FOUNDATIONS OF COMPUTING AND DECISION SCIENCES
Vol. 42 (2017) No. 3
ISSN 0867-6356
DE GRUYTER OPEN DOI: 10.1515/fcds-2017-0012 e-ISSN 2300-3405

KEYPOINT-LESS, HEURISTIC APPLICATION OF LOCAL 3D DESCRIPTORS

Bogdan HARASYMOWICZ-BOGGIO, Łukasz CHECHLIŃSKI *

Abstract. One of the most important topics in the research concerning 3D local descriptors is computational efficiency. The state-of-the-art approach addressing this matter consists in using keypoint detectors that effectively limit the number of points for which the descriptors are computed. However, the choice of keypoints is not trivial and might have negative implications, such as the omission of relevant areas. Instead, focusing on the task of single object detection, we propose a keypoint-less approach to attention focusing in which the full scene is processed in a hierarchical manner: weaker, less rejective and faster classification methods are used as heuristics for increasingly robust descriptors, which allows to use more demanding algorithms at the top level of the hierarchy. We have developed a massively-parallel, open source object recognition framework, which we use to explore the proposed method on demanding, realistic indoor scenes, applying the full power available in modern computers.

Keywords: RGBD, object detection, 3D descriptors, keypointless

1. Introduction and related works

Segmentation-based methods of 3D object recognition perform well in tidy, relatively simple environments and many state-of-the-art object recognition methods are based on pre-recognition scene segmentation [15, 16, 25, 26, 20, 14]. However, these methods are limited and prone to generate errors, as initial segmentation imposes a fixed structure of the scene before any understanding takes place. In complex environments with adjacent and semi-occluded objects the task of accurate segmentation without object-specific knowledge (i.e. before classification) becomes impossible. This issue is mentioned even by the authors of the most recent segmentation-based object recognition systems, such as [15].

*Faculty of Mechatronics, Warsaw University of Technology, Warsaw, Poland, {b.harasymowicz, l.chechlinski}@mchtr.pw.edu.pl

As an alternative, in recent years multiple techniques of neighborhood-based, local 3D surface description have been developed, such as PFH [23], FPFH [21], PFHRGB [4], CVFH [2]. The applications for such descriptors range from point cloud registration to object part recognition, fast orientation retrieval and, as descriptors become more accurate, raise the possibility of segmentation-less object detection. The use of neighborhood-based descriptors is usually paired with the use of keypoints [3, 7, 9, 8] as heuristics in order to limit the descriptor calculation and classification only to the most meaningful regions of the scene. As the cited papers show, the method of choice of keypoints influences the quality of classification of local descriptors (though not as much as the choice of descriptors). Different keypoint detectors are sensitive to different features and have varied repeatability [9].

The authors of the mentioned papers [3, 7] find that PFHRGB are the most robust, but also one of the most time-consuming descriptors. The same authors also provide an interesting comparison of category and instance recognition accuracy of descriptor classification using keypoints with descriptor classification applied for whole, sub-sampled point clouds views, using the points resulting from voxel grid filtering as keypoints. It was found that the sub-sampling method resulted in a higher accuracy at the cost of significantly higher processing time, since more descriptors had to be calculated – it took 1.6 to 5.7 times longer (depending on the leaf size of the voxel grid) to calculate PFHRGB (the best performing descriptor) for sub-sampled scenes than for keypoints. These conclusions have been drawn for experiments involving views of isolated objects belonging to known categories. Obviously, in realistic scenes, where occlusions and many unknown objects occur, we could expect lower recognition rates than the rates of the cited papers.

If we intend to apply local 3D descriptors for object detection in real-life environments, where known objects constitute only a fraction of the scene, an appropriate mechanism of attention focus (such as keypoint detection) is even more crucial. Keypoint finding algorithms are intended to detect general saliency before any classification occurs, and thus are oblivious of the objects of interest. In real-life applications, such as mobile robotics, an object recognition system is often concerned only with one or only a few object categories at the same time (e.g. for finding an object, performing an action using objects, etc.).

Taking this practical observation into consideration and focusing on the task of single object detection, in this article we propose a different approach to classification of local 3D detectors, which doesn't make use of keypoints. Instead, we apply voxel grid filtering and classify the resulting points (which we call *anchor* points, since they are not keypoints) in a hierarchical manner in three fully parallelized stages: first we remove the large smooth regions of the scene (which are prevalent in indoor environments), next we calculate and classify fast local descriptors, rejecting another part of the scene and finally we apply the most robust descriptors on the remaining anchor points. The difference between the anchor points used in this article and keypoints is that keypoints are calculated by a general-purpose algorithm (i.e. selected as being universally relevant), without any knowledge about the object of interest. Anchor points, on the other hand, are initially uniformly distributed all over the scene (resulting in their large quantity) and subsequently reduced in stages using information about a specific search object.

The idea of flat areas removal to reduce computation time is applied in complex systems of 3D semantic category recognition, such as [27]. The most common methods to achieve this are based on RANSAC plane detection [24, 5, 12, 17]. However, such methods are not very time-efficient (even in parallel implementations) and the processing times strongly depend on the complexity of the scene. Although on organized (2,5D) point clouds the plane segmentation task can be accomplished by simple image-based region growth using surface normal vectors [6], this technique is not practical for unorganized (non-image) clouds, such as the scenes used in the experimental section of this paper (the smallest useful neighborhoods tend to be large and not memory-aligned on these clouds). In the proposed system we have applied a massive-parallel region growth algorithm that we presented in a previous work [10].

In the experimental section we demonstrate that the proposed hierarchical, heuristic approach not only greatly reduces scene processing times (compared to dense descriptor classification), but also significantly increases detection accuracy by complementary action of the smooth area removal (which we refer to as non-flat filtering) and the descriptor classification. The presented approach could be combined with keypoint detection methods in order to further speed-up the scene processing as well as be extended to detect a limited number of objects simultaneously. However, to explore this possibilities is beyond the scope of this article.

