

Evaluation of the SIFT Object Recognition Method For a Mobile Robotic Platform

Arnau Ramisa[†], David Aldavert[§], Shrihari Vasudevan[‡], Ricardo Toledo[§]
and Ramon Lopez de Mantaras[†]

[†] *IIIA-CSIC, Campus UAB, 08193, Bellaterra, Spain, {aramisa, mantaras}@iiia.csic.es*

[§] *Computer Vision Center, Campus UAB, Bellaterra, E-08193, Spain. {aldavert, ricard}@cvc.uab.cat*

[‡] *Australian Center for Field Robotics, the University of Sydney, NSW 2006, Australia.*

shrihari.vasudevan@ieee.org

Abstract

This paper addresses object perception applied to mobile robotics. Being able to perceive semantically meaningful objects in unstructured environments is a key capability in order to make robots suitable to perform high-level tasks in home environments. However, finding a solution for this task is daunting: it requires the ability to handle the variability in image formation in a moving camera with tight time constraints. Here we extensively evaluate the well known SIFT object recognition method in a very challenging dataset of 30 objects captured while navigating with a mobile robot in a typical indoor setting. The obtained results are compared with those obtained in the same dataset by two other popular state of the art object recognition methods: the Vocabulary Tree and the Boosted Cascade of Weak Classifiers.

keywords: object detection, computer vision, mobile robotics

1 INTRODUCTION

Research in robotic systems is progressing towards machines with high level cognitive capabilities, able to conduct complex tasks even in unstructured and dynamic contexts such as a typical domestic environment. Getting around in this setting will require this robots to have advanced perception capabilities, able not just to help navigate but also to identify relevant objects for the task at hand.

Although different modalities of perception (e.g. laser range-finder, color camera, haptics) can be used, in this work we focus on passive vision, as it is interesting for several reasons, like an affordable cost, compatibility with human environments or richness of perceived information.

Object recognition is a very active research area in computer vision, that produced several very successful methods for certain specific tasks, such as face or car detection. Nevertheless, the usefulness of current generalist object recognition algorithms for cognitive robotics applications is largely unexplored.

In the present work, the popular SIFT object detection algorithm [6] is evaluated for this task, and compared to two other state of the art methods, the Vocabulary Tree method [12] and the boosted cascade of weak classifiers [14]. We show that the SIFT object recognition method obtains the best results for textured objects, while having a runtime below 1 second per frame in a not completely optimized implementation.

The paper is divided as follows: First, in Section 2, we present work related to ours. Next, the dataset used to evaluate the SIFT object recognition method is presented in Section 3. Follows a brief description of the SIFT object recognition method in Section 4, and the experiments performed and results obtained in Section 5 and Section 6 respectively. Finally, in Section 7 the conclusions of the evaluation are presented.

2 RELATED WORK

Probably the work most related with ours is the one of [7], where four methods (SIFT and KPCA+SVM with texture and color features) were combined in an object recognition/classification task for human-robot interaction. The appropriate method for each class of object was chosen automatically from the nine combinations of task/method/features available, and models of the learned objects were improved during interaction with the user (pictured as a handicapped person in the paper). This work was, however, more focused on building a working object classification method suitable for the particular task of human-robot interaction with feedback from the human user, and not in evaluating each particular method in a standardized way. Furthermore, no quantitative results were reported for the experiments with the robot.

Mikolajczyk et al. [9, 8] did a comprehensive comparison of interest region detectors and descriptors in the context of keypoint matching. Although this works are undoubtedly related with the one presented here, the objectives of the comparison are notably different: while Mikolajczyk et al. measured the repeatability of the region detectors and the matching precision of the region descriptors, here we focus on the performance of three well-known object recognition methods in the very specific setting of mobile robotics.

3 DATASETS AND PERFORMANCE METRICS

In order to evaluate the methods in a realistic mobile robots setting, we have created the IIIA30 dataset ¹, that consists of three sequences of different length acquired by our mobile robot while navigating at approximately 50 cm/s in a laboratory type environment and approximately twenty good quality images for training taken with a standard digital camera. The camera mounted in the robot is a Sony DFW-VL500 and the image size is 640x480 pixels. In Figure 1 the robotic platform used can be seen. The environment has not been modified in any way and the object instances in the test images

¹<http://www.iiia.csic.es/~aramisa/iiia30.html>

are affected by lightning changes, blur caused by the motion of the robot, occlusion and large viewpoint and scale changes.



Figure 1: Robotic platform used in the experiments.

We have considered a total of 30 categories (29 objects and background) that appear in the sequences. The objects have been selected to cover a wide range of characteristics: some are textured and flat, like the posters, while others are textureless and only defined by its shape. Figure 2.a shows the training images for all the object categories, and 2.b shows some cropped object instances from the test images. Each occurrence of an object in the video sequences has been manually annotated in each frame to construct the ground truth, along with its particular image characteristics (e.g. blurred, occluded...).

In order to evaluate the performance of the different methods we used several standard metrics that are briefly explained in the following lines. Precision is defined as the ratio of true positives among all the positively labeled examples, and reflects how accurate our classifier is.

$$Pre = \frac{TruePositives}{FalsePositives + TruePositives} \tag{1}$$

Recall measures the percentage of true positives that our classifier has been able to label as such. Namely,

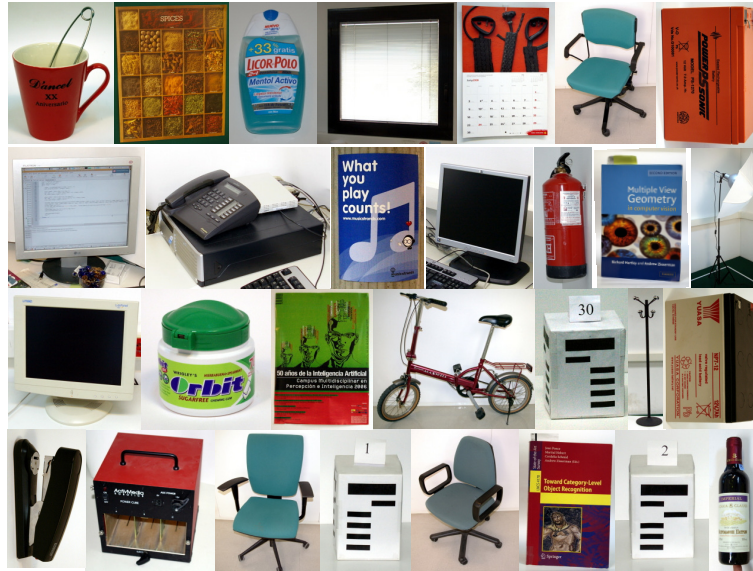
$$Rec = \frac{TruePositives}{FalseNegatives + TruePositives} \tag{2}$$

