation scenes are usually occluded by other objects or hands, making it impossible to fully extract global information. Xie *et al.*[10] projected local features back onto a three-dimensional (3D) point cloud and estimated the object pose using the random sampling consensus (RANSAC)[11] algorithm. This method is robust to occlusions since global information is not required. Therefore, we focused on local-feature-based approaches in this study.

Depending on the modality of the information represented by features, color features can be extracted from color images and geometric features can be extracted from point clouds. For instance, Collet *et al.*[12] detected multiple objects with scale-invariant feature transform (SIFT)[13] feature points extracted from RGB images, although this method is not suitable for objects with less texture. Vidal *et al.*[14] presented an object detection method based on the variations in point pair features (PPFs), which are not suitable for symmetrical yet textured objects. Since multimodal information can describe complementary properties of objects, it is natural to synthesize color and 3D features to deal with a wider range of objects. Tsai and Tsai[15] used the color-signature of histogram of orientation (CSHOT)[16] feature for object detection and pose estimation. This feature connects signatures of histograms in the spatial and color channels as a high-dimensional descriptor. However, the feature points were the results of downsampling the input point cloud,[15] rather than keypoints with invariance and repeatability. Another strategy is to determine the tangent plane of a local spatial surface and then extract color features.[17,18] These methods reduce the resolution of color images and leave 3D features unused. Motivated by these issues, a new method of object detection and pose estimation by mapping multimodal feature points into the same 3D space was proposed in this study.

Pose estimation based on feature point matching usually has medium accuracy. More importantly, the estimation method will fail when there are fewer than four pairs of matching points. Traditional approaches introduced a pose refinement process, making use of two entire point clouds.[19,20] However, it may contain many interference points belonging to the foreground occlusion and background regions. To address this problem, in this study, neighboring patches of feature points were used for pose refinement.

In summary, our contribution consists of two aspects. (1) We used multimodal features more effectively than before in tasks of object detection and pose estimation. (2) An accurate pose refinement method using feature points' neighboring patches was proposed.

The remaining part of this paper is organized as follows. In Sect. 2, datasets of objects, which contain multimodal feature points and their neighboring patches, are constructed. Then, the methods of object detection and pose estimation adopted in this paper are described fully in Sect. 3. Experiments are reported in Sect. 4, with conclusions given in Sect. 5.

2. Dataset Construction

The key to detecting known objects is to build datasets in advance. Our datasets were mainly composed of multimodal feature points and their neighboring patches.

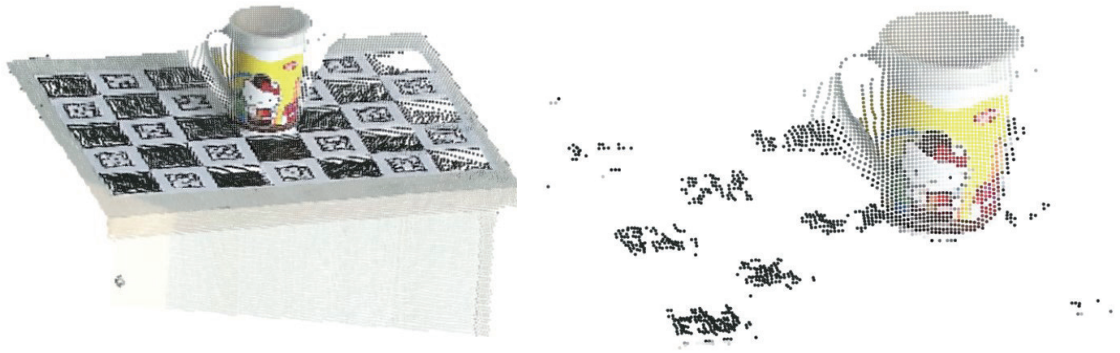
2.1 Segmentation of object on desktop

A scene point cloud was captured by a Microsoft Kinect 2.0 sensor, which is a widely used red-green-blue and depth (RGB-D) camera. The cuboid region near the desktop shown in Fig. 1(a) was selected as the region of interest (ROI).

Using the RANSAC plane fitting method, the maximum plane in the ROI was extracted as the desktop within an allowable error of 0.005 m. Points above the desktop were roughly extracted, as shown in Fig. 1(b). Since there were noisy points around the target object, we clustered the resulting points and retained the largest cluster.

