A Fast 3D Object Recognition Pipeline in Cluttered and Occluded Scenes

Liupo Zheng, Hesheng Wang(✉), and Weidong Chen

Shanghai Jiao Tong University, Shanghai 200240, China
{wanghesheng,wdchen}@sjtu.edu.cn

Abstract. In this paper we propose a framework for instance recognition and object localization in cluttered and occluded household environment for robot grasping task. The whole system bases on a coarse to fine pipeline in combination with the state-of-the-art methods of RGBD-based object detection. We build a sparse feature model by extracting structure key points incorporating texture cues in the train procedure. After that, the paper demonstrates how the algorithm decreases the time complexity and simultaneously guarantees the accuracy of the recognition and pose estimation. Quantitative experimental evaluations are presented using both acknowledged ground truth dataset and real-world robot perception system.

Keywords: Object recognition · RGBD-image · Clutter · Pose estimation

1 Introduction

One of the essential challenges in robot perception field is the object recognition and localization in unstructured scenes. The robot in such environment encounters many restrictions such as occlusions, clutters, illumination changes, multiple objects,real-time limits and etc. The conservative recognition methods using 2D image features like SIFT, SURF, ORB, HOG [1–3] are more and more unable to satisfy the requirements with the improvement of system accuracy and speed. Recently, the work based on RGBD images has been fostered thanks to the availability of low-price and high-performance 3D sensors such as Intel RealSence and Microsoft Kinect.

Different from general computer vision, robotic vision system has the particular characteristics [4]. We pay more attention to instance recognition rather than at category levels [5] and we should identify the object and simultaneously estimate 6-DOF pose for the robot grasping. Because the number of object instance is usually huge, the real-time performance should also be considered mainly in practical application. One way of solving these is to introduce hierarchy which executes the recognition at different levels [6]. The other direction of the research is trying to find more effective feature descriptors in local [7–9] or global [10,11] aspect. The local approaches construct the correspondence model in a scene by

Y. Huang et al. (Eds.): ICIRA 2017, Part II, LNAI 10463, pp. 588–598, 2017.
DOI: 10.1007/978-3-319-65292-4_51

extracting and matching feature points and afterwards these correspondences are clustered under certain rules to generate a object hypothesis [12]. The global algorithms on the other hand extract a single descriptor to describe the object in sight [13]. There is also some work dealing with performance improvements through combining multiple algorithms [14,15].

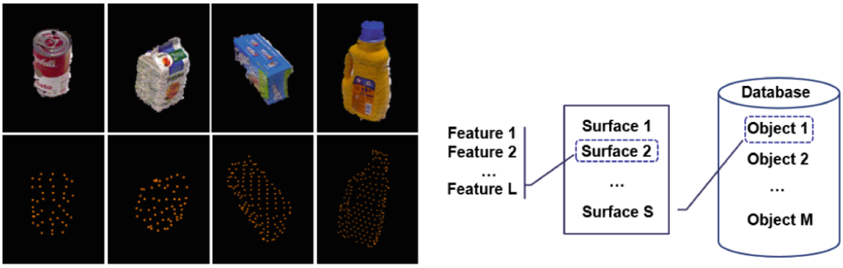
In this paper, we propose a coarse to fine object recognition and localization framework for robot perception in the complex environment. The first step is the off-line object modelling and training. We define a unique generalized eigenvector consisting of spatial and texture features to build a sparse model of the object. In order to search more effectively, we organize the training data in a specific structure and label each surface of the object. We conduct a novel search strategy to deliver the candidates in coarse pipeline after training. Then, several approaches are used to refine the initial result and estimate the 6-DOF pose of each object hypotheses. The framework ends with the output of object identity and transformation matrix with respect to camera frame.

The paper is structured as follows. The next section presents the steps of sparse feature model building and object training. Sect. 3 introduces more detailed components of the coarse to fine object recognition and pose estimation framework, followed by experiment and result in Sect. 4. Finally, the result is analysed and concluded in Sect. 5.

