Point Pair Feature based Object Detection for Random Bin Picking

Wim Abbeloos and Toon Goedemé

KU Leuven, Department of Electrical Engineering, EAVISE
Leuven, Belgium
Email: wim.abbeloos@kuleuven.be, toon.goedeme@kuleuven.be

Abstract—**Point pair features are a popular representation for free form 3D object detection and pose estimation. In this paper, their performance in an industrial random bin picking context is investigated. A new method to generate representative synthetic datasets is proposed. This allows to investigate the influence of a high degree of clutter and the presence of self similar features, which are typical to our application. We provide an overview of solutions proposed in literature and discuss their strengths and weaknesses. A simple heuristic method to drastically reduce the computational complexity is introduced, which results in improved robustness, speed and accuracy compared to the naive approach.**

Keywords-**Point Pair Feature, Object Detection, Pose Estimation, Bin Picking, Robotics**

I. INTRODUCTION

Nowadays, the automatic handling of objects by industrial robots is common practice. However, if the object's pose is unknown beforehand, it needs to be measured. Indeed, vision-guided industrial robots are one of the key ingredients of state-of-the-art manufacturing processes. Everyone is aware of the fact that future production processes will be increasingly flexible and less labor intensive. Purely mechanical singulation installations, such as vibration feeders, no longer meet flexibility requirements or are no longer profitable, and manual work is becoming more expensive. One very cost-effective and flexible solution is the supply of parts in bulk, as illustrated in Figure 1 and Figure 2, from which industrial robot arms pick out the objects one by one in order to feed them to the rest of the manufacturing chain. This application is referred to as random bin picking. The goal is to locate one pickable object instance at a time and determine its six degree of freedom (6D) pose, so that the robot's end effector can be moved towards the object and grasp it. This paper is focused on the object detection and localization task of such an industrial random bin picking application.

Pose estimation is a widely researched computer vision problem and various solutions based on 2D images, range images or 3D pointclouds exist. However, very few are suited for the specific conditions of the real-world industrial bin picking application at hand. As will be detailed in the next chapter's overview of relevant object detection algorithms, the lab examples mostly studied in literature differ quite a lot from real industrial bin picking scenarios. In
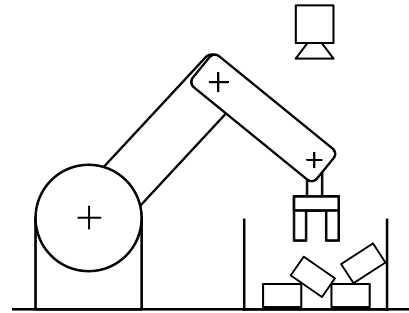

Figure 1. A typical example of a bin picking setup consists of some kind of 3D sensor mounted above a bin filled with randomly stacked parts.

Figure 2. Objects supplied in bulk are stacked randomly in a bin. An object detection and pose estimation algorithm is needed to determine the six degree of freedom pose of a pickable object, so that it can be grasped by the robot.

the latter case, objects with a wide gamut of characteristics are encountered: from nicely textured boxes to smooth shiny metal parts and from complex three dimensional free form shapes to rotationally-symmetric bolts. No satisfactory general solution to this problem exists yet. In this paper, point pair features are studied as a versatile object detection technique in random bin picking. Moreover, combined with the simple but very powerful heuristic search space reduction that is proposed in this paper, the technique's computational demands remain within manageable bounds. We also propose a generic method that enables to use industrially available CAD models of the objects to be detected as input to our detection pipeline.

In literature, a hodgepodge of different evaluation mechanisms for random bin picking are used. This paper proposes

a new universal way of evaluating pose estimation algorithms for random bin picking, necessary for fair comparison across different approaches. It consists of a completely automated procedure for the generation of realistic synthetic scenes and the evaluation of the detection algorithm. As the procedure is automatic, it can be used in a closed loop to optimize the detection algorithms parameters. The goal is to achieve optimal performance across a set of widely varying objects.

