Fast Self-supervised On-line Training for Object Recognition Specifically for Robotic Applications

Markus Schoeler, Simon Christoph Stein, Jeremie Papon, Alexey Abramov and Florentin Wörgötter
Georg-August University of Göttingen, III. Physikalisches Institut - Biophysik, Göttingen, Germany

Keywords: Object Recognition, On-line Training, Local Feature Orientation, Invariant Features, Vision Pipeline.

Abstract: Today most recognition pipelines are trained at an off-line stage, providing systems with pre-segmented images and predefined objects, or at an on-line stage, which requires a human supervisor to tediously control the learning. Self-Supervised on-line training of recognition pipelines without human intervention is a highly desirable goal, as it allows systems to learn unknown, environment specific objects on-the-fly. We propose a fast and automatic system, which can extract and learn unknown objects with minimal human intervention by employing a two-level pipeline combining the advantages of RGB-D sensors for object extraction and high-resolution cameras for object recognition. Furthermore, we significantly improve recognition results with local features by implementing a novel keypoint orientation scheme, which leads to highly invariant but discriminative object signatures. Using only one image per object for training, our system is able to achieve a recognition rate of 79% for 18 objects, benchmarked on 42 scenes with random poses, scales and occlusion, while only taking 7 seconds for the training. Additionally, we evaluate our orientation scheme on the state-of-the-art 56-object SDU-dataset boosting accuracy for one training view per object by +37% to 78% and peaking at a performance of 98% for 11 training views.

1 INTRODUCTION

Creating recognition systems which can quickly adapt to new and changing environments is not only a challenging but also highly desirable goal for the machine vision community. Solving this goal is especially important for creating machines (robots), which are able to assist humans in their daily life, as this task requires robots to interact with a multitude of objects it may encounter in a household. This, in turn, depends on successful detection and recognition of objects relevant for potential actions. Unluckily object recognition still remains one of the hardest tasks in computer vision, which leads to failures in todays robotic applications (Szeliski, 2010). One reason is that classification performance scales badly with the number of trained classes, which prohibits training the recognition system of a robot to deal with all possible objects it may encounter. One way to solve this problem is to reduce the objects to the most likely classes for a specific environment (a robot working in a kitchen will probably not need the knowledge about a hay-fork). However, this inevitably limits the robot to the most probable classes from the designers point of view. Furthermore recognizing specific instances (like the red coffee cup) is not possible. We, on the other hand, want to pursue a different path. We want to create a robot which is able to do quick, automatic and robust learning from scratch, enabling it to adapt to new or changing environments and only learning objects it encounters. Consequently our system needs to deal with the following problems in the training stage:

T1 Automatic detection and extraction of object candidates from the scene without prior object knowledge.

T2 Automatic training set generation with minimal human intervention.

T3 Dealing with a training set which is as small as possible and preferably just made of one observation per object (users should not spend their time rearranging objects for the robot to generate a large training set).

T4 Quick training of the recognition system.

For the recognition stage the system needs to deal with additional problems:

R1 Quick and robust recognition of objects in a scene (especially dealing with different distances, poses and occlusion of objects).

Schoeler M., Stein S., Papon J., Abramov A. and Woergoetter F..
Fast Self-supervised On-line Training for Object Recognition Specifically for Robotic Applications.
DOI: 10.5220/0004688000940103
In *Proceedings of the 9th International Conference on Computer Vision Theory and Applications* (VISAPP-2014), pages 94-103
ISBN: 978-989-758-004-8

R2 Determining the 3D coordinates of all objects for subsequent manipulations.

We address these issues by providing:

- A new two stage vision pipeline combining low resolution 3D information for object detection and high resolution 2D information for object recognition. 3D information is needed to make extraction of unknown objects on textured background possible (see Section 3.1). In addition using a high-resolution camera does significantly improve object recognition due to the much higher quality visual information as we show in section 4.2.
- A novel orientation scheme for local keypoints, denoted as **Radial**, which is rotation invariant but includes information about the object shape and thus making object signatures much more discriminative. We show that it outperforms state-of-the-art orientation schemes on two benchmarks in section 4.2 and 4.3.
- A fusion of two classifiers using Gray-SIFT (Lowe, 2004) and a simple local color feature (**CyColor**), which is based on the hue and saturation channels of the HSV-colorspace. This combination, called **Fused**, is not only much faster to extract than color versions of SIFT, but also significantly boosts recognition performance on the benchmarked datasets.

This enabled us to build a system which works online and highly automatically. It starts completely untrained, continues with fully automatic object extraction and leads to reliable object recognition.