Furthermore, to remove noisy points from the remaining point cloud, the average distance to the four nearest neighbors was calculated for each remaining point. A point was removed as an outlier when the average distance was greater than 0.005 m. In this way, the number of 3D points was reduced significantly without losing too much information, which was conducive to efficient computation. The denoised point cloud of the object is shown in Fig. 2(a).

Owing to the advantages of Kinect, it was convenient to project the object's point cloud onto the RGB image. Through a morphological close operation, the object's RGB mask was obtained and is shown in Fig. 2(b).



(a) (b)

Fig. 1. (Color online) Point clouds (a) within the ROI and (b) above the desktop.

(a) (b)

Fig. 2. (Color online) (a) Denoised point cloud and (b) RGB mask of the object.

2.2 Extraction of multimodal features from single view

For objects with rich texture, the speeded-up robust feature (SURF)[21] is usually used for object detection and pose estimation owing to its stability and reliability. On the basis of the object's RGB mask, feature points and descriptors of SURF were respectively detected and calculated. Using the parameters of Kinect, the 3D location of each SURF feature point was subsequently obtained.

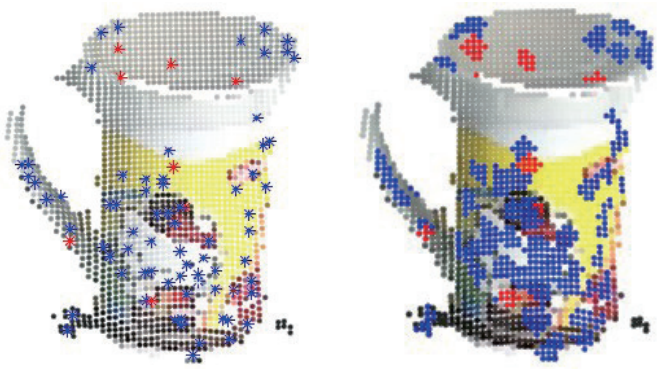
From the object's point cloud, the intrinsic shape signature 3D (ISS3D)[22] keypoints and the fast point feature histogram (FPFH)[23] were respectively detected and calculated. The ISS3D keypoint has an intrinsic reference frame enabling view-invariant feature extraction and fast pose registration. The FPFH is a robust spatial feature that describes the local geometry around the keypoint.

Keypoints of SURF (marked with blue asterisks) and ISS3D (marked with red asterisks) were simultaneously plotted within the object's point cloud, as shown in Fig. 3(a). For each keypoint, a local patch within the globular neighborhood was extracted and denoted as P_{patch}. 3D locations of points in P_{patch} were measured in the reference frame of the current camera and denoted as L_{patch}. Local patches of SURF (colored in blue) and ISS3D (colored in red) were plotted within the object's point cloud, as shown in Fig. 3(b).

For each view of the object, the whole point cloud was stored for reconstruction. Together with the descriptor and 3D location of each feature point, the name of the associated object and L_{patch} were stored.

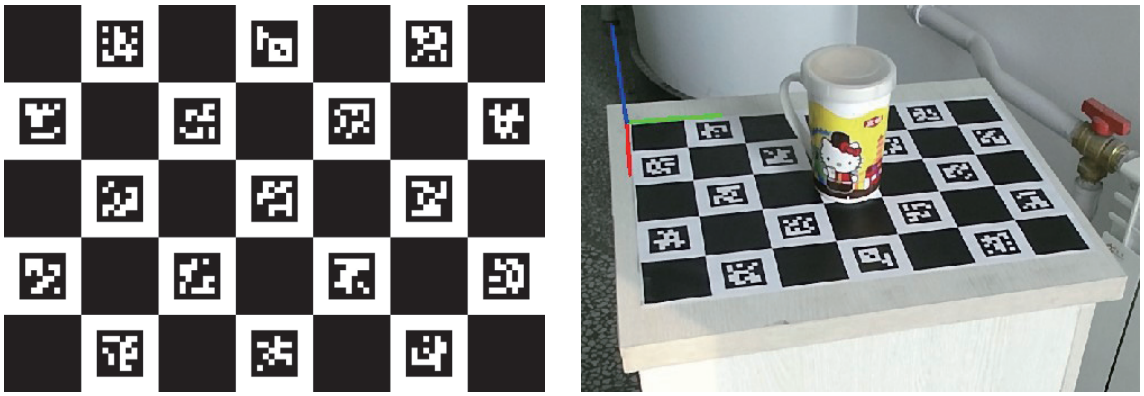
2.3 Registration of multiview information

To register different views reliably and conveniently, the ChArUco board[24] was adopted, as shown in Fig. 4(a). Given the parameters of the camera and ChArUco board, the pose of the reference frame S^b fixed on the ChArUco board could be determined, even if the board was partly obscured.



