A Learned Feature Descriptor for Object Recognition in RGB-D Data

Manuel Blum, Jost Tobias Springenberg, Jan Wülfing and Martin Riedmiller

***Abstract*— In this work we address the problem of feature extraction for object recognition in the context of cameras providing RGB and depth information (RGB-D data). We consider this problem in a bag of features like setting and propose a new, learned, local feature descriptor for RGB-D images, the *convolutional k-means descriptor*. The descriptor is based on recent results from the machine learning community. It automatically learns feature responses in the neighborhood of detected interest points and is able to combine all available information, such as color and depth into one, concise representation. To demonstrate the strength of this approach we show its applicability to different recognition problems. We evaluate the quality of the descriptor on the *RGB-D Object Dataset* where it is competitive with previously published results and propose an embedding into an image processing pipeline for object recognition and pose estimation.**

I. Introduction

Object recognition is an important problem in computer science appealing to researchers from different fields such as machine learning, computer vision and robotics. It has been widely studied resulting in a variety of methods, applications and standardized benchmark problems. The performance on these benchmarks (e.g. CIFAR [1] and NORB [2]) has constantly improved over the years. However, object recognition of everyday items in images of real-world scenes is still an open research problem. This is mainly due to the fact that many approaches performing well on the benchmarks do not scale to high-resolution data. Other reasons include the fact that real world objects often appear in cluttered scenes increasing the difficulty of the recognition problem. Objects are also often partially occluded and appear on different image scales. To further add to the problem set, object manipulation problems in robotics often require algorithms capable of object recognition in conjunction with pose detection. As a consequence the best performing algorithms for object recognition in real world scenes cannot reach the same performance as humans.

The state-of-the-art algorithms used in this setting try to avoid the aforementioned problems by extracting local feature descriptors around interest points. This method is commonly used in a BoW (short for *bag of words*) setting and has been successfully used in many different recognition problems. Most prominent methods within this class make use of hand-designed descriptors based on orientation histograms, such as SIFT [3] and SURF [4]. These methods, however, often discard available information. For example by using grayscale images only, despite the fact that most data sets provide color information as well. Since this restriction hampers the performance of all practical applications using these descriptors, different researchers have proposed to enrich their feature descriptors with additional, possibly more global information e.g. color histograms [5]. Recently, cheap RGB-D sensors like the Microsoft Kinect have been developed, which are capable of providing depth information for each pixel. This led to the availability of even richer image data, promising an increase in recognition performance. It is however unclear how this additional information can be exploited using traditional approaches.

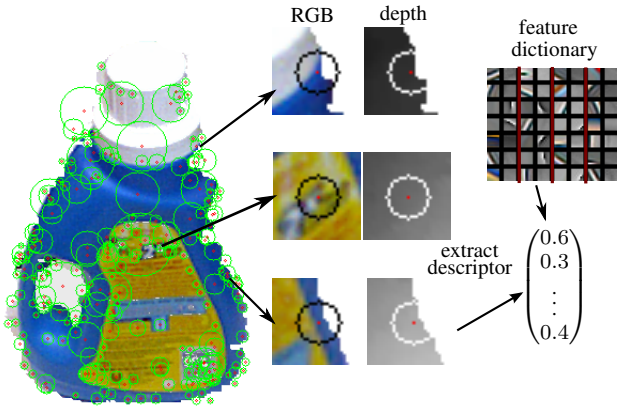


Fig. 1. The general descriptor extraction procedure for the *convolutional k-means descriptor*. First, a set of interest points is detected in the input image. Around each interest point a 16×16 px area is extracted. To build the feature descriptor for an image point feature responses from image patches of 6×6 px within this area are compared using the learned feature dictionary.

In this work we therefore propose a new, learned, feature descriptor based on recent advances in the machine learning community which we call the *convolutional k-means* descriptor (see Fig. 1). The machine learning approach treats all available information in a unified way. It can hence seamlessly integrate additional information such as the depth channel into one concise descriptor vector. We evaluate this feature descriptor in different object recognition settings on the *RGB-D Object Dataset* [6].