The experiments have been carried using our open-source object recognition system *Heuros* [1], which benefits from the Robot Operating System (ROS) [19] and Point Cloud Library (PCL) [22] – especially the parallel methods implemented by Anatoly Baksheev. All of the experimental data has been acquired using a Kinect sensor and pre-processed on-line applying the PCL implementation of Kinect Fusion [13]. All of the demanding processing was performed using CUDA implementations on a GPGPU, including voxel grid filtering, neighborhood search, flat areas detection, normal vector calculation, feature calculation and matching.

Interestingly, during the experiments we found that some of our descriptors (which are combinations of simple feature histograms and shape descriptors) significantly outperform the descriptors found in literature (including PFHRGB) in the task of segmentation-less object detection for the tested conditions of untidy, realistic indoor environments. Since these descriptors are based on different basic features than PFH, we decided to combine the best performing descriptor (ICAHSD) with PFHRGB in a naive manner, obtaining even higher correct recognition rates. In the following sections we describe the proposed hierarchical heuristics, the non-flat filtering parallel algorithm and the tested 3D local descriptors. The final sections concern the performed experiments and the general conclusions.

2. Hierarchical heuristics

The proposed method of cluster selection for descriptor calculation is presented in Fig. 1. The algorithm operates on point clouds of the full scene, which are increasingly reduced on each stage. The first stage consists in removing the large smooth areas of the scene, where no relevant features are expected. Next, the cloud is downsampled using a voxel grid filter in order to obtain a set of anchor points with relatively high frequency. The radial neighborhoods of these points are then retrieved from the full dense scene using an octree in order to calculate the first 3D local descriptors (denoted as "fast descriptors" on the diagram). For this classification stage we seek to apply low-cost descriptors, which give low false-negative rates, but for which we can tolerate moderately high false-positive rates.

We expect these two heuristic reduction stages to provide a region of interest of the scene, where most of the real positive anchor points are concentrated and for which we can calculate and classify more robust descriptors at a significantly reduced computational cost compared to calculation for all of the initial anchor points. Since the objects present on 3D point clouds are spatially consistent, we apply a dilation step after the first classification in order to prevent omission of important anchor points. This consists in including all of the radial anchor point neighborhoods of the positively classified points as input for the robust descriptors.

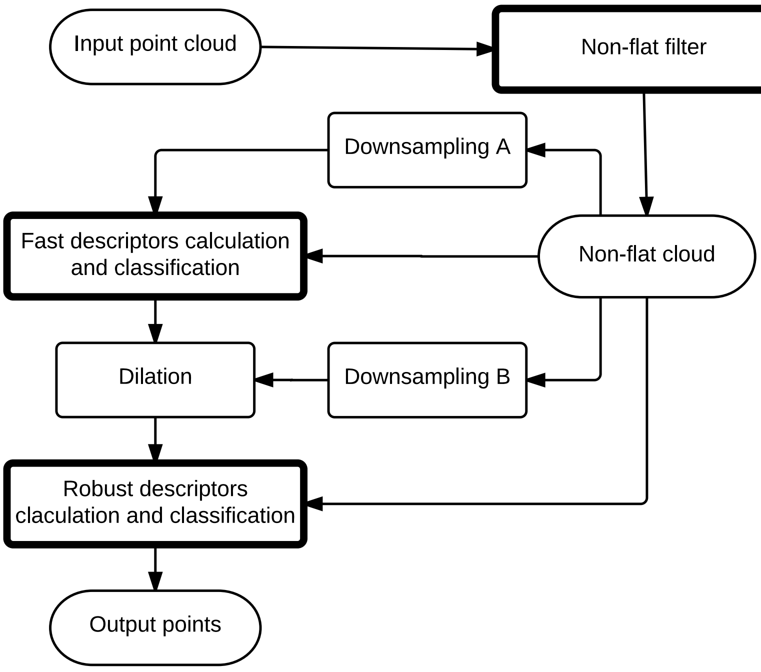
Input point cloud
Non-flat filter
Downsampling A
Fast descriptors calculation and classification
Non-flat cloud
Dilation
Downsampling B
Robust descriptors claculation and classification
Output points

Figure 1. Scheme of the proposed hierarchical scene reduction and descriptor calculation method. The three main stages of the procedure are highlighted.

3. Large smooth areas removal

Efficiently finding large smooth areas in unorganized point-clouds is not a trivial task, as simple image-based flood-fill methods are not applicable. Using a RANSAC approach (whether serial or parallel) to find planes results in data-dependent performance (i.e. we repeat the whole procedure for subsequent planes) and doesn't necessarily provide connected coplanar areas. In a previous work we have proposed a parallel region-growth algorithm [10]. As region-growth in its basic version is a highly sequential concept, our parallelization method is iterative and heuristic. The algorithm requires finding radial neighborhoods for each point and applying some criterion to create connections within these neighborhoods (e.g. normal vector similarity), which can easily be done in parallel. Each point is treated as a seed and is assigned a unique segment index. After that, the region growth parallel method (kernel) is launched multiple times, propagating the higher indices through the neighborhood connections and further-away, already interconnected regions. A simplified diagram of this algorithm is presented in Fig. 2.

