robotics

MDPI

Article

Visual Place Recognition for Autonomous Mobile Robots

Michael Horst * and Ralf Möller

Computer Engineering Group, Faculty of Technology, Bielefeld University, Bielefeld D-33615, Germany; moeller@ti.uni-bielefeld.de

* Correspondence: mhorst@ti.uni-bielefeld.de; Tel.: +49-521-106-5279

Academic Editor: Huosheng Hu

Received: 14 March 2017; Accepted: 12 April 2017; Published: 17 April 2017

Abstract: Place recognition is an essential component of autonomous mobile robot navigation. It is used for loop-closure detection to maintain consistent maps, or to localize the robot along a route, or in kidnapped-robot situations. Camera sensors provide rich visual information for this task. We compare different approaches for visual place recognition: holistic methods (*visual compass* and *warping*), signature-based methods (using Fourier coefficients or feature descriptors (able for binary-appearance loop-closure evaluation, *ABLE*)), and feature-based methods (fast appearance-based mapping, *FabMap*). As new contributions we investigate whether warping, a successful visual homing method, is suitable for place recognition. In addition, we extend the well-known visual compass to use multiple scale planes, a concept also employed by warping. To achieve tolerance against changing illumination conditions, we examine the NSAD distance measure (*normalized sum of absolute differences*) on edge-filtered images. To reduce the impact of illumination changes on the distance values, we suggest to compute ratios of image distances to normalize these values to a common range. We test all methods on multiple indoor databases, as well as a small outdoor database, using images with constant or changing illumination conditions. ROC analysis (receiver-operator characteristics) and the metric distance between best-matching image pairs are used as evaluation measures. Most methods perform well under constant illumination conditions, but fail under changing illumination. The visual compass using the NSAD measure on edge-filtered images with multiple scale planes, while being slower than signature methods, performs best in the latter case.

Keywords: visual place recognition; holistic image processing; visual compass; robot localization

1. Introduction

A common task of autonomous mobile robots is to build a map of an environment, and to navigate within this map (*simultaneous localization and mapping*, SLAM). The methods studied in this paper assume the use of a *topological* or *topometric* map. A *topological* map is a graph where each place in the environment is represented by a graph node [1]. Each node stores sensor information (e.g., a laser scan or a panoramic image) collected at the associated place. In the special case of a topological map with metric information (*topometric map*), nodes also carry metric position estimates. Links between nodes indicate that the robot can travel between these nodes on a direct path. The density of nodes required to describe the environment can vary depending on the robot's task, ranging from sparse (e.g., one node per room) to dense (e.g., a node every few centimeters) [2,3].

Identifying two places to be identical based on their associated sensor information is called *place recognition*. Commonly, a distance measure is computed based on the sensor information (e.g., a metric error after registration of laser scans, the number of correctly matched features between images, or a pixel-wise distance between camera images). The place with the minimum (or maximum)

distance can be selected as a single best match (e.g., for loop-closure detection). Alternatively, multiple similar places can be determined by thresholding the distance values (e.g., to add links to a graph). Possible applications of place recognition include: loop-closure detection, kidnapped-robot problem, localization along routes [4,5], and place discrimination. Particularly, *loop-closure detection* is an important prerequisite to build consistent maps. With loop-closure detection, the robot has to detect whether it has returned to a previously visited location. Errors during pose prediction can cause inaccurate pose estimates after driving long distances, thus loop closures cannot be detected based on metrical estimates alone (and in purely topological maps, pose estimates are not available at all). Instead, sensor information from range sensors or camera images is used for this task, as it captures information about the robot's actual surroundings instead of referring to its internal pose estimate.

Correctly identifying loop closures is particularly important for maps with metrical position estimates. If loop closures are not detected successfully, these maps might contain multiple disjoint representations of a single region. Such an inconsistent metric map is useless for navigation. Incorporating even correctly detected loop closures in metric maps can result in strong changes of the position estimates of the entire map if the estimated positions of matched regions are far apart. Early SLAM systems could not handle falsely identified loop closures, as these would permanently introduce constraints that are incompatible with a correct representation of the environment. Newer systems can validate loop-closure constraints or remove them again if they appear to be inconsistent [6–8].

When using purely *topological* maps [1], loop closures only add new links in the graph representing the map. Alternatively, nodes representing the same location can be fused, while the remaining nodes are not affected. *Missed* loop closures prevent the detection of shortcuts in the map, and may add redundant nodes representing the same location. However, navigation between nodes is still possible when following the links connecting them.

The *kidnapped-robot problem* arises when a robot has been displaced manually and has to localize itself within the map, due to missing information about the displacement. The current sensor information can be compared to the data stored in the nodes of the map to find the best-matching place. In [9], Fourier signatures together with a Monte Carlo method are used to localize the robot.