2 RELATED WORK

Although there are many recognition systems tackling some of the aforementioned problems, only few of them work fully automatic starting without object knowledge and with minimal human intervention. The reason is that most systems which try to extract objects from 2D images already need a trained classifier or rely on video streams and human manipulation to extract moving objects (Gall et al., 2011; Schiebener et al., 2011; Welke et al., 2010; Zhou et al., 2008). While there are methods which use a trained classification algorithm to semantically segment static images (Lai et al., 2012; Vijayanarasimhan and Grauman, 2011), few of them can extract unknown objects, like in (Iravani et al., 2011) where the authors threshold the spatial density of SIFT features or in (Ekvall et al., 2006) where a background subtraction algorithm is employed. Unfortunately both systems have their drawbacks. In the first case objects can only be placed on texture free ground and in the second case training requires a pick and place-back action by a human supervisor, thus being not fully automatic (see problem **T1** and **T2**). Furthermore, using just 2D images will not enable the robot to infer the absolute position of an object in the room, thus rendering it helpless when trying to execute an action and failing at problem **R2**.

Two other good approaches are presented in (Schiebener et al., 2011) and (Welke et al., 2010). The authors of the first work extract objects by physical robot interaction. Features are being tracked during the manipulation and simple geometrical models (planes and cylinders) are fitted to the point clouds for building object models. This method needs objects which are textured for reliable feature matching as well as objects which can be described by planes and cylinders. Furthermore, the robot needs to move all objects it encounters for training as well as for recognition, which dramatically slows down the system. In the second work objects are put into the hand of the robot and multiple images of the object are acquired while turning it. Since objects have to be segmented from the background using a stereo camera, problems with untextured objects or objects similar to the background emerge. Also holding an object in the hand can occlude important parts for the training, especially for small objects like the pen we use in our experiments.

To compare object recognition pipelines, researches often rely on publicly available benchmarks like the *RGB-D Object Dataset* (Lai et al., 2011) or the *KIT ObjectModels Web Database* (Kasper et al., 2012). We did not use them, because results for comparison are only available for turntable recordings, where objects are placed in the same spot and recorded from different inclinations. This is a very constrained scenario as objects are always placed upright and in-plane rotation is minimal. Instead, we used the SDU-dataset (Mustafa et al., 2013), which consists of single objects in arbitrary poses, but in a fixed distance and without occlusion. Robots, however, specifically also face objects in random distances and with occlusion, while working in human environments. Therefore, we recorded a new publicly available benchmark based on cluttered, high-resolution scenes with multiple objects partially occluding each other in random distances and poses[1]. This benchmark has been created using our proposed object detection pipeline.

[1] http://www.dpi.physik.uni-goettingen.de/~mschoeler/public/42-scenes/

3 METHODS

To automatically detect, extract and recognize objects in the scene, and thus solving problems **T1** and **R1**, we implemented a vision system which consists of two sensors:

1. RGB-D sensor for **object detection and extraction** (Section 3.1).
2. High-resolution 2D camera for the **object recognition** (Section 3.2).

Starting at an untrained recognition system the robot makes use of 3D information provided by the RGB-D sensor to automatically extract the object in front of it. Hereupon the vision system creates a mask and warps it to the reference frame of the high-resolution camera, takes an image and saves it for the training. The only job of the human supervisor is to actually tell the robot the names of the encountered objects, which addresses problem **T2**.

3.1 Object Detection and Extraction

All data from the RGB-D camera is processed in the form of point clouds. Creating object masks is done in the following way utilizing functions from the point cloud library (Rusu and Cousins, 2011):

1. The point cloud (see Figure 1 A and B) is downsampled for faster processing using a voxelgrid-filter.
2. The groundplane is subtracted (see Figure 1 C and D) by using a RANSAC plane fit to the voxelized cloud and deleting the respective inliers (This leaves a set of disconnected object candidates in our cloud, see Figure 1 C and D).
3. An Euclidean clustering scheme with a fixed distance threshold is applied to the cloud and all voxels within a cluster are treated as belonging to one object.

For all experiments a voxel resolution of 5 mm, a groundplane separation threshold of 5 mm and a clustering threshold of 4 cm have been used. The resulting labeled voxel cloud is then projected onto the high-resolution camera frame (see Figure 1 F), and for each indiviual cluster a 2D mask is created using the positions of the projected points belonging to that cluster. Since the number of projected voxels for one object is much smaller than the actual pixel count on the high-resolution image covering the object (due to the difference in resolution), we extend each projected voxel on the image by the average distance to the nearest neighboring voxel with the same label. This allows