(a) (b)

Fig. 3. (Color online) (a) Multimodal feature points and (b) local patches of the object.



(a) (b)

Fig. 4. (Color online) (a) ChArUco board and (b) sampling scene.

During the reconstruction process, the target object was fixed on the ChArUco board. By rotating the ChArUco board, feature points and point clouds obtained in each viewpoint were registered according to the pose of S^b. Figure 4(b) shows a sampling scene, in which the X, Y, and Z axes of S^b are marked with red, green, and blue short lines, respectively.

After the postprocessing of fusing and downsampling, the point cloud model of the object O was reconstructed and denoted as M. Finally, a local reference frame S was specified for M, whose origin was located in the centroid, and the coordinate axes were set parallel to the principal axes of inertia. S and M are shown in Fig. 5, where the origin is represented by S_O, and the X, Y, and Z axes are respectively denoted as S_X, S_Y, and S_Z. According to the registered pose of each viewpoint, the 3D location L of each feature point was transformed to L^S, and L_{patch} was transformed to L^S_{patch}.

To ensure the efficiency of feature matching in the process of pose estimation, we constructed two K-dimension trees (K-D trees), T^O_{SURF} and T^O_{FPFH}. T^O_{SURF} and T^O_{FPFH} were respectively based on SURF and FPFH features of object O. For each feature point in each K-D tree, the name of the associated object O, the descriptor f, the 3D location L^S, and the positions of points in the local patch L^S_{patch} were stored together as $info_p = \{f, L^S, O, L^S_{patch}\}$.

For each object O, two K-D trees of feature points were stored as $info_{obj} = \{T^O_{SURF}, T^O_{FPFH}\}$. Furthermore, to retrieve features more efficiently in the process of object detection, the T^O_{SURF} of each object was aggregated into one K-D tree denoted as T^{all}_{SURF}. T^{all}_{FPFH} was obtained in the same way.

3. Object Detection and Pose Estimation

In the testing phase, the process of feature extraction was similar to that in Sects. 2.1 and 2.2. The only difference was that we retained all reasonable clusters (containing more than 50 points), rather than just the largest ones, because there were multiple objects in the testing scene. We repeatedly applied our approach of object detection and pose estimation for each cluster. In this section, we take one cluster as an example to illustrate details of our method.

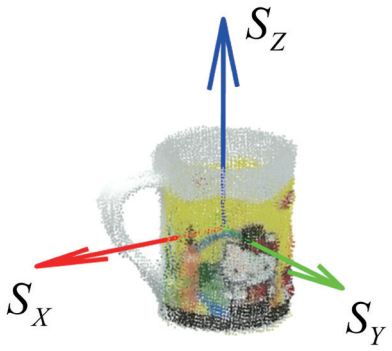


Fig. 5. (Color online) Point cloud model and local reference frame of the object.

3.1 Voting-based object detection

Common object detection methods based on feature points match scene feature points among the object dataset and then determine the object's name using a voting scheme. We retrieved multiple entries from the dataset for each feature point to achieve a more robust voting result, as shown in Fig. 6.

Multimodal feature points and their descriptors were extracted and calculated from each cluster in the testing scene. For each feature point x in the testing scene, the m nearest neighbors y_i (i = 1, 2, …, m) were retrieved from the K-D tree T^{all}_{SURF} or T^{all}_{FPFH} according to its feature descriptor $f(x)$. The neighboring point y_i cast a vote O_i, where O_i is the object's name stored with y_i. Objects with more votes than a threshold δ were the detection results.

3.2 Pose estimation based on multimodal feature points

According to the object detection result, we sequentially matched the feature dataset of each hypothetical object with the testing scene. Once matching pairs were determined, we no longer distinguished between the feature points of SURF and FPFH. These two types of feature points were used indiscriminately to estimate the object's pose, as shown in Fig. 7.