2 Object Modelling and Training

As the number of each point cloud data is huge, it is hard to guarantee the system real-time performance if we take all of the cues into consideration. For each object training process, we build a sparse model M_s including shape, size and texture information of an object which mainly characterizes its local feature. We define a novel generalized eigenvector $f_s : f = [cloud\_corrdinates, normals, feature\_descriptors]$ to describe these unique feature points. The item of $feature\_descriptors$ are extracted from two aspects: 3D points generated from the cloud data and 2D points which are back-projected from 3D space. More concretely, we use SHOT (Signature of Histograms of Orientations) [16] to describe the original 3D feature points and SIFT (Scale-invariant feature transform) to represent the 2D once.

The first step of our recognition pipeline is to build models M_s for each object. To do this, we use an off-the-shelf *Simultaneous Localisation and Mapping* approach [17,18] to merge the image data gathered by moving RGB-D camera around table-top object. In each frame, a color and a depth image are taken and the point cloud is generated through fusing both of information. The correspondences between two frames are estimated in color images, and the projected 3D points corresponding key points in 2D space are used to compute the transformation between two frames. The infinite points caused by the camera are first filtered, followed by the points which are too far away from the camera. To estimate the transformation, we use RANSAC (Random Sample Consensus) to eliminate outliers and ICP (Iterative Closest Point) to get more accurate results.



(a) key points and sparse model (b) structure of training database

Fig. 1. The sparse model building and data training

They are further optimized by the ParallaxBA (Parallax Bundle Adjustment) proposed by Zhao [19]. Finally all the frames are transformed into one object coordinate system. For each surface $S^i = \left\{S^i_1, S^i_2, S^i_1, S^i_3, \ldots, S^i_s\right\}$ of one object M_i, we unify the formation of feature vector f_i to describe them. These points are stored in the training database $\mathbf{M}_d = \{M^d_1, M^d_2, \ldots, M^d_m\}$. In Fig. 1, the sparse model building and data training are shown in detail.

3 Object Recognition and Pose Estimation

In order to fully exploit the occluded objects and simultaneously compute 6-DOF pose for robot grasping, we propose a coarse to fine framework which consists of three major parts, namely $Off-line\ training$, *Coarse pipeline* and *Fine pipeline*. The structure of the proposed method is outlined in Fig. 2. The output of the system is a cluster with recognized object label and its homogeneous transformation matrix with respect to the camera frame.

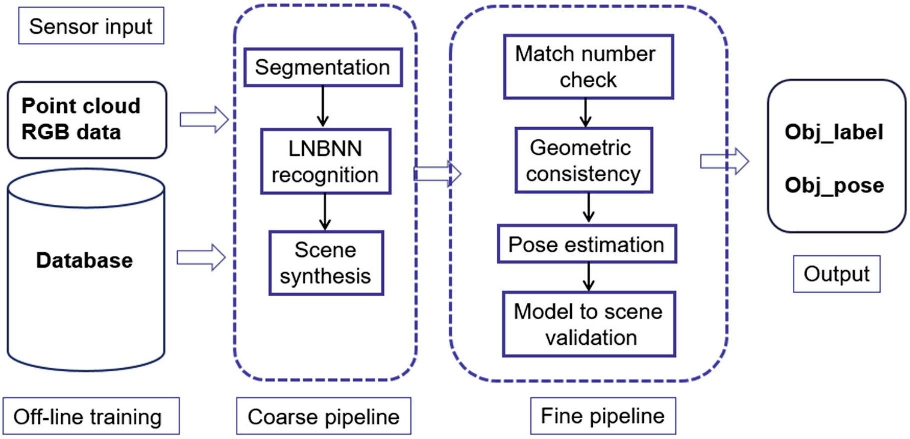


Fig. 2. Coarse to fine object recognition and pose estimation framework

3.1 Coarse Recognition Pipeline

Segmentation. In the testing case, the input point cloud P_j is firstly segmented into multiple object hypotheses $\mathbf{O}_j = \{O_1^j, O_2^j, \ldots, O_k^j\}$. We try to reduce the computational cost by focusing on the point cloud within a certain range on the supporting plane. Hence we use RANSAC to estimate the plane of the supporting table and cut the rest data below the plane.

