FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 42 (2017) No. 3
ISSN 0867-6356
DE GRUYTER OPEN
DOI: 10.1515/fcds-2017-0011 e-ISSN 2300-3405

3D OBJECT DETECTION AND RECOGNITION FOR ROBOTIC GRASPING BASED ON RGB-D IMAGES AND GLOBAL FEATURES

Witold CZAJEWSKI*, Krzysztof KOŁOMYJEC*

Abstract. This paper describes the results of experiments on detection and recognition of 3D objects in RGB-D images provided by the Microsoft Kinect sensor. While the studies focus on single image use, sequences of frames are also considered and evaluated. Observed objects are categorized based on both geometrical and visual cues, but the emphasis is laid on the performance of the point cloud matching method. To this end, a rarely used approach consisting of independent VFH and CRH descriptors matching, followed by ICP and HV algorithms from the Point Cloud Library is applied. Successfully recognized objects are then subjected to a classical 2D analysis based on color histogram comparison exclusively with objects in the same geometrical category. The proposed two-stage approach allows to distinguish objects of similar geometry and different visual appearance, like soda cans of various brands. By separating geometry and color identification phases, the applied system is still able to categorize objects based on their geometry, even if there is no color match. The recognized objects are then localized in the three-dimensional space and autonomously grasped by a manipulator. To evaluate this approach, a special validation set was created, and additionally a selected scene from the Washington RGB-D Object Dataset was used.

Keywords: 3D object detection and recognition, Kinect, point cloud analysis, RGB-D images, VFH, CRH, ICP

1. Introduction

The understanding of the observed environment based on computer registered images and, in particular, finding the number, the type, the properties and finally the pose of objects within this environment is one of the most profound problems and goals that face the machine vision community. Whereas the analysis and interpretation of images and extraction of key information contained therein are most often intuitive, effortless and instantaneous for humans, it is one of the crucial competencies that computer systems still

* Instytut Sterowania i Elektroniki Przemysłowej, Politechnika Warszawska,
ul. Koszykowa 75, 00-662 Warszawa, email: w.czajewski@isep.pw.edu.pl,
k_kolom@poczta.onet.pl

lack today. Despite numerous attempts to mimic human vision capabilities, computer algorithms in this field are in their early infancy due to the enormous complexity of the process and superficial knowledge of its progress in the human brain.

One of the key issues associated with the manipulation of objects is their detection, recognition and localization in the visual scene. The latter task seems to be particularly difficult, however, it became solvable in nearly real time with the application of depth images provided by sensors like the Microsoft Kinect. The Kinect-generated RGB-D image does not only contain the usual three color components of the observed scene for each pixel, but it also holds the distances of the observed points from the sensor. This opens up a whole new range of possibilities for analysis and processing of information, but at the same time, it creates new challenges that require new solutions.

In this paper, experiments and implementation results of a vision system for the detection and recognition of three-dimensional objects in RGB-D images provided by the Microsoft Kinect sensor are described. The focus is on using a single image for this purpose, however, the analysis of several consecutive frames is considered as well. Moreover, the recognized objects are localized in the 3D space and autonomously grasped by a manipulator. The experiments are conducted in a domestic-like environment and the objects to be recognized and manipulated are everyday items like soda cans, coffee mugs, etc. The applied algorithm classifies the observed items at two stages; at first, using only geometrical information and subsequently, analyzing visual cues. If an observed object is successfully recognized at the first stage, its final match is determined by color histogram comparison. This enables distinction of objects of similar geometry but different visual appearance, like soda cans of various brands. It also ensures correct geometrical categorization of objects of known geometry but of different appearance (a soda can of a brand not included in the object dataset will still be recognized as “some soda can”).

Eventually, the object is localized and grasped by the manipulator. In order for the above scenario to work, a database of objects’ point clouds and their RGB images is necessary. Objects outside of the database still might be recognized in the first phase, as long as their geometry is similar to an object in the database, but they will be most likely rejected in the color matching phase. A special objects’ database was created for the study, and the performance of the point cloud analysis part of the algorithm was additionally verified on a selected scene from the Washington RGB-D Object Dataset [26].

2. Related work

3D objects recognition methods have been extensively investigated in the last decade, but there are still many unresolved issues, opportunities and challenges. The development of affordable 3D acquisition systems (e.g Microsoft Kinect in 2010), sparked great interest in the field of 3D object identification, and the use of range images for object recognition has grown significantly. Numerous advantages, like easier segmentation, more accurate 3D pose estimation and new geometrical features shifted the focus of researchers from the classical 2D approach to the analysis of color-and-range data.

In 2010, Hinterstoisser et al. proposed a template based recognition method, which used a large RGB-D template database containing objects registered from different angles [21] and Papazov and Burschka developed a method for object recognition in cluttered scenes using point clouds [35]. Less than a year later, the Point Cloud Library – an open source library of algorithms for 3D point cloud processing – was introduced [38]. Since then, several approaches to object recognition using 3D data have been proposed. Rusu et al. developed a global descriptor named VFH [39], following its local predecessor FPFH [40]. This idea was soon extended in [2] to form a Clustered Viewpoint Feature Histogram, which is more robust to occlusions. The problem of 6DOF object pose estimation was solved by the Camera Roll Histogram Descriptor in [8]. Further research on the features of the 3D objects resulted in the appearance of OUR-CVFH descriptor based on Global Unique Reference Frames [5] computed for each cluster. The work described in [44] shows the application results of the Signature of Histograms of OrienTations (SHOT) – a local descriptor inspired by SIRF, which represents an intersection between Histograms and Signatures with high computational efficiency. In [3], a correspondence grouping algorithm and a fast descriptor matching accomplished by the FLANN Library [32] was proposed. In addition, an Iteractive Closest Point (ICP) algorithm for pose refinement introduced in [12] and extended in [48] was applied to point clouds. Finally, to reduce the number of false positives, a Hypotheses Verification (HV) step introduced in [3] was extended in [4] by additional hypotheses generation pipelines with the use of SIRF and OUR-CVFH descriptors. Aldoma et al. adapted the solution from [4] to a multi-view recognition system [7]. In [43] Tang et al. demonstrated advantages of simultaneous segmentation, object detection and 3D pose recovery in extracted global and local feature models acquired from multi-view color images and point clouds. His approach was improved in [47] by including additional cues: SIRF features, color and shape characteristic in hypotheses evaluation stage. Recently, Prankl at al., in [37], developed a method for creating full 3D models by fusing partial models obtained by different reconstruction sessions, and Narayanan and Likhachev, in [34], proposed a Perception via SeaRCH algorithm for detection and localization of multiple mutually occluded objects.