Our pose estimation process was implemented in two steps: rough and fine alignment stages. In the rough alignment stage, as in the traditional method, we estimated the rigid transformation between matching feature points using the RANSAC algorithm. Mismatched points were eliminated according to the consistency of the geometric transformation. In the fine alignment stage, we refined the pose by aligning two sparse point clouds composed of feature points' local neighbors using the iterative closest point (ICP) algorithm.

Moreover, we adjusted the mechanism used to find corresponding points in the ICP algorithm. Considering the situation shown in Fig. 8, we assumed that the keypoint p matches q according to their feature descriptors. The local patches P and Q are neighbors of p and q, respectively. For each point in P, the traditional ICP algorithm selected the closest point as the corresponding point, as shown in Fig. 8(a), which was sometimes unreasonable. According to the matching relationship between patches, the corresponding points of p's neighbors were

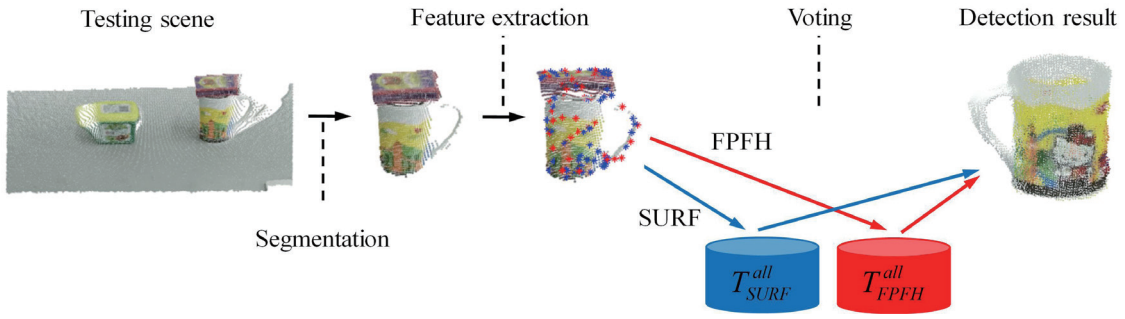


Fig. 6. (Color online) Illustration of our object detection method.

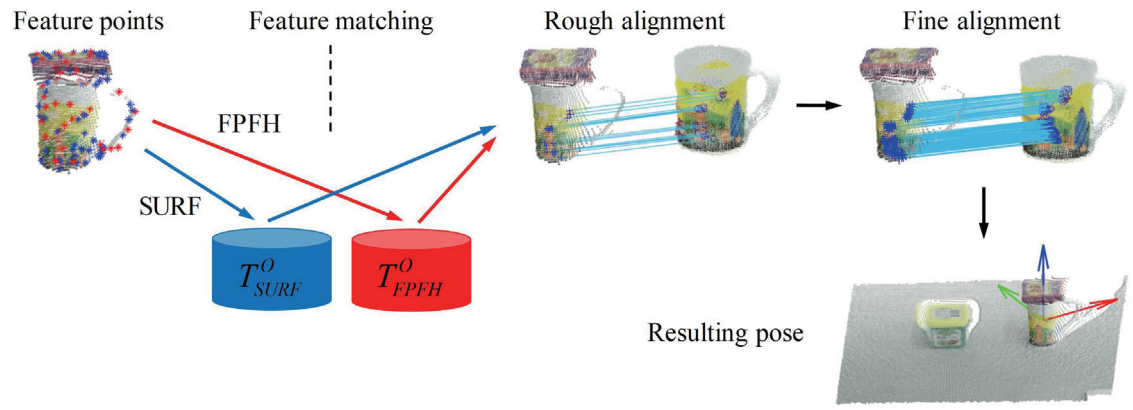


Fig. 7. (Color online) Illustration of our pose estimation method.

