An Experimental Comparison of Image Feature Detectors and Descriptors applied to Grid Map Matching

J.L. Blanco, J. Gonzalez, J.A. Fernández-Madrigal *

Technical Report

April 28, 2010

Abstract

Applying computer vision feature detectors and descriptors to occupancy grids has important practical applications for the problem of grid map matching in mobile robot localization and mapping, although this approach has received little attention by the community. This review presents a thorough performance evaluation for several combinations of detectors (Harris, KLT, SIFT and SURF) and descriptors (SIFT, SURF and circular patches) using maps obtained from real datasets. It is shown how a combination of the Harris or KLT detector with circular patch descriptors provides the best results in both computation time and classification success ratio.

Keywords – Occupancy grid maps, Image registration.

1 Introduction

Occupancy grid maps are one of the most widespread world representations employed in the mobile robot community [10, 11]. Most mainstream approaches to Simultaneous Localization and Mapping (SLAM) that rely on grid maps need to perform certain operations on these grids, namely: update them from sensory data and estimate the sensor observation likelihood [21]. Matching pairs of grid maps is also required when dealing with hybrid metric-topological models [3, 5, 8] or multi-robot mapping [2].

Several works have focused on problems such as building grids from raw sensory data [7, 20] and the more complex task of estimating observation likelihoods ([4, 18, 21]). In contrast, grid matching has received relatively little attention

*Authors are with the Department of System Engineering and Automation, University of Málaga, Spain.

and existing approaches ([2, 6, 11]) have not exploited yet the rich field of image interest point detection and matching. This is the point of view adopted in the present work, where grid maps are interpreted as grayscale images, named *map images*.

In this article we focus on the most fundamental issue faced when registering grids by matching map features: choosing a feature detector and a descriptor. We can find reviews in the literature for the general problem of image registration [22], but there are some peculiarities in grid matching that motivate our work: (i) grids are typically built with identical cell sizes in a way that scale is not an issue between different maps, (ii) a pair of map images can be related by a rigid transformation only (that is, translation and rotation), and (iii) interest points are highly ambiguous due to the poor distinctiveness of features found in practical scenarios (e.g. in the real world, many corners look alike locally).

In the next section we describe the interest point detectors that we have employed in our comparison, then we review the feature descriptors and in Section 4 we discuss the results of the benchmark. The datasets used in our experiments, as well as all the source code, are publicly available online [1].

2 Extraction of features

In this section we review some well-known image feature detectors and motivate the need for pre-processing the map images in order to improve the detection process.

2.1 Interest-point detectors

In a typical indoor occupancy grid map we can easily identify natural features produced by scene elements, like corners, columns or, in general, any sharp edge. They also appear in some outdoor maps originated by vertical poles, building corners, vehicle edges, etc. These natural landmarks are suitable for matching maps of the same areas since they naturally occur in the environment and they are typically static.

All those interest points can be detected by interpreting the grid map as a grayscale image, the map image, and applying existing key-point detectors. The most desirable property of any detector is its *repeatability*, that is, its ability to detect a given feature when it appears in different images.

We are interested in the performance of the following four methods:

- The Harris detector [12], which searches for points where the structure tensor has two large eigenvalues, revealing the existence of corners.
- The Kanade-Lucas-Tomasi (KLT) method ([16, 19]) also relies on the structure tensor. It detects salient points where one of the eigenvalues exceeds a given threshold.

[1] Refer to the website http://babel.isa.uma.es/mrpt/papers/charac-grids/

- The detection phase of the SIFT algorithm [15], which identifies scale-space extrema in pyramids of difference-of-Gaussians. This method aims at detecting *blobs* instead of corners [17].
- The detector of SURF, based on an approximation to the Hessian matrix [1].

There exists an issue in map images which affects the process of feature detection and needs to be handled appropriately. Grid mapping from laser range scans typically generates some artifacts in the maps which can be interpreted as high-frequency noise in the image (e.g. those arising from a single ray of the scans). To prevent the detection of spurious interest points in the middle of free-space, we propose to pre-process the images by applying first a Gaussian filter and then a median filter to attenuate most of the irregularities. Next we explain how we have tuned each filter for optimal detection performance.