The remainder of this paper is organized as follows. Section II gives an extensive overview of the different 3D object localization techniques for random bin picking that are proposed in literature, as well as the detection evaluation methods available. Our point pair feature-based random bin picking approach is introduced in section III, composed by the dataset preprocessing, object detection, heuristic search space reduction and evaluation steps we propose. Experimental results on representative industrial objects are presented and discussed in section IV, and a conclusion follows in section V.

II. RELATED WORK

An extensive review of the state of the art in 3D object detection and pose estimation algorithms is provided. This section is split into a part discussing the point pair feature based techniques and a part discussing algorithms based on other representations.

A. 3D Object Detection Methods

In some simple cases the problem of detecting and estimating the pose of objects has been addressed by segmenting the scene and applying a global feature descriptor to the segmented parts in order to recognize one of the segments as being the considered object. It is clear that, in a lot of cases, as in our random bin picking case, this approach will not work, as it is not possible to reliably segment an object in the presence of significant clutter and occlusion.

A more sophisticated approach is to detect and describe certain parts or features of the object. Some techniques have been proposed to detect 3D objects from regular images based on representations such as: 2D keypoints [1], 2D edge templates [2] [3] or line descriptors [4] [5]. Other techniques work on 3D representations such as: shape primitives (planes, cylinder, sphere, superquadrics, etc.) [6], 3D keypoints [7] [8] [9], range image templates [10] or color gradient and normal templates [11] [12] [13].

All these detection methods either create templates from several object views, or extract some kind of features. An important downside to the methods relying on multiple object views is that they require a large amount of dense templates and as such, are computationally expensive to match. There are two important issues with feature based methods, the first is that they are not very general: they can only represent objects that contain the specific type of feature they use. Another issue is that the feature descriptors (e.g. 3D keypoint descriptors) may be quite sensitive to noise.

B. Point Pair Features

In the previous section methods relying on several types of features were discussed. A lot of early work focused on describing the relations between a set of features of an object. The feature vector used in point pair features (see Section III-B) is very similar to some of these formulations [17], however, the important difference is that point pair features are calculated for all point pairs on a mesh, not just for specific features. This means they can be used to represent free form objects.

Point pair features were first introduced for 3D object classification [18]. A histogram of the features occurring on a surface allowed to classify the object. This technique was later combined with a clustering algorithm to allow the detection of objects in pointclouds [19]. An efficient procedure for storing the features in hashtables and calculating matches was proposed by Winkelbach et al. [20], which was later extended to an object detection technique [23] [24].

An efficient voting scheme to determine the object pose was described by Drost et al. [21]. The same authors later proposed a modified point pair feature using only points paired with geometric edge pixels, which are extracted from multimodal data [22]. This makes the method suitable for object detection (instead of recognition).

Papazov et al. [25] [26] reduced the dimensions of the hashtable and the number of points to match by only using point pairs within a certain relative distance range from each other. Kim et al. [27] extended the point pair feature with a visibility context, in order to achieve a more descriptive feature vector. Choi et al. proposed the use of more characteristic and selective features, such as boundary points and line segments [28] and extended the feature point vector with a color feature [29] [30]. A method to learn optimal feature weights was proposed by Tuzel et al. [31]. Several extensions are proposed by Birdal et al. [32]: they introduce a segmentation method, add a hypothesis verification stage and weigh the voting based on visibility.

C. Detection Evaluation

Several ways to measure the accuracy of detections have been suggested in literature.

- The Average Distance between all vertices in the 3D model in the estimated pose and the ground truth pose [33] [25] [34]. In some cases this corresponds to the residual error reported by the Iterative Closest Point (ICP) algorithm used to refine the pose.
- The fraction of points of the object model points transformed to the detected pose that are within a predefined distance to their closest point [23].