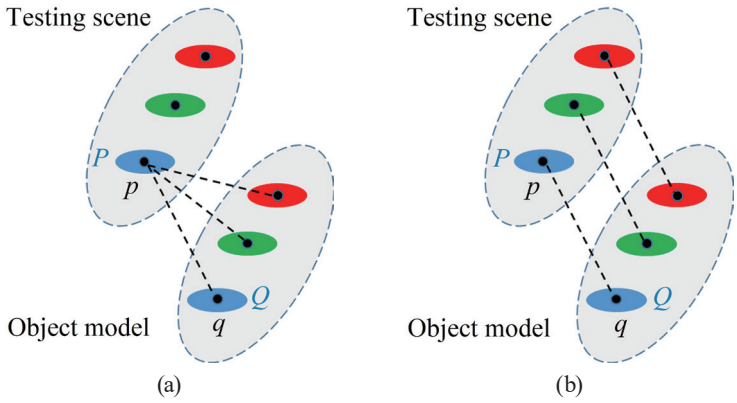


Fig. 8. (Color online) Mechanisms of selecting corresponding points by (a) ICP method and (b) our method.

mostly in Q. Therefore, when searching for corresponding points for members of P, only points inside Q rather than the entire point cloud were considered, as shown in Fig. 8(b). This approach effectively avoided most of the incorrectly corresponding pairs.

Finally, after eliminating the hypotheses whose ICP registration error was larger than a threshold τ, we obtained the identities and spatial poses of objects in the scene.

4. Experiments and Analysis

In this section, we verify the validity and applicability of our method by reporting three experiments.

4.1 Experimental settings

We reconstructed six objects and established seven datasets (one for each object and one for all of them) containing feature points and their neighboring patches. The resulting point clouds and labels of the six objects are shown in Fig. 9.

During the experiments, we estimated the poses of object A or B in 30 testing scenarios and removed unreasonable hypotheses based on the ICP registration errors. Texture-rich object A (in the first 15 scenarios) and texture-less object B (in the last 15 scenarios) were chosen as target objects. To test the robustness of our method to occlusion and background noise, we randomly placed irrelevant objects in scenes, which caused varying degrees of occlusion of object A or B. In each testing scene shown in Fig. 10(a), we removed other objects and used the ChArUco board to determine the ground-truth pose of object A or B, as shown in Fig. 10(b).

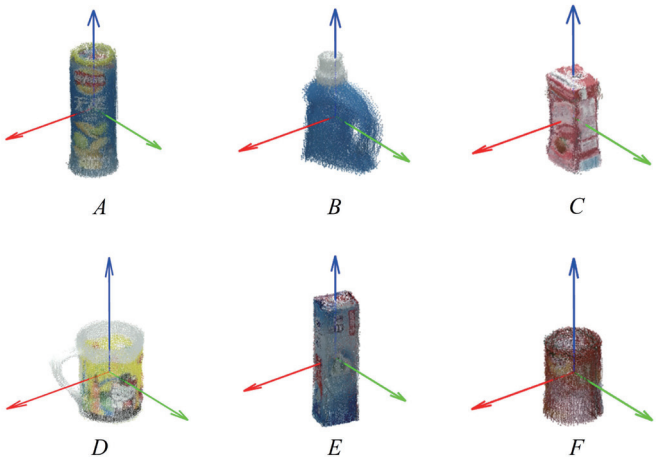


Fig. 9. (Color online) Objects in our dataset.

(a) (b)

Fig. 10. (Color online) (a) Testing scene and (b) ground-truth pose determination scene.

Considering the random deviation of the object's local reference frame, we kept object A or B fixed on the ChArUco board throughout the process of reconstruction and experiments. We set the number m of retrieval neighbors to 10, the threshold δ of object detection to 100, and the error threshold τ of pose estimation to 0.1 m.

4.2 Experiment 1

This comparative experiment was aimed at demonstrating the advantage of using multimodal features. On the basis of the proposed fine alignment method, we used three schemes of feature combinations for comparison. The precision, recall, and F1 scores of object detection in 30 testing scenes were calculated, as shown in Table 1.

By using multimodal features, our F1 score was at least 0.1349 higher than that of methods using a single feature in the 30 testing scenes. In the first 15 scenes (with a texture-rich object), the addition of FPFH feature points resulted in our method obtaining a 0.0339 higher F1 score than did the method only using the SURF feature. In the last 15 scenes (with a texture-less object), the addition of SURF feature points resulted in our method obtaining a 0.0551 higher F1 score than did the method only using the FPFH feature.

We transformed all points of the target object according to the estimated pose T_{est} and ground-truth pose T_{gt}. The average distance of all associated point pairs was obtained and denoted as *err*. A smaller err means a higher pose estimation accuracy. The *err* values were calculated in each scene, as shown in Fig. 11. It can be seen that our method achieved the highest estimation accuracy.