The paper is structured as follows: In Section II a brief survey of related work is presented. Then the proposed feature descriptor is described in Section III. In Section IV the performance of the descriptor is evaluated on the *RGB-D Object Dataset* [6] where it is empirically compared to several state-of-the-art algorithms. Finally, in Section V we propose to embed the descriptor into an image processing pipeline for object recognition and pose estimation in cluttered scenes in which the learned descriptor is accompanied by standard

This work was supported by the BMBF. All authors are with the Department of Computer Science, University of Freiburg, Germany.
{mblum,springj,wuelfj,riedmiller}@tf.uni-freiburg.de

methods for image segmentation and pose detection. This pipeline was tested as part of the *Solutions in Perception Challenge 2011*.

II. Related Work

This research focuses on object recognition using RGB-D data which is a newly established field mainly popularized by the availability of new, cheap sensors such as the *Microsoft Kinect*. Since we present a new feature descriptor for this setting we will briefly review current approaches to recognition in RGB-D data as well as the current state of feature extraction from images.

During the last years a number of different feature responses have been designed for object recognition tasks by the computer vision community. These feature responses are typically extracted from local image patches around detected interest points or using a fixed grid. The most prominent of these approaches are based on orientation histograms such as SIFT [3] and SURF [4]. While these feature responses are successfully used in numerous applications they are hard to design and cannot be easily adapted to incorporate additional information. Recent work on designing feature responses tries to approach this shortcoming by generalizing features based on orientation histograms to a broader class of so called kernel descriptors [7] which give a general design pattern for local feature responses and can make use of additional information in a unified way.

Work in the machine learning community has recently resulted in several different methods for learning low level feature responses from data. In particular the work on deep belief networks [8] and deep autoencoders [9] [10] resulted in object recognition architectures that achieve state of the art performance on several benchmarks. Work emphasizing the sparseness of the learned feature representations such as sparse coding [11] and local coordinate coding [12] comprise a related family of algorithms that have been successfully applied to object recognition tasks. Another interesting recent development is the work done by Coates et al. on unsupervised feature learning [13]. In this work the authors showed that good image features can be learned efficiently using standard unsupervised learning techniques, such as k-means clustering, if state-of-the-art image pre-processing and feature encoding is used.

In this work we consider a specific recognition setting in which the objects are represented using high resolution RGB-D data and propose to extract a feature histogram descriptor combining information from all 4 channels. To make our approach scalable to high resolution images we adapt the standard setting used by Hessian based approaches and chose to extract our learned feature responses around interest points, effectively substituting the hand designed Hessian descriptors. The descriptor is built from features, which are learned via a k-means approach that is adapted from the previously mentioned work in [13]. Our work is similar to work on kernel descriptors [7] in which a descriptor is built by comparing pixel orientations or color intensities. However, in contrast to this approach we do not explicitly design the used feature responses using pixel comparisons, but decide to learn a representative set of features which is then compared to the vicinity of the detected interest point.

III. Convolutional k-means descriptor

The extraction of image descriptors based on local patches around interest points is a successfully used approach in many computer vision tasks. It is however not straightforward to modify these methods to effectively use richer data representations. The idea of the *convolutional k-means descriptor* is to remedy this shortcoming by learning feature responses that make use of all available information.

To learn these features we adapt an unsupervised learning approach proposed by Coates et al. [13]. The algorithm was originally developed for feature extraction in object recognition benchmarks using small RGB or grayscale images (32×32 px for CIFAR [1], 96×96 px for NORB [2]). In contrast to this setting we however want to efficiently process large RGB-D images (e.g. images of up to 640×480 px containing additional depth information) as well as images of varying sizes which may contain objects that are partially occluded or appear in cluttered scenes. It is therefore necessary to modify the original algorithm in several ways.

First, to cope with the high dimensionality of the input signal, we propose to learn feature responses in the vicinity of interest points only. These feature responses are later combined into a feature descriptor (*the convolutional k-means descriptor*). Secondly, we add the depth information as a fourth channel to the vector representation of local image patches in order to make use of the data provided by the RGB-D sensor. Thirdly, we improve the unsupervised learning algorithm by modifying the pre-processing procedure and introducing a bootstrapping approach. These two modifications enable the algorithm to better cope with the high dimensionality of the input data and improve its convergence properties.

The procedure necessary to extract *convolutional k-means descriptors* from a given input image can be divided into three steps.

A. A set of feature responses must be learned capturing the image structure of patches around interest points.
B. An interest point detector is run on the input image.
C. *Convolutional k-means descriptor* responses are extracted around each detected interest point.