(a) Depth image (dark colors are closer)
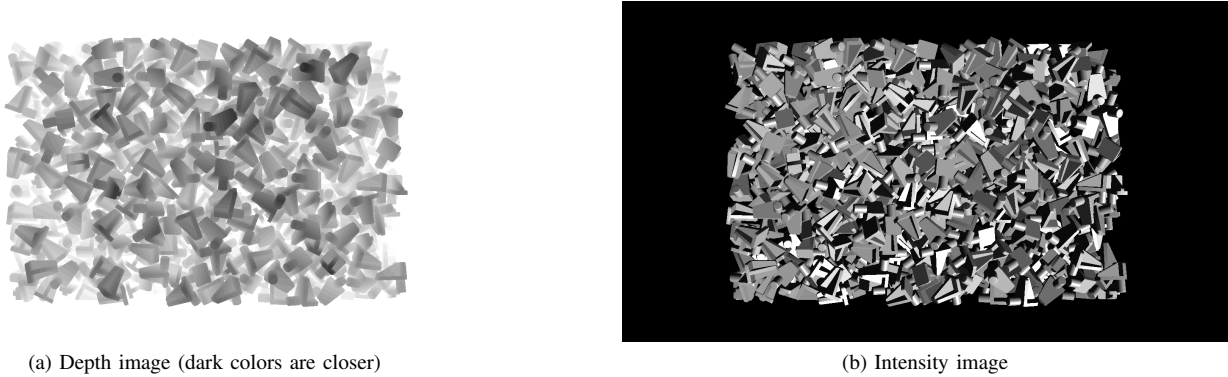

(b) Intensity image

Figure 3. An example of a bin containing parts, as simulated by the proposed method and used in the experiments. The bin itself was not rendered to allow better visibility, but was used in the physics simulation.

(a) Input CAD model (644 vertices)

(b) Object model decomposed into five convex parts using Approximate Convex Decomposition [14]. (185 vertices)

(c) Object model re-meshed using Approximated Centroidal Voronoi Diagrams [15] [16] to use as input for the generation of a PPF model. (1992 vertices)
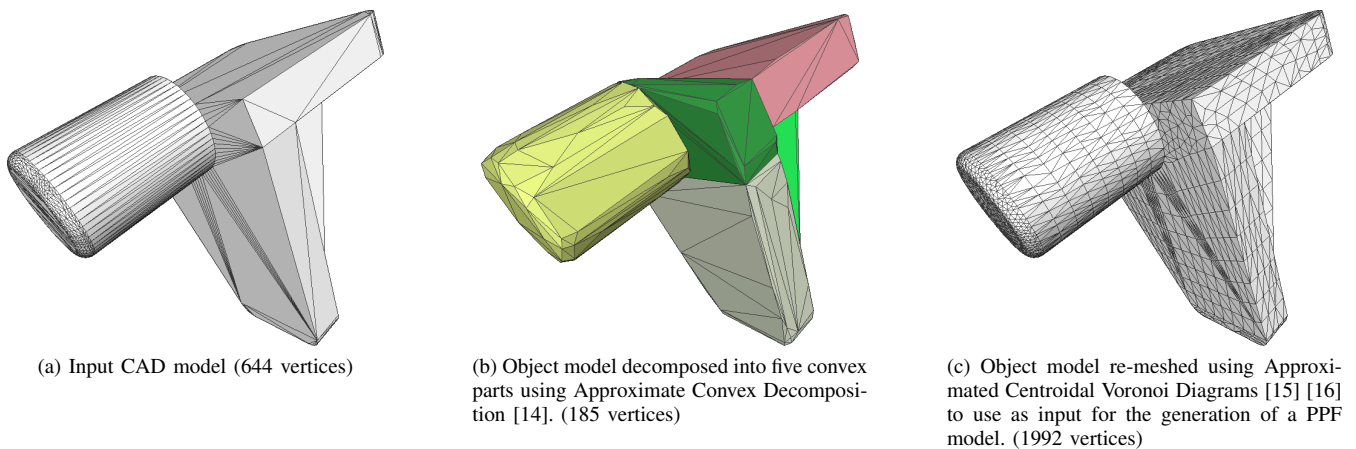
Figure 4. The used 3D model and the derived models used in the simulation, training and detection steps.