A recent interest in application of deep convolutional neural networks to 3D object recognition emerges as a natural consequence of their enormous successes in the field of 2D image analysis. Initially, the depth channel was treated on par with the RGB ones [9], but this approach could not grasp the entire complexity of the available 3D information. Therefore, it was soon enhanced to learn from volumetric data represented by occupancy grids [31], [46]. Another issue tackled by the CNN community was developing solutions for obtaining invariance to different kinds of transformations [28] and scale changes [18]. In [16], an adaptive, multi-scale CNN architecture was proposed to jointly perform depth prediction, surface normal estimation and semantic labeling. In contrast to these methods, a very recent approach described in [36], uses depth information to drive the scale selection of the convolutional filters. The deep learning methods have already outperformed most, if not all, classical approaches to model alignment [17], feature learning [45] and object recognition in cluttered 3D scenes [42].

The success of CNNs would not have been possible without large-scale annotated 3D datasets and deep learning frameworks. The most popular 3D datasets include the Washington Dataset [26], the Princeton ModelNet [46] and ShapeNets [14]. Deep learning frameworks such as Caffe [22], Theano [11], Torch [15] or TensorFlow [1] enable easy creation and training of various CNN architectures. They make extensive use of CUDA or OpenCL technology on GPUs to boost the processing speed.

Although deep learning methods are starting to outperform and replace classical ones, they require substantially bigger training sets and far more powerful and expensive computer resources, especially during the training phase. For example, the winning team of the Amazon Picking Challenge 2015, that did not use deep learning, needed just 161 training images to build a successful classifier of 25 objects [23], while the next year's winners, that applied CNNs, used 20 thousand images and a powerful and expensive NVIDIA TITAN X GPU [19]. Unfortunately, very few authors report training times of their CNNs, which can reach several days even on powerful GPUs. The system applied in the described study, on the other hand, requires a relatively small training set and is able to learn it within seconds or minutes. It runs in near real time on a single CPU core, so it could be also used in embedded solutions, not only for recognition, but for training as well, which is unthinkable in the case of CNNs. Therefore, in light of the above, evaluation of different approaches should not be based exclusively on their pure performance in the recognition stage (in terms of accuracy and speed), but also on the effort put during the training phase (to create a custom training set containing tens of thousands of samples) and its cost (primarily in terms of training time and/or direct equipment cost). If the recognition accuracy is the only goal, CNNs will definitely be the answer, but for cost-effective solutions, classical approach should still be considered.

3. Overview of the experimental setup

The objective of the described system is to identify certain objects placed on a flat surface, find their pose and grasp them with a manipulator (see Figure 1 for a general diagram of the system). The entire point cloud processing is performed with the Point Cloud Library (PCL) [38] and for the classical 2D image analysis (histogram comparison) the OpenCV library is used. The processing time, on a single core of a medium class CPU, is about 1-2 seconds per object in the scene (in verbose mode), with a dataset of 1000 model objects.

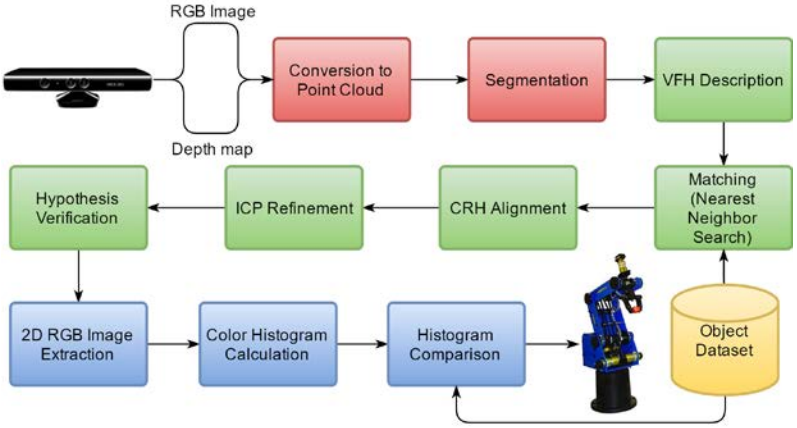

Figure 1. A general functional flow block diagram of the robotic vision system

An RGB-D image captured by the Kinect is first converted into a point cloud that is then filtered and segmented into separate clusters. The objects resting on a planar surface are described with a global Viewpoint Feature Histogram (VFH) [2], [30] that is matched with VFH histograms in the database. For promising candidates a Camera Roll Histogram (CRH) [5], [8] is then calculated in order to establish the right pose, which is further refined with the Iterative Closest Point (ICP) algorithm [48] to get the best match. The final stage of the analysis is the global Hypotheses Verification (HV) [3] during which object hypotheses are verified so as to reject false detections. Once an object is successfully recognized based purely on its geometry, a classical 2D image analysis is applied in order to classify objects of similar geometry and different visual appearance/colors. An RGB view of the observed object is compared by color histogram matching to all the objects in the database belonging exclusively to its geometrical category. Finally, the object is grabbed by the manipulator.

The approach presented above is based on improving and combining several prior works, mainly the ones presented in [39], [2], [8], [3], [32], [12], [33]. In contrast to many other methods (especially those based on deep learning), in this approach, a very large database of objects to learn a good classifier is not needed. Similarly to [33] global VFH descriptors are applied to identify objects, but a very rarely used CRH descriptor is added (less than 10 references in IEEEXplore database) to obtain their right poses. A slightly similar approach was applied in [20] where both descriptors were merged into one. Both solutions, however, require a large number of training objects acquired at nearly all possible poses in order to identify an object in an arbitrary pose. In the approach applied in the described study, both descriptors are dealt with independently and sequentially, and thus, a lot fewer object models are needed. Color and geometry in the recognition pipeline are also separated, unlike in [33], where both are combined. Such an approach allows recognition of objects of known geometry, but of unknown or altered colors, e.g. since the system has a model of a Cola can, it will classify any soda can as a soda can, but it will just not be able to tell its brand without additional color models.

All the VHF descriptors are organized in a form of a kd-tree so adding a new model to the database requires no time- and resource consuming processing, unlike in the case of deep neural networks, and their comparison at the beginning of the pipeline is fast. Moreover, multiple angles resulting from the CRH alignment phase are being considered in the applied approach, so the probability of skipping a valid candidate is significantly reduced. Additionally, ICP matching is applied for each of these candidates, which altogether, ensures very precise verification of each potential solution initially found in the VFH comparison phase.

The shortcomings of the proposed approach, which are inherent to the applied methods, include problems with proper object segmentations in heavily cluttered scenes, low capability of detecting partially occluded objects and a requirement of setting problem-specific thresholds.