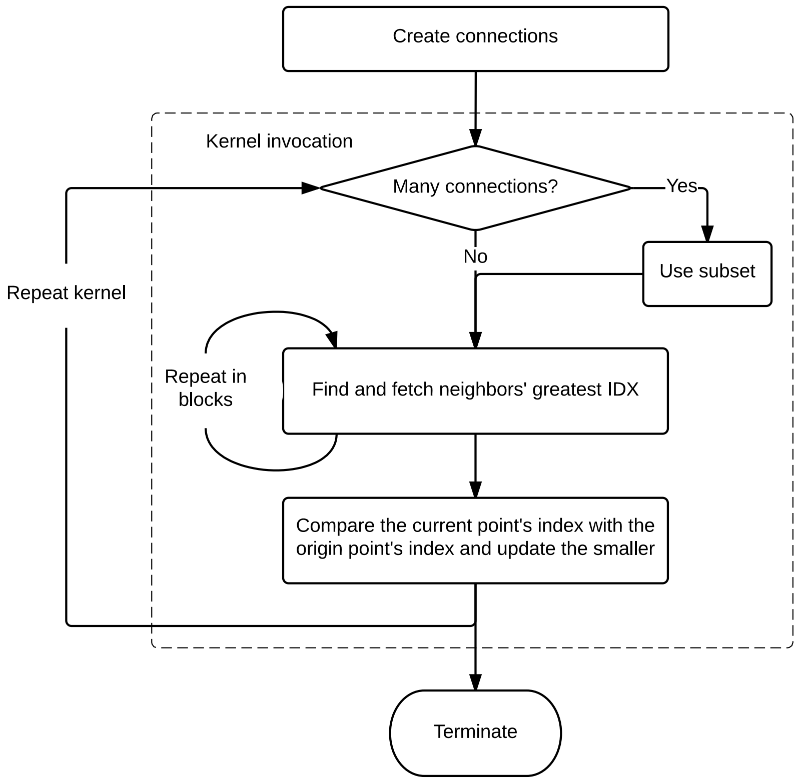
Create connections
Kernel invocation
Many connections?
Yes
No
Use subset
Repeat kernel
Repeat in blocks
Find and fetch neighbors' greatest IDX
Compare the current point's index with the origin point's index and update the smaller
Terminate

Figure 2. Region growth massively-parallel algorithm using for the non-flat filtering stage.

4. Local descriptors and negative models

In the experimental section we consider multiple 3D local descriptors (features), which can be calculated for the neighborhoods of the scene's anchor points – i.e. clusters of points found within a given radius. As descriptors corresponding to the "fast descriptors" stage (Fig. 1) we applied 20-bin 1D histograms and 20x20-bin 2D histograms of simple local features (inclination, convexity, anisotropy of convexity, hue and saturation) as well as shape distributions (D1-D3). For the "robust descriptors" stage more complex histogram features have been tested (PFH, FPFH, PFHRGB) as well as their combinations with the simpler descriptors. In this section we briefly introduce the used features, some of which we have proposed in previous works [11].

Inclination (I) of a local surface relative to the gravity vector measured with the Kinect's accelerometer is the first of the simple local features which we histogramize. This feature is invariant to translation of the object and to rotation around any vertical axis, but not invariant to rotation in other axes. However, since most household object have a limited number of possible bases (i.e. stable orientations), this feature proves highly useful in indoor environments.

Convexity (C) is defined as the local amount of outwards curvature, hence can be positive or negative. We calculate this feature at each scene point using the normal vectors within $1cm$ neighborhoods. To formally introduce this feature let us consider the local coordinate system attached to point $\mathbf{p}$ (in which $\mathbf{p} = \left[\begin{smallmatrix}0 & 0 & 0\end{smallmatrix}\right]^T$ and where $\mathbf{n} = \left[\begin{smallmatrix}0 & 0 & 1\end{smallmatrix}\right]^T$ is the surface normal vector at $\mathbf{p}$. Using this coordinate system we introduce the auxiliary quantity of convexity at $\mathbf{p}$ relative to point $\mathbf{p_i}$ defined with equation (1)

$$C(\mathbf{p}, \mathbf{p_i}) = (\widehat{\mathbf{e_{xy}p_i}}) \cdot (\widehat{\mathbf{e_{xy}n_i}}) \cdot \frac{\arccos(\mathbf{e_z n_i})}{\pi/2}, \tag{1}$$

where: $\mathbf{e_{xy}} = \left[\begin{smallmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{smallmatrix}\right]$, $\mathbf{e_z} = \left[\begin{smallmatrix}0 & 0 & 1\end{smallmatrix}\right]$, and $\mathbf{n_i}$ is the surface normal vector at point $\mathbf{p_i}$. Using this quantity we can define convexity at $\mathbf{p}$ as:

$$C(\mathbf{p}) = \frac{1}{K} \sum_{\mathbf{p_i} \in N_{\mathbf{p}}(r_1) \setminus N_{\mathbf{p}}(r_2)} C(\mathbf{p}, \mathbf{p_i}), \tag{2}$$

where $N_{\mathbf{p}}(r_1) \setminus N_{\mathbf{p}}(r_2), r_1 > r_2$ is the set-theoretic difference of two neighbourhoods of point $\mathbf{p}$ having radii of magnitude r_1 and r_2.

Anisotropy of convexity (A) is defined as the variation of convexity across different directions. We also calculate this feature for each point by retrieving the maximum and minimum convexity components of each neighbor within a given radius range:

$$A(\mathbf{p}) = \max_{\mathbf{p_i} \in N_{\mathbf{p}}(r_1) \setminus N_{\mathbf{p}}(r_2)} [C(\mathbf{p}, \mathbf{p_i})] - \min_{\mathbf{p_i} \in N_{\mathbf{p}}(r_1) \setminus N_{\mathbf{p}}(r_2)} [C(\mathbf{p}, \mathbf{p_i})] - 1 \tag{3}$$

Hue and saturation (H,S). We also apply histograms of color hue and saturation from the HSV color space (omitting V, as this dimension is the most affected by lighting variations).

D2, D3, D4 (abbreviated as D) [18] are shape distributions – i.e. normalized histograms of respectively: distances between any 2 cluster points, areas defined by any 3 points and volumes defined by any 4 points. These features are estimated using random point tuples of the descriptor clusters.