- The Intersection Over Union of the 2D axis aligned bounding boxes of the estimated and ground truth pose. This method could be used to compare the performance to algorithms that only provide a 2D position estimate. However, it obviously provides very little information on the depth and rotation accuracy. [35] [4]
- Picking success rate: the number of times a part is successfully grasped and extracted from the bin by the robot divided by the total number of attempts. This method requires implementation of the system using an actual sensor and robot. It can be argued that it is the most representative evaluation method to evaluate the actual system performance. The downside however, is that it is quite difficult to asses the detection algorithm performance, as variations in performance may be due to other parts of the system. It would also require users to have the exact same system, should they wish to compare their methods. [23] [28]
- The rotation and translation error, in absolute or relative measures. [21] [28] [31]) [32]

III. METHODS

A. Generating Synthetic Random Bin Picking Datasets

To obtain realistic test data that are representative for random bin picking, a method was developed to automatically generate synthetic random bin picking scenes from 3D object models. The 3D models used as input can be either the original CAD data, if available, or a reverse-engineered model from a combination of 3D scans of the object. A 3D model of a bin is loaded and a set of parts is dropped into it. The bullet physics library [36] is used to detect inter-object collisions and to calculate the resulting forces and dynamics. To achieve acceptable simulation times for the large number of objects being simulated, the objects are first decomposed into convex parts using Approximate Convex Decomposition [14] as shown in Figure 4b. The underlying reason for the speed gain is the fast methods that can be used to check for collisions between convex objects, versus the slower collision checking methods for general, free-form meshes.

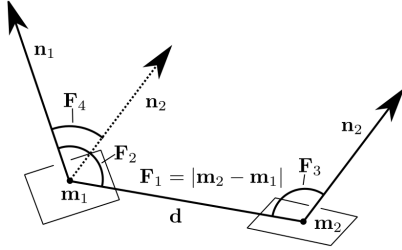Once all objects have reached a stable situation, meaning

Figure 5. Two surface points $\mathbf{m_i}$ and their normals $\mathbf{n_i}$ determine a point pair feature.
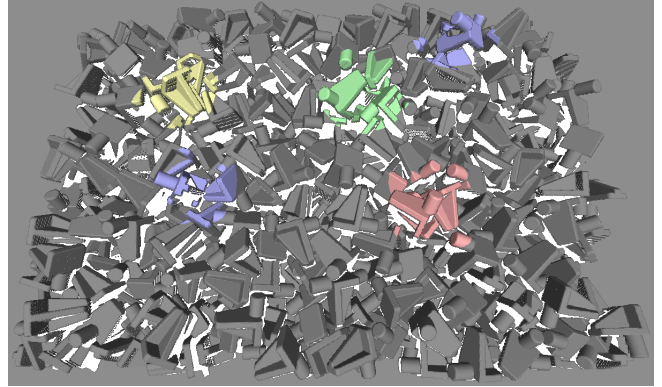

Figure 6. A view of the 3D mesh of the generated scene. The local neighborhoods of the five highest points in the scene are highlighted in color. When multiple hypotheses are tested, we propose picking the one that accumulated the highest number of votes in the detection process. To ensure better visibility, the gray scene points were very slightly translated downwards.

they are no longer in motion, their 6D poses are stored as the ground truth. To generate the corresponding range and intensity image (Figure 3a and 3b) and the surface mesh (Figure 6), the approximated models are replaced with the original models (Figure 4a) and the scene is rendered from the sensor's point of view.

B. Point Pair Feature based Object Detection

Point pair features describe the relative position and orientation of points on the surface of an object (Figure 5). Drost et al. define the feature [21] as: For two points $\mathbf{m_1}$ and $\mathbf{m_2}$ with normals $\mathbf{n_1}$ and $\mathbf{n_2}$, we set $\mathbf{d} = \mathbf{m_2} - \mathbf{m_1}$ and define the feature $\mathbf{F}$ as:

$$\mathbf{F(m_1, m_2)} = (\|\mathbf{d}\|_2, \angle(\mathbf{n_1}, \mathbf{d}), \angle(\mathbf{n_2}, \mathbf{d}), \angle(\mathbf{n_1}, \mathbf{n_2})) \quad (1)$$

where $\angle(\mathbf{a}, \mathbf{b}) \in [0 \quad \pi]$ denotes the angle between two vectors.