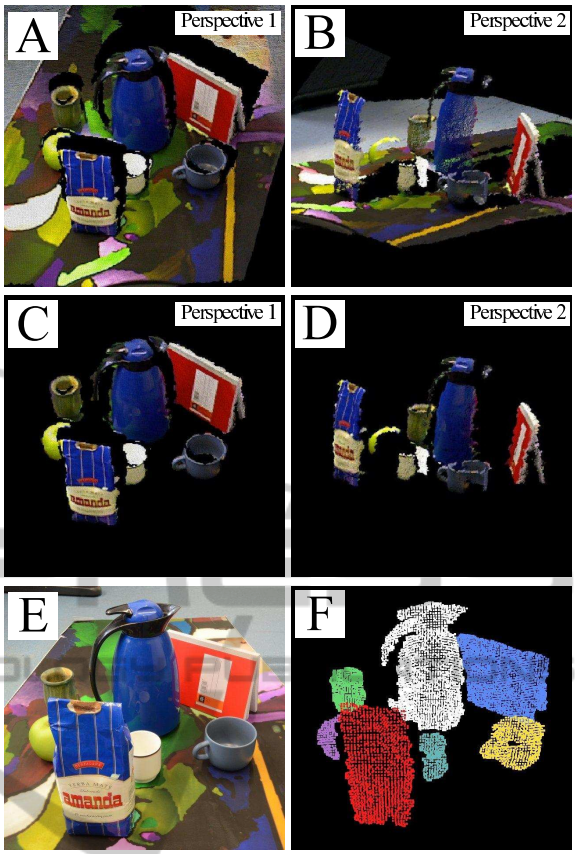

Figure 1: Process chain for extracting objects from the scene. A and B: The acquired point cloud from the RGB-D camera from different perspectives. C and D: The point cloud after groundplane subtraction. E: High-resolution camera image. F: Projected voxels on high-resolution image.

us to create a full mask for each object in the high-resolution image instead of just having a sparse set of pixels from the projection. Note that this simple scheme can provide us with fast, robust and accurate segmentation even for scenes which are cluttered in 2D or with textured background, as long as the visible parts of the objects are not touching in 3D space. Since we already possess complete 3D information for all objects, unlike systems which are working in 2D solely, we automatically solved problem **R2** as well.

3.2 Object Recognition

We implemented two recognition pipelines to incorporate full RGB information. One based on color versions of SIFT and a faster version fusing two disjoint classifiers: Gray-SIFT with a three dimensional **CyColor** feature. This combination will be denoted as **Fused**. We chose SIFT features as they are con-

sidered state-of-the-art and are widely used in recent works of object recognition (Silberman and Fergus, 2011; Zhou et al., 2010; Van de Sande et al., 2010; Binder et al., 2011; Bo et al., 2011). In all cases a bag-of-words algorithm with k-means clustering (Csurka et al., 2004) is employed to generate compact signatures for the objects. We use all descriptors of up to 5 images per object for the vocabulary generation (about 4000 to 30000 for all objects). We cluster them to 300 visual words using k-means and generate signatures by binning each descriptor to the nearest visual word in L2-distance. The resulting signatures are used with one-versus-rest support vector machines (SVM) (Vapnik, 1998) using a histogram intersection-kernel (Barla et al., 2003) for the classification.

3.2.1 Radial Orientation Scheme

While we leave the SIFT descriptors untouched, we do adapt the detector step (determining the location, size and orientation of the keypoints) to leverage on the additional information provided by the first part of our pipeline 3.1. Keypoint locations are placed on a regular grid within each object mask with a stepsize of $\Delta D/d_{Step}$, with d_{Step} being a fixed number and ΔD being the diagonal of the mask size ($\Delta D = \sqrt{\Delta h^2 + \Delta w^2}$, Δh and Δw denoting the height and width of the mask's bounding box, respectively). In our experiments a value of 14 for d_{Step} yielded a good trade-off between classification performance and speed. An overview of how we are locating the keypoints can be seen in Figure 2.

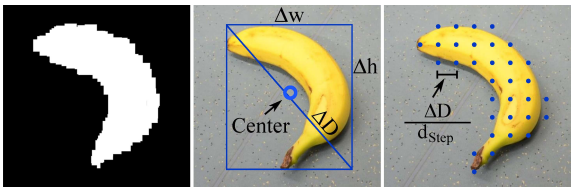

Figure 2: Defining keypoint location and object center. Left: Extracted Object mask, Middle: Determined object size and center, Right: Keypoint locations.

For each location we extract SIFT features on four different scales $\frac{\Delta D}{80} 2.5^l (l = 0,1,2,3)$. Using four different sizes makes our signatures more robust to small errors in object size estimation in case of occlusion and is a common technique in the literature (Gehler and Nowozin, 2009; Bosch et al., 2007b). In contrast to the aforementioned works we are not using a fixed scale but instead scale our SIFT features with the dimension of the mask, which makes our classification scheme completely robust to scale variations, even for unknown object, which addresses problem **R1**.

As the orientation of the keypoint decides, if the resulting signatures are invariant to in-plane rotation, we want to briefly discuss two popular approaches found in the literature: **Local** gradient and **Fixed** orientation. The **Local** gradient scheme orients features along the dominant local brightness gradient of the image patch around each keypoint. This is by far the most widely used orientation scheme (Lowe, 2004; Zhou et al., 2010; Silberman and Fergus, 2011; Bosch et al., 2007b; Bosch et al., 2007a) as it makes image signatures invariant to in-plane rotation. This unfortunately sacrifices discriminative power (Calonder et al., 2010), as important information about the object shape, encoded in the orientation of the dominant local gradient, is lost. Consequently an important cue, describing the object shape, is missing. The **Fixed** orientation scheme on the contrary orients all keypoints in a fixed direction (Calonder et al., 2010; Bay et al., 2008), thus incorporating the shape information into the signature and as a result making it more discriminative. This however comes at the cost of making it variant to in-plane object rotation. To make our signatures robust to inplane-rotation, but still keeping their discriminative power, we introduce a simple, but powerful novel orientation scheme named **Radial**. For this we approximate the center of the object by determining the middle of the object mask and orient all keypoints in a radial manner, pointing away from the center. An example of the three orientation schemes is depicted in Figure 3. Note that using the **Radial** orientation scheme requires knowledge about each object's location, which we retrieve by segmenting in 3D directly.