4. Image acquisition and point cloud extraction

A point cloud of the entire scene, generated by the Kinect sensor, has to be preprocessed in such a way that only relevant points belonging to unique objects of interest are grouped together, and stored for later processing. It is relatively easy to detect and remove certain known structures like tables or the floor or any other planar surface of a major size. In this way, all the remaining points can be clustered into separate objects. The pipeline of this process is shown in Figure 2 and processing results at each stage are depicted in Figure 3.

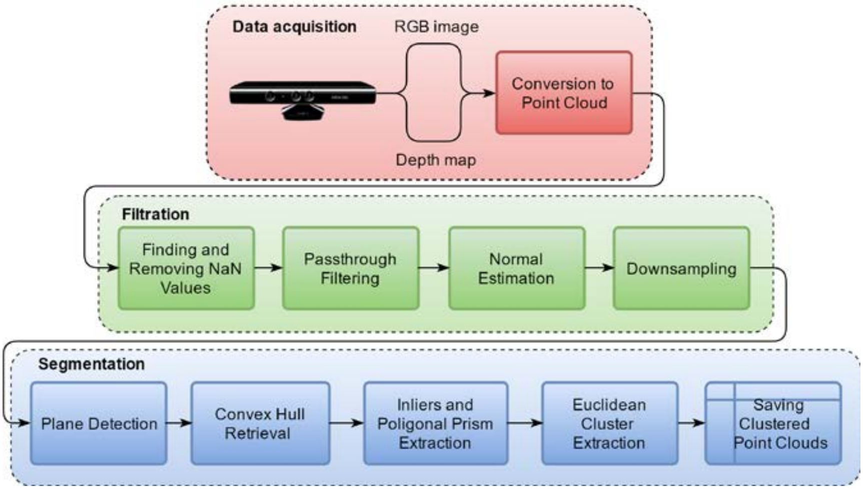
Data acquisition
RGB image
Conversion to Point Cloud
Depth map
Filtration
Finding and Removing NaN Values
Passthrough Filtering
Normal Estimation
Downsampling
Segmentation
Plane Detection
Convex Hull Retrieval
Inliers and Poligonal Prism Extraction
Euclidean Cluster Extraction
Saving Clustered Point Clouds

Figure 2. Processing pipeline of object detection and point cloud extraction

A 640 by 480 pixels RGB-D image acquired by the Kinect sensor is processed with the PCL library and converted into a point cloud containing exactly 307 200 points (see Figure 3a). It requires initial filtration before it can undergo the segmentation process. At first, points that are of no use for 3D processing as having no information about depth (NaN) due to occlusions, transparent or specular surface etc. are removed. Subsequently, a passthrough filter is used to remove all the points lying outside of the user-defined range (see Figure 3b). Experiments have shown that reliable recognition of small objects is not possible beyond 1.5 m from the sensor and, thus, the passtrough filter cut-off distance along the Z axis was set to this value. For the remaining points in the cloud, normal vectors are computed. Despite the removal of some of the points, the point count in the cloud is still high, which may slow down further processing. Optionally, the point cloud may be downsampled using a voxelized grid method in order to increase performance. Nevertheless, downsampling may have a negative impact on recognition quality, especially in the case of small or distant objects. Therefore, in the system used in the described study, both versions of the point cloud are used; the full point cloud for descriptor computation and the reduced one for segmentation, centroid calculation and precise pose estimation with the ICP algorithm.

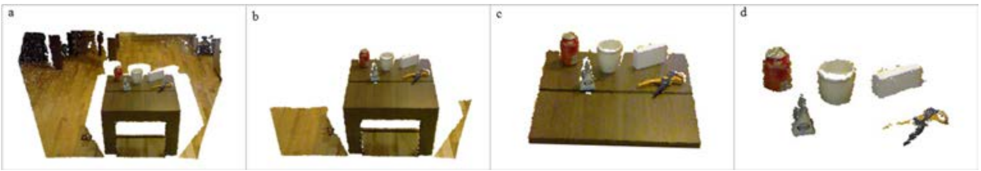

Figure 3. Object segmentation stages (point count in brackets): a) initial point cloud (307 200), b) after removal of NaN points, outliers and passthrough filtering (149 262), c) dominant plane with touching objects (81 764), d) finally segmented objects (13995)

After filtration is completed, the segmentation process may be applied. Since a large, dominant plane in the scene can be easily detected with a RANSAC algorithm, it is used as a supporting plane to segment object standing on top of it. All the points not connected to that plane are removed (see Figure 3c). By subtracting the points belonging to the plane itself from the remaining cloud, all the objects of interest are extracted (see Figure 3d). Finally, the cloud is segmented into clusters corresponding to separate objects.

5. Dataset construction

The dataset created for the described study contains visual and depth images of 5 different objects taken from different elevations in a way similar to the one described in [26]. The dataset was acquired with a Kinect sensor and a computer controlled turntable. It made the registration process fairly quick and easy – only the Kinect elevation angle was controlled manually [24].

Each object in the dataset was scanned by the Kinect sensor from a distance of approximately 0.8 m at 5 different angles of inclination on a turntable at 25 angular positions (approx. every 15 degrees). Moreover, three objects (i.e. processed cheese, box and soda can) were acquired twice: in horizontal and vertical orientation respectively. Eventually, 1000 models, each consisting of an object's point cloud and a 2D RGB image were registered (see Figure 4 for samples). For each of them, global VFH and CRH descriptors as well as a hue-saturation color histogram were calculated. In order to advance later descriptor search, all the VHF descriptors were organized in the form of a k-dimensional tree (KDT). The dataset is not as rich as the one mentioned in [26], both in terms of number of objects and, what is important, samples per object. Nevertheless, experiments have shown that it is sufficient in the case of regular items like soda cans or coffee mugs, however more complex objects like insulation stripper would require many more samples to reach a comparable recognition rate.

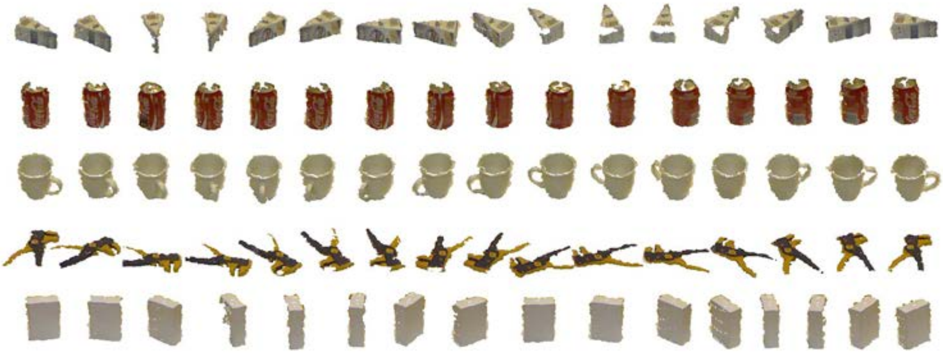

Figure 4. Examples of point clouds from the dataset created for the described study: processed cheese, soda can, coffee mug, insulation stripper, box

6. Object description and recognition

6.1. Selection of descriptors

The existing, classical methods for 3D object recognition can be divided into two general categories: global feature-based methods and local feature-based methods. Each of them represents a different approach to the description of the observed objects' geometry.