To learn a new object model, the four dimensional feature vectors are discretized and stored in a hashtable. When using a CAD model as input, some model preprocessing steps are usually needed, as CAD models typically have very non-uniform meshes (Figure 4a). We apply a re-meshing technique based on Approximated Centroidal Voronoi Diagrams [15] [16] to obtain a much more uniform mesh with a suitable resolution (Figure 4c). Note that this method works well both for input models that have too little vertices, or too many vertices. This method also offers an improvement in usability compared to rendering the object from a large number of viewpoints, calculating the resulting pointcloud and then stitching all generated pointclouds into a new object model. After re-meshing, a voxel based uniform subsampling procedure is used to retain the desired number of points.

The used detection procedure is nearly identical to Drost et al. [21]. To detect the object model in the scene, the uniform subsampling procedure is applied to the scene mesh. An additional subsampling procedure is used to select reference points among the scene points. The point pair features between these reference points and all other subsampled scene points are calculated, and their feature vector is compared to the ones stored in the object models

hashtables. All corresponding features are obtained from the model and vote for an object pose. In a pose clustering procedure, similar poses are grouped and their poses are averaged.

C. Heuristic Hypothesis Generation

Due to the high computational complexity of detecting objects using point pair features, it is not feasible to detect small objects in large pointclouds directly. Therefore, a heuristic approach is used to to drastically reduce complexity. A low pass filter is applied to the range image, and iteratively (five iterations in our application), the closest point is selected. After each iteration, the local neighborhood of the selected point is excluded. The assumption made is that the highest objects will be the easiest to detect and grasp, as they are less likely to be occluded by other objects. This method does not rely on segmentation (e.g. [32]) or feature (boundary) detection (e.g. [24]), which would strongly limit the applicability, as they depend on features being present and detectable. An example of the hypothesized object positions is shown in Figure 6.

D. Evaluation

The detection algorithm was evaluated on the synthetic scenes generated from a 3D object model. The detection is performed in the local neighborhood of the hypothesized positions. For each hypothesis, the cluster with the highest number of votes is selected, the contributing votes are averaged and the resulting pose is compared to the ground truth.