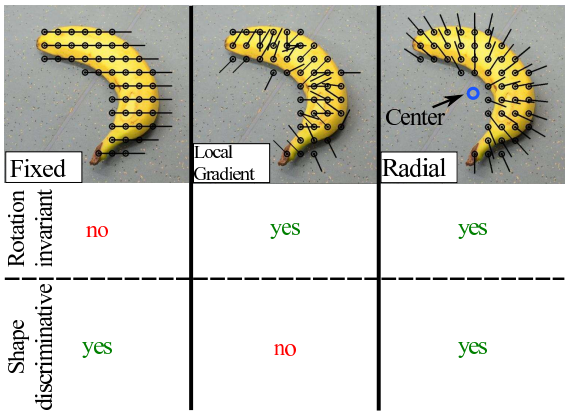

Figure 3: Comparing the three keypoint orientation schemes: **Fixed** orientation, **Local** gradient orientation and our **Radial** orientation.

3.2.2 CyColor Feature

Traditionally SIFT descriptors are extracted on gray-scale images. One popular possibility to use full RGB

information is to extract SIFT descriptors on all channels of an image separately and concatenating each channel's descriptors to one big descriptor (Van de Sande et al., 2010; Bosch et al., 2007a). While this generally boosts recognition performance all operations for SIFT have to be repeated on three channels, which makes the feature extraction slow. Additionally, SIFT-based feature can not deal with textureless objects as SIFT only considers gradients. To speed up the feature extraction and to cope with textureless objects, we employed a second much faster feature which we will call **CyColor**. This feature is extracted using the local pixel value at a keypoint location in HSV-colorspace. To account for the cyclic nature of the hue channel, we defined the three dimensional feature vector $\vec{f}_C$ in the following way:

$$\vec{f}_C = [sin(2\pi H), cos(2\pi H), S],$$

with H and S denoting the hue and saturation value [0,1]. Using this feature vector one easily gets rid of the problematic cyclic nature of the hue channel, while still being mostly robust to illumination variations, since we ignore the value channel. Since our **CyColor** feature itself does not cover shape of the object, we always fuse it with Gray-SIFT as described in the next section.

3.2.3 Fused Classifier

There are multiple ways to fuse different classifiers (Rodriguez et al., 2007; Gehler and Nowozin, 2009), but most of these methods need a large training set to determine meaningful weights via cross-validation. We on the other side want to keep the training set as small as possible (see problem **T3**). Consequently one robust weighting scheme (when intra object variance for the individual features is unknown), is averaging the classification results. For this we train two independent classifiers: One using Gray-SIFT and one using aforementioned **CyColor** feature. Each classifier extracts features on the same keypoints. For the classification we use one-against-rest SVMs and average their scores such that class j gets the score:

$$Score(j) = \frac{1}{2}\left(Score_{CyColor}(j) + Score_{SIFT}(j)\right)$$

4 EXPERIMENTAL EVALUATION

Our main goal is to create a system which can be trained as fast as possible with minimal human intervention. Consequently we investigate how the different orientation schemes and features deal with a limited number of training samples. This is important as it shows how many observations the robot needs to robustly recognize the objects and therefore how fast the robot learns to distinguish between objects starting from an untrained system. We tested our system on two datasets: Our own publicly available scene benchmark and on the SDU-dataset which was kindly provided by the authors (Mustafa et al., 2013).

4.1 Experiment on 42-Scenes Benchmark

For the 42-Scenes Benchmark we recorded about 60 images per object in different poses and under different lighting conditions using the proposed object extraction pipeline. All objects are shown in Figure 4. For object recognition the robot was only allowed to select a fixed number of images per object from this pool for the training. After the classifier was trained, we exposed it to a new scene with several objects being placed in random orientation, distance and pose with partial occlusion up to 50%. Each object was shown in 15 scenes. Example scenes together with masks and classification results are depicted in Figure 5.

Figure 4: All objects used in the experiments.

To simulate the effect of a reduced image resolution, as one would encounter when directly using the RGB-D camera images for the recognition, we reduced the high-resolution images and masks from their original size $R_{full} = 2464 \times 1632$ pixels to the maximum Microsoft Kinect resolution $R_{low} = 1280 \times 1024$ pixels using bilinear interpolation. First note that the result of this operation still yields much higher quality images (less noise, sharper contrasts) as compared to the images you can retrieve with the Microsoft Kinect and second that using the depth channel limits your resolution to 640×480 pixels.

