Evaluation of the SIFT Object Recognition Method in Mobile Robots

Arnau RAMISA [a,1], Shrihari VASUDEVAN [b], David ALDAVERT [c], Ricardo TOLEDO [c], and Ramon LOPEZ DE MANTARAS [b],

[a] *IIIA-CSIC, Spain*
[b] *ACFR, Australia*
[c] *CVC (UAB) Spain*

Abstract. General object recognition in mobile robots is of primary importance in order to enhance the representation of the environment that robots will use for their reasoning processes. Therefore, we contribute reduce this gap by evaluating the SIFT Object Recognition method in a challenging dataset, focusing on issues relevant to mobile robotics. Resistance of the method to the robotics working conditions was found, but it was limited mainly to well-textured objects.

Keywords. Computer Vision, Object Recognition, Mobile Robots

Introduction

As can be seen in recently published literature [6], currently there is a big push towards semantics and higher level cognitive capabilities in robotics research. One central requirement towards these capabilities is being able to identify higher level features like objects, doors etc. in perceptual data. Although impressive results are obtained by modern object recognition and classification methods, still a lightweight object perception method which allows them to interact with the environment in a human cognitive level is lacking. Furthermore, the system should be able to learn new objects in an easy, and preferably automatic, way.

Recently methods have been proposed that are quite successful in particular instances of the general object classification problem, such as detecting frontal faces or cars , or in datasets that concentrate on a particular issue (e.g. classification in the scale-normalized and segmented Caltech-101 dataset). However in more challenging datasets, like the general object detection competition of the Pascal VOC 2007, the methods presented achieved a lower average precision[2].

This low performance is not surprising, since object recognition in real scenes is one of the most challenging problems in computer vision [5]. The visual appearance of objects can change enormously due to viewpoint variation, occlusions, illumination changes or sensor noise. Furthermore, objects are not presented alone to the vision system, but they are immersed in an environment with other elements, which clutter the

[1]Corresponding Author: Arnau Ramisa, IIIA-CSIC, Campus de la UAB, E-08193 Bellaterra, Catalonia (Spain); E-mail: aramisa@iiia.csic.es

[2]http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2007/

scene and make recognition more complicated. In a mobile robotics scenario a new challenge is added to the list: computational complexity. In a dynamic world, information about the objects in the scene can become obsolete even before it is ready to be used if the recognition algorithm is not fast enough.

In order to help reduce a bit this gap, this paper contributes an evaluation of the SIFT object recognition method from [3] on a realistic mobile robotics scenario, that includes many of the typical problems that will be encountered when roboticists try to use this method on practical matters. Additionally, and more importantly, several modifications and improvements of the original method are proposed in order to adapt it to the domain of mobile robotics.

1. Lowe 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 at the same time in an unsegmented image.

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. False matches are rejected if the distance of the first nearest neighbor is not distinctive enough when compared with that of the second. 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.

This approach has been modified in several ways in our experiments. The breakdown point (i.e. ratio of outliers in the input data were the model fitting method fails) for the least squares method is at 0% of outliers, 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, instead of the least squares, we have used the Iteratively Reweighted Least Squares (IRLS). Furthermore we have added the RANdom SAmple Consensus (RANSAC), another well-known model fitting algorithm that iteratively tests the support of models estimated using minimal subsets of points randomly sampled from the input data. Finally, we have manually defined a set of heuristic rules on the parameters of the estimated affine transformation to reject those clearly beyond plausibility.

In order to evaluate the methods in a realistic mobile robots setting, we have created the IIIA30 database[3], that consists of three sequences of different length acquired by our mobile robot while navigating in a laboratory type environment. Image size is 640x480 pixels. The environment has not been modified in any way and the object instances in the test images are affected by lightning changes, blur caused by the motion of

[3] http://www.iiia.csic.es/~aramisa/iiia30.html

the robot, occlusion and large scale and viewpoint changes. 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. Training images have been acquired with a standard digital camera and reduced to the same resolution as testing. Figure 1.f shows some examples from the training and testing sets. 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...).