Since it is equally important to perform well in both metrics, we also considered the f —measure metric:

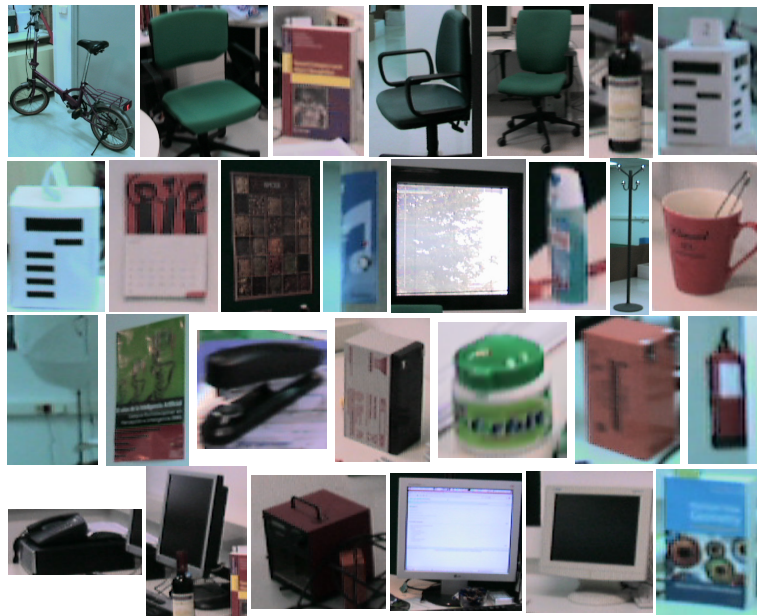
$$f - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3}$$

This measure assigns a single score to an operating point of our classifier weighting equally precision and recall, and is also known as f_1 —measure or balanced f —score. If the costs of a false positive and a false negative are asymmetric, the general f —measure can be used by adjusting the β parameter:

$$f_g - measure = \frac{(1 + \beta^2) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{4}$$



(a)



(b)

Figure 2: (a) Training images for the IIA30 dataset. (b) Cropped instances of objects from the test images.

In the object detection experiments, we have used the Pascal VOC object detection criterion [2] to determine if a given detection is a false or a true positive. In brief, to consider an object as a true positive, the bounding boxes of the ground truth and the detected instance must have a ratio of overlap equal or greater than 50% according to the following equation:

$$\frac{BB_{gt} \cap BB_{detected}}{BB_{gt} \cup BB_{detected}} \geq 0.5 \quad (5)$$

where BB_{gt} and $BB_{detected}$ stand for the ground truth and detected object bounding box respectively. For objects marked as occluded only the visible part has been annotated in the ground truth, but the SIFT object recognition method will still try to adjust the detection bounding box for the whole object based only in the visible part. Since the type of annotation is not compatible with the output of the SIFT algorithm, for the case of objects marked as occluded, we have modified the above formula in the following way:

$$\frac{BB_{gt} \cap BB_{detected}}{BB_{gt}} \geq 0.5 \quad (6)$$

As can be seen in the previous equation, it is only required that the detected object bounding box overlaps 50% of the ground truth bounding box.

4 LOWE’S SIFT OBJECT RECOGNITION METHOD

Lowe’s SIFT object recognition approach is a view-centered object detection and recognition system with some interesting characteristics for mobile robots, most significant of which is the ability to detect and recognize objects in an unsegmented image. Another interesting feature is the Best-Bin-First algorithm used for approximated fast matching, which reduces the search time by two orders of magnitude for a database of 100,000 keypoints for a 5% loss in the number of correct matches [6]. Follows a brief outline of the algorithm.

The first stage of the approach consists on matching individually the SIFT descriptors of the features detected in a test image to the ones stored in the object database using the Euclidean distance. As a way to reject false correspondences, only those query descriptors for which the best match is isolated from the second best and the rest of database descriptors are retained. In Figure 4, the matching features between a test and model images can be seen. The presence of some outliers (incorrect pairings of query and database features) can also be observed.

Once a set of matches is found, the Generalized Hough Transform is used to cluster each match of every database image depending on its particular transformation (translation, rotation and scale change). Although imprecise, this step generates a number of initial coherent hypotheses and removes a notable portion of the outliers that could potentially confuse more precise but also more sensitive methods. All clusters with at least three matches for a particular training object are accepted, and fed to the next stage: the Least Squares method, used to improve the estimation of the affine transformation between the model and the test images.

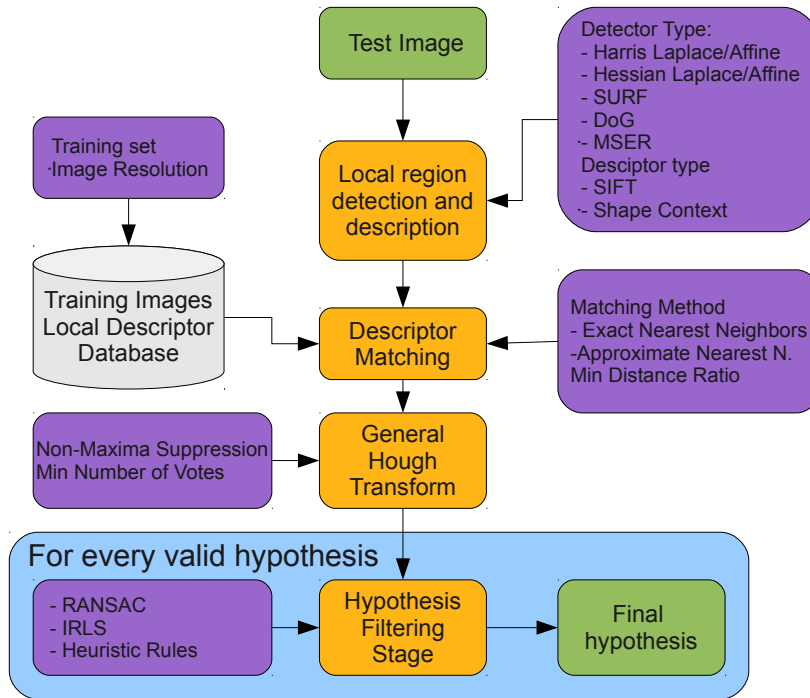


Figure 3: Diagram of the Lowe’s SIFT method with all the tests performed shown as purple boxes, Orange ones refer to steps of the method and green to input/output of the algorithm.