We compare three different features HSV-SIFT (Bosch et al., 2007b; Bosch et al., 2007a; Gehler and Nowozin, 2009), Opponent-SIFT (Van de Sande et al., 2010) and **Fused** (Section 3.2.3), as they all in-

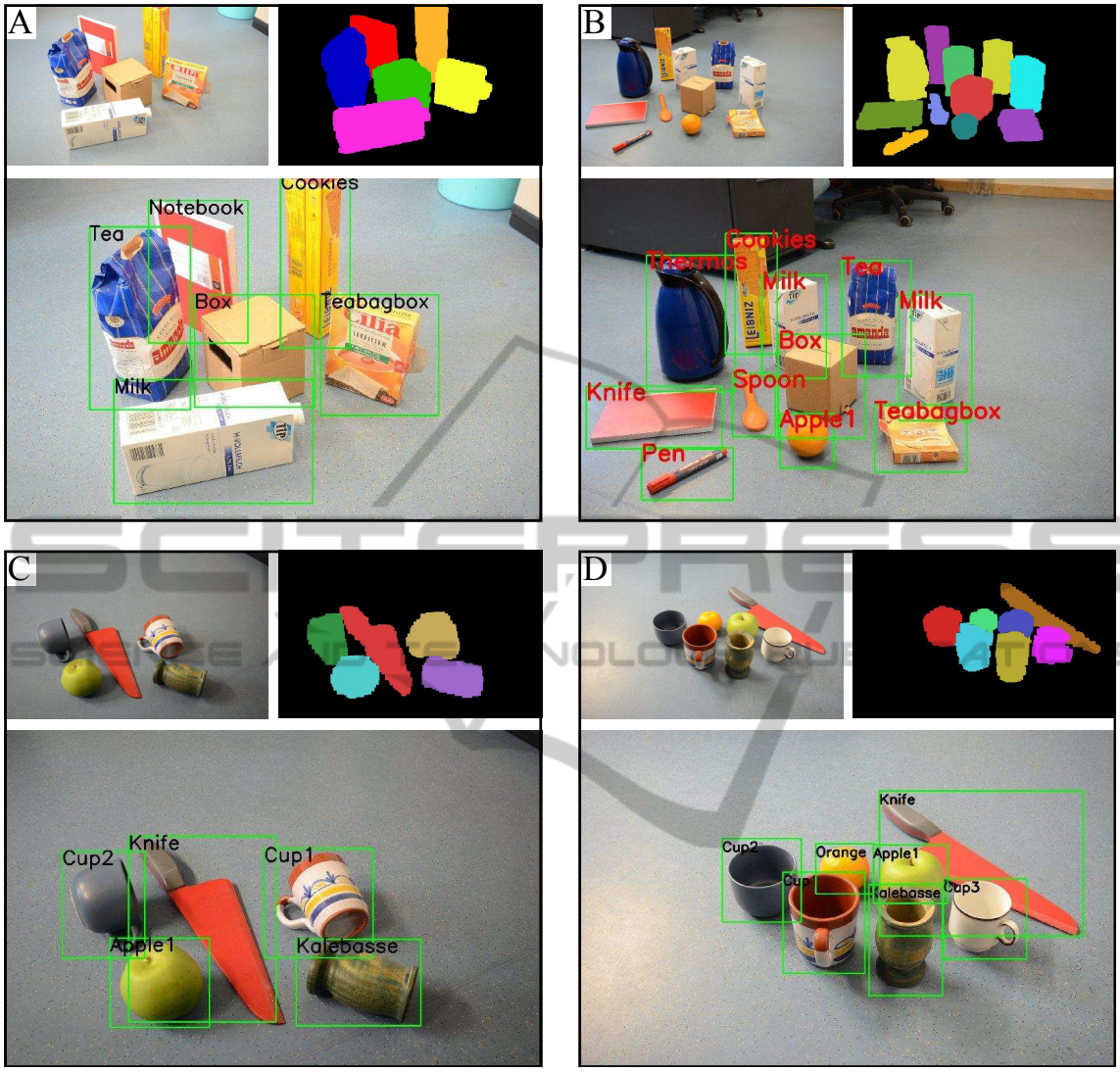

Figure 5: Four scenes from our 42-Scenes Benchmark with cluttered objects in various poses which the robot had to solve. (A-D): Top left: High-resolution scene image, Top right: Automatically extracted masks. Bottom: Example outcome using **Radial-Fused**.

corporate color information. For HSV-SIFT, features are extracted on each of the three HSV-channels separately and concatenated to form a $3 \times 128 = 384$ dimensional vector. Opponent-SIFT does the same but on the three CIELAB channels. Altogether we compared the six following classification algorithms:

- **Fixed-HSV-Low.** Fixed keypoint detector with HSV-SIFT features on reduced scene resolution R_{low}.
- **Fixed-HSV.** Fixed keypoint detector with HSV-SIFT features on full scene resolution R_{full}.
- **Local-HSV.** Local gradient detector with HSV-SIFT features on full scene resolution R_{full}.
- **Radial-HSV.** Radial keypoint detector with HSV-SIFT features on full scene resolution R_{full}.
- **Radial-Opponent.** Radial keypoint detector with Opponent-SIFT features on full scene resolution R_{full}.
- **Radial-Fused.** Radial keypoint detector with Gray-SIFT features combined with **CyColor** features (as described in Section 3.2.3) on full scene resolution R_{full}.