The translation error is defined as the euclidean distance between the detection and ground truth pose. The translational error can be expressed as a percentage of the object model diameter to allow comparison among objects of different sizes. The rotational error is obtained by calculating
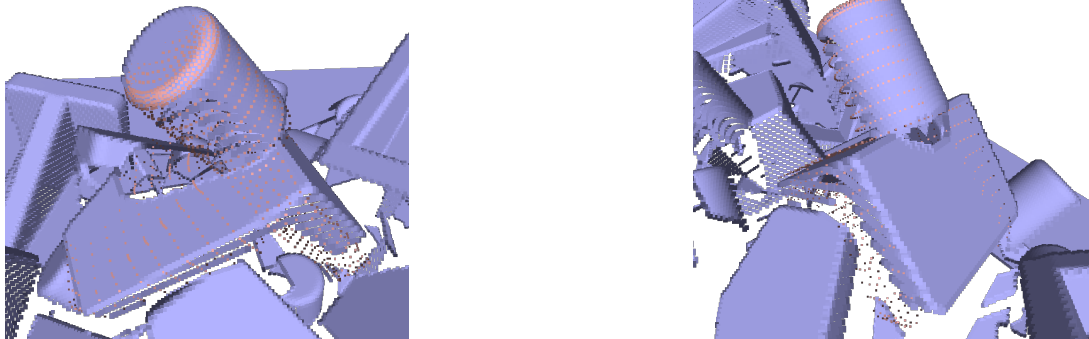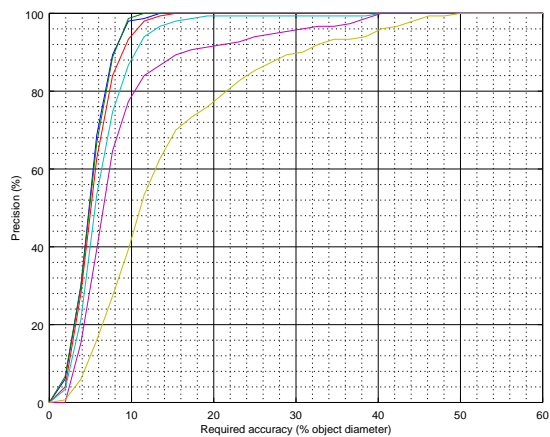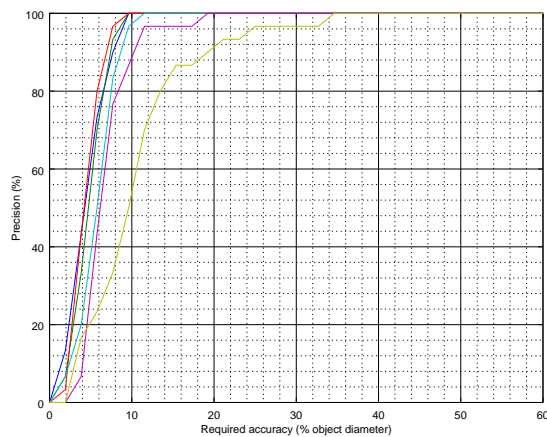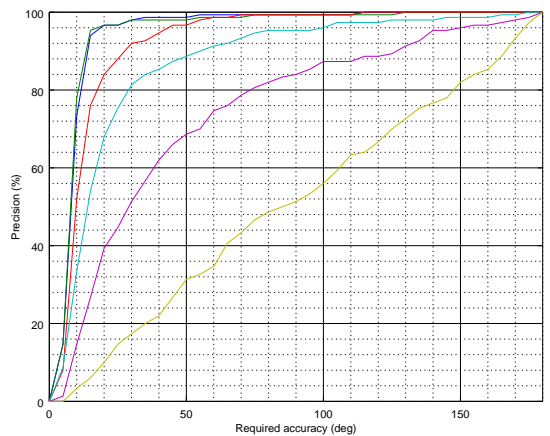
Figure 7. A detected object model (pink points) in the scene pointcloud (blue points), as seen from two viewpoints.
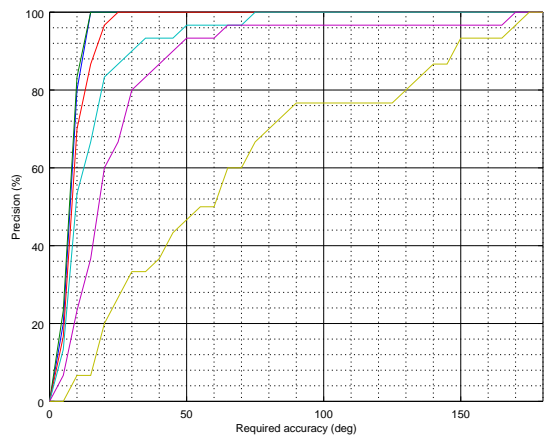

(a) Varying translation threshold (all detections)

(b) Varying translation threshold (detections with highest number of votes)

(c) Varying rotation threshold (all detections)

(d) Varying rotation threshold (detections with highest number of votes)

Figure 8. Detection precision as a function of increasingly tight error thresholds and an increasing amount of gaussian noise (color coded, with σ 0 to 5% of the object diameter).

the rotation matrix to align the frames of the detection and ground truth pose . The rotation matrix is converted to Angle-Axis representation and the resulting angle is used as an error metric. This means the rotation error is in the range $[0 \quad \pi]$.

As the procedure is automatic, it can be used in a closed loop to optimize the detection algorithm parameters.

IV. Experimental Results

A synthetic dataset of 30 bins is generated to test the point pair feature based object detection. The objects are dropped in an array arrangement of five by seven, at a time. This is repeated ten times per bin, so each bin contains 350 objects. The initial object orientation is random, and the initial velocity is zero. The objects fall into the bin as a result of the applied gravitational force. The rendering resolution was set to 1600x900 pixel. The used object has a diameter of about 5.2 and the bin dimensions are 60x40x30 (width x height x depth). The objects are dropped into a bin from a height of 60 to the bin floor, while the camera is positioned at 100 from the bin floor.