2.2 Characterization

As already introduced in Section 1, the dataset employed in this work (available online) consists of 10 pairs of grid maps from real robot data. We must remark that the maps represent real loop-closure situations with partial overlap and small differences in the grids caused by noise and different viewpoints of the robot. Since hundreds of key points are detected in each of these grids, our overall characterization can be considered significant from a statistical point of view.

In order to evaluate the repeatability of each interest point detector we have applied it to both maps in each pair, and then counted the number of common detected features, i.e. the same feature must be detected in *both* grid maps. The correct pairings were obtained then from ground truth transformations between the pairs of maps, computed manually. To avoid a bias in our results due to the number of detected points, we have limited the number of interest points to a fixed value proportional to the extension of each grid map (a typical value of 0.015 features per square meter is appropriate for all the maps employed in our comparison).

The results are summarized in Figure 1 for each detector and for different values of W_g and W_m, the sizes of the Gaussian and the median filter, respectively. The values $W_g = 0$ and $W_m = 1$ correspond to a null filter in each case, thus the cases of applying just one of the filters (or none of them) have been also accounted for.

Observe how blob detectors (SIFT and SURF) perform well for large filter sizes (that lead to more "softened" images), whereas corner detectors (Harris and KLT) have good repeatability for slightly filtered images or even for maps not filtered at all (refer to KLT results in Figure 1). Figure 2 shows an example of the different filters required by each detector to perform optimally. The best filter configuration for each detector has been employed in the benchmark presented in Section 4, and the corresponding overall number of matches can be seen in Figure 5(f).

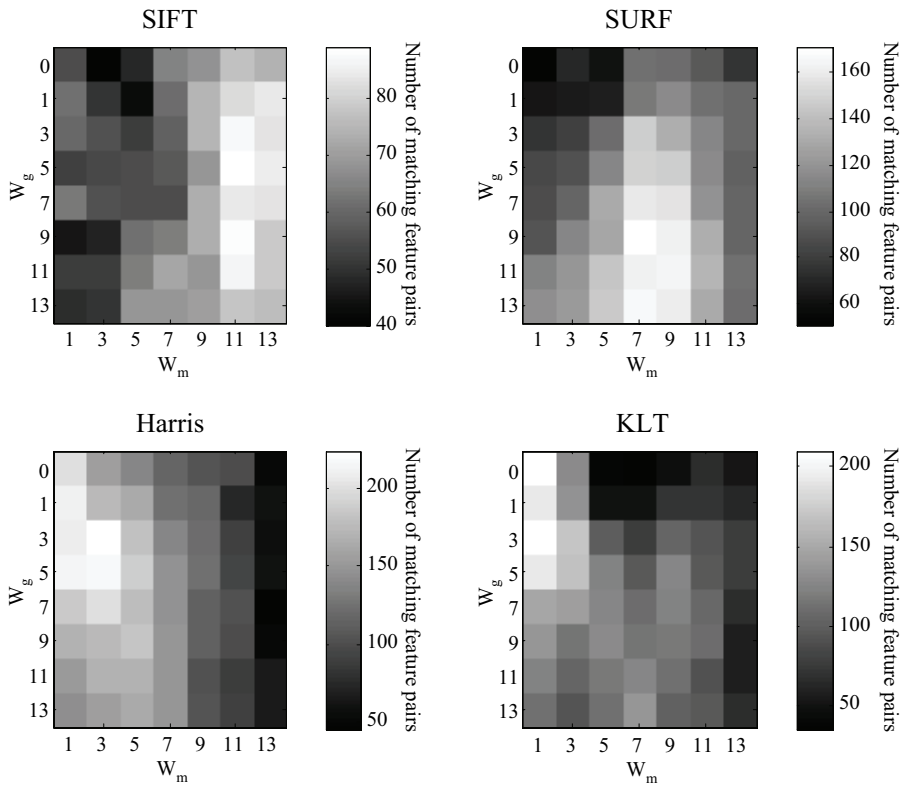

Figure 1: A measure of the repeatability for each detector and for different sizes of the Gaussian (W_g) and median (W_m) filters used to smooth the map images. Brighter colors indicate a higher number of common features detected in both maps.