For the objects on the plane, we conduct a bottom-up segmentation method proposed by Richtsfeld [20]. We over-segment the point cloud into supervoxels and build up a supervoxel adjacency graph using Voxel Cloud Connectivity Segmentation [21], followed by a pre-merging process, in which all the adjacent supervoxels are merged into patches based on their normal similarity. The clustered patches are fitted to a object hypotheses according to the local convexity and sanity criterion and using noise filtering procedure to merge the small noisy patches into the neighboring segment with the greatest size.

Local Naive Bayes Nearest Neighbor. The next step for coarse pipeline is to search object hypothesis candidates in the train set $\mathbf{M}_d = \{M_1^d, M_2^d, \ldots, M_m^d\}$ for each segmented point clusters $\mathbf{M}_s = \{M_1^s, M_2^s, \ldots, M_k^s\}$ which are the feature models extracted from $\mathbf{O}_j = \{O_1^j, O_2^j, \ldots, O_k^j\}$ using the same sparse expression as training.

The naive Bayes Nearest Neighbor algorithm is widely used in the image searching and classification [22]. The goal of the approach is to determine the most probable class $\hat{C}$ of a query image Q according to

$$\hat{C} = arg\ max\ P(C|Q) \tag{1}$$

Refer to this thought, we define our problem as flows. Each train model M^d is essentially a set of generalized eigenvectors $M^d = \{f_i : f_i = [cloud\_cor, normals, feature\_descriptors],\ i = 1, 2, \ldots, L\}$. In the on-line object recognition pipeline, we extract feature descriptors from segmented object hypotheses and obtain the sparse model M^s which contains N feature vectors $\{f_j,\ j = 1, 2, \ldots, N\}$. Then the recognition issue is transformed into the maximization of posterior probability shown in Eq. 2.

$$\hat{M^d} = arg\ max\ P(M^d|M^s) \tag{2}$$

Assuming a uniform prior probability over objects and independence of the descriptors f_j extracted from cluster M^s, applying Bayes' rules:

$$P(M^d|M^s) \propto P(M^d)P(M^s|M^d) \propto \prod_{j=1}^{N} P(f_j|M^d) \tag{3}$$

According to kernel density estimation and the descriptors f_j are highly dimensional, therefore distribute sparsely, the $P(f_j|M^d)$ can be rewritten approximately

$$P(f_j|M^d) = \frac{1}{L}\sum_{i=1}^{L} K\left(f_i - f_j\right) \approx \frac{1}{L} e^{-\|f_j - f_{NN}(f_j)\|^2} \tag{4}$$

where the $K(\cdot)$ is the $Gaussian - Parzen\ kernel$ and $f_{NN}(f_j)$ is the nearest neighbor to f_j in the train database $M^d = \{f_i : i = 1, 2, \ldots, L\}$. Substituting Formula 4 into 3 and taking logarithm both sides of 3, we can get

$$\hat{M^d} = arg\ min \sum_{j=1}^{N} \|f_j - f_{NN}(f_j)\|^2 \tag{5}$$

We notice that the feature cluster M_m^s is actually one of the object surfaces, in order to match more accurately and provided more refined cues later for the pose estimation, we label each surface S^d in the train database, rewrite the generalized eigenvector f_s : f = [$surface\_label$, $cloud\_corrdinates$, $normals$, $feature\_descriptors$] and search target in the category of S^d. Suppose the number of object is M, each object trains S surfaces and each surface has L features averagely. We conduct KD (K-Demensional) search strategy in a single loop and the complexity of the NBNN is $O(M \cdot S \cdot N \cdot log(L))$ and the time consumption increases linearly with the number of train database elements. The real-time will be influenced heavily with the object number increasing.