PFH, FPFH, PFHRGB [23, 21, 4]. The former are multi-dimensional histograms of parameters describing the relations of point pairs position and normal vectors, whereas FPFH is an approximate speed-up version of PFH and PFHRGB is PFH with three added histograms representing the RGB color relations between these point pairs.

While many other descriptors can be found in literature, we have chosen only the mentioned for experimental evaluation. We have implemented the first 8 features used for local description, due to their simplicity, in an optimized, integrated calculation procedure. The GPU module of PCL provides massively-parallel implementations of PFH, FPFH, PFHRGB. In the experimental section we explore classifiers which combine the simple features I, C, A, H, S, D by averaging similarity scores for all of their histogram components (in our case, these scores are Pearson's correlation coefficients to model histograms). As shown in the next section, such combined descriptors as ICAHS and ICAHSD give surprisingly good results and can be further combined in the same manner with the complex descriptors. This simplistic combination method could, of course, be further investigated and developed (e.g. using weighted averages or a higher-dimensional descriptor). However, due to the limited, self-collected test data available for the experiments presented in this article, we decided to avoid automated fine-tuning the new combined descriptors in order to avoid overfitting, and use the naive combination method instead.

In the system used for the experimental section, we classify the calculated descriptors by comparing them to model descriptors (i.e. descriptors calculated for a higher density of anchor points in training object views). In order to prevent replication of almost identical model descriptors, each new calculated training descriptor is compared in terms of Pearson's correlation to the already acquired descriptors and discarded if it's too similar to an existing descriptor (for this we apply a correlation threshold set to 0.99). Even though the previously described non-flat filtering method removes vast amounts of smooth areas, we still observed that flat points were left out, some of them giving false-positive detections. To prevent this, at the training stage we apply negative models consisting of a few samples of flat areas (which we further call *antimodels*). Beside being compared to the positive models, each new training descriptor is also compared to the antimodels with a lower threshold and discarded if a similarity is found.

5. Experiments

5.1. Methodology

In the carried experiments we consider multiple parameters of the proposed method evaluating its usefulness for candidate object detection. In particular, we discuss the choice of the applied descriptors, the influence of the non-flat heuristics, the use of antimodels, and the descriptor calculation radiuses.

These parameters have been evaluated using realistic train and test datasets of scenes we acquired with the Kinect sensor applying the Kinect Fusion SLAM technique [13] and processed on a PC equipped with a Nvidia GTX Titan Black GPGPU. Since there are no available public datasets containing labelled Kinect Fusion-quality 3D scenes known to the authors, the used datasets are self-collected. The test data is intended to reflect difficult conditions of untidy indoor environments, which include multiple occlusions, overturned, adjacent and stacked objects, as well varied visibility and acquisition distance. The training set contains 6 different household "prop" objects in different poses (6 views each): *book, cup, ironer, ketchup bottle, brake fluid bottle* and *mouse.* The test set consists of 21 scenes containing the objects present in the training data, as well as many other objects, unknown to the system. The test data is limited – however, it presents demanding and fairly varied conditions. The known objects of the test data have been manually labeled for calculation of detection rates, providing a grid of points belonging to their respective surfaces. Example detection results for the test set are presented in figures 3 and 4. On the scenes from Fig. 4 beside true positive detections we can see some false positives: On the book detection scene, despite applying flat area removal and flat antimodels, the algorithm produces some missclassifications because of the low uniqueness of the book's shape features. On the ironer detection scene we can see how a part of the ironer is detected despite the heavy cluttering from other objects. Some of it's features, however, are also erroneously detected on a duct tape. On the mouse detection scene the algorithm failed to provide any detections. The mouse presented difficulties often when the sensor was not close enough, since the mouse is small and black, which resulted in unstable feature values.

As motivated before, we have focused on the single-class detection task, for which we apply a nearest neighbor approach: we classify a descriptor as belonging to the object of interest if the proximity measure to the nearest neighbor form the training set doesn't exceed a given threshold. As the proximity metric we have applied Pearson's correlation coefficient. This approach is justified in the considered single-object detection task, as the number of training descriptors is small and the search is fully parallelized (the nearest neighbor search times are small compared to the feature calculation times).

In the presented experiments, in order to compare classification quality of different methods, we consider (1-precission) x recall plots, as well as $F1$ scores, which are widely used for evaluation of binary classifiers:

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + reccall} \tag{4}$$

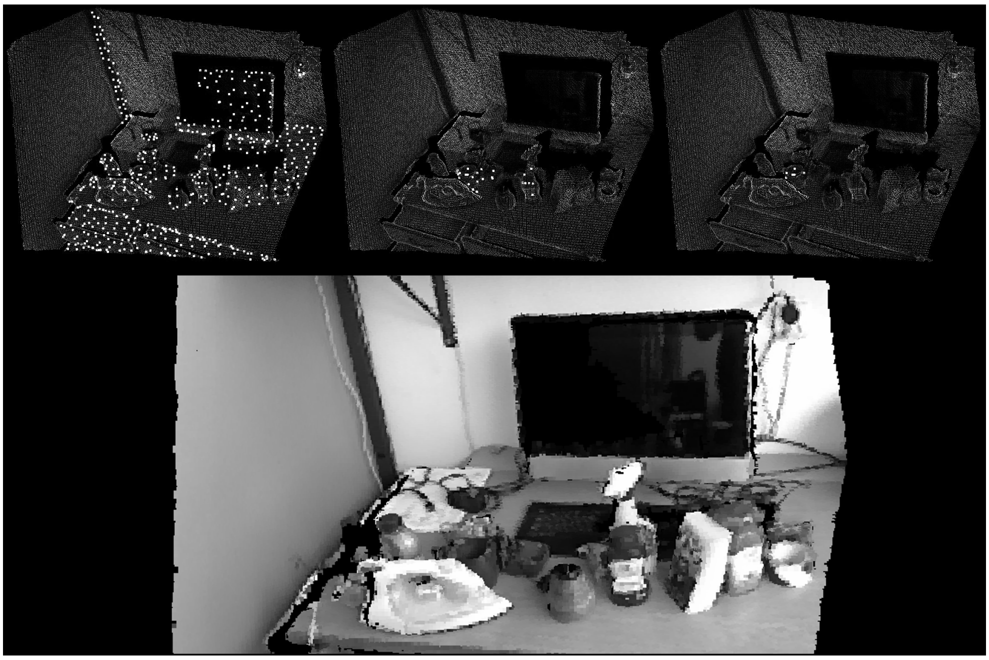

Figure 3. Example test scene from our Kinect Fusion dataset processed in the proposed hierarchical manner. The white dots represent the anchoring points for (from left to right) non-flat filtering, ICAHSD and Combined descriptor classification for the task of cup detection. An additional view of RGB colors converted to grayscale is provided.

where $precision = \frac{TP}{TP+FP}$ and $recall = \frac{TP}{TP+FN}$. To obtain the true positive (TP), false positive (FP), true and false negative rates (TN, FN) we applied approximate estimations based on the neighborhoods of the positive detection (D) and positive label sets (L) of anchor points:

$$\begin{aligned} TP &= N_b(D) \cap N_s(L) \\ FP &= N_s(D) \cap N'_b(L) \\ TN &= N'_b(D) \cap N'_b(L) \\ FN &= N'_b(D) \cap N_s(L) \end{aligned} \tag{5}$$

where N_b and N_s are anchor point neighborhoods retrieved respectively for 0.6 and 0.4 factors of the descriptor calculation radius. These "relaxed" neighborhood definitions are intended to partially decrease the impact of imperfect correspondence between the detections and the labels, focusing instead on whether the positive detections hit or miss a labeled object as a whole.

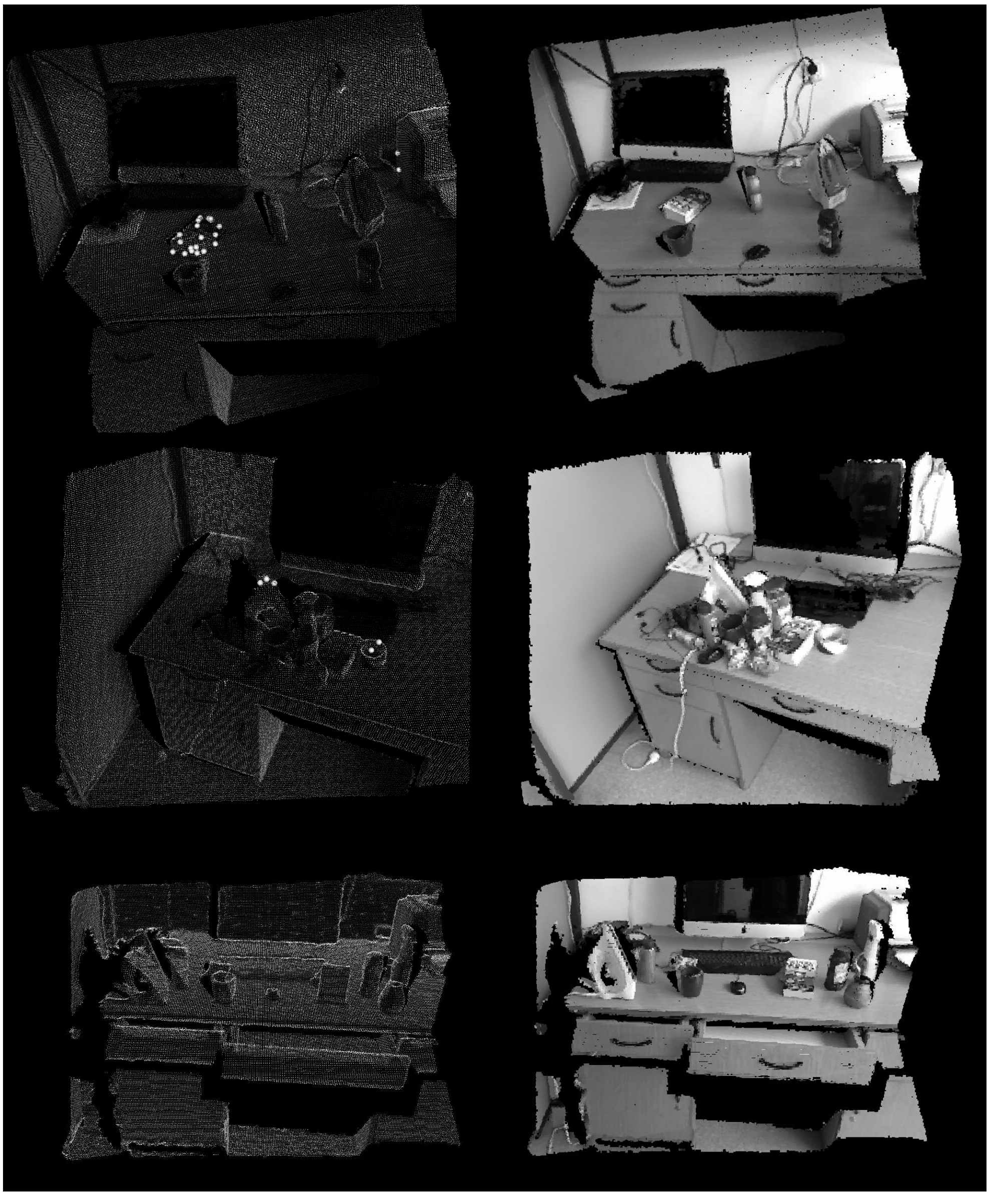

Figure 4. Example test scenes imperfectly processed (from higher to lower) for *book*, *ironer* and *mouse* detection (on the last scene the algorighm failed to provide any detections). The right column shows the grayscale representation of the scene colors.