This approach has been modified in several ways in our experiments: The least squares method has a 0% breakdown point (i.e. any false correspondence will make the model fitting method fail or give sub-optimal results), which is a rather unfeasible restriction since we have found it is normal to still have some false matches in a given hypothesis after the Hough Transform. To alleviate this limitation, instead of the least squares, we have used the Iteratively Reweighted Least Squares (IRLS), which we have found to perform well in practice at a reasonable speed. Furthermore we have evaluated the RANdom SAmple Consensus (RANSAC), another well-known model fitting algorithm, to substitute or complement the IRLS. The RANSAC algorithm iteratively tests the support of models estimated using minimal subsets of points randomly sampled from the input data. Finally, we have incorporated some domain knowledge by defining several heuristic rules on the parameters of the estimated affine transformation to reject those clearly beyond plausibility. Namely:

- Hypotheses with object centers that are too close.
- Hypotheses that have a ratio between the x and y scales below a threshold.

Figure 3 shows an overview of our implementation of the SIFT object recognition algorithm steps.

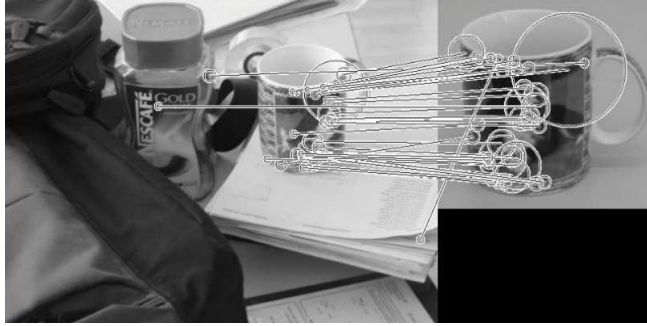


Figure 4: Matching stage in the SIFT object recognition method.

5 EXPERIMENTS

For evaluating the method, one image per category from the training image set is used. As there are several parameters to adjust in this method, we used the first sequence of the IIA30 dataset (IIA30-1) as test data to perform an extensive cross-validation over detector and descriptor type, training image size, matching method, distance ratio to the second nearest neighbor for rejecting matches, non-maxima suppression, minimum number of votes in the Hough Transform, hypothesis verification and refinement methods. The f -score is used to assign a single score to each parameter configuration, but as not all situations tolerate both error types equally, we also discuss precision and recall individually where possible.

Nonetheless, speed is probably the most relevant performance measure in our setting, and therefore we searched for the parameter combinations that perform as close as possible to real-time while retaining a good precision and recall.

What follows is a detailed discussion of the results obtained for every parameter dimension. The evaluated parameters are: Detector and descriptor type, training image size, matching method, distance ratio to the second nearest neighbor for rejecting matches, non-maxima suppression and minimum number of votes in the Hough Transform and hypothesis verification and refinement methods.

A. Feature Detectors and Descriptor: A handful of feature detectors have been proposed in the literature that find different structures in images. These feature detectors vary in number of detected regions and robustness to image variations and, a priori, it is difficult to choose one or a combination of various among the available options. We have evaluated seven feature detectors: Harris Affine, Hessian Affine, Harris Laplace, Hessian Laplace, MSER[9], SURF[1] and DoG[6].

To compute the SIFT descriptor for the feature regions detected with the first six feature detectors in the list, we have used the Oxford implementation of the descriptor². In the other hand, for the seventh feature detector, the DoG, Lowe’s original implementation of SIFT was used³. It is important to notice that both descriptor implementations give significantly different results as can be appreciated in Figure

²<http://www.robots.ox.ac.uk/~vgg/research/affine/descriptors.html>

³<http://www.cs.ubc.ca/~lowe/keypoints/>

5.

As our objective is to obtain the best results and compare the object recognition methods, we have used the best performing implementation with the DoG features and, as it was not possible to use it with other detectors, the Oxford implementation with the rest.

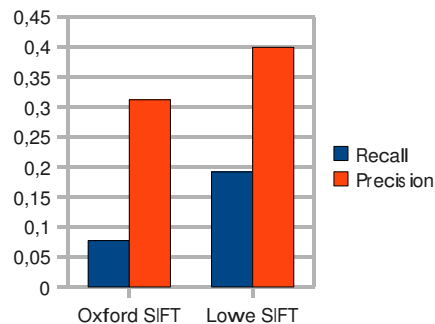


Figure 5: Results obtained with the two SIFT descriptor implementations using the DoG feature detector.

As can be seen in Figure 6.a, Hessian based detectors (Hessian Affine and Hessian Laplace) obtained the highest recall but also suffered from a low precision. Harris-based detectors obtained results on the line of the Hessian-based ones, but with a slightly lower recall and precision. Overall, the best f -measure has been obtained by the DoG detector followed by SURF. Finally, the MSER detector had a very low recall. The explanation for these results seems to be in the number of features found by each detector (see Figure 6.b). Harris and Hessian based detectors find enough features to achieve high recall rates, but without additional filtering of hypotheses, precision drops below 10%. Furthermore, the computational cost of matching the features and processing the hypotheses increases notably. On the other hand, the MSER detector finds very discriminative features but not sufficient to recognize most of the object instances. The best compromise is achieved by DoG and SURF. Additionally to feature detectors, we have considered the Shape Context descriptor [10], but results were not competitive and therefore are not displayed.

B. Training Image Size: The original SIFT object recognition method[6] is designed to be a one shot object recognition method, which makes the choice of the training image an important decision. In our experiments we have considered both: images extracted from a sequence acquired with the robot cameras –to enhance the similarity between the training and the test data– and good quality images of the objects acquired with a conventional digital camera to maximize the number of detected regions. Training images selected from a different sequence acquired with the robot did not have a competitive result, so they were discarded. For the good quality images, we have considered four different image sizes: 320x240, 640x480, 800x600 and 1024x768 pixels.