A. Learning feature responses

Our approach focuses on learning good feature representations for high resolution RGB-D images using unsupervised learning. We adapt the unsupervised learning formulation from [13] where the goal is to learn a set of feature responses $D \in \mathbb{R}^{N \times k}$ given a set of input vectors $X = \{x^{(1)}, \ldots, x^{(m)}\}$ with $x^{(i)} \in \mathbb{R}^N$. The input vectors are patches of $w \cdot w$ pixels extracted from a training set represented as column vectors. Each pixel is represented using d channels (4 in the case of RGB-D data) and hence $N = w \cdot w \cdot d$. Since we are interested in extracting a descriptor around detected interest points we generate the

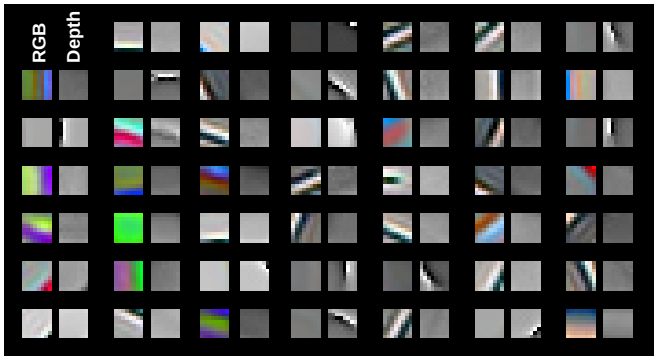
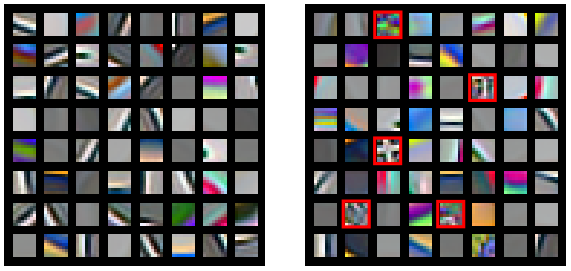


Fig. 2. Example features learned by the k-means approach. The displayed 48 features are part of a larger dictionary of 961 features that were learned in our experiments. We visualize the RGB as well as the depth channel for each feature next to each other.



(a) with bootstrapping (b) without bootstrapping

Fig. 3. Comparison of features learned from the *RGB-D object dataset* with and without bootstrapping. (a) When bootstrapping is enabled all learned centroids correspond to nicely localized features. (b) Without bootstrapping several cluster centers, marked in red, do not represent good feature responses due to the high dimensionality of the space in which k-means clustering is performed. Note that the depth channel is omitted in this picture to make visual comparison of the feature quality easier. Patches corresponding to features with depth selective responses thus appear in gray.

set of training patches X from the training images by first detecting interest points and then extracting a region around each interest point. As in the extraction of orientation histogram descriptors, we chose the size of this region to be 16×16 pixels (see Fig. 4). From this region random patches of size $w \cdot w$ pixels are extracted to build the training set X. Once X is known we apply a pre-processing step followed by the unsupervised learning algorithm.

1) Pre-processing: As a pre-processing step we first normalize all image patches contained in X by subtracting their mean and dividing by the standard deviation. Afterwards a PCA whitening transformation [14] is applied to the image patches. The purpose of the whitening transformation is to ensure that pixel values are decorrelated and have unit variance. This step is crucial to ensure a good quality of the learned feature responses as shown in [13]. We chose to use PCA whitening instead of the originally proposed ZCA whitening. This adaption to the original algorithm opens the opportunity to drop insignificant dimensions from the input data, which can result in increased feature extraction speed and improved feature quality. If dimensionality reduction is used, we chose to keep the first n components thereby projecting each extracted patch $x \in \mathbb{R}^N$ to a lower dimensional vector $x' \in \mathbb{R}^n$.

2) Unsupervised learning: We use a k-means approach to learn k centroids building the feature response dictionary D by clustering the extracted patches X. Although k-means is a very simple unsupervised learning algorithm that is easy to implement, it has recently been shown that it is competitive to other unsupervised learning algorithms when learning local, low-level features from pre-processed image data [13]. Apart from its simplicity the main advantage of using k-means over other algorithms is that it is very fast and scales well to a large amount of centroids. It can therefore be trivially parallelized on current computer hardware in a map-reduce manner and allows us to learn large, over-complete feature dictionaries that can be expensive to learn using other unsupervised learning approaches.