2. Parameter Tuning

In order to find the best set of parameters for the SIFT object recognition system, a series of experiments were done. Each experiment aims to evaluate a particular aspect of the method. Nonetheless, speed is probably the most relevant performance measure in our setting, and therefore we search for the parameter combinations that perform as close as possible to real-time while retaining a good precision and recall.

To consider an object as a true positive, the intersection of the ground truth and detected bounding boxes divided by its union must be greater or equal to 0.5. For objects marked as occluded, the detected bounding box is only required to overlap at least 50% of the ground truth bounding box. What follows is a detailed discussion of the results obtained for every parameter dimension.

Feature Detectors and Descriptor: Seven feature detectors are evaluated: Harris Affine, Hessian Affine, Harris Laplace, Hessian Laplace, MSER. We have used the Oxford SIFT implementation[4] To compute the descriptor of feature regions detected with the first six feature detectors, while the descriptors for the DoG regions have been computed with Lowe's original implementation of SIFT that comes with the DoG detector. It is important to understand that both implementations give significantly different results as can be appreciated in Figure 1.d. As can be seen in Figure 1.c, 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 1.e). 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.

Matching Method: Various approximate nearest neighbors alternatives have been proposed in the literature [3, 4, 1] 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 [4] proposed an improved approach, coined FLANN, which we compare with exact nearest

[4] http://www.robots.ox.ac.uk/~vgg/research/affine/descriptors.html

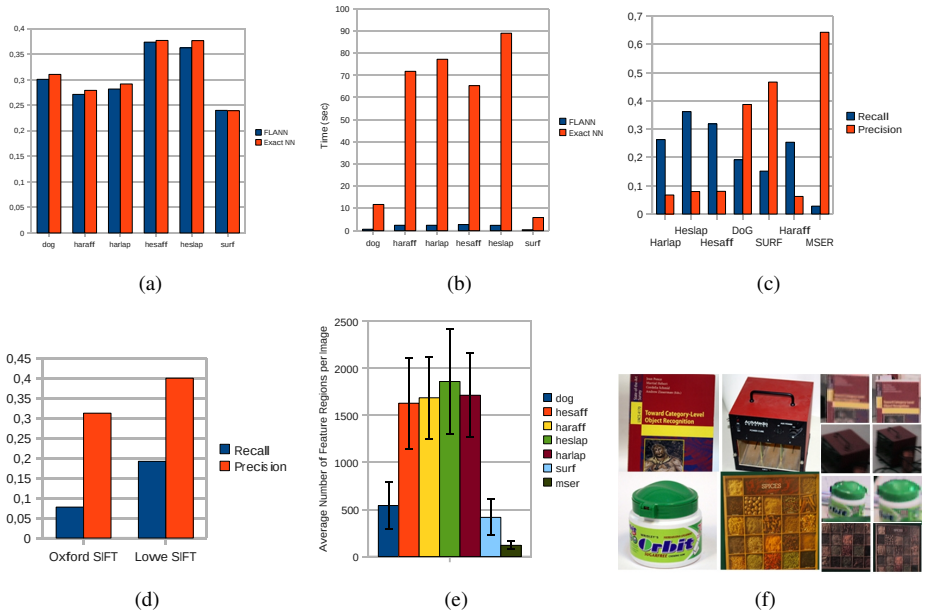
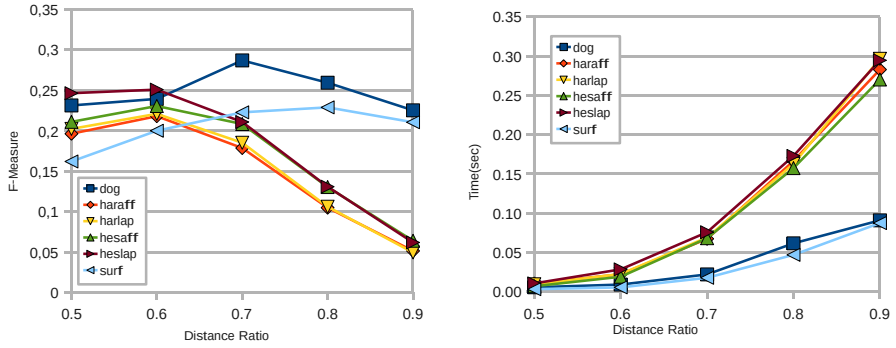