Considering the $Gaussian - Parzen\ kernel$ $K(f_i - f_j) = e^{-\frac{\|f_i - f_j\|^2}{2\sigma^2}}$, $P(f_j|M^d)$ decreases exponentially with respect to $\|f_i - f_j\|^2$ and there is no need to search every surface S^d in the database, we just care several nearest neighbors of the f_j. Under this point, we merge all of the trained surfaces S^d into a new structure $\{f_k^{DB}\} = \{f_k^{S_1^d}\} \bigcup \{f_k^{S_2^d}\} \bigcup \ldots \bigcup \{f_k^{S_{M \cdot S}^d}\}$ and conduct the KD search in the merged database $\{f_k^{DB}\}$. Compare with the previous NBNN, our method just search the nearest neighbors in one structure instead of every object. We call this method *Local Naive Bayes Nearest Neighbor* (summarized in Algorithm 1). The complexity of LNBNN is $O(N \cdot log(M \cdot S \cdot L))$ and the time consumption increases logarithmically with the number of objects.

Scene Synthesis. After LNBNN search, we get candidates $\mathbf{C}_{candi}^{M^s} = \{C_1^{M^s}, C_2^{M^s}, \ldots, C_t^{M^s}\}$ of certain object cluster M^s. Since we have marked the category and surface label for each candidate, it is easy to distinguish which candidate belongs to the same object. We conduct a voting scheme to synthesize the coarse recognition result and select the final candidate set with the uniform classification label.

3.2 Fine Recognition Pipeline

Match Number Check and Geometric Consistency. The coarse pipeline output a set of most likely candidates of the object hypotheses, we next determine which is the best recognition result in the fine pipeline. Given the descriptors of object hypotheses and an appropriate candidate, we first use FLANN (Fast Library for Approximate Nearest Neighbors) strategy to discover the correspondent points and define a matching threshold to discarded correspondences with large distance. We then execute *match number check* stage to decide whether it is meet the requirement or not. Afterwards *geometric consistency*

Algorithm 1. Local Naive Bayes Nearest Neighbor

```
Input: Train dataset $\{f_k^{DB}\}$ and object hypotheses feature cluster $M^s$
Output: Candidates of object hypotheses $\{C_1^{M^s},\ C_2^{M^s}, ..., C_t^{M^s}\} \in \{f_k^{DB}\}$
Initialize $dist_m = 0$ $(m = 1, 2, ..., M \cdot S)$ where $dist_m$ represents the distance
between object hypotheses $M^s$ and trained surface $S^d$
for $f_j \in M^s$ do
    search in $\{f_k^{DB}\}$ and get $r + 1$ nearest neighbors $\{f_{NN1}, f_{NN2}, ..., f_{NNr+1}\}$
    merge eigenvectors belong to one surface and obtain set $\{C_1^{M^s},\ C_2^{M^s}, ...C_s^{M^s}\}$
    set $dist_0$ equal $\|f_j - f_{NNr+1}\|^2$
    for $C_i \in \{C_1^{M^s},\ C_2^{M^s}, ...C_s^{M^s}\}$ do
        $dist_m = dist_m + \|f_j - f_{NN_i}\|^2 - dist_0$
    end
end
sort $dist_m$ in ascending order and take the first $t$ $\{C_i^{M^s}\}$ $(i = 1, 2, ..., t)$
as candidates
return $\{C_i^{M^s}\}$
```

clustering algorithm will be conducted to enforce geometric constraints between pairs of correspondences and remove the mismatching points.

Pose Estimation. Since the correspondences of the key points have been extracted, we conduct SVD (singular value decomposition) to get the initial transformation T_m^c between the candidate surface C_i and object M^s. Because we use nearest neighbors of the descriptors to generate point-to-point relationships in 3D space, some of our correspondences are likely to be incorrect. We account for this using RANSAC algorithm and through dozens of iteration to get a better result which will be used in ICP stage as initial value to increase the accuracy of the transformation by $T_m^c = T_{ICP} \cdot T_m^c$. Therefore, the 6-DOF pose of the object hypotheses M^s with respect to camera is $T_s = T_{cam}^m \cdot T_m^c$, where the T_{cam}^m is the transformation matrix between object frame and camera frame.