To further improve upon the feature quality that can be achieved using standard k-means, as well as the required runtime until convergence, we propose to use a bootstrapping learning approach to train the k centroids.

Since the data is clustered in PCA whitened space we can apply a bootstrapping learning scheme. We first cluster in the subspace spanned by the first p principal components and fill the learned centroids with zeros for all other $n - p$ dimensions. These centroids are then used to *warm start* the clustering procedure in the n dimensional PCA whitened space.

The application of this procedure is presented in Fig. 3. Without bootstrapping some features are badly localized, which is an artifact of clustering in a high dimensional space (e.g. 144 dimensions if image patches of 6×6 px are used). This effect is visible in Fig. 3 where affected features are marked red. When the bootstrapping procedure is used the features are well distributed over the whole feature space by pre-training the centroids on the major principal components. The consecutive clustering procedure in the complete feature space is thus simplified and the badly localized features disappear. An example set of features that is learned using data from the *RGB-D object dataset* is depicted in Fig. 2.

B. Interest point detection

Approaches to detect interest points traditionally comprise methods such as SIFT [3], SURF [4] and FAST [15] which mainly try to detect corners that are robust to image distortions and small transformations. More recently the problem of interest point detection in depth images has been addressed [16]. In our approach we chose to extract SURF corners as interest points since they are reasonably fast to compute and detect interest points of high quality. In some experiments, where computation time was no issue, we also used a fixed grid of interest points. It should be noted that in principle any interest point detector can be used for training the feature responses as well as extracting the descriptors. Especially the combination of our learned feature descriptor with 3D interest point detectors such as the one used in the NARF [16] feature extraction procedure is an interesting possibility.

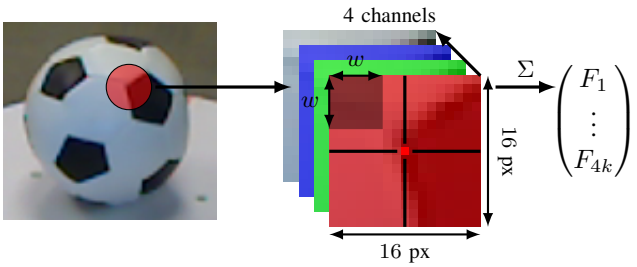


Fig. 4. The extraction procedure for the *convolutional k-means descriptor*. After an interest point is detected at position (p_x, p_y) the feature response $f(x)$ is computed for each $w \times w$ patch x in a 16×16 px region around the interest point. The feature responses are then summed in four regions around the interest point, thereby building the feature histograms that constitute the final descriptor vector.

C. Descriptor extraction

The extraction of the *convolutional k-means descriptor* around a given set of interest points is similar to the extraction of SURF descriptors. First, a 16×16 pixel region around the interest point is extracted. Then the feature response $f(x) \in \mathbb{R}^k$ of each $w \times w$ sub-patch $x \in \mathbb{R}^{w \times w \times d}$ in the region of interest is computed. For this study, we chose $w = 6$. If standard hard k-means is used the $f(x)$ is a sparse vector indicating the closest centroid

$$f_i(x) = \begin{cases} 1 & \text{if } c^i = \arg\min_{c^i \in D} \|c^i - x\| \\ 0 & \text{else} \end{cases}. \quad (1)$$

To maximize the information content of each feature response, a different function can be computed that keeps the information about the distance of the current patch to all centroids $c^i \in D$ that are closer than the average distance $\mu(z) = \frac{1}{k} \sum_{i=1}^{k} z_i$ where $z \in \mathbb{R}^k$ with $z_i = \|c^i - x\|$. In [13] this is called the triangular response. In this case $f(x)$ can be defined as

$$f_i(x) = \max(0, \mu(z) - z_i). \quad (2)$$

This is also the response we use for the *convolutional k-means descriptor* as it has been shown to yield better performance than the hard case. Finally, the feature responses are summed into feature histograms covering four regions as depicted in Fig. 4. The computation of the feature descriptor of an interest point at pixel position $p = (p_x, p_y)^T$ can thus be formalized as a vector $F(p_x, p_y) \in \mathbb{R}^{4 \cdot k}$ with