The superior performance in both object detection and pose estimation can be explained from the following three aspects. Firstly, the use of multimodal features extended the application scope of our method and introduced more feature points, both of which increased the reliability of object detection and the accuracy of pose estimation. Secondly, all feature points were obtained by the corresponding feature extraction methods, which maintained their inherent invariance and repeatability. Lastly, color features were extracted directly from the original color image, which guaranteed the feature quality.

Table 1
Object detection performance of methods using different features.

Scene number	Feature used	Precision	Recall	F1 score
1–15	SURF	0.7778	0.9333	0.8485
	FPFH	0.2000	0.0667	0.1000
	SURF & FPFH	0.7895	1.0000	0.8824
16–30	SURF	0.7143	0.6667	0.6897
	FPFH	0.7895	1.0000	0.8824
	SURF & FPFH	0.8824	1.0000	0.9375
1–30	SURF	0.7500	0.8000	0.7742
	FPFH	0.6667	0.5333	0.5926
	SURF & FPFH	0.8333	1.0000	0.9091

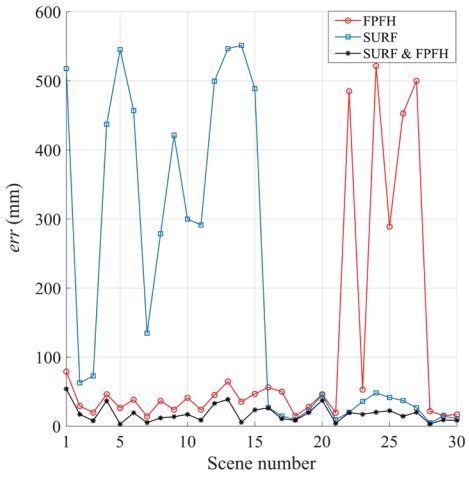


Fig. 11. (Color online) *err* values of different feature combination schemes.

4.3 Experiment 2

We designed this comparative experiment to show the advantage of our pose refinement strategy. On the basis of multimodal features, we used four pose refinement strategies for comparison. The precision, recall, and F1 scores of object detection in 30 testing scenes were computed, as shown in Table 2. "Key points" refers to a pose estimation method without the refinement process. "Whole points" refers to the strategy of using all the points of the object and testing scene for pose refinement. "Patches (mix)" refers to the strategy of using feature points' neighboring patches. "Ours" refers to our strategy.

Taking "Key points" as a benchmark, we determined the increase in F1 score for each pose refinement strategy. "Whole points" had worse performance than the benchmark. "Patches (mix)" was 0.0667 higher than the benchmark. "Ours" was 0.1633 higher than the benchmark and had the top F1 score.

Similarly to Experiment 1, according to the ground-truth poses, we obtained the *err* value for the resulting pose in each testing scene. The experimental results in Fig. 12 show that, taking "Key points" as a benchmark, "Ours" achieved the largest improvement in the accuracy of pose estimation.

For "Whole points", invisible points in the object model often biased the resulting pose to the object's interior. In addition, occlusion and a noisy background also introduced many irrelevant points. These problems resulted in the worst performance of "Whole points". In "Patches (mix)", only points near matching keypoints were involved. Points in the unseen views and occluded regions of the object and testing scene were eliminated adaptively while maintaining a sufficient number of points for accurate ICP alignment. On this basis, our method avoided incorrectly corresponding points among the mismatched patches and thus obtained the highest matching accuracy. These advantages resulted in greater improvement in the F1 score of object detection and the accuracy of pose estimation.

Table 2
Object detection performance of methods using different pose refinement strategies.

Align strategy	Precision	Recall	F1 score
Whole points	0.6667	0.4000	0.5000
Key points	0.7586	0.7333	0.7458
Patches (mix)	0.7647	0.8667	0.8125
Ours	0.8333	1.0000	0.9091

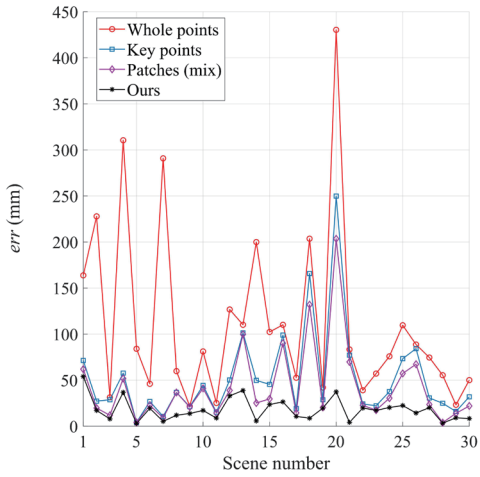


Fig. 12. (Color online) *err* values of different pose refinement strategies.