In *place discrimination*, the results of place recognition are used to decide when a new node should be inserted into a map: If the current sensor information indicates a place already present in the map, no new node is inserted. Only if no good match with an existing place can be found, a new node is added to the map [1].

While *following a route* in a topological map, place-recognition techniques can be used to check whether the next node along the route has been reached [4]. The next node along the route can then be selected and the process repeats until the goal is reached.

In this paper, we only consider place-recognition methods using visual information, specifically panoramic images. Our previous work has been concerned with the visual control of cleaning robots for systematic room coverage using a topometric map [3,10]. Aside from map consistency, loop-closure detection in this task is crucial to avoid cleaning an area multiple times, or leaving areas uncleaned. The navigation of our cleaning robot employs a holistic visual homing method (Min-Warping [11]). Here the term "holistic" refers to methods considering the entire image content rather than detecting some local features. Our goal in this work is to investigate whether holistic methods can also be used for place recognition as an alternative to the widely used feature-based methods. We are currently working on a complete-coverage method for cleaning robots that employs a dense purely topological map (with only relative pose information stored in the links). Nodes identified by place recognition, together with their neighbors in the graph, are used to triangulate relative pose estimates to move adjacent to previously traversed areas. False loop closures could be identified by comparing multiple triangulations for consistency.

The paper is structured as follows: Section 1.1 gives an overview of the related literature. The methods used in this paper are presented in Section 2. Section 3 describes the datasets and experimental setups used to evaluate the different place-recognition methods. The results are presented in Section 4, a discussion follows in Section 5. Final conclusions are presented in Section 6.

1.1. Related Work

In the following, we focus on visual methods for place recognition, which can be divided into multiple categories: holistic, feature-based, and signature-based. Holistic methods use all information contained in an image, usually based on pixel-wise comparisons, without selecting any special regions of interest. Feature-based methods detect keypoints in an image (e.g., corners) and describe the surrounding pixels using a descriptor; these descriptors are matched between images. Signature-based methods describe whole images using a low-dimensional representation.

A recent survey on visual place recognition, focusing on feature- and signature-based methods, is available in [12].

1.1.1. Holistic Methods

Holistic methods for place recognition, especially for outdoor applications, have been studied in [13–15]. Visual place recognition usually relies on interrelating image content appearing in multiple images. To avoid a limited field of view resulting from standard cameras and lenses, panoramic images showing a 360° view around the robot are preferred to guarantee overlapping image content. Panoramic images are often captured using catadioptric mirrors or ultra-wide-angle fisheye lenses and mapped to a spherical panoramic image (example images are provided in Section 3). Here we assume that robot motion is restricted to the plane, such that only azimuthal rotations of the camera are possible, which correspond to horizontal shifts of the panoramic images. Two images are most similar when they are rotationally aligned in azimuthal direction [13]. Therefore, a pixel-wise distance measure between the panoramic images is computed for all rotations and the minimum is determined. The rotation angle at the minimum can be used as a *visual compass* [13,14]. Image dissimilarity can be deduced from the image distance at this minimum. Minimal image dissimilarity typically increases with increasing spatial distance between the capture points of the two images [13]. This relationship is caused by the fact that image changes caused by translation (both vertical and horizontal distortions) are not captured by the rotational alignment. Phase correlation [16,17] based on Fourier transformation can also be used to estimate the rotation.

A local visual homing method called Min-Warping [11] utilizes two extensions to this simple visual compass to consider distortions caused by translations. Its first phase computes all pairwise distances between image columns and can be used to estimate the relative orientation between images, similar to the visual compass. To account for *vertical distortions* that are caused when the distance to an object changes, the images are scaled around the horizon using a discrete set of scale factors. *Horizontal distortions* also arise from changes of the relative order of objects: A close object appearing on one side of a more distant object in the first image may appear on its other side after movement of the camera. In the second phase of Min-Warping the image columns are compared independently to counter this effect, instead of retaining the relative order of the image columns during the comparison (as in the 2D-Warping method [18]). We explore how these two extensions affect the performance of Min-Warping as a method for place recognition.

Holistic methods are also used in SeqSLAM [5,19], where the sum of absolute differences between low-resolution, patch-normalized images is computed. Sequences of images are compared to identify segments captured previously along a route. An extension to two-dimensional maps (in contrast to routes) is available in 2D SeqSLAM [20].