The five highest points in the scene were selected as hypotheses to run the detection algorithm on. The performance of the detection with respect to increasing levels of gaussian noise was tested (Figure 8). Note that no effort was made to reliably estimate the surface normals in the presence of noise, as this is not the key issue addressed in this paper. Adding such a method would drastically reduce the negative influence of the applied noise.

The point pair features were discretized into 30 steps for the normal angles, resulting in an angle of $12°$ per step. The distance between the point pairs was discretized into 20 steps, and the maximum distance corresponds to the object diameter. The resolution of the subsampling applied to both the model and scene was set to five percent of the object diameter. The experiments were performed using 20% of the subsampled scene points as reference points. All poses within a distance of 0.75 and with a rotation angle of 20 degrees or less are clustered. Their poses are averaged to increase the precision of the final pose estimate. Using these settings, the average detection time per hypothesis is 726 milliseconds. Note that the percentage of reference points can be reduced to increase speed, while still achieving high performance [21]. When 2% of the subsampled points are used as reference points, the required processing time is only 70 milliseconds per hypothesis.

The precision of using any of the five hypotheses was compared to using the one with the highest number of supporting points (points that vote for that particular pose). The improvements can be seen by comparing Figures 8a and 8c to Figures 8b and 8d. The results indicate that the number of votes received from supporting points is a valuable indicator of whether a detection is correct. Thus, testing more hypotheses and selecting the best one will result in better performance. As the processing time required increases linearly with respect to the number of hypotheses tested, it is up to the user to find a balance between performance and processing time.

V. Conclusions

An object detection and pose estimation algorithm based on point pair features was proposed. Several enhancements to optimize performance in a random bin picking context were made. Experiments on synthetic data show that we can drastically reduce the complexity of the problem by reducing the search space using simple heuristics. This offers huge benefits, as it increases robustness, speed and accuracy. The accuracy of the estimated poses was measured and found to be sufficient for our purpose, without requiring an additional ICP step.

References

[1] D. G. Lowe, "Local feature view clustering for 3d object recognition," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. IEEE, 2001, pp. I–682.

[2] M. Ulrich, C. Wiedemann, and C. Steger, "Cad-based recognition of 3d objects in monocular images." in *ICRA*, vol. 9, 2009, pp. 1191–1198.

[3] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa, "Fast object localization and pose estimation in heavy clutter for robotic bin picking," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 951–973, 2012.

[4] D. Damen, P. Bunnun, A. Calway, and W. W. Mayol-Cuevas, "Real-time learning and detection of 3d texture-less objects: A scalable approach." in *BMVC*, 2012, pp. 1–12.

[5] F. Tombari, A. Franchi, and L. Di Stefano, "Bold features to detect texture-less objects," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1265–1272.

[6] T. Rabbani, F. van den Heuvel, and G. Vosselmann, "Segmentation of point clouds using smoothness constraint," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. 5, pp. 248–253, 2006.

[7] F. Tombari and L. D. Stefano, "Object recognition in 3d scenes with occlusions and clutter by hough voting," in *Image and Video Technology (PSIVT), 2010 Fourth Pacific-Rim Symposium on*. IEEE, 2010, pp. 349–355.

[8] F. Tombari, S. Salti, and L. Di Stefano, "Performance evaluation of 3d keypoint detectors," *International Journal of Computer Vision*, vol. 102, no. 1-3, pp. 198–220, 2013.

[9] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *International Journal of Computer Vision*, vol. 89, no. 2-3, pp. 348–361, 2010.

[10] I. K. Park, M. Germann, M. D. Breitenstein, and H. Pfister, "Fast and automatic object pose estimation for range images on the gpu," *Machine Vision and Applications*, vol. 21, no. 5, pp. 749–766, 2010.

[11] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit, "Gradient response maps for real-time detection of textureless objects," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 876–888, 2012.

[12] W. Kehl, F. Tombari, N. Navab, S. Ilic, and V. Lepetit, "Hashmod: A hashing method for scalable 3d object detection," in *Proceedings of the British Machine Vision Conference (BMVC)*. BMVA Press, September 2015, pp. 36.1–36.12.