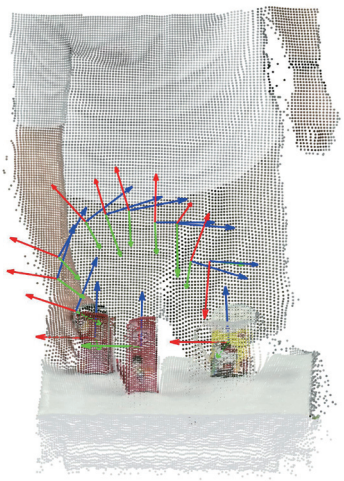


Fig. 13. (Color online) Experiment on our method in a human demonstration.

4.4 Experiment 3

This experiment served as a qualitative demonstration. We dealt with objects in a human demonstration based on the proposed method. In this scenario, the demonstrator poured food from object F into object D and finally placed object F back on the table. We drew the local coordinate system of each detected object in the same point cloud, as shown in Fig. 13. This experiment showed that the accuracy of our proposed method can basically meet requirements of demonstration learning.

5. Conclusions

We proposed a new method of manipulated-object detection and pose estimation. By using multimodal features, the number of matching features was increased for texture-less as well as texture-rich objects. A 0.1349 higher F1 score than that in methods using a single feature was demonstrated in Experiment 1. Our pose refinement strategy eliminated most of the scene interference and incorrectly corresponding points in the ICP process. This was proved by Experiment 2, in which our F1 score was 0.1633 higher than that of the benchmark. Furthermore, Experiment 3 qualitatively showed that our method can be used for manipulated-object detection and pose estimation.

Acknowledgments

This work was funded by the National Natural Science Foundation of China (61673136) and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 51521003).

