The bins for the x- and y-location in the accumulator, i_x and i_y, are calculated as shown in Eq. 4 and Eq. 5,

$$i_x = \left\lfloor \frac{c_{x,s}}{w} b_x \right\rfloor \tag{4}$$

$$i_y = \left\lfloor \frac{c_{y,s}}{h} b_y \right\rfloor \tag{5}$$

where w and h refer to the image width and height, and b_x and b_y is the number of bins in the x- and y-dimension of the Hough-accumulator, respectively. Apart from i_x and i_y also the corresponding bins with position indices $i_x + 1$ and $i_y + 1$ are incremented to reduce discretization errors.

The index for the rotation bin is calculated using the difference between the angles of the key point correspondences α and the total number of bins reserved for the rotation in the accumulator b_r. Originally, α is in $[-\pi, \pi]$, thus Eq. 6 normalizes the angle to be in $[0, 2\pi]$.

$$i'_r = \frac{(\alpha + \pi) b_r}{2\pi} \tag{6}$$

To allow for the rotations by $-\pi$ and π to be close together in the accumulator the final index for the rotation bin is calculated according to Eq. 7.

$$i_r = \lfloor i'_r \rfloor \mathrm{mod} b_r \tag{7}$$

Again, a second bin is used to reduce discretization errors. It's index is calculated according to Eq. 8:

$$i_r = \lfloor i'_r + 1 \rfloor \mathrm{mod} b_r. \tag{8}$$

The fourth dimension in the accumulator encodes the scale the point of interest was found at. To determine the accumulator bin for scale, first the ratio q between the scales of the key point in the scene σ_s and in the learned object σ_o is needed (Eq. 9):

$$q = \frac{\sigma_s}{\sigma_o}. \tag{9}$$

Further, the index is determined by the total number of bins used to represent scale b_s and the number of octaves n used for SURF extraction and is calculated according to Eq. 10:

$$b_s = \left\lfloor \frac{\log_2(q)}{2(n-1)} + 0.5 \right\rfloor b_s. \tag{10}$$

As before, discretization errors are reduced by using a second bin with the index $b_s + 1$. All scales that go beyond the range represented by the last bin are subsumed in the bin for the biggest scale of the accumulator.

As a result of the Hough-transform clustering all features with consistent poses are sorted into bins, while most outliers are removed because they do not form maxima in Hough-space (Fig. 3). So far, all features were processed independently of all other features without taking into account the geometry of the whole object. With the resulting possible object poses from the Hough-transform clustering, the goal in the next step is to find the best geometric match with all features in one accumulator bin.

3) Homography Calculation: In this step, bins representing maxima in Hough-space are inspected in order to find the bin that matches best the object pose. All bins containing five key point correspondences or more are considered as maxima. A perspective transformation is calculated between the features of a bin and the corresponding points in the database under the assumption that all features lie on a 2D plane. As most outliers were removed by discarding minima in Hough-space, a consistent transformation is obtained here. Random Sample Consensus (RANSAC) is used to identify the best homography for the set of correspondences. The homography with most point correspondences is considered to be the correct object pose. Using the obtained homography the recognized object can be projected into the scene (Fig. 8). Since homography calculation is computationally expensive the runtime of the object recognition algorithm would increase considerably if a homography was calculated for each bin. To speed up the algorithm all bins are sorted in descending order considering their number of features. A homography is calculated starting with the bin containing the highest number of features. The calculation terminates if the next bin contains less features than the number of found point correspondences in the calculation of the best homography so far. The result is a homography describing the relative position, orientation and scale of the best fitting training image for a test image, as well as the number of features supporting this hypothesis.

4) Verification of Results: The last step of our object recognition pipeline verifies the results. Using a threshold of a minimal matched feature number to verify the presence of an object in the scene is futile since large and heterogeneously structured objects contain more features than small and homogeneously structured objects. Instead, an object presence probability p is calculated as

$$p = \frac{f_m}{f_t} \tag{11}$$

where f_m is the number of matched features of that object and f_t is the total number of features that are present in the area of the object. The number of features in the object area is calculated by projecting the object into the scene using the homography and then counting all features in the bounding box of the projected object.

IV. Evaluation

This Section describes the different experiments performed to evaluate the presented object recognition approach. Experiments were performed to test the influence of the accumulator size, variable background and light conditions, as well as partial occlusion on the performance of the classification.