Figure 1. (a) F-Measure depending on the training image size. (b) Time per image depending on training image size with exact nearest neighbor matching. (c) Precision and recall depending on feature type (640x480 pixels training images). (d) Results obtained with the two SIFT descriptor implementations using the DoG feature detector. (e) Average detected feature regions per image in testing data. (f) Some examples from the IIIA30 dataset.

neighbor matching. As can be seen in Figures 1.a and 1.b, the approximate nearest neighbors method drastically improves the time per image without affecting significantly the performance.

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 2.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 2.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.

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. Table 1 shows the results of three different experiments with both the standard and the NMS approaches. In the standard configuration



(a) (b)

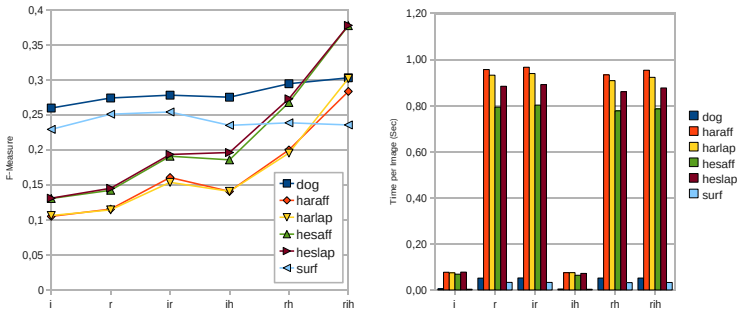
Figure 2. (a) F-Measure depending on the distance ratio. (b) Time spent in the Hough Transform and IRLS depending on the distance ratio.

	DoG			Hessian Affine			DoG2		
	HT	IRLS	Pre	HT	IRLS	Prec	HT	RIH	Pre
Standard	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.

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.

Hypotheses Verification and Refinement: We evaluated the impact of introducing other robust model fitting and filtering methods, besides IRLS, to discard a higher number of false positives. Specifically we used 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 3.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 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 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 con-



(a) (b)

Figure 3. (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.

Method	DR	Det	MM	HT	R	IRLS	H	Time	Rec.	Prec.	F-M
Config 1	0.8	SURF	5	NMS	No	Yes	No	0.37s	0.15	0.51	0.23
Config 2	0.8	SURF	3	NMS	Yes	Yes	Yes	0.42s	0.14	0.87	0.24
Config 3	0.8	DoG	10	NMS	No	Yes	No	0.52s	0.17	0.47	0.25
Config 4	0.8	DoG	10	NMS	Yes	Yes	Yes	0.55s	0.17	0.9	0.28
Config 5	0.8	DoG	5	NMS	Yes	Yes	Yes	0.60s	0.19	0.87	0.31
Config 6	0.8	HesLap	10	NMS	Yes	Yes	Yes	2.03s	0.28	0.64	0.39

Table 2. Detailed configuration parameters for the six chosen configurations in increasing time order. *DR* stands for Distance Ratio, *Det* for Detector, *MM* for Minimum number of Matches to accept an hypothesis, *R* for RANSAC and *H* for Heuristics. All combinations used Aproximated Nearest Neighbors. Performance of the configurations is also shown. *Rec* stands for recall, *Prec* for precision and *F-M* for F-Measure

figuration 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 configurations that had a best balance between recall and precision (best f-measure), the top performing obtained 0.4 and 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.