To compare each classifier's performance we averaged the F1-score (Hu et al., 2009) across all objects and across all scenes for 20 runs with a random draw of training images (see Figure 6). We decided to use the F1-score, as it puts equal weights on precision and recall and therefore describing the overall perfor-

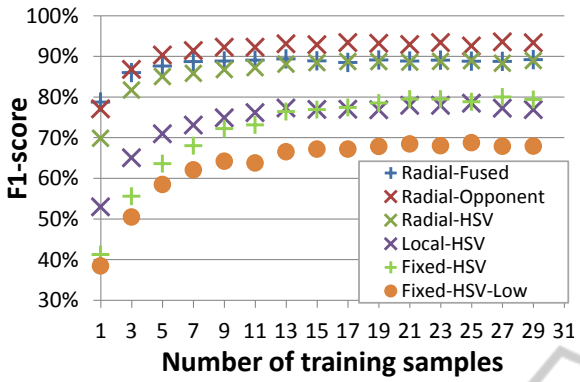

Figure 6: Averaged F1-score across all objects in all 42 scenes using different classification algorithms versus number of training images per object.

mance of the recognition system. It ranges from 0 to 1 (0 being worst and 1 being best) and is defined as

$$F1 = 2\frac{PR}{P+R},$$

with P and R denoting precision and recall, respectively. For comparing the processing time of the different classifiers we used an Intel i7 hexacore processor with 3.2 GHz per core.

4.2 Discussion 42-Scenes Benchmark

Influence of Image Resolution
When comparing **Fixed-HSV-Low** and **Fixed-HSV**, the lower resolution R_{low} in general decreased the classification results significantly. While the difference for one training image per object is negligible (as both have a bad performance), an increasing number of training images shows the influence of the lower resolution. Due to the low resolution visual information of the object is lost for the SIFT features, which decreased their discriminative power. We found out, that reducing the image resolution to 640×480, the average F1-score again decreased by roughly 6% compared to the resolution R_{low}. This is consistent with the findings of Ekvall et al. (Ekvall et al., 2006) who also noted a decrease in performance when decreasing resolution. This strongly emphasizes the importance of image resolution and justifies our approach to combine a low resolution RGB-D sensor and a high-resolution RGB camera.

Influence of Orientation Scheme
When comparing the three orientation schemes (**Fixed-HSV**, **Local-HSV**, **Radial-HSV**), two regions can be analyzed separately:

1. Few training images per object ($\leq$ 11 images per object)
2. Enough training images per object, such that an increase does not improve classifier performance significantly ($>$ 11 images per object)

Few Training Images: Having only a few images puts very high emphasis on the signature itself. Orientation schemes which produce signatures invariant to in-plane rotation like **Radial** and **Local** are superior to orientation schemes which are fragile to object rotation like **Fixed**. Accordingly one needs only a few images to generalize to the full object using the former scheme. The distinction between **Radial** and **Local** is caused by the poor discriminative power of the **Local** scheme as described in Section 3.2.1.
Many Training Images: Using many images the robustness of the signature becomes less important, because different object poses are known for the training. Here the performance is more dictated by the power of the SVM, which uses all signatures as input to separate the classes. Consequently orientation schemes which lead to discriminative signatures (**Radial** and **Fixed**) work better in this regime. This is the reason why the performance of **Local** drops below the performance of **Fixed** when increasing trainingset size.

Since our orientation scheme is robust to in-plane rotation, but still discriminative (see Figure 3) , it is by far the best choice for robotic applications using local features and improves classification by 16.9% for one training image per object and about 10% for the saturated region. Please also note that the **Radial** orientation scheme is very fast to calculate (in average 4 ms per object).

Influence of Feature Selection
Comparing **Radial-HSV** and **Radial-Opponent** one clearly sees that the Opponent version of the SIFT descriptor is superior to the HSV version. This confirms the findings of (Van de Sande et al., 2010) where the authors compared several color extensions of SIFT. When only a few images are available the classifiers **Radial-Fused** and **Radial-Opponent** perform equally good, with **Radial-Fused** being slightly better for one training image (2%) and **Radial-Opponent** being better for more than 5 images per object (3%). The reason for the fused classifier to perform better for a very small number of training images is that the **CyColor** descriptor is fully invariant to any object rotation in 3D as long as the same side is visible or the object color distribution does not change too much when rotating the object. Consequently we achieve results of 79% for **Radial-Fused**, 77% for **Radial-Opponent** and 70% for **Radial-HSV** when using only one image per object. In the saturated region ($>$ 11 images per object) the

performance of **Radial-Fused** and **Radial-HSV** are identical, because we train already with a variety of different poses. This means that being completely pose invariant and being only invariant to in-plane rotation does not make a big difference any more. Here **Radial-Opponent** shows better results than **Radial-Fused** with an average score of 93.1%, but at the cost of being slower as shown below. Noteworthy is that using the red-green and yellow-blue channels of the Opponent-color space as base for our **CyColor** feature decreased the performance of the **Fused** classifier to 76% for one training image. The reason is, that the Opponent-color space is not invariant to lighting variations in contrast to the H and S channels of the HSV color space (Van de Sande et al., 2010). This is a severe problem when using the absolute values of these channels, whereas considering gradients, as SIFT does, circumvents this problem.