For the verification step we used a threshold of $p = 15\,\%$ (Eq. 11) and a minimum of 5 matched features per object. These two values have the most influence on the number of false positive recognitions. If they are not chosen restrictively

(a) (b) (c)

Fig. 4: Object view used for time measurement, green arrows indicate key points (a). Key point correspondences after nearest neighbor matching: some key points of the learned object view are matched to the background (b). Recognition result after eliminating erroneous correspondences (c).

TABLE I: Calculation time for each algorithm step, measured with the object from Fig. 4.

Algorithm Step	# Key Points	Time [ms]
Detection	500	697
NN-Matching	139	61
Hough-clustering	102	13
Homography	98	12

enough, the number of false positive recognitions increases. On the other hand, if they are chosen too restrictively, no objects would be recognized or a higher number of training views per object would be required to provide enough features for matching.

When not stated otherwise the accumulator size is 10 bins in each dimension. For the evaluation objects from the database presented in [13] were used. All images in this database have a resolution of 640×480 pixels.

A. *Performance*

The evaluation was performed on an off-the-shelf notebook with an Intel Core 2 processor with 2 GHz and 2048 MB RAM. We measured the processing time of the algorithm for one object view and a scene image with weak heterogeneous background and the learned object in the same pose. The object view used in this test is depicted in Fig. 4. The processing time needed for each step of our object recognition algorithm is presented in Tab. I.

Initially, 500 key points are detected in the scene image. The key point detection step takes the most time, since key points are not only extracted from the object, but also from the background. For instance, the nearest neighbor matching yields 139 key point correspondences between the scene image and an example object view (Fig. 4). However, some of these correspondences are erroneous as some of them are matched with the background. After the Hough-transform clustering only 102 correspondences remain. Finally, after calculating two homographies in 12 ms the best is found with 98 remaining correspondences to form the object recognition result (Fig. 4).

A total of 783 ms is needed for the calculation. This time increases if multiple object and object views are loaded into the object database, as the extracted features have to be compared with all data in the database. It is therefore crucial not to learn too many views of an object (see next Subsection). However, the step consuming the most time (key point extraction) has to be performed only once per scene image, regardless of the number of learned objects in the database.

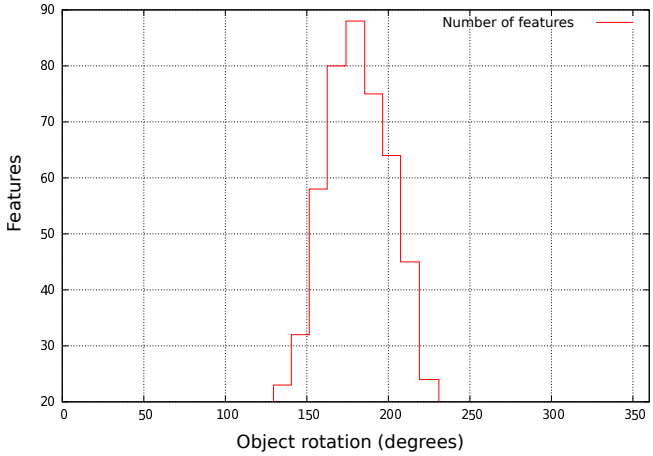


Fig. 5: Number of matched features of an object view depending on the object's rotation. The object view was acquired at a rotation of 180°.

TABLE II: Object recognition results on images with different backgrounds. The numbers in brackets indicate the number of false positive recognitions.

Object	hom. back.	weak het. back.	strong het. back.
bscup	100 %	90 % (1)	100 %
nizoral	100 %	100 % (2)	90 %
perrier	100 %	100 % (1)	100 % (2)
ricola	100 %	100 % (1)	100 %
truck	100 %	90 %	70 % (1)

B. *Learning Object Views*

We performed experiments to determine how many object views are necessary for reliable object recognition. In order to do this it has to be determined by what angle an object can be rotated without loosing too many key points. For the evaluation, a single object view was acquired. Without loss of

Fig. 6: Objects from [13] used for evaluation. From left to right: *bscup*, *nizoral*, *perrier*, *ricola*, *truck*. All objects are shown with homogeneous background.

generality the object view was defined as depicting a pose with a rotation of 180° about its vertical axis. Subsequently, images from the database showing the object at different rotations were used for testing. As shown in Fig. 5 the number of matched features decreases rapidly for rotations beneath 150° and above 220°. Thus, a rotation of 30° between subsequently acquired image views is a good trade-off between the number of found features and image views.