Figure 7.b shows that the time spent in the matching process increases significantly with the training image size, this was expected as the number of detected features also increases. Contrarily, the f -

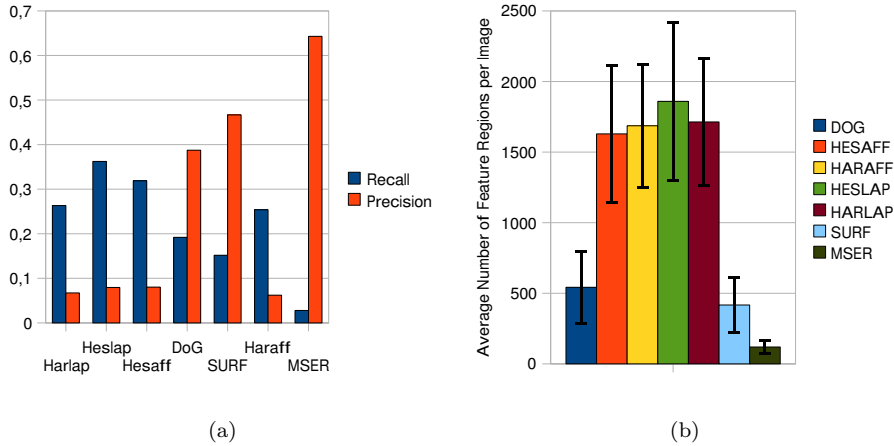


Figure 6: (a) Precision and recall depending on feature type (640x480 pixels training images). (b) Average detected feature regions per image in testing data.

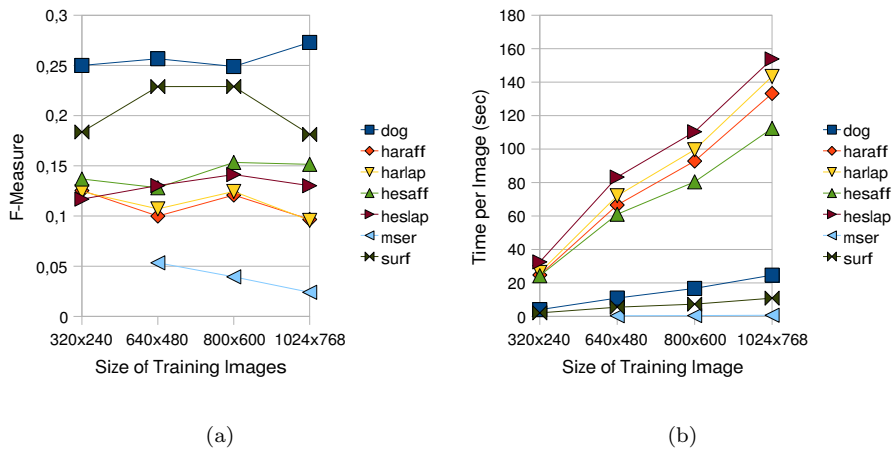


Figure 7: (a) f -Measure depending on the training image size. (b) Time per image depending on training image size with exact nearest neighbor matching.

measure did not improve noticeably with the increase of training image size. The cause of this erratic behavior is that, although more true positives were found with bigger training images, the number of false positives increased as well.

As no clear advantage was observed in using larger training images, we fixed the 640×480 size for the remaining experiments. This image size is commonly used in mobile robotics, and offers a good compromise between speed and results and, as can be seen in Figure 6.b, 320×240 was not sufficient for MSER, as it was not able to find enough features.

C. Matching Method: Various approximate nearest neighbors alternatives have been proposed in the literature [6, 11, 4] in order to accelerate the matching process between feature descriptors. As mentioned before, in the original article of the SIFT object recognition algorithm a K-D tree was used with the Best-Bin-First algorithm. Later [11] proposed an automatic approximate nearest neighbor

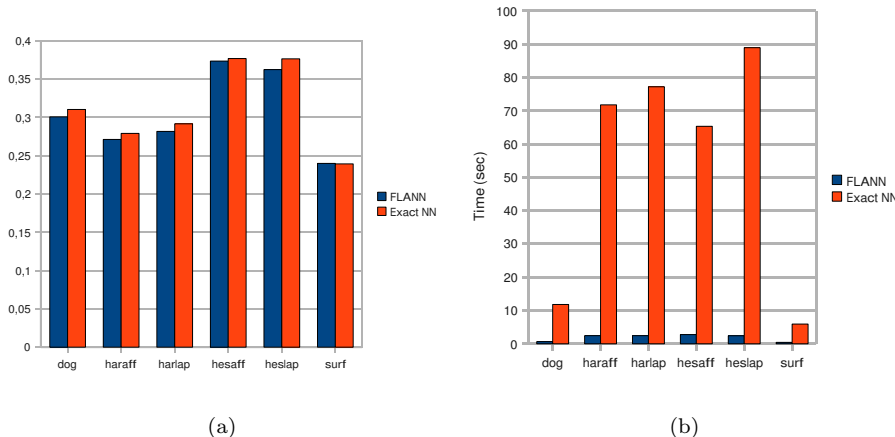


Figure 8: (a) f —Measure depending on the matching method. (b) Time per image depending on matching method.

algorithm selection method, coined FLANN, that decides between hierarchical k-means trees with a priority search order and multiple randomized K-D trees depending on the characteristics of the database to be indexed. As can be seen in Figure 8, the FLANN library drastically improves the time per image without significantly affecting the performance.

D. Distance Ratio: The distance ratio between the first and the second nearest neighbor required to accept a match is a critical choice, as it will directly influence the amount of false positive hypotheses generated (and consequently processing time) if too permissive, and the recall if too restrictive. In the original SIFT object recognition approach, the distance ratio between the first and the second nearest neighbor was required to be inferior to 0.8 in order to accept a match. However, as can be seen in Figure 9.a we found that different feature types have different optimal values for this threshold: for the Hessian and Harris based detectors, the best value for f —measure is 0.6, while DoG attains the best results at 0.7 and SURF at 0.8. As can be seen in Figure 9.b, time spent in the Hough Transform and IRLS stages increases rapidly as more potentially false matches are accepted. Keeping in mind that our aim is producing good enough results within tight time constraints, the choice of a restrictive distance ratio seems attractive.