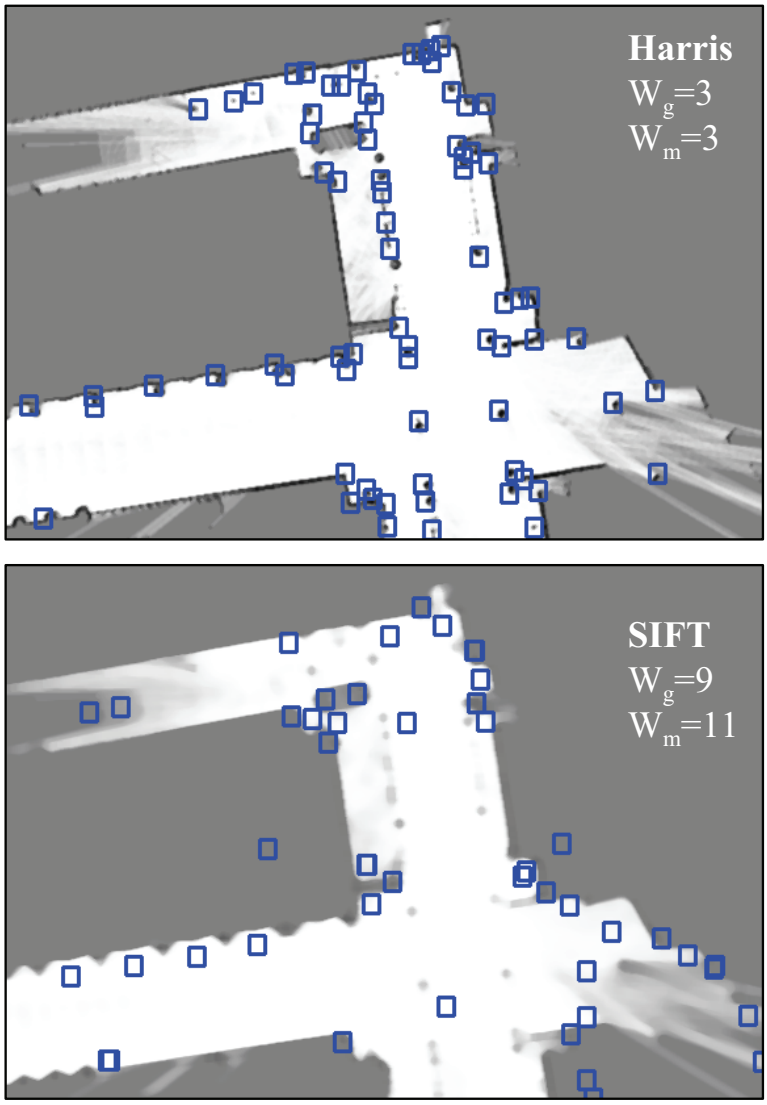

Figure 2: One of the maps from the dataset, filtered with a Gaussian and median filter of sizes W_g and W_m, respectively. Detected interest points are marked with small squares for the Harris and SIFT detectors. Notice how each method detects a different kind of features (corners or blobs), hence the different filtering requirements.

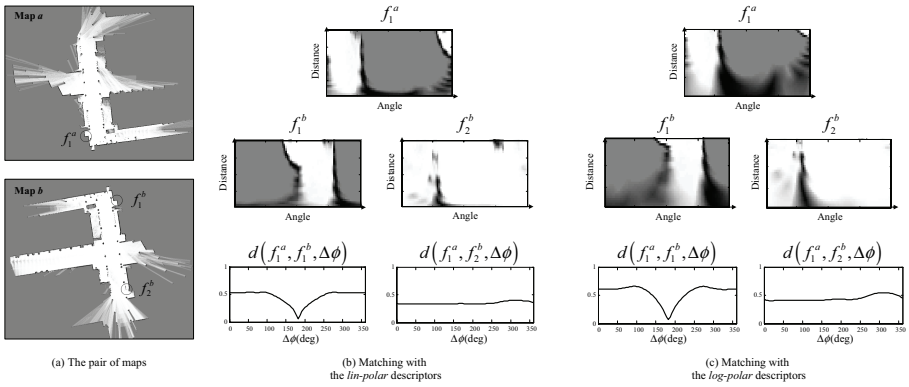

Figure 3: Example of matching features with different orientation. (a) An arbitrary reference feature f_1^a is highlighted in map a, and two potential pairings f_1^b (the real correspondence) and f_2^b are marked in map b. (b)–(c) The similarity between the feature descriptors is displayed as the distance function $d(f_i, f_j, \Delta\phi)$ for the cases of using the *lin-polar* and *log-polar* descriptors, respectively. Notice the pronounced minimum of the distance for the case of the real correspondence $f_1^a \leftrightarrow f_1^b$ close to the 180° relative rotation.