Local descriptors are calculated based on geometrical features extracted from the individual keypoints found on the object. Although this approach does not require prior segmentation and generally deals well with clutter and occlusions [10], it usually involves time-consuming keypoint detection, description and comparison.

Global descriptors, on the other hand, are based on the geometry of the entire object [2]. They require a prior segmentation of the object from the scene and may ignore minor shape details, but they have the ability to generalize an entire object with a single feature vector. As a result, the recognition process is faster and more robust to noise, which is important in applications designed to operate in near real-time [33]. Hence, in the application described in this paper, a global recognition method based on VHF and CRH descriptors was used.

6.1.1. Viewpoint Feature Histogram (VFH) Descriptor

The Viewpoint Feature Histogram (VFH) [2], [39] is based on the local Fast Point Feature Histograms (FPFH) [40]. It consists of two parts: the viewing direction component and the extended FPFH component. For the calculation of the first part of the descriptor, it is necessary to find the centroid of the object and a normalized vector between the centroid and the viewpoint. Next, the angular difference between this vector and normal vector for each point in the point cloud is determined. Each of its three angles is binned into a 45-bin histogram. The remaining part is calculated in the same way as in the case of the FPFH descriptor. As a result, a global descriptor of a total length of 308 bins is created. Since the bins are normalized using the total number of points in the cloud, the VFH descriptor becomes scale invariant. According to [39], this descriptor proved to be very effective and therefore was used in the described system.

6.1.2. Camera Roll Histogram (CRH) Descriptor

As global descriptors are roll-invariant and one of the goals of the applied system is to retrieve the full 6DOF pose of the recognized objects, the sixth degree of freedom must be determined with another descriptor: Camera Roll Histogram. It is calculated as follows: for every point, its normal is projected onto a plane orthogonal to the vector defined by the camera center and the centroid of the point cloud. Then, the angle between the projected normal and the up vector of the camera is computed and added to the histogram. For a resolution of 4 degrees, the CRH descriptor will have 90 bins [2].

6.2. Object recognition pipeline

Object recognition pipeline (see Figure 5) starts with comparing the unknown object's VFH descriptor that was just calculated with all the VHF descriptors of models from the dataset. Matching is performed using a k-Nearest Neighbor search from the FLANN library with a Chi-Square metric. It produces approximate results, but is much faster than a full search, especially with large datasets. In the case of the described study, *k* is equal to 5, but when four nearest candidates in the entire dataset belong to a single geometrical category, as the fifth one the next nearest from any other category is selected. This is to avoid situations where the object being recognized might accidentally and incorrectly resemble an object of a different geometrical category. In such a case, it will be rejected later during the processing. By taking the fifth candidate from another geometrical category, the chance of finding the right model is increased.

The value of the calculated distance *d* between the object and a model determines further workflow. The thresholds for *d* used in the described algorithm were established experimentally. A distance of less than 40 means that the object being recognized and the model are almost identical and the object was correctly categorized. There is no need for CRH calculation and ICP fitting as both point clouds are already well aligned. This may happen if an object is observed from nearly the same viewpoint as during the dataset creation (a common occurrence in very rich datasets). In the case of the rather small dataset applied in the study, there was no situation where models from different categories would report a distance smaller than 40. It did happen, however, for a few symmetrical models belonging to a single category (soda can).

If the above mentioned distance *d* is greater than 200, it is assumed that no model in the dataset is close enough to the object being examined and the recognition result is false. No further processing of this object is attempted.

The third case, when $40 \leq d \leq 200$ is the most common one for objects that actually have a model in the described, rather small, dataset and should be recognized. Such a value of *d* denotes uncertainty about the VFH-based object matching to a model. It could be due to small misalignment of the two point clouds belonging, in fact, to the same object or due to some similarity of two, in fact different, geometries. It is quite a common occurrence that among 5 nearest neighbors there are models from different categories and therefore further steps are necessary to select the right model or reject all of them. To this end, CRH descriptors are used. A CRH descriptor of the current object is aligned with the CRH descriptor of a model, which produces a list of probable common roll angles [2]. Then the object's original point cloud is rotated by all the roll angles respectively and its pose is refined with the use of the Iterative Closest Point (ICP) algorithm. It will attempt to realign the clouds in order to improve the transformation until the termination condition is satisfied (maximum iterations or error threshold). These clouds that cannot be aligned are disregarded and the remaining ones are subjected to two more verification steps. The first one is based on geometrical dimensions of the objects and is applied due to the fact that the VFH descriptor is scale invariant. Simple constraints set on objects width and height and their ratio will eliminate situations where, for example, any cylindrical object is mistakenly identified as a soda can. Those models that pass the dimensions test are very strong candidates for final geometrical recognition result. All these hypotheses about the right match undergo the last verification procedure (HV) described in [3]. The first model that passes this stage is considered as the recognition result. If a given model fails at any stage, it is disregarded and a next candidate is examined.

Successful shape recognition is followed by the very last stage in the entire processing pipeline: color matching, where visual appearance of objects is compared by color histogram matching.