E. Non-Maxima Suppression in Hough Transform: As in Lowe’s SIFT object recognition method each match votes for 16 bins in the Hough Transform, multiple neighboring bins can easily be activated for the same object, leading to false or *shadow* hypotheses that consume processing time in successive stages to end up being finally rejected or, even worse, generating false positives. To alleviate this we evaluated the effect of introducing a Non-Maxima Suppression (NMS) step to the Hough Transform (i.e. the score of all bins of the Hough Transform with more than the minimum required number of votes is compared to its 80 neighbors in the four dimensions of the Hough space, and is only accepted as an hypothesis if it is higher than all the other bins). Table 1 shows the results of three experiments

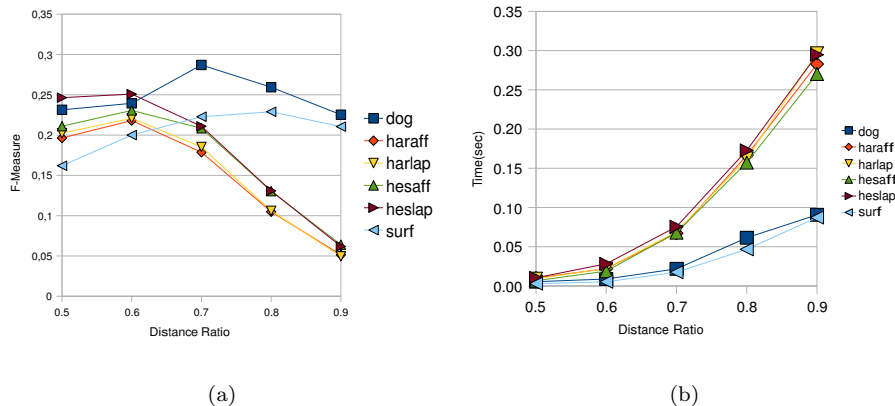


Figure 9: (a) f —Measure depending on the distance ratio. (b) Time spent in the Hough Transform and IRLS depending on the distance ratio.

	DoG			Hessian Affine			DoG2		
	HT	IRLS	Pre	HT	IRLS	Pre	HT	RIH	Pre
Std	14 ms	11 ms	0.14	27 ms	229 ms	0.02	16 ms	84 ms	0.82
NMS	56 ms	5 ms	0.40	89 ms	68 ms	0.08	73 ms	48 ms	0.87

Table 1: Three experiments with the two different Hough Transform approaches: Standard and with non-maxima suppression. The first two columns of each experiment show the time spent in the Hough clustering and in the hypotheses refinement stages respectively, and the third column shows the precision achieved (recall varies at most 0.01 between both HT approaches). In the third parameter combination, RIH stands for the combination of RANSAC, IRLS and Heuristics filtering stages.

(details can be found in the caption of the table) with different feature types and filtering methods, and both the standard (Std) and the NMS approaches for the Hough Transform.

In the base configuration with DoG features, the NMS step does not pay off in terms of computational complexity, but increases significantly the precision. However, if the number of false matches is high such as in the case of Hessian Affine with a 0.8 distance ratio, the time savings of the IRLS step are considerable. In the last experiment, additional hypothesis filtering steps are added in order to raise the precision of the standard approach to a value similar to that of the NMS. However, this extra steps increase the time to a similar value also.

F. Minimum number of votes in the Hough Transform bins: Although three matches are sufficient to estimate the pose of an object up to an affine transformation, often this number of points is low enough to be a product of chance and the introduced hypotheses will have to be discarded in subsequent steps. In spite of the theoretical justification, empirically we did not find much advantage on increasing the minimum number of feature points regarding performance as can be seen in Figures 10.a and 10.c. Precision did not increase significantly and recall degraded after losing hypotheses with few support. However, regarding computational cost, increasing the minimum number of features decreased

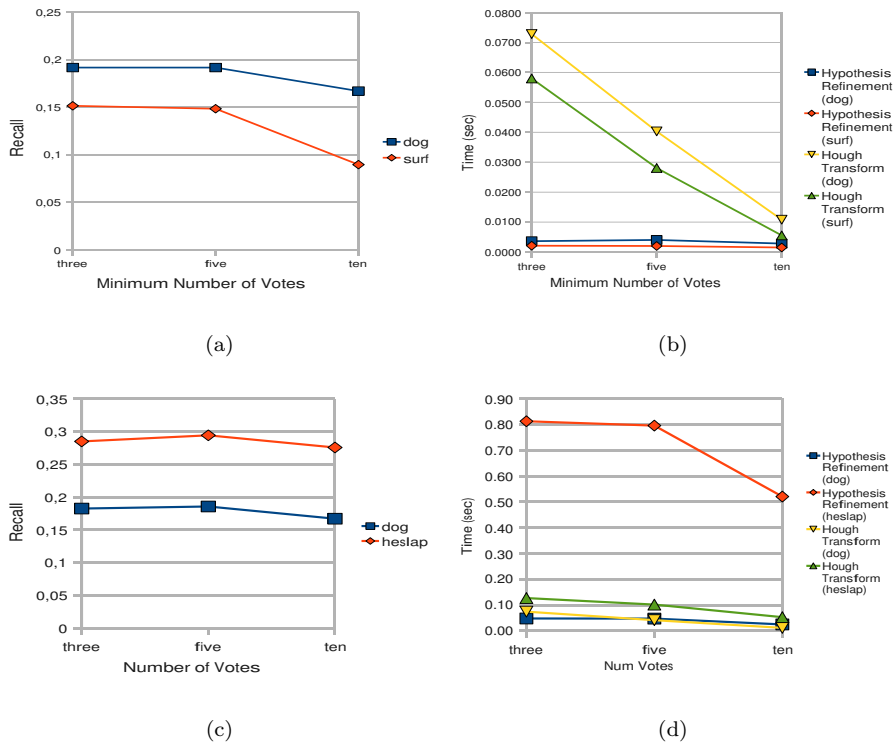


Figure 10: Performance depending on minimum number of votes in a Hough Transform bin to accept an hypothesis. (a-b) Shows results for the DoG and SURF detectors with only IRLS hypothesis filtering stage. (c-d) Shows results for DoG and Hessian Laplace detectors with all hypothesis filtering methods proposed (IRLS, RANSAC and Heuristics).

time spent in both Hough clustering (there were less putative bins to perform non-maxima suppression) and hypotheses refinement stages (less hypotheses to verify) as shown in Figures 10.b and 10.d.