5.2. Radiuses

We have tested multiple fixed radiuses for descriptor calculation, ranging from 3 to 15 cm as well as variable radiuses determined using the model views as follows:

$$R = s \cdot r\sqrt{N/n} \tag{6}$$

where n is the average number of neighbors in a r radius from any surface point, N is the number of points in a model object and s is a scaling factor. The detailed performance comparisons are not presented in the article – however, we have concluded that the variable radiuses with $s = 1$ worked best (settling the radius using just the first model's number of points N and calculating n for $r = 1$). Fixed radiuses around 6-7 cm were only slightly worse on average and would be more computationally efficient if we wanted to compare the descriptors with models from different semantic classes. The presented experimental results in this section use these variable radiuses. To get the descriptors' anchoring points, scene point clouds are downsampled using parallel voxel grid filtering with a voxel size calculated as:

$$v = \frac{2R}{3} \tag{7}$$

meaning that on a flat surface a descriptor's neighborhood covers about 7 anchoring points, which we consider to be fairly dense.

5.3. Non-flat filtering and antimodels

In order to validate the first heuristic scene reduction mechanism, we have compared the detection quality of descriptor classification applied on full scenes with the classification applied only on non-flat areas and classifiers trained using antimodels (also applied on non-flat areas). We have found that both of these mechanisms improve the classification quality in terms of precision and recall (especially non-flat filtering), which is demonstrated only for a selected case on Fig. 5 – for the PFHRGB descriptor (which was found to perform best in [8, 3]). Some important performance parameters of the non-flat filter algorithms are also provided in Tab. 1.

Table 1. Results for non-flat filtering

Avgerage scene anchor points	1975
Avgerage anchor points after non-flat	1010
Avgerage non-flat filtering time	38.5 ms

5.4. Features comparison

Applying these conclusions (i.e. using non-flat filtering and antimodels), we have compared multiple descriptors: FPFH, PFH, PFHRGB, and many combinations of

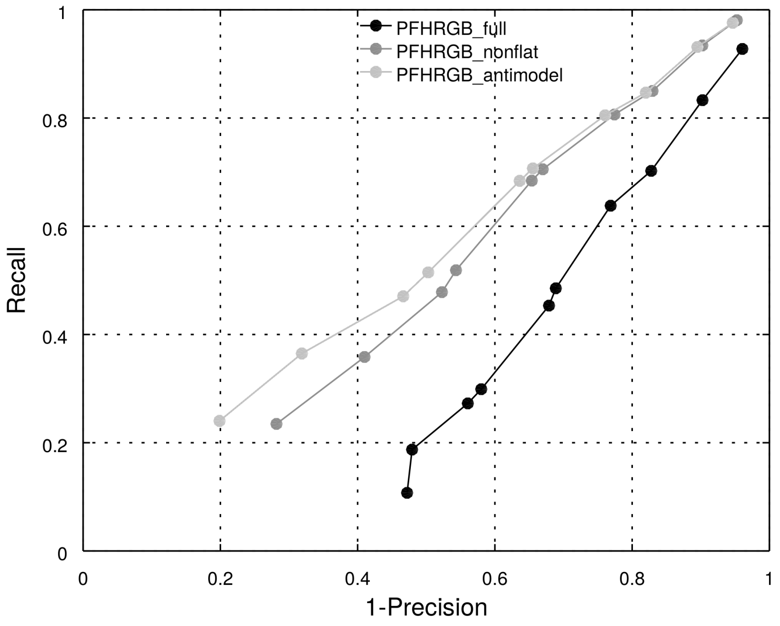

Figure 5. Comparison of PFHRGB classification for full scenes, non-flat areas and non-flat areas trained with antimodels.

the earlier described simple features I, C, A, H, S, D2, D3, D4 and their 2D histogram combinations as well (even though we did not find any benefit from using 2D over 1D histograms). If Fig. 6 we present the (1-precision) x recall plots for a selected subset of the tested descriptors. The descriptor combining all of the 1D simple features (which we refer to as ICAHSD) achieved the best results (significantly better than PFHRGB). However, when combined with PFHRGB in a naive manner (we simply took the average correlation of both descriptors), the resulting descriptor (denoted as Combined) surpasses both of its components. As we can see, the curves don't always cover the higher precision part of the graph, which indicates that after some correlation threshold no positive detections are returned and no higher precision is achieved.

5.5. Hierarchical heuristics

Based on these measurements, we decided to demonstrate the full hierarchical classification applying non-flat filtering, antimodels and using ICAHSD descriptors (which took on average slightly above $100ms$, to calculate on non-flat areas, excluding non-flat filtering itself) as a heuristic for the slower PFHRGB (with an average of $440ms$ on nonflat areas), and then calculating the correlation scores of the Combined de-