Time Performance

Since we are also interested in the speed of the recognition system (problems **T4** and **R1**), we measured the time for object extraction and compared the training and recognition time for the three best scoring classifiers. The average time for the object extraction in a scene with 6 objects is 30 ms. Table 1 shows that **Radial-HSV** and **Radial-Opponent** are 3 times slower for the recognition and about 2 times slower for the training compared to the combination of Gray-SIFT and **CyColor**. This result is not surprising as the SIFT feature extraction step is by far the slowest part of the whole recognition process and consequently doing it on three channels instead of just one increases the processing time significantly. The training time grows approximately linearly with the number of images used for the training, again showing the advantage of a classification algorithm which can deal with a small number of training images. To reach the maximum performance (**Radial-Opponent** with 13 images per object) training takes 201 s. Consequently a decision has to be made, whether a high recognition rate or a fast system is preferred.

Table 1: Comparing training time and average object recognition time for **Radial-HSV**, **Radial-Opponent** and **Radial-Fused** using a single training image per object.

Classifier	Training [s]	Recognition [s]
Radial-Fused	6.8	0.23
Radial-Opponent	14.7	0.66
Radial-HSV	15.7	0.68

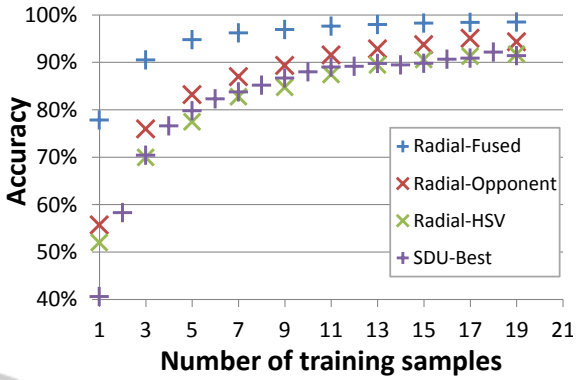

Figure 7: Averaged classification accuracy versus number of training images for the SDU-dataset.

4.3 Experiment and Discussion SDU-dataset

For the SDU-dataset (Mustafa et al., 2013) we followed the same experimental procedure as described in the paper. Since we are interested to see how well our classifier deals with a limited number of training samples, we mainly compared to Figure 7 in their paper. Figure 7 shows accuracy (mean of the confusion matrix) of our three highest scoring **Radial** classifiers on the SDU-dataset as well as results of their best scoring classifier in the paper (named SDU-Best which uses a combination of a point cloud feature and a hue-saturation histogram, (Mustafa et al., 2013)-Figure 7-pink curve). All parameters are left unchanged. As can be seen, all **Radial** classifiers are supperior to the SDU-Best classifier, altough we did not use the depth channel. **Radial-Fused** is by far the best scoring classifier (especially for few training samples) with an accuracy increase of 37% compared to SDU-Best (78% versus 41%) for a single training view. Using 11 and more training views per object, accuracy increases to 98%, which is a big improvement over the state-of-the-art as presented in the paper. This is a significant result as it shows how valuable absolute color information (only provided by the **CyColor** descriptors) is for object recognition especially for few training samples.

As stated by the authors knowing about the shape is indispensable for robust recognition. While they need 3D information to calculate shape descriptors, our **Radial** orientation scheme includes shape information in a natural way directly from 2D data.

5 CONCLUSIONS

This work presented a recognition system, which can

adapt to new and changing environments. This is especially important for robots assisting humans in their daily life. To achieve this a system needs to deal with problems **T1** - **R2** as described in our introduction. Therefore we designed a two stage pipeline, featuring fast, automatic and robust learning of objects with minimal human intervention. In the first stage (**Object detection and extraction**) the robot uses 3D information from the RGB-D sensor to automatically retrieve objects from cluttered scenes. Projecting all object masks to a high-resolution camera, we were able to provide the second stage of the recognition system (**Object recognition**) with accurate and detailed visual information.