G. Hypotheses Verification and Refinement: After the clustering of the found matches in the Hough Transform bins, the candidate object hypotheses are subject to a pose estimation up to an affine transformation with an iterative least squares method. This step also reduces the number of false positives by discarding those whose support falls below the minimum number of matches specified (three by default). We evaluated the impact of introducing other robust model fitting and filtering methods to discard a higher number of false positives. Specifically we used, in addition to the Iterative Reweighted Least Squares (IRLS), the RANdom Sample Consensus (RANSAC) and a set of manually defined heuristics on the detected object bounding box to eliminate repetitions and hypotheses which described unrealistic transformations. As can be seen in Figure 11.a, the f -measure increases as more strict filtering methods are applied. The best result is obtained combining all the filtering methods with the Hessian-based feature detectors. This is not surprising as these detectors obtained the best recall but suffered from a high number of false positives. Adding better hypotheses verification methods the precision and therefore the f -measure are improved. The false positives that IRLS alone is not able to filter are mainly due to untextured or repetitively textured objects. The major drawback of these extra

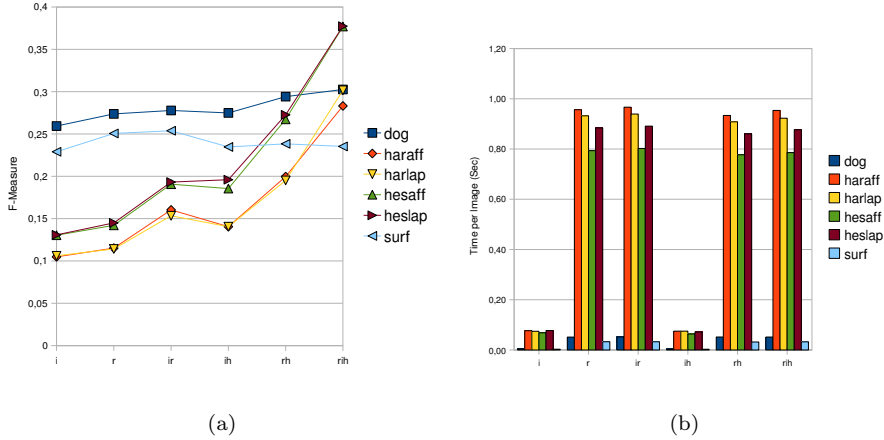


Figure 11: (a) f —Measure depending on the hypotheses filtering methods and (b) time spent in the filtering stage per image. i stands for IRLS, r for RANSAC and h for heuristics.

methods is an increase of the processing time in the hypotheses verification stages, especially in the case of RANSAC due to its Monte Carlo nature. Taking this extra computational cost into account, the best choice seems to be to use all three methods for the Harris and Hessian based detectors, and only the IRLS and heuristics for the DoG and the SURF, as the improvement introduced by the RANSAC is more modest in this case.

6 RESULTS

In this section we describe the results of the cross-validation tests conducted using sequence 1 of the IIIA30 dataset (IIIA30-1) with the different parameter combinations considered and, for a few configurations that performed best, the results obtained in the full dataset. Taking into account all combinations, the best recall obtained has been 0.45 with the Hessian Laplace detector and the less restrictive settings possible. However this configuration suffered from a really low precision, just 0.03.

The best precision score has been 0.94, and has been obtained also with the Hessian Laplace detector, with a restrictive distance ratio to accept matches: 0.5. The recall of this combination was 0.14. The same precision value but with lower recall has been obtained with the SURF and Hessian Affine detectors.

Looking at the combinations that had a best balance between recall and precision (best f —measure), the top performing ones obtained 0.39 also with the Hessian Laplace detector (0.29 recall and 0.63 precision). However, even though approximate nearest neighbors is used, each image takes around 2 seconds to be processed.

Given the objectives of this work, the most relevant way to analyze the results consists in prioritizing the time component and select the fastest parameter settings. As a runtime greater than one second is not acceptable for our purposes, the combinations that improved the f —measure with respect to faster combinations for those close to one second for image have been selected as interesting. Table 2 shows the parameters of the chosen combinations.

Method	Distance Ratio	Detector	Min. Matches	HT Method	RANSAC	Approx-NN	IRLS	Heuristics	Time (sec)	Recall	Precision	F -Measure
Config 1	0.8	SURF	5	NMS	No	Yes	Yes	No	0.37	0.15	0.51	0.23
Config 2	0.8	SURF	3	NMS	Yes	Yes	Yes	Yes	0.42	0.14	0.87	0.24
Config 3	0.8	DoG	10	NMS	No	Yes	Yes	No	0.52	0.17	0.47	0.25
Config 4	0.8	DoG	10	NMS	Yes	Yes	Yes	Yes	0.55	0.17	0.9	0.28
Config 5	0.8	DoG	5	NMS	Yes	Yes	Yes	Yes	0.60	0.19	0.87	0.31
Config 6	0.8	HesLap	10	NMS	Yes	Yes	Yes	Yes	2.03	0.28	0.64	0.39

Table 2: Detailed configuration parameters and results for the six representative configurations in increasing time order. They have been chosen for providing the best results in a sufficiently short time.

6.1 Evaluation of Selected Configurations

This section presents the results obtained applying the parameter combinations previously selected to all the sequences in the dataset.

In general all possible combinations of parameters performed better in well textured and flat objects, like the books or posters. For example the *Hartley book* or the *calendar* had an average recall across the six configurations (see Table 2 for the configuration parameters) of 0.78 and 0.54 respectively. This is not surprising as the SIFT descriptor assumes local planarity, and depth discontinuities can severely degrade descriptor similarity. On average, textured objects achieved a recall of 0.53 and a precision 0.79 across all sequences. Objects only defined by shape and color were in general harder or even impossible to detect, as can be seen in Table 3. Recall for this type of objects was only 0.05 on average. Configuration 6, that used the Hessian Laplace detector, exhibited a notably better performance for some objects of this type thanks to its higher number of detected regions. For example the *chair* obtained a recall of 0.54, or the *rack* that obtained a 0.77 recall using this feature detector. Finally, and somewhat surprisingly, objects with a repetitive texture such as the *landmark cubes* (see Figure 2) had a quite good recall of 0.46 on average; the result becomes even better if we take into consideration that besides the self-similarity, all three *landmark cubes* were also similar to one another.

Regarding the image quality parameters (see Table 4), all combinations behaved in a similar manner: the best recall, as expected, was obtained by images not affected by blur, occlusions or strong illumination changes. From the different disturbances, what was tolerated best was occlusion, followed by blur and then by illumination. Combinations of problems also had a demolishing effect in the method performance as seen in the last three rows of the table, being the worst case the combination of *blur* and *illumination* that had 0 recall. Object instance size (for objects with a bounding box defining an area bigger than 5000 pixels) did not seem to have such an impact in performance as image quality has. The performance with objects of smaller area has not yet been rigorously analyzed and is left for future work. As can be seen in the results, RANSAC and the heuristics significantly improved precision without affecting recall.