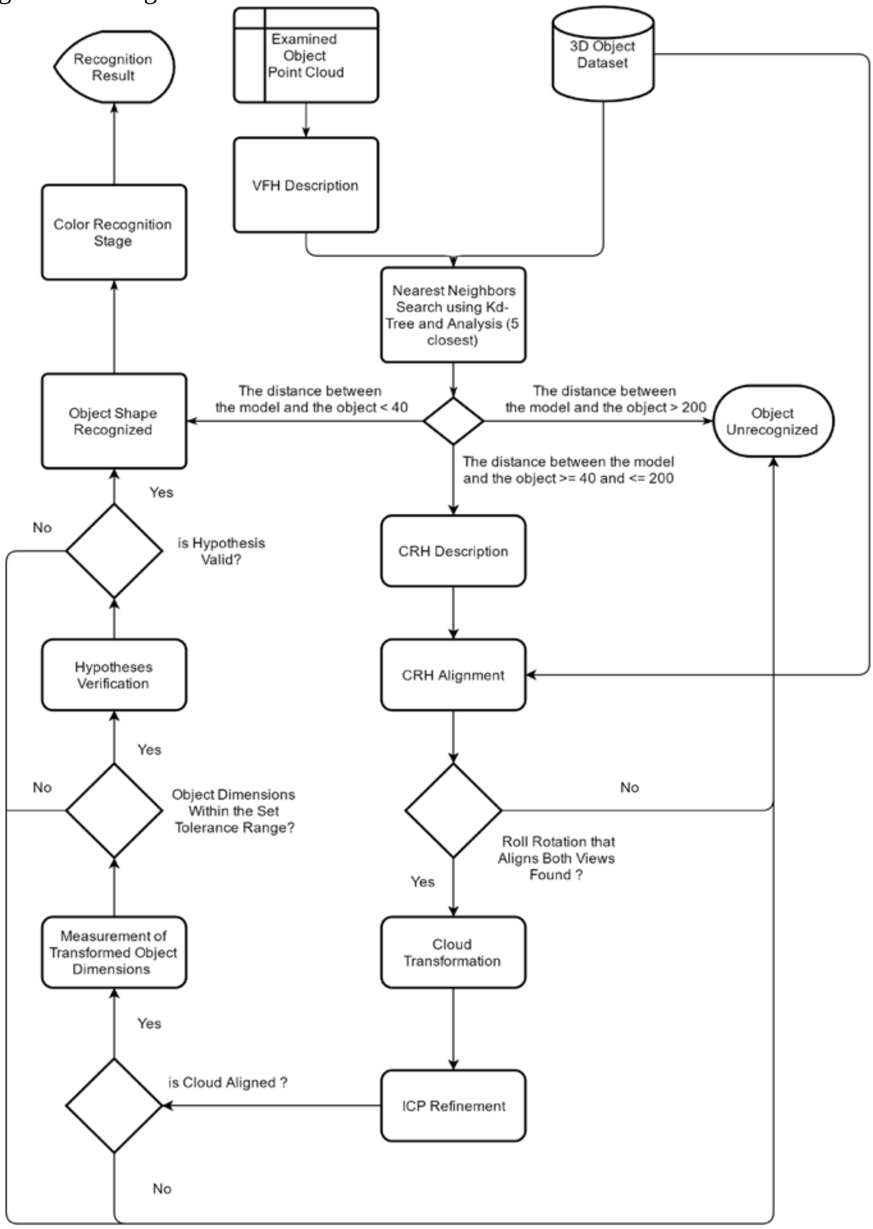
Recognition Result
Examined Object Point Cloud
3D Object Dataset
VFH Description
Color Recognition Stage
Nearest Neighbors Search using Kd-Tree and Analysis (5 closest)
The distance between the model and the object < 40
The distance between the model and the object > 200
Object Shape Recognized
Object Unrecognized
The distance between the model and the object >= 40 and <= 200
Yes
No
is Hypothesis Valid?
CRH Description
Hypotheses Verification
CRH Alignment
Yes
No
Object Dimensions Within the Set Tolerance Range?
No
Roll Rotation that Aligns Both Views Found ?
Yes
Measurement of Transformed Object Dimensions
Cloud Transformation
Yes
is Cloud Aligned ?
ICP Refinement
No

Figure 5. Simplified object recognition pipeline

6.3. Color matching

In contrast to the approach proposed in [29], the color matching algorithm applied in the described study, is based on histogram comparison. A 2D hue and saturation histogram is computed for the color view of the recognized object. Next, it is compared with all the histograms stored in the database only for that particular geometrical category of objects. Histogram correlation and Bhattacharyya distance are used as similarity metrics and the model holding the highest value thereof, if above a certain threshold, is assumed as the final recognition result (e.g. geometrical category: soda can, color subcategory: Coca-Cola).

The undisputable advantage of the approach described above is its speed. Unfortunately, it is influenced by relatively low precision and resolution of the Kinect sensor and, most importantly, lighting conditions. A minor difference in light color, during the training and recognition stages may lead to false color classification. The experiments have shown that histogram comparison works nearly perfectly in the case of objects of clearly different colors, but it fails when the colors are similar. Due to the above, the color classification stage is separated from the geometry identification stage so as not to affect the latter.

7. Experiments and results

In order to verify the effectiveness of the 3D object recognition approach described above, four sets of experiments were conducted. The focus was set on the recognition of point clouds, but the color identification was tested as well.

In the first experiment, objects from the dataset created for the study were captured in various arrangements and the quality of recognition was evaluated. Next, an attempt to identify several soda cans of different brands was made. In the third experiment, a robotic manipulator was added to the system and it executed a grasping task upon successful recognition of objects. In the final experiment, the performance of the point cloud based object recognition system was verified on a selected scene from the Washington RGB-D Object Dataset [26]. Although the system performed relatively well in all the experiments, it would require adaptation of its parameters for more populated and cluttered scenes than the ones being tested. The meta-parameters of the applied algorithm were determined heuristically based on a number of similar scenes with the same objects, until satisfactory results were obtained.

7.1. Validation dataset

The performance of the recognition system was first verified on the validation dataset containing objects described in section 5. Five objects were captured in 12 different arrangements, each at 3 different distances (see the first 2 rows of Figure 6 for a few examples). This yields 36 different scenes and each of them was captured 5 times, which results in 180 test frames and 900 object classification attempts.

Average classification quality measures are given in Table 1. The overall results for all 5 objects are quickly deteriorating with distance, however recall is dropping substantially in contrast to precision that shows only a slight decline. This is caused by many false negatives for the insulation stripper, especially at larger distances. Apparently, this tool does not have enough models in the dataset and/or is too small and/or too complex to be recognized successfully by the described system. Other, bigger and more regular objects do not suffer from so many negative detections and, if the stripping tool is excluded from the statistics, the recognition measures reach satisfactory levels, especially under the distance of 1 m.

Table 1. Average classification quality for 5 (or for 4, without the insulation stripper, in brackets) objects for different object-Kinect distances

Distance [m]	Precision	Recall	F1
0.50 – 0.80	0.97 (0.99)	0.88 (0.94)	0.92 (0.96)
0.81 – 1.10	0.97 (0.97)	0.78 (0.85)	0.86 (0.91)
1.11 – 1.50	0.87 (0.96)	0.58 (0.71)	0.66 (0.79)