Model to Scene Validation. For each object hypotheses M^s conducts these steps above, all the scene objects have been recognized and we will get a cluster of most likely candidates $\mathbf{C} : \{C_1, C_2, \ldots, C_k\}$ correspond to the segmented point cluster $\mathbf{M}_s : \{M_1^s, M_2^s, \ldots, M_k^s\}$. If more than one candidates C_i contain the same label, two conditions probably lead to this. One is the over segmentation and the other is false recognition. We distinguish this two situations by means of checking whether object hypotheses bounding volumes with same label are overlap. If so, we merge those hypotheses and estimate the pose again, otherwise we discard this false result and restart the recognition pipeline.

After checking scene consistency, we project each candidate into the scene using estimated pose matrix and conduct overlap ratio verification. We construct a two-dimensional histogram of $20 \cdot 20$ bins to represent the distribution and orientation of surface normals. Since the surface normals are normalized, the term n_z is determined by n_x and n_y, there is no need to contain n_z in the histogram. A direct method to evaluate the similarity of two histograms is

$$D(A, B) = \sum_{i,j} |a_{ij} - b_{ij}| \tag{6}$$

Normalizing the histogram and utilizing the following equation

$$min(a, b) = \frac{1}{2}(a + b) - \frac{1}{2}|a - b| \tag{7}$$

we can get a more effective way to compute the similarity of the histogram

$$S(A, B) = 1 - \frac{1}{2}D(A, B) = \sum_{i,j} min(a_{ij} - b_{ij}) \tag{8}$$

where $S(A, B)$ is the metric which has a positive correlation between the value and the similarity of histograms. Further more, it is also a indirect indication of the overlap ratio between the candidate model and the scene object. The bigger of the metric value, the more accurate of the recognition and pose estimation.

4 Experiment and Result

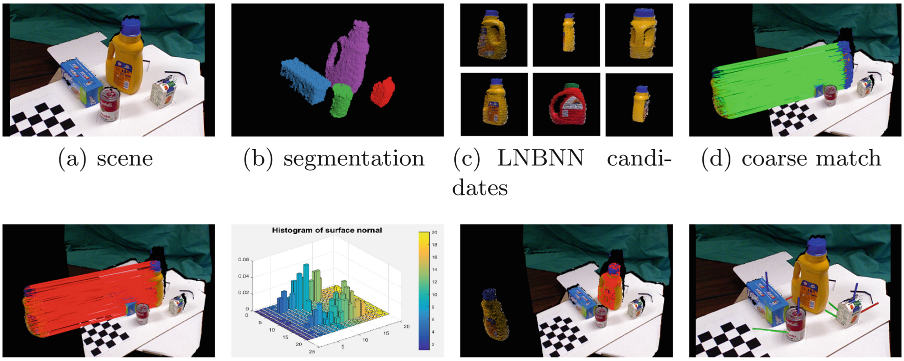
4.1 Recognition Experiment

We evaluated the efficiency of recognition framework on a famous household dataset *Willow Challenges* from ICRA *Perception Challenge* 2011. This dataset contained 35 rigid object instances and 39 scenes including both simple and complex case, each object consisted of 37 frames from different views. Our training pipeline built the sparse feature model using these point cloud instance while recognition pipeline was implemented with the given ground truth scenes using OpenCV and PCL.

After training all of the objects in dataset, we selected 14 scenes randomly for testing and each scene contained 4 different view ports. The whole process steps are shown in Fig. 3. After LNBNN searching and fine pipeline optimizing, the final recognition precision is summarized in Table 1. We also analysed the pose recovery of detected object with the benchmark, the statistics indicate the average errors of translation and rotation are under 4 cm and 8°. Figure 4 shows the line chart of translation and rotation errors around X-axis, Y-axis and Z-axis.