We tested our recognition system in two scenarios: First with 18 objects with varying poses, illumination and distances in 42 scenes with partial occlusion and second on the SDU-dataset with 56 objects in arbitrary poses. The former is made publicly available. Comparing results of the SDU-Benchmark to our 42-Scenes Benchmark, one can see that our benchmark is more challenging, although the SDU-dataset uses more objects. The reason is twofold: First, we did not put any constrains on object pose, distance as well as illumination, and second, we evaluate on a collection of labeled and masked scenes which show occlusion, making the recognition more difficult. In both benchmarks our novel **Radial** orientation scheme achieved better than state-of-the-art results. This is because our orientation scheme leads to signatures which do incorporate shape information in contrast to widely used local gradient orientation schemes. Furthermore, using a simple fusion of Gray-SIFT and our three dimensional **CyColor** feature did not only speed up the recognition pipeline (7 s for the full training in our 42-Scenes Benchmark), but also boosts classification accuracy for the SDU-dataset significantly. This shows the value of absolute color information for object recognition, especially for few training samples. The combination of our **Radial** orientation scheme with our **CyColor** features leads to an improvement over the state-of-the-art on the SDU-dataset by +37% to a total of 78% for only a single training view and to 98% for 11 training views.

REFERENCES

Barla, A., Odone, F., and Verri, A. (2003). Histogram intersection kernel for image classification. In *Image Processing(ICIP), 2003 International Conference on*, volume 3, pages III–513–16 vol.2.

Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (surf). *Computer Vision Image Understanding*, 110(3):346–359.

Binder, A., Wojcikiewicz, W., Müller, C., and Kawanabe, M. (2011). A hybrid supervised-unsupervised vocabulary generation algorithm for visual concept recognition. In *Proceedings of the 10th Asian conference on Computer vision - Volume Part III*, ACCV'10, pages 95–108, Berlin, Heidelberg. Springer-Verlag.

Bo, L., Ren, X., and Fox, D. (2011). Depth kernel descriptors for object recognition. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 821–826.

Bosch, A., Zisserman, A., and Munoz, X. (2007a). Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, CIVR '07, pages 401–408, New York, NY, USA. ACM.

Bosch, A., Zisserman, A., and Muoz, X. (2007b). Image classification using random forests and ferns. In *Computer Vision (ICCV), 2007 IEEE 11th International Conference on*, pages 1–8.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. (2010). Brief: binary robust independent elementary features. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 778–792, Berlin, Heidelberg. Springer-Verlag.

Csurka, G., Dance, C. R., Fan, L., Willamowski, J., and Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22.

Ekvall, S., Jensfelt, P., and Kragic, D. (2006). Integrating active mobile robot object recognition and slam in natural environments. In *Intelligent Robots and Systems (IROS), 2006 IEEE/RSJ International Conference on*, pages 5792–5797.

Gall, J., Fossati, A., and Van Gool, L. (2011). Functional categorization of objects using real-time markerless motion capture. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1969–1976.

Gehler, P. and Nowozin, S. (2009). On feature combination for multiclass object classification. In *Computer Vision (ICCV), 2009 IEEE 12th International Conference on*, pages 221–228.

Hu, X., Zhang, X., Lu, C., Park, E. K., and Zhou, X. (2009). Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 389–396, New York, NY, USA. ACM.

Iravani, P., Hall, P., Beale, D., Charron, C., and Hicks, Y. (2011). Visual object classification by robots, using on-line, self-supervised learning. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1092–1099.

Kasper, A., Xue, Z., and Dillmann, R. (2012). The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research (IHRR)*, 31(8):927–934.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. In

Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1817–1824.

Lai, K., Bo, L., Ren, X., and Fox, D. (2012). Detection-based object labeling in 3d scenes. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1330–1337.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Mustafa, W., Pugeault, N., and Krger, N. (2013). Multi-view object recognition using view-point invariant shape relations and appearance information. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*.

Rodriguez, B., Peterson, G., and Agaian, S. (2007). Multi-class classification averaging fusion for detecting steganography. In *System of Systems Engineering, 2007 IEEE International Conference on*, pages 1–5.

Rusu, R. and Cousins, S. (2011). 3d is here: Point cloud library (pcl). In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1–4.

Schiebener, D., Ude, A., Morimotot, J., Asfour, T., and Dillmann, R. (2011). Segmentation and learning of unknown objects through physical interaction. In *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*, pages 500–506.

Silberman, N. and Fergus, R. (2011). Indoor scene segmentation using a structured light sensor. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 601–608.

Szeliski, R. (2010). Computer vision: Algorithms and applications. In *Computer Vision: Algorithms and Applications*, page 657. Springer.

Van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M. (2010). Evaluating color descriptors for object and scene recognition. 32(9):1582–1596.

Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, 1 edition.

Vijayanarasimhan, S. and Grauman, K. (2011). Efficient region search for object detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1401–1408.

Welke, K., Issac, J., Schiebener, D., Asfour, T., and Dillmann, R. (2010). Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2012–2019.

Zhou, F., Torre, F., and Hodgins, J. (2008). Aligned cluster analysis for temporal segmentation of human motion. In *Automatic Face Gesture Recognition, 2008 IEEE International Conference on*, pages 1–7.

Zhou, X., Yu, K., Zhang, T., and Huang, T. S. (2010). Image classification using super-vector coding of local image descriptors. In *Proceedings of the 11th European conference on Computer vision: Part V*, ECCV'10, pages 141–154, Berlin, Heidelberg. Springer-Verlag.