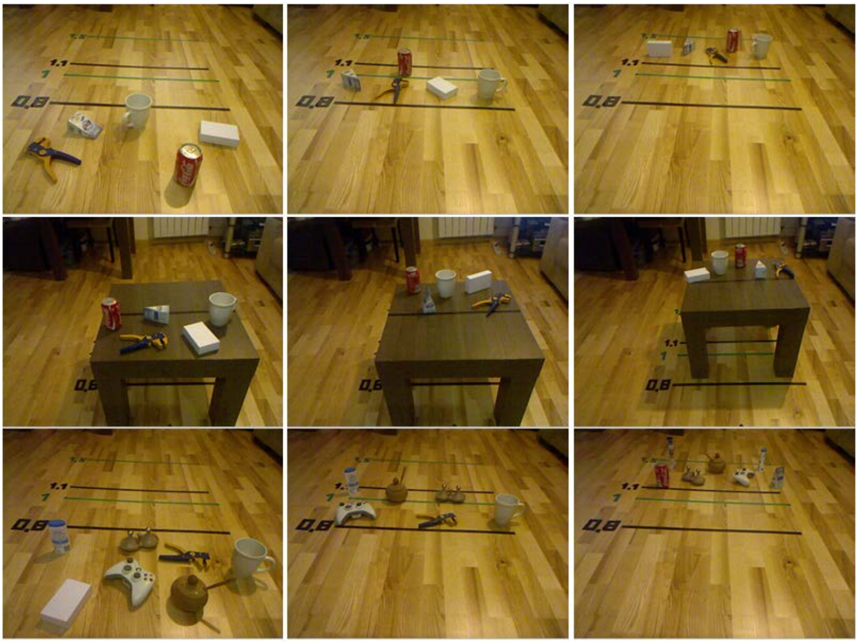

Figure 6. Examples of test scenes

In the second experiment, the recognition system was evaluated on 18 scenes containing additional objects outside of the dataset (see the bottom row of Figure 6 for a few examples). A total of 595 recognition attempts were executed. The soda can, as the best-recognized object, was not used at the shortest range, and the insulation stripper and the box were not used at the longest range. This time, due to more numerous false positives, precision of the system dropped significantly, especially at long ranges. Nevertheless, for distances up to 1 m, the recognition measures are still satisfactory (see Table 2).

Table 2. Average classification quality for scenes containing additional objects outside of the dataset for different object-Kinect distances

Distance [m]	Precision	Recall	F1
0.50 – 0.80	0.81	1.00	0.88
0.81 – 1.10	0.87	0.96	0.88
1.11 – 1.50	0.70	0.67	0.69

7.2. Color identification

Histogram-based color comparison tests were conducted on detected coffee mugs of uniform colors and soda cans with dominant colors in more or less constant lighting conditions. For 3 coffee mugs, 6 different scenes were captured 5 times each, which results in 90 color classification attempts. Only in 3 cases, a light green mug was mistakenly classified as a light blue one. The remaining classification results were correct.

In the next test, 3 soda cans (Cola, Sprite, Fanta) were identified. They are not uniform in color, but the distribution of dominant colors makes them look quite different. Again, 90 classification attempts were evaluated. The results were perfectly good in all but one scene, where a can was in a horizontal position facing the Kinect with its silver lid. In this sole case, a Cola can was identified as a Fanta.

The last experiment involved two cans of similar appearance (Dr Pepper and Coca-Cola). They are both red, with a slightly different hue and different, although white, logos. As expected, in this case, the histogram matching classifier failed and the recognition results seemed to be completely random. Thus, this technique of comparing images is an effective and fast tool only if examined objects have significantly different color distributions and its performance will deteriorate with growing number of relevant objects in the dataset.

7.3. Robotic vision system

The 3D recognition system was also tested in a robotic environment. The objective of the test was to detect and identify certain objects placed on a flat surface, find their pose and grasp them with a manipulator. The test objects (a soda can and a box) were placed at a short distance from the Kinect so that the recognition rate was high. After identification and localization of objects, they were grasped and handed over to the operator. Due to poor hand-eye calibration and/or minor errors in pose estimation, sometimes the gripper would miss the object by a small distance. This was particularly apparent in the case of a soda can, where incorrect gripper position would cause gripper fingers to slip over the surface of the can. Naturally, this problem did not occur for the box due to its flat shape. The experiment can be watched at: https://youtu.be/lPQZIrdIo0g

7.4. The Washington Dataset

The final verification of the described approach was performed on one of the scenes from the Washington RGB-D Object Dataset [26]. One hundred frames numbered from 21 through 120 from the *table_small_1* sequence were used. Each scene contains 4 objects standing on top of a table and some objects in the background. The objects to be recognized include a bowl, a coffee mug, a soda can and a cereal box. The dataset contains a huge number of their models (given in brackets), among which there are 6 slightly different bowls (3884 models), 8 coffee mugs (4866), 6 soda cans (3553) and 5 cereal boxes (2929). The total number of captured partial models per object category is enormous in comparison to the dataset having a mere 125 or 250 models per object.

In the first experiment, all of these models were used for object recognition (the color identification was skipped in all tests with this dataset) in 100 frames from the sequence. Initially, no geometrical restrictions on objects' dimensions described in Section 6.2 were applied. The results are given in Table 3 (before slash).

Table 3. Classification quality for all models without / with geometrical constraints

	Bowl	**Coffee mug**	**Soda can**	**Cereal box**	**Average**
Precision	1.00 / 1.00	0.82 / 0.83	0.82 / 0.99	0.54 / 0.77	0.79 / 0.90
Recall	0.95 / 0.87	1.00 / 1.00	0.76 / 0.72	0.89 / 0.70	0.90 / 0.82
F1	0.97 / 0.93	0.90 / 0.91	0.79 / 0.83	0.67 / 0.73	0.83 / 0.85

The recognition quality of the system is rather disappointing. This is mainly due to the fact that the algorithm registered a lot of false positives by identifying a vertical power rail as a cereal box (74 times) or a soda can (16 times). The power rail was not removed by the passthrough filter as it is very close to the table and appeared to the segmentation algorithm as an object connected to the table. Since the power rail is much bigger (longer) than a cereal box (not to mention a soda can) and the VFH is scale invariant, geometrical constraints described in Section 6.2 were introduced into the algorithm. Any initially recognized object not complying with these constraints was rejected. The results are presented in Table 3 (after slash). The overall performance of the system slightly improved as the total number of false positives dropped. The power rail was identified only once as a soda can and 22 times as a cereal box. It is better but still not as good as expected.