4.2 Grasping Experiment

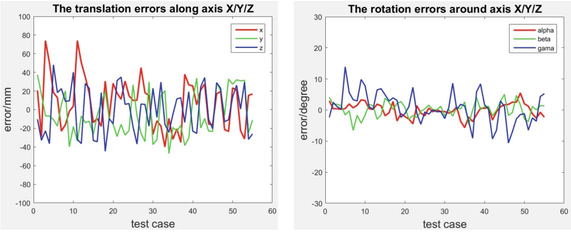
The proposed method was then tested on the real-world environment under the ABB industrial manipulator with a calibrated Kinect sensor (shown in Fig. 5(a)). The objects were placed on a plane table in clutter and the goal was to recognize all instances in the scene and estimate the pose at the same time. For the robot-grasp planning, we followed the work of *Dogar* [23], by means of NGR (negative goal region) and relevant methods to deliver the motion trajectory and grasp gesture.



(a) scene (b) segmentation (c) LNBNN candidates (d) coarse match

(e) geometric consistency (f) histogram (g) model to scene (h) pose estimation

Fig. 3. Overview of object recognition and pose estimation pipeline



(a) translation error (b) roation error

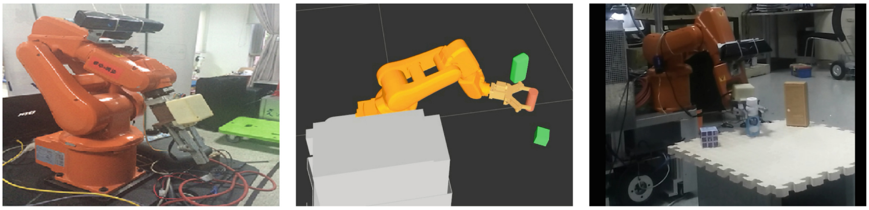
Fig. 4. The errors of translation and rotation around X-axis, Y-axis and Z-axis

Table 1. Quantitative recognition precision

Item	Value
True positive	96.76%
False positive	3.24%
False negative	2.18%
Recall	97.80%
Precision	96.76%

The experiment was designed as follows:

- Training the objects which are commonly used in household environment.
- Captured the original scene in clutter and conducted the segmentation stage to deliver the object hypotheses $\{M_1^s, M_2^s, \ldots, M_k^s\}$.
- Applied LNBNN and *scene synthesis* to obtained candidates in coarse pipeline.
- Conducted fine pipeline and iterated until all the hypotheses are recognized.



(a) robot with Kinect (b) motion planning (c) grasp action

Fig. 5. Snapshots for grasping experiment of the robot

- Combined the output of the recognition framework, the robot executed the motion planning under ROS (Robotic Operation System) and grasped objects to verity the correctness of the result.

The Fig. 5(b) shows the robot motion planning in Rviz and Fig. 5(c) is a snapshot of robot grasping. The experiment result illustrates that the recognition and pose estimation satisfies the robot grasping requirement.

5 Conclusions

In this paper, we present a coarse to fine object recognition and pose estimation framework for robot grasping in the cluttered and occluded household environment. Our system combines the texture and spatial cues constructing a generalized eigenvector and it is robust for occlusion and illumination changes. We exploit the segmented clusters by means of LNBNN to ensure the real time performance and through fine recognition pipeline to improve the accuracy of the pose estimation. The experiment illustrates the efficiency of the proposed method. In the future, we aim to extend our algorithm to non-rigid objects and a more effective segmentation approach is also need to explore deeply.

Acknowledgement. This work was supported in part by the Natural Science Foundation of China under Grant U1613218, 61473191 and 61503245, in part by the Science and Technology Commission of Shanghai Municipality under Grant 15111104802, in part by Shanghai Sailing Program under Grant 15YF1406300, in part by State Key Laboratory of Robotics and System (HIT).