3 Descriptors

3.1 Review

Once the key-points are detected they are assigned distinctive descriptors in order to establish correspondences. We have studied the performance of the following five image descriptors [2] :

- **SIFT**: This method is based on histograms of image gradients [15], obtaining a 128-length descriptor vector.
- **SURF**: Based on the responses of Haar-wavelets as described in [1].
- **Intensity-domain spin images (Spin)**: A 2D histogram of intensities and distances [14], with the maximum radius from the interest point determined by the parameter R_{max}. The usage of distances (disregarding angles), makes this descriptor rotation invariant.
- **Linear or logarithmic circular patches**: These two descriptors have many similarities, hence we discuss them here together. Both map a circular region of radius R_{max} centered at the interest point into a 2D matrix (the descriptor) of polar coordinates. Let this matrix be denoted by

[2] OpenCV implementations have been used for all the feature detectors and descriptors mentioned in this paper, except for: (i) the SIFT method for which we rely on the implementation [13] and (ii) the *lin-polar* descriptor, coded by the authors and submitted for publication in OpenCV 2.0.

$\mathbf{f}(u, v)$, where the indices u and v stand for different values of the distance and the angle from the feature, respectively. The idea is to extract a circular patch of the neighborhood of the feature in a representation which is not invariant to rotations, but where these rotations become just shifts in the angle dimension (v), as illustrated with the examples in Figure 3(b)–(c). The only difference between the linear polar descriptor (*lin-polar* for short) and its logarithmic version (*log-polar*) is the usage of a linear or logarithmic scale in the distances.

Next we address the problem of measuring the similarity between descriptors, a requisite to evaluate their distinctiveness.

3.2 A Similarity Function Between Descriptors

Given a pair of descriptors $\mathbf{f}_i^a$ and $\mathbf{f}_j^b$ for two keypoints i and j from maps a and b, respectively, we are interested in measuring their similarity. For the SIFT, SURF and Spin descriptors the most natural measure is the Euclidean distance between the descriptor vectors. However, the cases of *lin-polar* and *log-polar* deserve more discussion since they are not directly invariant to orientation.

As illustrated in Figure 3, the descriptors of two matching features only differ by a shift in the angular dimension. Therefore, we propose to measure the distance between two descriptors $\mathbf{f}_i$ and $\mathbf{f}_j$ by their Euclidean distance, given a rotation $\Delta\phi$, that is:

$$d\left(\mathbf{f}_i, \mathbf{f}_j, \Delta\phi\right) = \left(\sum_u \sum_v \left| \mathbf{f}_i(u, v) - \mathbf{f}_j(u, v + \Delta\phi) \right|^2 \right)^{\frac{1}{2}} \tag{1}$$

where the angular polar coordinate v is taken modulo the corresponding size of the matrix.

By computing the distance in Eq. (1) to a pair of descriptors $\mathbf{f}_i^a$ and $\mathbf{f}_j^b$ we obtain a distance vector for each possible shift in orientation $\Delta\phi$. As shown in Figure 3, these distance vectors have pronounced minimums for the true orientation when two features do really match, thus we propose to measure the inter-feature distance in the cases of *lin-polar* and *log-polar* as:

$$d\left(\mathbf{f}_i, \mathbf{f}_j\right) = \min_{\Delta\phi} d\left(\mathbf{f}_i, \mathbf{f}_j, \Delta\phi\right) \tag{2}$$

For all the descriptors in our comparison we have normalized distances to the range $[0, 1]$ in order to keep homogeneity in the results presented in the next section.