$$F_i(p_x, p_y) = \sum_{(p_{x'}, p_{y'})^T \in P} f_i(x_{(p_{x'}, p_{y'})}) \cdot \delta(p_x, p_y, p_{x'}, p_{y'}, i), \quad (3)$$

where P is the set of all pixels within the 16×16 region of interest, $x_{(p_i, p_j)}$ is the $w \times w$ patch around the pixel (p_i, p_j) and δ is an indicator function computing the corresponding descriptor region of the pixel given as

$$\delta(p_x, p_y, p_{x'}, p_{y'}, i) = \begin{cases} 1 \text{ if} & \max(0, \text{sgn}(p_{x'} - p_x)) + \\ & \max(0, \text{sgn}(p_{y'} - p_y)) \cdot 2 \\ & = \lfloor i/k \rfloor \\ 0 \text{ else} & \end{cases}. \quad (4)$$

Since the number of learned features k is typically high (common values range from 600 to 4000) the size of the descriptor vector also tends to be large. If the recognition architecture in which the descriptor extraction is embedded requires a smaller feature vector (for example when building a vocabulary tree for a large amount of real world objects) it can be beneficial to reduce its dimensionality. This can again be done by performing a PCA over the descriptors extracted from the training images and dropping all but the most significant principal components.

IV. Evaluation

We evaluate the performance of the *convolutional k-means descriptor* on the *RGB-D Object Dataset* [6]. We conduct three experiments. The first two focus on the influence of the PCA dimensionality reduction and the importance of the depth channel respectively. In a third experiment we empirically compare the object recognition performance of our *convolutional k-means descriptor* to several state-of-the art algorithms.

A. RGB-D Object Dataset

The *RGB-D Object Dataset* which is presented in [6] is a novel, large scale RGB-D dataset containing image sequences of 300 objects, grouped in 51 categories. Images are captured with a Kinect style camera at a resolution of 640×480 px. Each object is recorded from three viewing heights (30°, 45° and 60° above the horizon) while it rotates on a turntable resulting in approximately 150 views per object and 207920 RGB-D images in total. The images are already cropped and segmented, obviating the need for object detection and segmentation.

For our experiments, we use the same setup as in [6], distinguishing between category and instance recognition. For object recognition on the instance level, the task for the recognition algorithm is to detect the exact physical instance of an object that was previously presented as a training image. The object may appear in a different setting as the training instance, i.e. in a different scene, different lighting conditions and changing perspective. In the object recognition setting on the category level a set of training images with corresponding labels is presented. The labels group object instances into a set of categories (e.g. all instances of chairs are assigned the label *chair*). The objects presented in the recognition phase are new instances of a given category. The task for the algorithm then is to assign the correct label to the presented object.

To generate training and test sets, first, the data is subsampled by taking every fifth video frame resulting in 41942 RGB-D images. For category recognition, from each

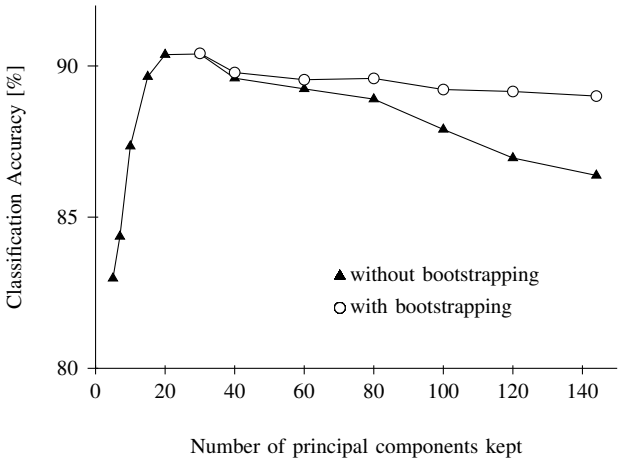


Fig. 5. Classification accuracy in the leave-sequence-out setting for different numbers of retained principal components with and without boostrapping.

category one object is selected randomly for testing and the classifier is trained on all remaining objects. For instance recognition there are two scenarios:

- Alternating contiguous frames: Each video is divided into 3 contiguous sequences of equal length. For each object there are three heights, giving a total of 9 video sequences for each instance. These are split randomly into 7 parts for training and 2 parts for testing.
- Leave-sequence-out: The classifier is trained on the video sequences from the 30° and 60° perspective and evaluated on the 45° video sequence.