References

1. 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 2009, pp. 48–55. IEEE (2009)
2. Martinez, M., Collet, A., Srinivasa, S.S.: Moped: a scalable and low latency object recognition and pose estimation system. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 2043–2049. IEEE (2010)

3. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 2564–2571. IEEE (2011)
4. Tang, J., Miller, S., Singh, A., Abbeel, P.: A textured object recognition pipeline for color and depth image data. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), pp. 3467–3474. IEEE (2012)
5. Janoch, A., Karayev, S., Jia, Y., Barron, J.T., Fritz, M., Saenko, K., Darrell, T.: A category-level 3D object dataset: putting the kinect to work. In: Fossati, A., Gall, J., Grabner, H., Ren, X., Konolige, K. (eds.) Consumer Depth Cameras for Computer Vision. Advances in Computer Vision and Pattern Recognition, pp. 141–165. Springer, London (2013). doi:10.1007/978-1-4471-4640-7_8
6. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view RGB-D object dataset. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 1817–1824. IEEE (2011)
7. Mian, A., Bennamoun, M., Owens, R.: On the repeatability and quality of keypoints for local feature-based 3D object retrieval from cluttered scenes. Int. J. Comput. Vis. **89**(2–3), 348–361 (2010)
8. Papazov, C., Burschka, D.: An efficient RANSAC for 3D object recognition in noisy and occluded scenes. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010. LNCS, vol. 6492, pp. 135–148. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19315-6_11
9. Petrelli, A., Di Stefano, L.: On the repeatability of the local reference frame for partial shape matching. In: 2011 International Conference on Computer Vision, pp. 2244–2251. IEEE (2011)
10. Aldoma, A., Tombari, F., Rusu, R.B., Vincze, M.: OUR-CVFH – oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6DOF pose estimation. In: Pinz, A., Pock, T., Bischof, H., Leberl, F. (eds.) DAGM/OAGM 2012. LNCS, vol. 7476, pp. 113–122. Springer, Heidelberg (2012). doi:10.1007/978-3-642-32717-9_12
11. Wohlkinger, W., Vincze, M.: Ensemble of shape functions for 3D object classification. In: 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 2987–2992. IEEE (2011)
12. Jiang, D., Wang, H., Chen, W., Wu, R.: A novel occlusion-free active recognition algorithm for objects in clutter. In: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), pp. 1389–1394. IEEE (2016)
13. Aldoma, A., Tombari, F., Stefano, L., Vincze, M.: A global hypotheses verification method for 3D object recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 511–524. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33712-3_37
14. Aldoma, A., Tombari, F., Prankl, J., Richtsfeld, A., Di Stefano, L., Vincze, M.: Multimodal cue integration through hypotheses verification for RGB-D object recognition and 6DoF pose estimation. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 2104–2111. IEEE (2013)
15. Lutz, M., Stampfer, D., Schlegel, C.: Probabilistic object recognition and pose estimation by fusing multiple algorithms. In: 2013 IEEE International Conference on Robotics and Automation (ICRA), pp. 4244–4249. IEEE (2013)
16. Tombari, F., Salti, S., Stefano, L.: Unique signatures of histograms for local surface description. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6313, pp. 356–369. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15558-1_26

17. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments. Int. J. Robot. Res. **31**(5), 647–663 (2012)
18. Herbst, E., Henry, P., Fox, D.: Toward online 3-D object segmentation and mapping. In: 2014 IEEE International Conference on Robotics and Automation (ICRA), pp. 3193–3200. IEEE (2014)
19. Zhao, L., Huang, S., Sun, Y., Yan, L., Dissanayake, G.: Parallaxba: bundle adjustment using parallax angle feature parametrization. Int. J. Robot. Res. **34**(4–5), 493–516 (2015)
20. Richtsfeld, A., Mörwald, T., Prankl, J., Zillich, M., Vincze, M.: Segmentation of unknown objects in indoor environments. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4791–4796. IEEE (2012)
21. Papon, J., Abramov, A., Schoeler, M., Worgotter, F.: Voxel cloud connectivity segmentation-supervoxels for point clouds. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2027–2034. IEEE (2013)
22. Tuytelaars, T., Fritz, M., Saenko, K., Darrell, T.: The NBNN kernel. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1824–1831. IEEE (2011)
23. Dogar, M., Srinivasa, S.: A framework for push-grasping in clutter. Robot.: Sci. Syst. VII **1** (2011)