References

1 B. D. Argall, S. Chernova, M. Veloso, and B. Browning: Rob. Auton. Syst. **57** (2009) 469. https://doi.org/10.1016/j.robot.2008.10.024
2 X. Ye, Z. Lin, and Y. Yang: Rob. Auton. Syst. **116** (2019) 126. https://doi.org/10.1016/j.robot.2019.03.011
3 J.-F. Lafleche, S. Saunderson, and G. Nejat: IEEE Rob. Autom. Lett. **4** (2018) 193. https://doi.org/10.1109/LRA.2018.2885584
4 W. Kehl, F. Manhardt, F. Tombari, S. Ilic, and N. Navab: Proc. IEEE Int. Conf. on Computer Vision (IEEE, 2017) 1521. https://doi.org/10.1109/ICCV.2017.169
5 E. Brachmann, F. Michel, A. Krull, M. Ying Yang, and S. Gumhold: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (IEEE, 2016) 3364. https://doi.org/10.1109/CVPR.2016.366
6 J. Wang, H. Yin, S. Zhang, P. Gui, and K. Xu: Sens. Mater. **31** (2019) 2089. https://doi.org/10.18494/SAM.2019.2307
7 S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab: Asian Conf. on Computer Vision (Springer, 2012) 548. https://doi.org/10.1007/978-3-642-33885-4_60
8 S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit: IEEE Trans. Pattern Anal. Mach. Intell. **34** (2011) 876. https://doi.org/10.1109/TPAMI.2011.206
9 T. Hodaň, X. Zabulis, M. Lourakis, Š. Obdržálek, and J. Matas: 2015 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) (IEEE, 2015) 4421. https://doi.org/10.1109/IROS.2015.7354005
10 Z. Xie, A. Singh, J. Uang, K. S. Narayan, and P. Abbeel: 2013 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IEEE, 2013) 2214. https://doi.org/10.1109/IROS.2013.6696666
11 M. A. Fischler and R. C. Bolles: Commun. ACM **24** (1981) 381. https://doi.org/10.1016/B978-0-08-051581-6.50070-2
12 A. Collet, M. Martinez, and S. S. Srinivasa: Int. J. Rob. Res. **30** (2011) 1284. https://doi.org/10.1177/0278364911401765
13 D. G. Lowe: Int. J. Comput. Vision **60** (2004) 91. https://doi.org/10.1023/B:VISI.00000
14 J. Vidal, C.-Y. Lin, and R. Martí: 2018 4th Int. Conf. on Control, Automation and Robotics (ICCAR) (IEEE, 2018) 405. https://doi.org/10.1109/ICCAR.2018.8384709
15 C.-Y. Tsai and S.-H. Tsai: IEEE Access **6** (2018) 28859. https://doi.org/10.1109/ACCESS.2018.2808225
16 F. Tombari, S. Salti, and L. Di Stefano: 2011 18th IEEE Int. Conf. on Image Processing (IEEE, 2011) 809. https://doi.org/10.1109/ICIP.2011.6116679
17 M. Karpushin, G. Valenzise, and F. Dufaux: Image Vision Comput. **71** (2018) 1. https://doi.org/10.1016/j.imavis.2017.11.007
18 Q. Yu, J. Liang, J. Xiao, H. Lu, and Z. Zheng: Comput. Vision Image Understanding **167** (2018) 109. https://doi.org/10.1016/j.cviu.2017.12.001
19 A. Zeng, K.-T. Yu, S. Song, D. Suo, E. Walker, A. Rodriguez, and J. Xiao: 2017 IEEE Int. Conf. on Robotics and Automation (ICRA) (IEEE, 2017) 1386. https://doi.org/10.1109/ICRA.2017.7989165
20 H. Liu, Y. Cong, C. Yang, and Y. Tang: Pattern Recognit. **92** (2019) 135. https://doi.org/10.1016/j.patcog.2019.03.025
21 H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool: Comput. Vision Image Understanding **110** (2008) 346. https://doi.org/10.1016/j.cviu.2007.09.014
22 Y. Zhong: 2009 IEEE 12th Int. Conf. on Computer Vision Workshops, ICCV Workshops (IEEE, 2009) 689. https://doi.org/10.1109/ICCVW.2009.5457637
23 R. B. Rusu, N. Blodow, and M. Beetz: 2009 IEEE Int. Conf. on Robotics and Automation (IEEE, 2009) 32127. https://doi.org/10.1109/ROBOT.2009.5152473
24 G. An, S. Lee, M.-W. Seo, K. Yun, W.-S. Cheong, and S.-J. Kang: Electronics **7** (2018) 421. https://doi.org/10.3390/electronics7120421

About the Authors

Xunwei Tong received his B.S. degree from Taiyuan University of Technology, China, in 2012 and his M.S. degree from Harbin Institute of Technology, China, in 2014. Since 2014, he has been a Ph.D. student at Harbin Institute of Technology, China. His research interest is in computer vision. (tong1137@163.com)

Ruifeng Li received his B.S. degree from Harbin Institute of Technology, China, in 1988 and his M.S. and Ph.D. degrees from Harbin Institute of Technology, China, in 1991 and 1997, respectively. Since 2004, he has been a professor at Harbin Institute of Technology, China. He is currently a vice leader at the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, China. His research interests are in intelligent service robot systems, advanced industrial robot technology, and humanoid robot mechanisms. (lrf100@hit.edu.cn)

Lijun Zhao received his B.S. degree from Beijing Institute of Technology, China, in 1996 and his M.S. and Ph.D. degrees from Harbin Institute of Technology, China, in 2002 and 2009, respectively. Since 2012, he has been a senior engineer at the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, China. His research interests are in intelligent robot motion control, SLAM, and object recognition technology. (zhaolj@hit.edu.cn)

Lianzheng Ge received his Ph.D. degree from Harbin Institute of Technology, China, in 2009. He is currently a lecturer at the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, China. His research interests are in mobile robots and control theory. (gelz@hit.edu.cn)

Ke Wang received his B.S. and Ph.D. degrees from Dalian University of Technology, China, in 2002 and 2008, respectively. He is currently a lecturer at the State Key Laboratory of Robotics and Systems, Harbin Institute of Technology, China. His research interests are in computer vision, image processing, human machine interaction, and estimation theory and their applications in robotics. (wangke@hit.edu.cn)