B. Experiments

If not stated otherwise we use the same setup in all experimental scenarios (leave-sequence-out, alternating contiguous frames and category): We use a representative subset of the training images to learn 1200 features with a patch size of $w = 6$ px. The k-means algorithm for learning the feature dictionaries is run for a maximum of 100 episodes. To be able to compare our results with previously published studies ([17], [6], [18], [7]), we chose to extract feature descriptors for each object on fixed grid positions with a grid size of 4×4 px, instead of using an interest point detector. After the feature extraction we sum-pool the descriptor vectors over four quadrants to reduce the dimensionality of the final representation to 4 times the descriptor size and use a linear SVM for classification.

1) Influence of dimensionality reduction and bootstrapping: In the first experiment we evaluate the influence of the PCA dimensionality reduction and the bootstrapping procedure. For this purpose we consider the leave-sequence-out setting since good performance in this setting most directly correlates with discriminative feature representations for each object instance as well as robustness to image distortions.

We vary the number of principal components n that are kept. The maximum number of components is the original dimensionality of the patches extracted for learning the features, which is $N = w \cdot w \cdot d = 144$. The number of kept principal components is reduced step by step until $n = 5$. We conduct the experiment with and without bootstrapping.

Fig. 5 depicts the results of this experiment. By reducing the number of kept principal components (without bootstrapping) classification accuracy increases significantly from 86.4% ($n = 144$) to 90.4% ($n = 20$). Bootstrapping stabilizes classification accuracy in all experiments. Combining bootstrapping with dimensionality reduction also results in a performance gain which is, however, less pronounced. Apart from this performance increase, the computation time required for clustering and descriptor extraction is also reduced by the dimensionality reduction.

2) Contribution of depth information: We conducted a second experiment to test whether the additional depth information improves recognition accuracy. We applied the convolutional k-means approach on the dataset in two experimental conditions, one using all available 4 channels (RGB-D) and one using only color information (RGB) from the images. In both experiments, 961 features were learned while keeping the first 60 principal components. Ignoring depth information (RGB only), the recognition accuracy was 82.9%. Using all available channels, accuracy was significantly higher (89.5%) (RGB-D). This shows that the additional depth channel does carry additional information which is utilized by our algorithm.

A representative set of the resulting learned features is visualized in Fig. 2. There are two important observations: One, distinctive features for both the RGB and the depth channel were learned and, two, most features are either selective for the depth or the RGB channel.

3) Performance on the RGB-D Object Dataset: In a third experiment, we compare the performance of the *convolutional k-means descriptor* on the RGB-D Object Dataset [6] to several state-of-the-art algorithms. Except for the leave-sequence-out setting, results were averaged over 10 trials.

The *convolutional k-means descriptor* outperforms the currently best performing approaches in all three scenarios. The results are shown in Table I. In the leave-sequence-out setting, the best result was achieved using Kernel Descriptors [7] which reached an accuracy of 84.5%. Using the *convo-*

TABLE I
CLASSIFICATION ACCURACY ON THE RGB-D OBJECT DATASET. (A) LEAVE-SEQUENCE-OUT (B) ALTERNATING CONTIGUOUS FRAMES; ACCURACIES ARE AVERAGED OVER 10 TRIALS.

Method	Instance		Category
	(a)	(b)	
Linear SVM [6]	73.9	90.2 ± 0.6	81.9 ± 2.8
Nonlinear SVM [6]	74.8	90.6 ± 0.6	83.8 ± 3.5
Random Forest [6]	73.1	90.5 ± 0.4	79.6 ± 4.0
IDL [17]	-	91.3 ± 0.3	85.4 ± 3.2
HKDES [18]	82.4	-	84.1 ± 2.2
Kernel Desc. [7]	84.5	-	86.2 ± 2.1
CKM Desc. (this work)	**90.4**	$\mathbf{92.1} \pm 0.4$	$\mathbf{86.4} \pm 2.3$

lutional k-means descriptor this result could be improved by a large margin (5.9%). There is also an improvement on the other two scenarios compared to the performance of previously published methods.