4 Benchmark

After defining a similarity measure for pairs of descriptors in Section 3.2, we are interested in obtaining a set of *candidate correspondences* between the features

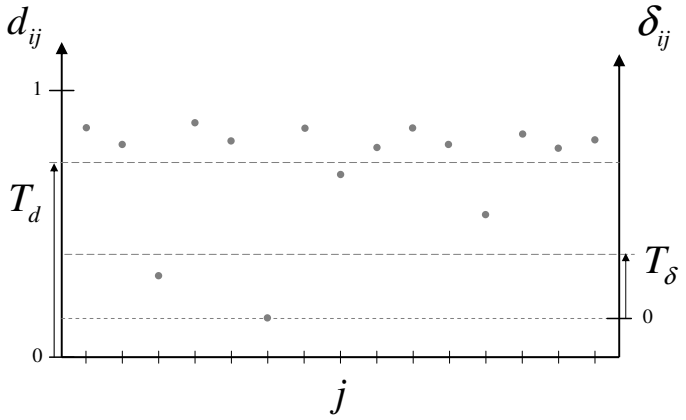

Figure 4: A schematic illustration of the distance between descriptors d_{ij} and the index δ_{ij}, which measures those distances relative to the closest one for each given feature i. Note that, by definition, the best pairing is always assigned a value $\delta_{ij} = 0$. A pairing will be accepted only if it is below both thresholds T_d (absolute) and T_δ (relative to the minimum distance).

of two maps a and b, given their descriptors $\mathbf{f}_i^a$ and $\mathbf{f}_j^b$. The goodness of all the potential correspondences must be evaluated such as only the most promising pairings (those passing a given test) are considered as candidates. It is acceptable for each feature to have multiple potential correspondences in the other map, since a subsequent robust matching step (such as RANSAC [9]) can easily manage that ambiguity.

The arguably simplest test for selecting matchings is thresholding, which in our case means to accept a potential match between $\mathbf{f}_i^a$ and $\mathbf{f}_j^b$ only if the distance d_{ij} between their descriptors is below a fixed value T_d. However, this simple scheme has some drawbacks in the context of grid matching, because distance values between actually corresponding pairs may vary in a relatively large range. Thus, any permissive threshold T_d which covers most of the good correspondences would suffer from a high rate of false positives.

Following an idea similar to Lowe's proposal in [15] we introduce a second condition for establishing candidate pairings: the associated distance d_{ij} must be not only below the threshold T_d, but also sufficiently close to the best matching of $\mathbf{f}_i^a$ in map b, that is, the minimum of d_{ij} for all values of j (see Figure 4). This restriction is characterized by a second threshold T_δ which states the maximum acceptable distance δ between a potential pairing and the best one, that is, $\delta_{ij} = d_{ij} - \min_j d_{ij}$. Notice that for the extreme case $T_\delta = 0$ each feature will be associated to only one in the other map: the one with the closest descriptor. Both measures d_{ij} and δ_{ij} are illustrated with an example in Figure 4 for clarity.

A benchmark has been carried out to obtain the optimal values for the thresholds T_d and T_δ from a training set of 10 pairs of submaps with known