Finally, we have validated the detection accuracy by the ratio of overlap between the ground truth

Object	Config 1		Config 2		Config 3		Config 4		Config 5		Config 6		Voc Tree		Cascade	
	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre
Grey battery	0	0	0	0	0	0	0	0	0	0	0	0	0.36	0.01	0.36	0.24
Red battery	0	0	0	0	0.02	0.05	0	0	0	0	0	0	0.17	0.01	0.37	0.82
Bicycle	0.54	0.52	0.52	1.00	0.33	0.52	0.36	0.89	0.38	0.90	0.33	0.62	0.49	0.01	0	0
Ponce book	0.67	0.75	0.69	0.93	0.79	0.87	0.78	0.94	0.83	0.91	0.72	0.84	0.41	0.01	0.81	0.88
Hartley book	0.58	0.93	0.58	0.93	0.86	0.77	0.88	0.88	0.95	0.85	0.81	0.73	0.21	0	0.66	0.94
Calendar	0.44	0.65	0.35	0.86	0.56	0.66	0.56	0.79	0.56	0.79	0.79	0.71	0.12	0	0.33	0.08
Chair 1	0.03	0.08	0.02	0.33	0	0	0	0	0.01	1.00	0.54	1.00	0.78	0.06	0	0
Chair 2	0	0	0	0	0	0	0	0	0	0	0	0	0.11	0.03	0	0
Chair 3	0	0	0	0	0.01	0.25	0	0	0	0	0.05	0.50	0.02	0.05	0	0
Charger	0.03	0.20	0.03	0.50	0	0	0	0	0	0	0.18	0.14	0	0	0.12	0.08
Cube 1	0.11	0.05	0.18	0.50	0.11	0.08	0.07	0.40	0.18	0.50	0.32	0.28	0.43	0.01	0.22	0.43
Cube 2	0.62	0.28	0.67	0.67	0.71	0.11	0.76	0.59	0.76	0.55	0.52	0.38	0.17	0	0.23	0.11
Cube 3	0.53	0.22	0.31	0.50	0.50	0.25	0.59	1.00	0.66	1.00	0.66	0.45	0.09	0	0.28	0.53
Extingisher	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Monitor 1	0	0	0	0	0.01	0.05	0.01	1.00	0.04	0.75	0.15	0.63	0.15	0.02	0	0
Monitor 2	0	0	0	0	0	0	0	0	0	0	0	0	0.57	0.08	0.23	0.57
Monitor 3	0	0	0	0	0	0	0	0	0	0	0.02	0.33	0.71	0.09	0.04	0.13
Orbit box	0	0	0	0	0	0	0	0	0	0	0	0	0.14	0	0.15	0.03
Dentifrice	0	0	0	0	0	0	0	0	0	0	0	0	0.13	0	0	0
Poster CMPI	0.18	0.44	0.26	1.00	0.31	0.63	0.41	1.00	0.46	0.95	0.23	0.82	0.26	0.02	0.11	0.34
Phone	0	0	0	0	0	0	0	0	0	0	0	0	0.04	0	0.05	0.09
Poster Mys-trands	0.20	0.56	0.20	0.71	0.40	0.43	0.36	0.75	0.44	0.65	0.36	0.60	0.24	0.03	0	0
Poster spices	0.38	0.77	0.42	0.94	0.54	0.79	0.53	0.87	0.58	0.87	0.56	0.92	0.46	0.03	0.04	0.38
Rack	0.26	0.59	0.26	1.00	0.10	0.80	0.10	1.00	0.23	1.00	0.77	0.79	0.58	0.06	0	0
Red cup	0	0	0	0	0	0	0	0	0	0	0.22	0.29	0	0	0.89	0.89
Stapler	0	0	0	0	0	0	0	0	0	0	0.03	0.33	0.24	0.01	0.24	0.21
Umbrella	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Window	0.10	0.53	0.04	0.90	0.08	0.28	0.02	0.67	0.02	0.71	0.27	0.42	1.00	0.07	0.03	0.40
Wine bottle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.10	0.06

Table 3: Object-wise recall and precision for all combinations. The best results for the Vocabulary Tree method and the Boosted Cascade of Weak Classifiers are included for comparison.

Object	Config 1	Config 2	Config 3	Config 4	Config 5	Config 6
Normal	0.26	0.25	0.26	0.28	0.3	0.33
Blur	0.1	0.1	0.16	0.15	0.18	0.25
Occluded	0.16	0.14	0.14	0.12	0.14	0.34
Illumination	0	0	0.06	0.06	0.06	0.06
Blur+Occl	0.06	0.04	0.08	0.06	0.09	0.14
Occl+Illum	0.08	0.08	0.08	0.08	0.08	0.06
Blur+Illum	0	0	0	0	0	0

Table 4: Recall depending on image characteristics. *Normal* stands for object instances with good image quality and *blur* for blurred images due to motion, *illumination* indicates that the object instance is in a highlight or shadow and therefore has low contrast. Finally the last three rows indicate that the object instance suffers from two different problems at the same time.

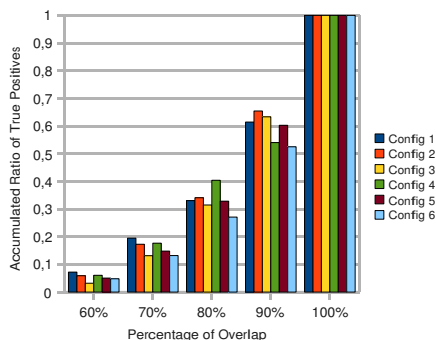


Figure 12: Accumulated frequencies for ratio of overlap between the ground truth bounding box and the detected bounding box for correctly found objects (true positives). An object is considered correctly detected if the ratio of overlap between the bounding boxes computed with equation 5 is 50% or more.

bounding box and the detected object instance as calculated in Equation 5. As can be seen in Figure 12, on average 70% of true positives have a ratio of overlap greater than to 80%, regardless of the parameter combination. Furthermore, we found no appreciable advantage on detection accuracy for any object type or viewing conditions.