In the next attempt at achieving better results, the models that caused so many false positives were browsed. It turned out that some of them were incomplete or acquired at awkward angles or had many missing points or, on the contrary, had some extra points (see Figure 7 for a couple of samples). For example, a cereal box model containing just its narrowest side could be a good fit to many flat objects seen in the scene and not exclusively to a cereal box. Therefore, many such models were arbitrarily removed from the dataset leaving only models that ensured a good geometrical representation of original objects. Simultaneously, many similar instances of objects were removed as they did not provide useful additional information above what already was included with previous models. This was a particularly frequent occurrence for round objects with a vertical axis of symmetry (see Figure 7). Since the original dataset was huge, this manual, heuristic-based selection of "good models" is by no means optimal and requires further investigation. Nevertheless, results obtained with the reorganized dataset turned out to be clearly better than previously (see Table 4). The training dataset contains 5,785 elements (including 3,553 models of soda can, which were not altered) out of available 15,232 and can be downloaded from https://www.dropbox.com/s/ioahfe1wrst73kf/Czajewski_Kolomyjec_dataset.zip?dl=0

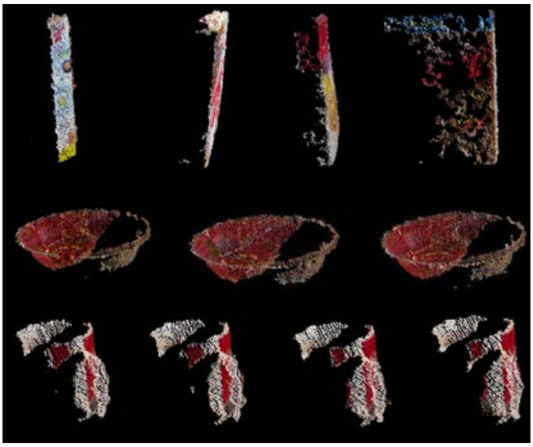

Figure 7. Examples of objects that were removed from the training set due to incompleteness and ambiguity (top row) and nearly identical appearance providing no valuable additional information (middle and bottom rows)

Table 4. Classification quality for reorganized dataset with geometrical constraints

	Bowl	**Coffee mug**	**Soda can**	**Cereal box**	**Average**
Precision	1.00	0.88	0.99	0.83	0.93
Recall	0.91	1.00	0.83	0.86	0.90
F1	0.95	0.93	0.90	0.85	0.91

All of the quality measures were higher in this test, which proves that too many models are not necessarily helpful in achieving good recognition results, especially if some of the models might be a little defective. This time the vertical power rail was identified only once as a soda can and 17 times as a cereal box.

The result described above, has been so far the best result for a single snapshot of a scene. However, one does not have to rely on a single view only. A mobile robotic system that is capable of changing its position or just position of the Kinect sensor may easily acquire a series of point clouds of the same scene from slightly different points of view. There is always a chance that incorrect results will not occur systematically in consecutive frames. Therefore two more experiments were conducted, during which a classification decision was made based on a simple voting mechanism for 5 and 9 consecutive frames respectively. As expected, the recognition results (presented in Table 5) were further improved, but there were still cases of false positives.

In the case of 5-frames long sequence, the power rail was still detected as a cereal box 16 times and for the 9-frame long sequence this false positive count dropped to 9. Classification quality of all the objects significantly increased and the overall performance of the system was nearly perfect. It must be noted, however, that the test scene was not cluttered and contained just one foreign object that caused many false positives. Therefore, one should not expect this kind of recognition quality for more populated and cluttered scenes. In such cases, the recognition parameters (especially during the final hypotheses verification stage) must be more restrictive so as not to accept false positives, which will, unfortunately, lead to a higher number of false negatives.

Table 5. Classification quality for sequences of 5 / 9 consecutive frames of point clouds with reorganized dataset and geometrical constraints

	Bowl	**Coffee mug**	**Soda can**	**Cereal box**	**Average**
Precision	1.00 / 1.00	0.92 / 1.00	1.00 / 1.00	0.84 / 0.90	0.94 / 0.98
Recall	0.92 / 0.96	1.00 / 1.00	0.92 / 1.00	0.91 / 0.91	0.93 / 0.97
F1	0.96 / 0.98	0.96 / 1.00	0.96 / 1.00	0.87 / 0.91	0.94 / 0.97

Although the proposed method was tested only on a video sequence and not on the evaluation set of turntable images and thus it cannot be directly compared to other algorithms shown in Table 6, it’s recognition performance of 90% for such a demanding sequence seems to be a promising indicator for the simpler evaluation set.

Table 6. Category recognition performance of various classifiers on the Washington RGB-D Object Dataset

Method	**Recognition performance [%]**
Linear SVM [26]	53
Nonlinear SVM [26]	65
Random Forest [26]	67
Sparse Distance Learning [27]	70
CNN-RNN [41]	79
RGB-D Kernel Descriptors [13]	80
Hierarchical Matching Pursuit [13]	81

8. Conclusions and Future Work

In this paper, the results of experiments on detection and recognition of three-dimensional objects in RGB-D images provided by the Microsoft Kinect sensor were described. Although the focus was put on using a single image for that purpose, utilizing a series of frames was also considered and evaluated. Experiments performed on hundreds of test scenes show that the proposed and rarely used approach based on the global VFH and CRH descriptors combined with the ICP method and final hypotheses verification can be successfully applied to recognition and localization of objects. However, in order to achieve high recognition rates, the model dataset must be optimized and the distance between the Kinect and the objects being recognized should be relatively small (preferably less than 1 m). The histogram based color identification method proved to be successful for objects of distinctly different colors and failed, as expected, for similar objects.

In light of the above, it seems necessary to use 2D images with higher resolution, which is possible with the Kinect sensor (maximum image resolution is 1280x1024 RGB) or an optional high-resolution cameras. This would allow to apply more sophisticated 2D image comparison techniques like keypoint detectors and descriptors for image matching.

Another important issue is the optimization of the dataset. Different objects require different numbers and kinds of models in order to provide the same level of recognition. Having more models does not always result in an increase in the overall system quality. Therefore, as part of further work, it is planned to develop a system consisting of a turntable and a manipulator on a linear slidebase, which will allow automatic model capture and verification of the recognition and pose estimation quality from almost any viewpoint, both for single captures as well as for series of acquisitions. Not less important is verification of other available point cloud descriptors and comparison with a deep learning approach. It seems interesting to find out how much bigger the training set and training time must be in the case of CNNs, in order to achieve comparable results to a classical approach.

References

[1] Abadi M., et al., Tensorflow: Large-scale machine learning on heterogeneous systems, *Software available from tensorflow. org*, 2015.

[2] Aldoma A. et al., CAD-model recognition and 6DOF pose estimation using 3D cues, *IEEE International Conference on Computer Vision Workshops*, 6-13 November 2011.

[3] Aldoma A., et al., A global hypotheses verification method for 3D object recognition, *Computer Vision–ECCV*, 511–524, 2012.

[4] Aldoma A., et al., Multimodal cue integration through Hypotheses Verification for RGB-D object recognition and 6DoF pose estimation, *IEEE International Conference on Robotics and Automation (ICRA)*, 2104–2111, 2013.

[5] Aldoma A., et al., OUR-CVFH – Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation, *Pattern Recognition. DAGM/OAGM 2012. Lecture Notes in Computer Science, vol. 7476*, 113–122, 2012.

[6] Aldoma A., et al., Three-Dimensional Object Recognition and 6 DoF Pose Estimation, *IEEE Robotics & Automation Magazine*, 80–91, September 2012.