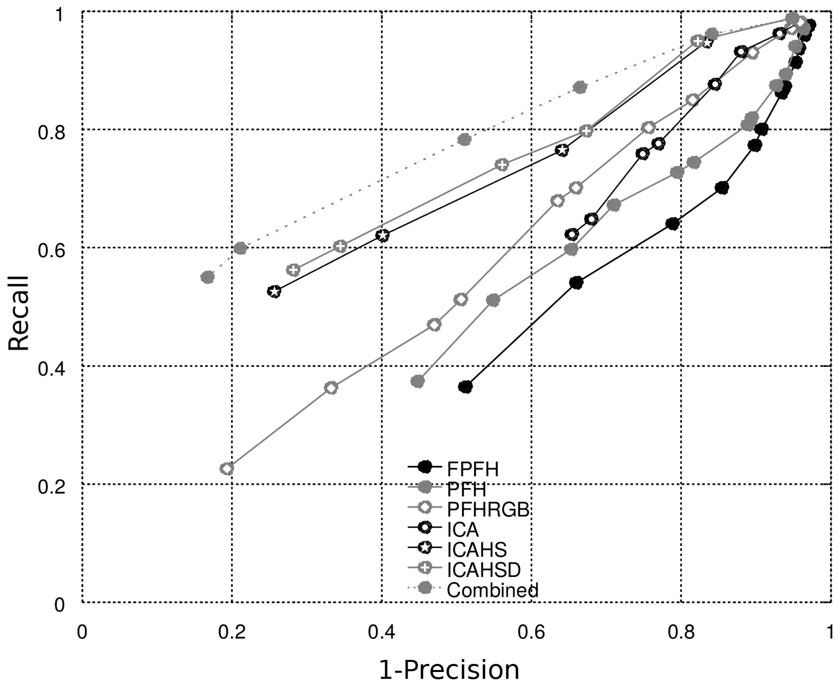

Figure 6. Comparison of selected descriptors (non-flat heuristics and antimodels were applied for all the presented descriptors).

scriptor to obtain a final output classification. In Fig. 7 we present the measured impact of the proposed heuristic mechanism on the scene processing times considering different classification thresholds for the lower-level stage classifier (using ICAHSD descriptors). Tab. 2 summarizes the performed experiments in terms of times and max F1 scores, providing also the applied heuristic level (0 – full scene processing, 1 – processing only non-flat areas, 2 – processing only the anchor points resulting from non-flat filtering and the first descriptor classification stage).

As we can see, the initial processing time was cut to less than half just by applying non-flat filtering (improving also the F1 scores). The processing time was further reduced twice again by applying the next heuristic stage (ICAHSD). The maximum F1 score was negligibly affected until we have increased the heuristic threshold to 0.87 – thus, the value 0.85 seems optimal (this was also visible on the (1-precision) x recall plots, which are not presented here). However, if we desired a high precision level, the threshold could be further increased to speed up the scene processing without affecting performance. As we can see in Tab. 2, the simpler descriptors' classification time doesn't benefit from the application of non-flat filtering, but still increases accuracy by reducing false-positive detections.

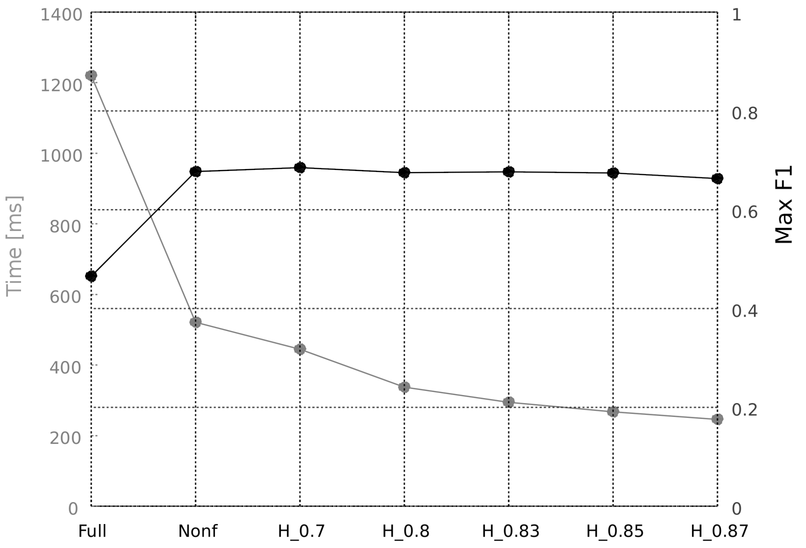

Figure 7. Processing times and maximum F1 scores for the heuristic Combined descriptor without heuristics, with non-flat heuristics and full descriptor heuristics for different ICAHSD classification thresholds.

6. Conclusions

In this paper we have presented a hierarchical, heuristic approach to descriptor-based object detection, useful for applications with limited numbers of interest objects. We implemented and validated this method using parallel computing on a high-end GPGPU. The two stages of anchor point reduction have been shown to decrease the processing time over 4.5 times, while also significantly increasing classification accuracy. Furthermore, we have explored multiple 3D local descriptors and found that straightforward combinations of simple feature histograms give better results that state-of-the-art complex descriptors for the considered task – we also combined these histograms with PFHRGB, obtaining an even better descriptor. The presented descriptor combination method is simplistic, as it was not in the primarily intended scope of the article. However, it opens interesting exploration possibilities for weighted or boosting local descriptor mixtures.

In the experimental setup we have applied exhaustive nearest neighbor classification, which is justified for the established single-object detection goal. However, for the simultaneous recognition of multiple objects, fast-learning techniques like SVM could be applied. Also, it is not given that there is a single descriptor that suits all semantic classes and object-specific descriptor usage could be considered.

Although it has not been tested, it is possible that we could achieve further time

Table 2. Comparison of heuristic levels, max F1 scores and average processing times for the selected descriptors in several configurations: no heuristics (full), non-flat filtering (N), antimodels (A), All heuristic stages (H)

Descriptor	H lvl	Max F1	Avg T [ms]
FPFH full	0	0,229	**254,3**
FPFH N + A	1	0,367	**166,2**
PFH full	0	0,261	671,3
PFH N + A	1	0,369	**290,8**
PFHRGB full	0	0,283	1053,1
PFHRGB N	1	0,403	439,1
PFHRGB N + A	1	0,460	440,2
ICA full	0	0,279	**94,5**
ICA N + A	1	0,327	**103,5**
ICAHS full	0	0,422	**105,3**
ICAHS N + A	1	0,542	**109,1**
ICAHSD full	0	0,382	**192,6**
ICAHSD N + A	1	0,567	**147,2**
Combined full	0	0,466	1220,3
Combined N + A	1	**0,677**	521,5
Combined H 0.7	2	**0,685**	445,2
Combined H 0.8	2	**0,675**	337,7
Combined H 0.83	2	**0,677**	**294,9**
Combined H 0.85	2	**0,674**	**267,6**
Combined H 0.87	2	**0,663**	**246,6**

reduction by adding more descriptor-based heuristic processing stages (e.g. using the somewhat faster ICAHS descriptors before ICAHSD). Also, since our method is based on a different heuristic mechanism, it could probably benefit from the application of keypoints, which can be tested in future works. Object detection methods based on local descriptors are also suitable to be combined with more discriminative (but more time-consuming) techniques such as point cloud of feature cloud registration. In order to gain a better understanding of these possibilities it would be appropriate to carry larger experiments, as well as tests involving scenes captured with newer depth sensors, such as the Kinect 2.