As a means to provide a context to the results obtained with the six selected configurations (i.e. how good are they with respect to what can be obtained without taking into account the execution time), we compare them to the best overall recall and precision values obtained with the SIFT object recognition method. Table 5 displays the averaged precision and recall values of the four configurations that obtained the overall best recall and the four that obtained the overall best precision, as well as the six selected configurations. As can be seen in the table, the attained recall in the selected configurations was 20% lower than the maximum possible, independently of the type of objects. Precision is more affected by the amount of texture, and differences with respect to the top performing configurations ranged from 17% to 38%.

The step of the algorithm that takes most of the processing time is the descriptor matching, as it has a complexity of $O(N \cdot M \cdot D)$ comparisons, where N is the number of features in the new test image,

	Best Recall		Best Precision		Selected Config.	
	mean	std	mean	std	mean	std
Repetitively textured objects						
Recall	0.65	0.09	0.16	0.01	0.46	0.05
Precision	0.02	0.01	0.75	0.15	0.43	0.24
Textured objects						
Recall	0.70	0.03	0.28	0.03	0.53	0.10
Precision	0.05	0.02	0.96	0.02	0.79	0.09
Not textured objects						
Recall	0.21	0.01	0.01	0.01	0.05	0.04
Precision	0.03	0.01	0.62	0.32	0.24	0.21

Table 5: Average recall and precision of the configurations that were selected for having the best values according to these two measures in the last section. Also average results among the six selected configurations are shown for comparison. Standard deviation is provided to illustrate scatter between the selected configurations. Objects are grouped in the three “level of texture” categories in the following way: the three cubes form the repetitively textured category, the two books, the calendar and the three posters form the textured category, and the rest fall into the non textured category.

M is the number of features in the training dataset and D is the dimension of the descriptor vector. Approximate matching strategies, such as the one by [11] used in this work, make the SIFT object recognition method suitable for robotic application by largely reducing its computational cost. In our experiments we experienced only a 0.01 loss in the f —measure for an up to 35 times speed-up.

6.2 COMPARISON TO OTHER METHODS

In [13] the authors evaluated the performance of the Vocabulary Tree object classification method [12] on the IIA30 object detection dataset. Their best results (tree with branch factor 9 and depth 4, Hessian Affine features and intensity segmentation) are included in Table 3, where can be compared to the ones obtained in this work. As can be seen, the Vocabulary Tree method had a much better recall for untextured objects, being able to recover on average 0.29 of them, but with a precision of only 0.03.

Finally, in [] the popular boosted cascade of weak classifiers object detector from Viola and Jones [14] is evaluated with the IIA30 dataset. Again, for comparison purposes, their best results are included in Table 3. With this method objects that were completely invisible for the SIFT object recognition method (like the batteries) obtained a reasonable recall and precision. However, the overall performance of the Viola and Jones detector is slightly lower than the method used in this work, and it required a much higher amount of training data.

7 CONCLUSIONS AND FUTURE WORK

In this work we have evaluated the well known SIFT object recognition system in a very challenging dataset. Experiments show that using the SIFT object recognition approach with the proposed modifications, it is possible to precisely detect, considering all image degradations, around 60% of well textured

object instances with a precision close to 0.9 in our challenging dataset. The framerate achieved has been approximately one frame per second in 640×480 pixel images, with our not fully optimized implementation. Even feature detectors known to sacrifice repeatability (probability of finding the same feature region in slightly different viewing conditions) for speed such as the SURF obtain reasonable results. Performance degrades for objects with repetitive textures or no texture at all. Regarding image disturbances, the approach resisted occlusions well, since the SIFT object recognition method is able to estimate a reliable transformation (as long as a minimum number of correct matches is found, three by default), but not so well blur due to motion or deficient illumination.

Compared to the two object recognition methods, the Vocabulary Tree [12] and the Boosted Cascade of Weak Classifiers [14], the method evaluated in this work is able to attain a higher precision, but has a significantly lower performance with non-textured objects.

In terms of runtime, with our current non-optimal implementation and the approximate nearest neighbor matching schema of [11], the method is able to run in less than one second in a single core, and an implementation tailored to performance should be able to achieve faster rates. A drawback of the SIFT object recognition method is that it is not robust to viewpoint change. It would be interesting to evaluate how enhancing the method with 3D view clustering as described in [5] affects the results, as it should introduce robustness to this type of transformation.

The main limitation of the SIFT object recognition method is that only the first nearest neighbor of each test image feature is considered in the subsequent stages. This restriction makes the SIFT method very fast, but at the same time makes it unable to detect objects with repetitive textures. Other approaches with direct matching, like that of [3], overcome this by allowing every feature to vote for all feasible object hypotheses given the feature position and orientation. Evaluating this type of methods, or modifying the SIFT to accept several hypotheses for each test image feature, would be an interesting line of continuation of this work.

Finally, the heuristics proposed in this work to improve the SIFT object recognition method have been manually designed. Nevertheless, it would be much better if the system itself was able to learn and generalize which bounding boxes parameters constitute valid hypotheses, for instance using Reinforcement Learning.

REFERENCES

- [1] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. *LECTURE NOTES IN COMPUTER SCIENCE*, 3951:404, 2006.
- [2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>, 2007.

- [3] Bastian Leibe, Ales Leonardis, and Bernt Schiele. Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, 77(1-3):259–289, 2008.
- [4] Vincent Lepetit, Julien Pilet, and Pascal Fua. Point matching as a classification problem for fast and robust object pose estimation. In *CVPR (2)*, pages 244–250, 2004. URL <http://dblp.uni-trier.de/db/conf/cvpr/cvpr2004-2.html#LepetitPF04>.
- [5] David G. Lowe. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, page 1150, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0164-8.
- [6] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. ISSN 0920-5691.
- [7] Al Mansur and Yoshinori Kuno. Specific and class object recognition for service robots through autonomous and interactive methods. *IEICE - Trans. Inf. Syst.*, E91-D(6):1793–1803, 2008. ISSN 0916-8532. doi: <http://dx.doi.org/10.1093/ietisy/e91-d.6.1793>.
- [8] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.
- [9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.
- [10] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.
- [11] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISAPP'09)*, October 2009.
- [12] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. *Conf. Computer Vision and Pattern Recognition*, 2:2161–2168, 2006.
- [13] A. Ramisa, D. Aldavert, S. Vasudevan, R. Toledo, and R. Lopez de Mantaras. Evaluation of three vision based object perception methods for a mobile robot. *arXiv: 2011arXiv1102.0454R*, February 2011.
- [14] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 1, page 511, 2001.