[7] Aldoma A., Fäulhammer T., Vincze M., Automation of “Ground Truth” Annotation for Multi-View RGB-D Object Instance Recognition Datasets, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 5016–5023, 2014.

[8] Aldoma A., Rusu R. B., Vincze M., 0-Order Affordances through CAD-Model Recognition and 6DOF Pose Estimation, *Active Semantic Perception and Object Search in the Real World Workshop, IROS*, 2011.

[9] Alexandre L. A., 3d object recognition using convolutional neural networks with transfer learning between input channels, *13th International Conference on Intelligent Autonomous Systems*, 2014.

[10] Bayramoglu N., Alatan A., Shape index SIFT: Range image recognition using local features, *20th International Conference on Pattern Recognition*, 352–355, 2010.

[11] Bergstra J., et al., Theano: Deep learning on gpus with python, *NIPS Big Learning Workshop*, Granada, Spain, 2011.

[12] Besl P., McKay N., A Method for Registration of 3-D Shapes, IEEE Transactions on *Pattern Analysis and Machine Intelligence, vol. 14, no. 2*, 1992.

[13] Bo L., Ren X., Fox D., Unsupervised feature learning for rgb-d based object recognition. *Experimental Robotics*, 387–402. Springer, 2013.

[14] Chang A. X., et al., Shapenet: An informationrich 3d model repository, *arXiv preprint arXiv:1512.03012*, 2015.

[15] Collobert R., Kavukcuoglu K., Farabet C., Torch7: A matlab-like environment for machine learning, in BigLearn, *NIPS Workshop, no. EPFL-CONF-192376*, 2011.

[16] Eigen D., Fergus R., Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture, *International Conference on Computer Vision*, 2650–2658, 2015.

[17] Gupta S., et al., Aligning 3D models to RGB-D images of cluttered scenes, *Inte. Conference on Computer Vision and Pattern Recognition*, 4731–4740, 2015.

[18] He K., et al., Spatial pyramid pooling in deep convolutional networks for visual recognition, *13th European Conference on Computer Vision*, 346–361, 2014.

[19] Hernandez C et al., Team Delft's Robot Winner of the Amazon Picking Challenge, *ArXiv eprints. arXiv: 1610.05514 [cs.RO]*, 2016.

[20] Hernandez-Vela A. et al., BoVDW: Bag-of-Visual-and-Depth-Words for gesture recognition, *International Conference on Pattern Recognition (ICPR)*, 2012.

[21] Hinterstoisser S, et al., Dominant orientation templates for real-time detection of texture-less objects, *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[22] Jia Y., et al., Caffe: Convolutional architecture for fast feature embedding, *arXiv preprint arXiv:1408.5093*, 2014.

[23] Jonschkowski R. et al., Probabilistic multi-class segmentation for the Amazon Picking Challenge, *International Conference on Intelligent Robots and Systems (IROS)*, 2016

[24] Kołomyjec K., Czajewski W., Identification and localization of objects in RGD-D images for the purpose of manipulation, in Polish, *Prace Naukowe Politechniki Warszawskiej. Elektronika, 195, 2*, 377-386, 2016.

[25] Kurban R., Skuka F., Bozpolat H., Plane Segmentation of Kinect Point Clouds using RANSAC, *7th International Conference on Information Technology, ICIT*, Amman, Jordan, 2015, 545–551.

[26] Lai K., et al., A Large-scale Hierarchical Multi-view RGB-D Object Dataset, *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011.

[27] Lai K., et al., Sparse distance learning for object recognition combining rgb and depth information. *IEEE International Conference on Robotics and Automation,* (ICRA) May 2011.

[28] Laptev D. et al., Ti-pooling: Transformation-invariant pooling for feature learning in convolutional neural networks, *International Conference on Computer Vision and Pattern Recognition*, 289–297, 2016.

[29] Łępicka M., Kornuta T., Stefańczyk M., Utilization of colour in ICP-based point cloud registration, *9th International Conference on Computer Recognition Systems,* 821–830, 2015, 2016.

[30] Martínez L., Loncomilla P., Ruiz-del-Solar J., Object recognition for manipulation tasks in real domestic settings: A comparative study, *18th RoboCup International Symposium, Lecture Notes in Computer Science. Springer*, Joao Pessoa, Brazil, 2014.

[31] Maturana D., Scherer S., Voxnet: A 3d convolutional neural network for real-time object recognition, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[32] Muja M., Lowe D.G., Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration, *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2009.

[33] Nafouki K., Object recognition and pose estimation from an RGB-D image, *Technical report*, Technical University of Munich, 2016.

[34] Narayanan V., Likhachev M., PERCH: Perception via Search for Multi-Object Recognition and Localization, *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[35] Papazov C., Burschka D., An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes, *Asian Conference on Computer Vision*, Part I, 135–148, 2010.

[36] Porzi Z. et al, Learning Depth-Aware Deep Representations for Robotic Perception, *IEEE Robotics and Automation Letters*, Volume 2, Issue 2, April 2017.

[37] Prankl J., et al., RGB-D Object Modelling for Object Recognition and Tracking, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[38] Rusu R.B., Cousins R., 3D is here: Point Cloud Library (PCL), *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, 9–13 May 2011.

[39] Rusu R.B., et al. Fast 3D Recognition and Pose Using the Viewpoint Feature Histogram, *23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, 10/2010, 2155–2162.

[40] Rusu, R.B., Blodow N., Beetz M., Fast point feature histograms (fpfh) for 3d registration, *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.

[41] Socher R., et al., Convolutional-recursive deep learning for 3d object classification. *Advances in Neural Information Processing Systems*, 665-673, 2012.

[42] Song S., Xiao J., Deep sliding shapes for amodal 3d object detection in rgb-d images, *29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] Tang J., et al., A textured object recognition pipeline for color and depth image data, *IEEE International Conference on Robotics and Automation*, 3467–3474, 2012.

[44] Tombari F., Salti S., Di Stefano L., Unique signatures of Histograms for local surface description, *European Conference on Computer Vision (ECCV)*, 2010.

[45] Wang A., et al., MMSS: Multi-modal sharable and specific feature learning for RGB-D object recognition, International Conference on Computer Vision, 1125–1133, 2015.

[46] Wu Z., et al., 3dshapenets: A deep representation for volumetric shapes, *IEEE Conference on Computer Vision and Pattern Recognition*, 1912–1920, 2015.

[47] Xie Z., et al., Multimodal blending for high-accuracy instance recognition, *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2214–2221, 2013.

[48] Zang Z., Iterative point matching for registration of free-form curves and surfaces, *International Journal of Computer Vision*, Volume 13, Issue 2, 119–152, 1994.

This is an extended version of the paper presented at the 14th National Conference on Robotics (KKR 2016), Polanica Zdrój, Poland, September 14-18, 2016.