C. Accumulator Size

The size of each dimension of the accumulator is a crucial parameter for the performance of the algorithm. In our approach 10 bins per dimension proved to be a good trade-off between quantization errors (if too many bins are used) and insufficient accuracy (if too little bins are used). More than 10 bins lead to a shorter runtime as the features are distributed among a greater bin number, thus leading to less bins with a sufficiently large number of features for further processing. However, at the same time less features can be matched leading to unreliable recognition results.

D. Background Variation

The experimental results of our algorithm with different backgrounds are presented in Tab. II. A comparison of our algorithm with a statistical object recognition approach was given in [2]. The algorithm was trained with 5 different objects (Fig. 6) and 5 views per object from [13]. The classification was performed on the same 5 objects, but with 10 different views per object. The employed database contains images of the same objects with homogeneous, weak heterogeneous and strong heterogeneous backgrounds (Fig. 7). Different light conditions are present in the images with non-homogeneous backgrounds.

With increasing heterogeneity of the background, more erroneous correspondences are matched. If their number is very high, a false positive recognition occurs. A challenge in recognition is posed by the objects *perrier* and *truck* as they are small compared to the overall image size. With the low image resolution only few pixels remain for the object and thus only a little number of features is extracted. During the Technical Challenge of the RoboCup we used a higher image resolution. Please refer to Sec. IV-F for more details.

E. Partial Occlusion

Another experiment was performed to test the algorithm with partially occluded objects. Occlusion was simulated by partially replacing the object in the test data with the corresponding background. The results are presented in Fig. 8. The

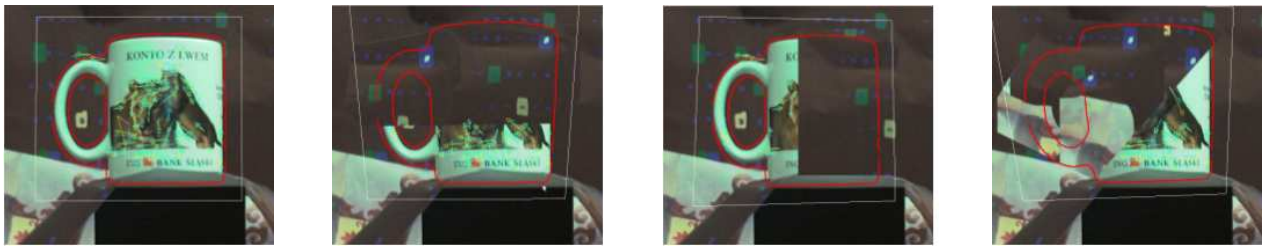
Fig. 7: Object *nizoral* from [13] with different backgrounds. From left to right: homogeneous, weak heterogeneous, and strong heterogeneous backgrounds.

unoccluded object is recognized with a total of 98 matched features and a confidence of 38 % (Eq. 11). With increasing occlusion the number of features decreases, but is still high enough for a correct recognition of the object. However, with increasing occlusion the accuracy of the computed homography (red lines in Fig. 8) and thus of the bounding box decreases.

F. RoboCup@Home: Technical Challenge

This object recognition approach was also applied during the Technical Challenge in the @Home league of the RoboCup world championship that took place in Mexico-City in 2012. 50 objects were placed on a table containing randomly selected 15 of 25 previously known objects. Our robot could correctly identify 12 of the 15 present known objects correctly, while at the same time having no false positive recognitions. This recognition result was achieved with a single scene view. With this result our robot places first in the Technical Challenge and won the Technical Challenge Award. The input image for object recognition as well as the recognition results are shown in Fig. 9.