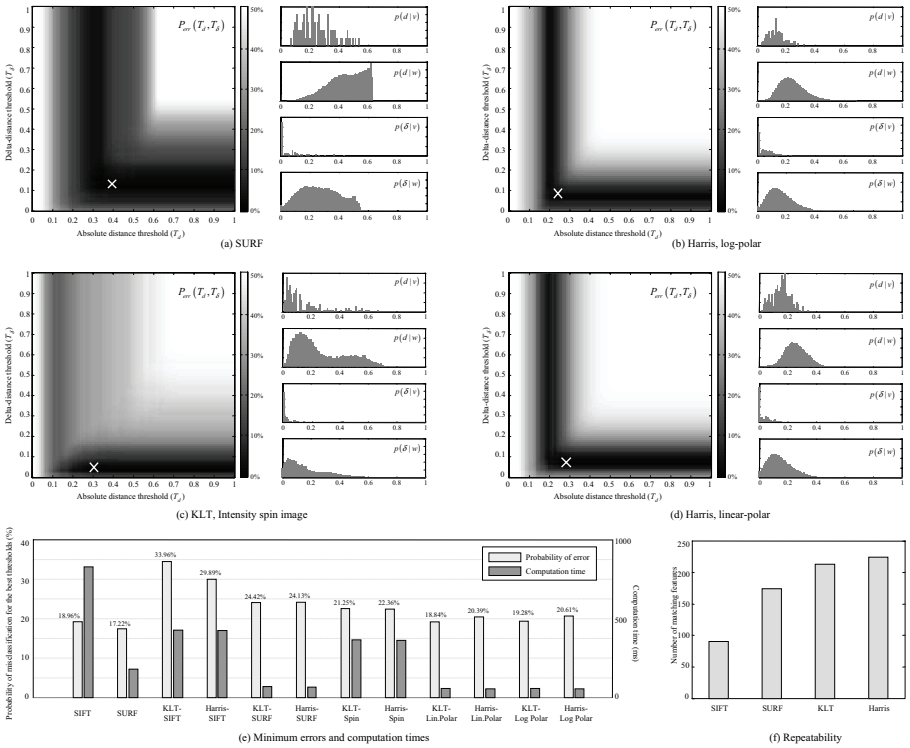

Figure 5: The benchmark of feature detectors for grid map matching. (a)–(d) Four examples of the expected P_{err} for different values of thresholds T_d and T_δ. The point with the minimum P_{err} is marked with a cross in each figure. We have also shown the marginal conditional distributions for the distance d and the distance-difference δ for valid (v) and wrong (w) associations are shown on the right hand of each subfigure. (e) For each combination of detector and descriptor, the resulting overall probability of classification error P_{err} for its best thresholds, i.e. that marked with a cross in (a)–(d), along with its average computation time for one map. (f) A measure of the repeatability for each detector.

ground-truth and for several combinations of detectors and descriptors. Optimal thresholds have been determined by minimizing the probability P_{err} of misclassifying a correspondence as a valid or an invalid candidate, given by:

$$\begin{aligned} P_{err}(T_d, T_\delta) &= P(w)P_{err}(T_d, T_\delta|w) + P(v)P_{err}(T_d, T_\delta|v) \\ &= P(w)P(d_{ij} < T_d, \delta_{ij} < T_\delta|w) \\ &+ P(v)\left[1 - P(d_{ij} < T_d, \delta_{ij} < T_\delta|v)\right] \end{aligned} \quad (3)$$

which can be evaluated given knowledge of the joint densities $p(d, \delta|v)$ and $p(d, \delta|w)$, where v and w stand for valid and wrong pairings, respectively. The expression above can be easily derived by noticing that a misclassification will occurs when: (i) a distance d_{ij} passes both thresholds and it was a wrong association (first term in the sum), or (ii) a valid pairing does not pass the thresholds (second term). For our analysis we assume no a priori information about the probability of being in a valid or invalid pairing, thus we have $P(v) = P(w) = 1/2$. The joint conditional densities $p(d_{ij}, \delta_{ij}|v)$ and $p(d_{ij}, \delta_{ij}|w)$ have been estimated from histograms generated by evaluating all the potential pairings in the 10 pairs of submaps, which amounts to 220 valid and 240,000 invalid correspondences.

The results of the benchmark are summarized in Figure 5(e) which shows the minimum classification error P_{err} attainable by each combination of feature detector and descriptor, along the associated average computation time (for one whole submap). These times include detection, descriptor extraction and distance computations, but they do not include the preprocessing filters discussed in Section 2.2. This preprocessing would add an average of 10 to 200ms, with larger computational burdens associated to SIFT and SURF since they require larger filter kernels than the Harris or KLT methods.

Please, notice that for those descriptors parameterized by a maximum radius R_{max} (see Section 3.1) we present the results only for the value that minimizes the classification error. However, this is a non-critical parameter since any value in the range $1 - 3$ meters gives similar results.

5 Discussion

The first important conclusion we can extract from our comparison is that no descriptor can tell valid pairings from wrong ones with a classification error below $\sim 20\%$, which is clearly a consequence of the ambiguity of features in map images where many ones look quite similar locally. Still, discarding $\sim 80\%$ of the wrong pairings provides an invaluable improvement to any subsequent robust matching algorithm (such as [**?**]), since it will have to deal with a reduced fraction of outliers.