V. Solutions in Perception Challenge

In order to solve the task of object recognition in cluttered scenes in conjunction with pose detection, we embed our descriptor into a complete RGB-D image processing pipeline. It was used at the Solutions in Perception Challenge, held at the 2011 IEEE International Conference on Robotics and Automation in Shanghai, China, where our method ranked fourth. The task of this challenge was to recognize the identity and pose of rigid, Lambertian, textured objects using the *Microsoft Kinect* sensor. Training and test sequences of 35 objects, mostly supermarket articles, were provided beforehand. Fifteen additional objects produced by NIST were introduced at the contest.

The pipeline can roughly be divided into three parts: Image segmentation, object recognition and pose detection. Since test scenes may contain multiple objects at a time it is necessary to first segment the images. Using contextual knowledge (e.g. objects are placed on a table), we solve the segmentation task by using RANSAC plane fitting to find the table in the scene and discard all points underneath it. Assuming that there is enough free space between the objects, the remaining points are assigned to object clusters, which is achieved efficiently using a k-d tree representation of the input point cloud combined with a flood fill algorithm.

We consider object recognition as a two-step process. First, *convolutional k-means descriptors* are extracted at interest points detected by a standard method. Secondly, we train a vocabulary tree [19] in conjunction with a database of object views for classification. In the recognition phase the database can then be queried for the best matching view of an object. This view is then used to estimate the 6DoF pose in the pose detection phase.

For 6DoF pose detection, we rely mainly on the depth information (3D point clouds) provided by the Microsoft Kinect. We first compute an initial, rough pose estimate using the 3D coordinates of corresponding interest points. Correspondences are determined based on descriptor similarity. The 6DoF pose estimate is further refined by applying the Iterative Closest Point algorithm [20] on the point clouds.

VI. Conclusion

In this work we have presented a new descriptor, the *convolutional k-means descriptor*, that relies on learned feature responses. The approach is able to learn meaningful features from RGB as well as depth data automatically. Its performance was evaluated on the *RGB-D Object Dataset* in three scenarios, where it was able to improve on previously published results. Especially in the instance recognition problem, the best performing algorithms to date were outperformed by a large margin, underlining the capabilities of the machine learning approach. To the best of our knowledge our descriptor therefore has the highest accuracy on this dataset in all three recognition settings. We believe that a learned feature descriptor is a valuable tool for efficiently combining different sensory modalities, such as RGB-D data and will help to improve the object recognition performance in real world applications.

References

[1] Alex Krizhevsky. *Learning Multiple Layers of Features from Tiny Images*. Master thesis, Department of Computer Science, University of Toronto, 2009.

[2] Yann LeCun, Fu-Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[3] David G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, 2006.

[5] Alaa E. Abdel-Hakim and Aly A. Farag. CSIFT: A SIFT Descriptor with Color Invariant Characteristics. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[6] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. A Large-Scale Hierarchical Multi-View RGB-D Object Dataset. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[7] Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Depth Kernel Descriptors for Object Recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[8] Geoffrey E Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11:428–34, October 2007.

[9] Dan C. Ciresan, Ueli Meier, Jonathan Masci, Luca M. Gambardella, and Jürgen Schmidhuber. High-Performance Neural Networks for Visual Object Classification. Technical report, Dalle Molle Institute for Artificial Intelligence, Manno, Switzerland, 2011.

[10] Quoc V Le, Jiquan Ngiam, Zhenghao Chen, Daniel Chia, Pang Wei Koh, and Andrew Y. Ng. Tiled convolutional neural networks. *Advances in Neural Information Processing Systems*, 2010.

[11] Adam Coates, Andrew Y. Ng, and Serra Mall. The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2011.

[12] Kai Yu and T. Zhang. Improved local coordinate coding using local tangents. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, 2010.

[13] Adam Coates, Honglak Lee, and Andrew Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[14] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–30, 2000.

[15] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*, 2006.

[16] Bastian Steder, R.B. Rusu, Kurt Konolige, and W. Burgard. Point Feature Extraction on 3D Range Scans Taking into Account Object Boundaries. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[17] Kevin Lai, Liefeng Bo, Xiaofeng Ren, and Dieter Fox. Sparse Distance Learning for Object Recognition Combining RGB and Depth Information. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[18] Liefeng Bo, Kevin Lai, Xiaofeng Ren, and Dieter Fox. Object Recognition with Hierarchical Kernel Descriptors. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2011.

[19] David Nister and Henrik Stewenius. Scalable recognition with a vocabulary tree. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[20] P.J. Besl and N.D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on pattern analysis and machine intelligence*, pages 239–256, 1992.