We use a difference of 30° between object views to minimize training time and the number of images in the database. Objects were trained and recognized with an off-the-shelf digital camera (Canon PowerShot SX100 IS) and an image resolution of 8 megapixels (MP). Since the object recognition took a long processing time, further tests with the RoboCup@Home objects were performed after the Technical Challenge (Tab. III). The total recognition time depends on the resolution in the training phase as well as on the resolution of the scene image. However, the resolution in the training has a greater influence on the total recognition time. According to Tab. III it is sufficient to create an object database where features are extracted from 4 MP images, but use a resolution of 8 MP for recognition. This is not surprising since the object



(a) (b) (c) (d)

Fig. 8: Example images for detection of partially occluded objects. The unoccluded object is recognized with 98 matched features and 38 % confidence (a). The occluded object have less features, but are still recognized correctly: 49 with 33 % (b), 17 with 17 % (c), and 34 with 34 % (d).

(a)

(b)

Fig. 9: The input image for object recognition as acquired by our robot during the Technical Challenge of the RoboCup (a) and the output image depicting the recognition results (b). During training and recognition an image resolution of 8 MP was used.

to camera distance is usually smaller during training than during recognition. Thus, even with a lower image resolution a sufficient number of features can be extracted and saved in the database during training.

V. Conclusion

We presented our object recognition approach that we use in the RoboCup@Home league. Our approach is based on SURF features that are clustered in Hough-space. Hough-clustering allows us to discard most erroneous correspondences in the detection step. Subsequently, homographies are calcu-

TABLE III: Comparison of different image resolutions and their effect on recognition time and recognition quality.

Resolution Training [MP]	Resolution Scene [MP]	Recognition Time [s]	True Positives	False Positives
4	4	20	5	1
4	8	26	12	0
8	4	53	6	1
8	8	117	12	0

lated to take into account the relative positions of all features on the object. The homography that best matches the object geometry is back projected into the scene image to indicate the position of the recognized object.

Our recognition approach performs well on images with cluttered background and partially occluded objects. Objects at different scales and with arbitrary poses in the scene image are recognized reliably. For best results, it is recommended to use high resolution images in order to extract sufficient features for object representation.

This object recognition algorithm is available as an open source ROS package and can be downloaded from [3]. Apart from the recognition algorithm itself, the ROS package provides a convenient GUI to learn new object models and test the recognition. With this GUI new object models can be learned in less then a minute.

Our future work will concentrate on further evaluating and optimizing our approach. We plan to test several key point detectors and descriptors, as well as test different Hough-clustering techniques.

REFERENCES

[1] H. Bay, T. Tuytelaars, and L. J. V. Gool, "SURF: Speeded up robust features," *ECCV*, pp. 404–417, 2006.

[2] P. Decker, S. Thierfelder, D. Paulus, and M. Grzegorzek, "Dense Statistic Versus Sparse Feature Based Approach for 3D Object Recognition," *Pattern Recognition and Image Analysis*, vol. 21, no. 2, pp. 238–241, 2011.

[3] AGAS. (2014, Jul.) Ros package for object recognition based on hough-transform clustering of surf. [Online]. Available: http://wiki.ros.org/agas-ros-pkg

[4] B. Leibe and B. Schiele, "Interleaved object categorization and segmentation," in *BMVC*, 2003.

[5] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *ECCV' 04 Workshop on Statistical Learning in Computer Vision*, 2004, pp. 17–32.

[6] C. Harris and M. Stephens, "A combined corner and edge detector," in *Fourth Alvey Vision Conference*, Manchester, UK, 1988, pp. 147–151.

[7] L. A. Pineda, C. Rascon, G. Fuentes, V. Estrada, A. Rodriguez, I. Meza, H. Ortega, M. Reyes, M. Pena, J. Duran *et al.*, "The golem team, robocup@ home 2014."

[8] A. Collet, M. Martinez, and S. S. Srinivasa, "The moped framework: Object recognition and pose estimation for manipulation," *The International Journal of Robotics Research*, p. 0278364911401765, 2011.

[9] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] J. Stuckler and S. Behnke, "Integrating indoor mobility, object manipulation, and intuitive interaction for domestic service tasks," in *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*. IEEE, 2009, pp. 506–513.

[11] M. Muja, "Flann, fast library for approximate nearest neighbors," 2009, http://mloss.org/software/view/143/.

[12] W. E. L. Grimson and D. P. Huttenlocher, "On the sensitivity of the hough transform for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-12, no. 3, pp. 255–274, 1990.

[13] M. Grzegorzek and H. Niemann, "Statistical object recognition including color modeling," in *2nd International Conference on Image Analysis and Recognition*. Toronto, Canada: Springer, 2005, pp. 481–489.