It is interesting to note that the SIFT and SURF descriptors have a much poorer performance when computed for interest points localized by the Harris or the KLT detectors (third to sixth values in the bar graph) than when computed

as proposed in their original methods (the first two values in the graph). As commented in Section 2.1 and illustrated in Figure 5(f), this has important consequences for the practical applicability of those descriptors to grid matching, since the original SIFT and SURF detectors have poorer repeatability than the Harris and KLT methods. Subsequently, we discard the usage of these two descriptors as the optimal solution since they lead to quite similar error ratios (P_{err}) than the other descriptors while severely reducing the number of matched points and implying a higher computational burden, as can be seen in Figure 5(e).

In Figure 5(a)–(d) it is represented the computed $P_{err}(T_d, T_\delta)$ for some selected methods along the marginal distributions obtained in our benchmark. Observe how the marginal $p(\delta_{ij}|v)$ presents a clear peak at the origin ($\delta_{ij} = 0$) for all the methods, which indicates that the closest feature is often the actual correspondence [3]. However, this is not always the case, hence the optimal T_δ values are not exactly zero.

Notice that the worst obtained value for P_{err} (0.5, represented in white in the graphs) is obtained for a wide range of threshold values, while more reduced error ratios only appear for a certain band of the parameters (represented by darker areas). The thickness of these bands is related to the distinctiveness of the descriptors, as can be observed in the densities of descriptor distances for valid and wrong pairings (the histograms at the right hand of each P_{err} graph). For instance, compare the histograms $p(d|v)$ and $p(d|w)$ for the SURF and the Spin descriptors in Figure 5(a)-(c), where it is clear that in SURF the histograms concentrate in relatively different areas (easing the decision of where to place the threshold) whereas this is definitively not the case for the Spin descriptor.

As a final conclusion from our benchmark, the *lin-polar* and *log-polar* descriptors, both with virtually identical performance, emerge as the best choices for grid matching in combination with either Harris or KLT detector, due to their reduced misclassification probability and faster computation time.

References

[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Lecture Notes in Computer Science*, 3951:404, 2006.

[2] A. Birk and S. Carpin. Merging occupancy grid maps from multiple robots. *IEEE Proceedings*, 94(7):1384, 2006.

[3] J.L. Blanco, J.A. Fernández-Madrigal, and J. Gonzalez. Towards a Unified Bayesian Approach to Hybrid Metric-Topological SLAM. *IEEE Transactions on Robotics*, 24(2):259–270, 2008.

[4] J.L. Blanco, J. Gonzalez, and J.A. Fernández-Madrigal. A Consensus-based Approach for Estimating the Observation Likelihood of Accurate Range

[3] Recall that, by definition, $\delta_{ij} = 0$ means that feature $\mathbf{f}_j$ has the minimum distance to feature $\mathbf{f}_i$.

Sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4032–4037, 2007.

[5] M. Bosse, P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller. An Atlas framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 2, pages 1899–1906, 2003.

[6] Tom Duckett and Ulrich Nehmzow. Mobile robot self-localisation using occupancy histograms and a mixture of gaussian location hypotheses. *Robotics and Autonomous Systems*, 34(2–3):119–130, 2001.

[7] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989.

[8] C. Estrada, J. Neira, and J.D. Tardos. Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments. *IEEE Transactions on Robotics*, 21(4):588–596, 2005.

[9] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[10] G. Grisetti, G.D. Tipaldi, C. Stachniss, W. Burgard, and D. Nardi. Fast and accurate slam with rao-blackwellized particle filters. *Robotics and Autonomous Systems*, 55(1):30–38, 2007.

[11] J.S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 318–325, 1999.

[12] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of Alvey Vision Conference*, volume 15, 1988.

[13] Rob Hess. C implementation of SIFT feature detector, 2009.

[14] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using affine-invariant regions. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, 2003.

[15] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, 1999.

[16] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *Proc. DARPA Image Understanding Workshop*, 121:130, 1981.

[17] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of European Conference on Computer Vision*, volume 1, pages 128–142. Springer, 2002.

[18] C. Plagemann, K. Kersting, P. Pfaff, and W. Burgard. Gaussian beam processes: A nonparametric bayesian measurement model for range finders. In *Robotics: Science and Systems (RSS)*, 2007.

[19] J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[20] S. Thrun. Learning Occupancy Grid Maps with Forward Sensor Models. *Autonomous Robots*, 15(2):111–127, 2003.

[21] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, September 2005.

[22] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, 2003.