[13] R. Rios-Cabrera and T. Tuytelaars, "Discriminatively trained templates for 3d object detection: A real time scalable approach," in *Computer Vision (ICCV), 2013 IEEE International Conference on*. IEEE, 2013, pp. 2048–2055.

[14] K. Mamou and F. Ghorbel, "A simple and efficient approach for 3d mesh approximate convex decomposition," in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 3501–3504.

[15] S. Valette and J.-M. Chassery, "Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening," in *Computer Graphics Forum*, vol. 23, no. 3. Wiley Online Library, 2004, pp. 381–389.

[16] S. Valette, J.-M. Chassery, and R. Prost, "Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 2, pp. 369–381, 2008.

[17] T. Stahs and F. Wahl, "Object recognition and pose estimation with a fast and versatile 3d robot sensor," in *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, Aug 1992, pp. 684–687.

[18] E. Wahl, U. Hillenbrand, and G. Hirzinger, "Surflet-pair-relation histograms: a statistical 3d-shape representation for rapid classification," in *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*. IEEE, 2003, pp. 474–481.

[19] E. Wahl and G. Hirzinger, "Cluster-based point cloud analysis for rapid scene interpretation," in *Pattern Recognition*. Springer, 2005, pp. 160–167.

[20] S. Winkelbach, S. Molkenstruck, and F. M. Wahl, "Low-cost laser range scanner and fast surface registration approach," in *Pattern Recognition*. Springer, 2006, pp. 718–728.

[21] B. Drost, M. Ulrich, N. Navab, and S. Ilic, "Model globally, match locally: Efficient and robust 3d object recognition," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 998–1005.

[22] B. Drost and S. Ilic, "3d object detection and localization using multimodal point pair features," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*. IEEE, 2012, pp. 9–16.

[23] D. Buchholz, S. Winkelbach, and F. M. Wahl, "Ransam for industrial bin-picking," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*. VDE, 2010, pp. 1–6.

[24] D. Buchholz, M. Futterlieb, S. Winkelbach, and F. M. Wahl, "Efficient bin-picking and grasp planning based on depth data," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3245–3250.

[25] C. Papazov and D. Burschka, "An efficient ransac for 3d object recognition in noisy and occluded scenes," in *Computer Vision–ACCV 2010*. Springer, 2010, pp. 135–148.

[26] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka, "Rigid 3d geometry matching for grasping of known objects in cluttered scenes," *International Journal of Robotics Research*, vol. 31, no. 4, pp. 538–553, 2012.

[27] E. Kim and G. Medioni, "3d object recognition in range images using visibility context," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*. IEEE, 2011, pp. 3800–3807.

[28] C. Choi, Y. Taguchi, O. Tuzel, M.-Y. Liu, and S. Ramalingam, "Voting-based pose estimation for robotic assembly using a 3d sensor," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 1724–1731.

[29] C. Choi and H. I. Christensen, "3d pose estimation of daily objects using an rgb-d camera," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 3342–3349.

[30] ——, "Rgb-d object pose estimation in unstructured environments," *Robotics and Autonomous Systems*, vol. 75, pp. 595–613, 2016.

[31] O. Tuzel, M.-Y. Liu, Y. Taguchi, and A. Raghunathan, "Learning to rank 3d features," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 520–535.

[32] T. Birdal and S. Ilic, "Point pair features based object detection and pose estimation revisited," in *2015 International Conference on 3D Vision, 3DV 2015, Lyon, France, October 19-22, 2015*, 2015, pp. 527–535.

[33] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," in *Computer Vision–ACCV 2012*. Springer, 2012, pp. 548–562.

[34] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Symposium on geometry processing*, vol. 2, no. 3, 2005, p. 5.

[35] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "Pascal visual object classes challenge results," *Available from www. pascal-network. org*, 2005.

[36] E. Coumans, "Bullet physics simulation," in *ACM SIG-GRAPH 2015 Courses*, ser. SIGGRAPH '15.   New York, NY, USA: ACM, 2015.