References

[1] *Heuros 3D object recognition system.* https://bitbucket.org/rrgwut/heuros. Accessed: 04-05-2015.

[2] Aldoma, A., et al. *CAD-model recognition and 6DOF pose estimation using 3D cues.* In *IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011*, 585–592. 2011.

[3] Alexandre, L. A. *3D descriptors for object and category recognition: a comparative evaluation.* In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Vilamoura, Portugal, 2012.

[4] Alexandre, L. A. *Set distance functions for 3D object recognition.* In *18th Iberoamerican Congress on Pattern Recognition*, volume LNCS 8258 of *Lecture Notes in Computer Science*, 57–64. Springer, Havana, Cuba, 2013.

[5] Cupec, R., et al. *Detection of planar surfaces based on ransac and lad plane fitting.* In Petrovic, I., Lilienthal, A. J., editors, *ECMR*, 37–42. KoREMA, 2009.

[6] Farid, R. *Region-Growing Planar Segmentation for Robot Action Planning*, 179–191. Springer International Publishing, Cham, 2015.

[7] Filipe, S., Alexandre, L. A. *A comparative evaluation of 3d keypoint detectors in a rgb-d object dataset.* In *2014 International Conference on Computer Vision Theory and Applications (VISAPP).* 2014.

[8] Filipe, S., Itti, L., Alexandre, L. A. *BIK-BUS: Biologically motivated 3D keypoint based on bottom-up saliency.* IEEE Transactions on Image Processing, 24(1):163–175, 2015.

[9] Ghorpade, V. K., et al. *Performance evaluation of 3d keypoint detectors for time-of-flight depth data.* In *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 1–6. 2016.

[10] Harasymowicz-Boggio, B., Chechliński, L., Siemiatkowska, B. *Nature-inspired, parallel object recognition.* In Szewczyk, R., Zieliński, C., Kaliczyńska, M., editors, *Progress in Automation, Robotics and Measuring Techniques. Control and Automation. Advances in Intelligent Systems and Computing vol. 350.* Springer, 2015.

[11] Harasymowicz-Boggio, B., Chechliński, u., Siemiatkowska, B. *Significance of features in object recognition using depth sensors.* Optica Applicata, 45(4):559–571, 2015.

[12] Holz, D., et al. *Robot soccer world cup xv.* chapter Real-time Plane Segmentation Using RGB-D Cameras, 306–317. Springer-Verlag, Berlin, Heidelberg, 2012.

[13] Izadi, S., et al. *Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera.* In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, UIST '11, 559–568. ACM, New York, NY, USA, 2011.

[14] Kalogerakis, E., Hertzmann, A., Singh, K. *Learning 3D mesh segmentation and labeling.* ACM Trans. Graph., 29(4).

[15] Lai, K., Bo, L., Fox, D. *Unsupervised feature learning for 3D scene labeling.* In *IEEE International Conference on on Robotics and Automation.* 2014.

[16] Nathan Silberman, P. K., Derek Hoiem, Fergus, R. *Indoor segmentation and support inference from RGBD images.* In *ECCV*. 2012.

[17] Neves, A. J. R., et al. *Object detection based on plane segmentation and features matching for a service robot.* International Journal of Computer, Electrical, Automation, Control and Information Engineering, 10(4):745 – 752, 2016.

[18] Osada, R., et al. *Shape distributions.* ACM Transactions on Graphics, 21(4):807–832, 2002.

[19] Quigley, M., et al. *Ros: an open-source robot operating system.* In *ICRA Workshop on Open Source Software*. 2009.

[20] Ren, X., Bo, L., Fox, D. *RGB-D scene labeling: Features and algorithms.* In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2759–2766. 2012.

[21] Rusu, R. B., Blodow, N., Beetz, M. *Fast point feature histograms (FPFH) for 3D registration.* In *in In Proceedings of the International Conference on Robotics and Automation (ICRA*. 2009.

[22] Rusu, R. B., Cousins, S. *3d is here: Point cloud library (pcl).* In *International Conference on Robotics and Automation*. Shanghai, China, 2011.

[23] Rusu, R. B., et al. *Learning Informative Point Classes for the Acquisition of Object Model Maps.* In *Proceedings of the 10th International Conference on Control, Automation, Robotics and Vision (ICARCV), Hanoi, Vietnam, December 17-20*. 2008.

[24] Rusu, R. B., et al. *Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments.* In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1–6. 2009.

[25] Rusu, R. B., et al. *Detecting and segmenting objects for mobile manipulation.* In *Proceedings of IEEE Workshop on Search in 3D and Video (S3DV), held in conjunction with the 12th IEEE International Conference on Computer Vision (ICCV)*. Kyoto, Japan, 2009.

[26] Rusu, R. B., et al. *Fast 3d recognition and pose using the viewpoint feature histogram.* In *Proceedings of the 23rd IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Taipei, Taiwan, 2010.

[27] Song, S., Xiao, J. *Deep sliding shapes for amodal 3d object detection in RGB-D images.* CoRR, abs/1511.02300, 2015.

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