Finally, we have sorted the configurations in reverse time complexity order, and those combinations that improved the f-measure with respect to faster combinations for those below 1 second for image have been selected as interesting. Table 2 shows the parameters of the chosen combinations and also performance results.

3. 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 combintions 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 disconti-

Object	Config 1		Config 2		Config 3		Config 4		Config 5		Config 6	
	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre	Rec	Pre
Bicicle	0.54	0.52	0.52	1.00	0.33	0.52	0.36	0.89	0.38	0.90	0.33	0.62
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
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
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
Chair 1	0.03	0.08	0.02	0.33	0	0	0	0	0.01	1.00	0.54	1.00
Charger	0.03	0.20	0.03	0.50	0	0	0	0	0	0	0.18	0.14
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
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
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
Monitor 1	0	0	0	0	0.01	0.05	0.01	1.00	0.04	0.75	0.15	0.63
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
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
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
Red cup	0	0	0	0	0	0	0	0	0	0	0.22	0.29
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

Table 3. Recall and precision of some selected objects. Complete results available online at: `http://www.iiia.csic.es/~aramisa/iiia30.html`

nuities 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, for example the *chair*, obtained a recall of 0.54, or the *rack* that obtained a 0.77 recall. Finally, and somewhat surprisingly, objects with a repetitive texture such as the *landmark cubes* had a quite good recall of 0.46 on average. Furthermore, 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, 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,being the worst case the combination of *blur* and *illumination* that had 0% recall. As predicted in Section 2, RANSAC and the heuristics significantly improved precision without affecting recall. Finally, we have evaluated the exactitude in the detection of the objects by the ratio of overlap between the ground truth bounding box and the detected object instance.On average 70% of true positives have a ratio of overlap superior to 80% regardless of the parameter combination. In order to put into context the results obtained with the selected configurations, we have also evaluated the four configurations that obtained the overall best recall and the four that obtained the overall best precision. The attained recall in the selected configurations was 20% lower than the maximum obtained, 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%.

4. Conclusions

In this work we have performed a careful evaluation of the SIFT object recognition method in a mobile robotics setting. Also we have proposed some modification to the original schema to improve the results. 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. Even 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 well occlusions, since the SIFT object recognition method is able to estimate a reliable transformation as long as the visible part of the object contains enough texture (and a minimum number of correct matches, three by default) but not so well blur due to motion or deficient illumination.

As can be seen in Table 2, all but one of the selected methods had a running time lower to one second, which makes them suitable for robotic applications. 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, 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 [4] used in this work, are able to reduce this cost. In our experiments we experienced only a 0.01 loss in the f-measure for an up to 35 times speed-up. Furthermore, an implementation tailored to performance should be able to achieve even 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 [2] affects the results, as it should introduce robustness to this type of transformation.

Acknowledgements

This work was supported by the FI grant from the Generalitat de Catalunya, the European Social Fund, and the MID-CBR project grant TIN2006-15140-C03-01 and FEDER funds and the grant 2005-SGR-00093 and the MIPRCV Consolider Imagennio 2010.

References

[1] V. Lepetit, J. Pilet, and P. Fua. Point matching as a classification problem for fast and robust object pose estimation. In *CVPR (2)*, pages 244–250, 2004.

[2] D. Lowe. Local feature view clustering for 3d object recognition. In *CVPR (1)*, pages 682–688. IEEE Computer Society, 2001.

[3] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[4] 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.

[5] N. Pinto, D. D. Cox, and J. J. Dicarlo. Why is real-world visual object recognition hard? *PLoS Computational Biology*, 4(1):151–156, January 2008.

[6] S. Vasudevan, S. Gächter, V. Nguyen, and R. Siegwart. Cognitive maps for mobile robots - an object based approach. *Robotics and Autonomous Systems*, 55(5):359–371, 2007.