1.1.2. Feature-Based Methods

A large variety of feature-based methods is available which can also be applied in the context of place recognition. Feature-based methods usually detect keypoints (e.g., corners) and describe the surrounding patches using a descriptor (*feature extraction*), which is matched to feature descriptors from other images. The descriptor can contain, for example, information about pixel values, gradients, or relative pixel intensities. Scale-invariant feature transform (SIFT) [21] and speeded-up robust features (SURF) [22] were among the first widely used feature detectors and descriptors. Binary feature descriptors became popular in recent years due to their faster processing speed. To match features, their descriptors can be compared using the Hamming distance [23] which just counts differing bit values in the descriptors. Examples of such detectors and descriptors include features from accelerated segment test (FAST) [24], binary robust independent elementary features (BRIEF) [25], oriented FAST and rotated BRIEF (ORB) [26], binary robust invariant scalable keypoints (BRISK) [27], and local difference binary (LDB) [28].

A well-known example of a feature-based technique for place recognition is FabMap. FabMap [29] and its extensions [30,31] provide a full appearance-based SLAM system. It uses the concept of a bag of visual words [32], where each feature descriptor is associated to a visual word using a clustering method. An image dissimilarity value can be derived from comparing histograms of these visual words. An open source implementation called "OpenFabMap" is available [33] and has been used for this work.

The clustering method used for FabMap has to fit with the chosen image descriptor type. While k-means clustering is suitable for floating-point descriptors like SURF, it cannot be used for binary descriptors. A majority-based clustering for binary vectors has been proposed in [34] and has also been used here.

1.1.3. Signature-Based Methods

Signature-based methods are also referred to as parameter-based methods, or global image descriptors. They describe an entire image using a low-dimensional parameter vector or descriptor (signature). Using rotation-invariant signatures allows to match images captured with different azimuthal orientations without aligning them explicitly. Alternatively, a signature can describe multiple horizontal regions of an image and pairwise matches can be computed to account for orientation differences. Different combinations of signatures and dissimilarity measures have been evaluated in [35,36], where signatures included histograms, image moments, and Fourier coefficients. The measures used to compare signatures include maximum norm, Manhattan and Euclidean distance, and Kullback–Leibler divergence for histograms.

Other methods use single feature descriptors to represent images. A descriptor is usually computed for a small patch, for example a 32×32 pixel neighborhood of a keypoint. To compute a single feature descriptor for a complete image, the image is first scaled down to this patch size. Then a "keypoint" is placed in the center of this patch, and the feature descriptor is computed. The image can also be split into multiple regions (for example on a regular grid), each of which is then scaled down individually. A feature descriptor for each patch is then computed and the results are stacked into a single signature vector.

BRIEF-Gist [6] uses a single or multiple patches, and computes the BRIEF descriptor of each patch. Image comparisons are based on the Hamming distance between the resulting feature descriptors. The method works on panoramic and standard images, but they have to be aligned correctly. Splitting a panoramic image into multiple regions horizontally and calculating the distance between all pairs of these regions in two images allows rotated images to be matched, to compensate for orientation changes of the robot. The method called ABLE (Able for Binary-appearance Loop-closure Evaluation) [37–40] uses this approach. The authors identified the LDB feature descriptor [28] to work well with ABLE. An open-source implementation of this method called OpenABLE [40] is available online and has been used here.

2. Evaluated Methods

In the experiments described below, the robot is assumed to have three degrees of freedom, only moving in a horizontal plane without tilt or changing height. Rotations are restricted to azimuthal orientation changes within this plane. Except for FabMap, all methods evaluated in this paper rely on this assumption.

The methods presented in this paper use omni-directional images that show a 360° view of the surroundings. Such images can be captured using catadioptric or fisheye imaging systems. Some methods need panoramic images that are spherical mappings of the captured images. They are cyclic in horizontal direction, so azimuthal rotations correspond to horizontal shifts of the columns (example images are provided in Section 3). Any indexing based on horizontal coordinates (x) is assumed to be wrapped accordingly. The images have width w and height h.

Most methods presented in this section compute a dissimilarity value between a pair of images (we use the term image dissimilarity in this paper to avoid confusion with the actual metric distance between the corresponding capture locations). In the context of place recognition, one image usually corresponds to the robot's current location and is called *current view (CV)*. It is compared to previously captured images, each referred to as *snapshot (SS)*. The method presented in Section 2.4 also considers neighbors of a snapshot for the dissimilarity computation.

2.1. Holistic Methods

We include two holistic methods in our comparison: a *visual compass* and *warping*.

2.1.1. Visual Compass

Computing the dissimilarity value for all relative azimuthal orientations of two images (see below for details) results in a *rotational image dissimilarity function (R-IDF)* [13]. The position of its minimum is an estimate for the actual relative orientation between the images (*compass estimate*), while the value at the minimum gives the dissimilarity value for the image pair. The R-IDF between an image and all rotated images captured at the same position shows a pronounced minimum for unshifted images, leveling off at higher values for large rotations. For images taken further away from the snapshot, the R-IDF becomes more shallow and the minimum increases [13] (see Section 2.1.3). This is caused by changes in the image content resulting from translations, since in this simple form, the visual compass can only consider image changes caused by rotations. The computation can be done by calculating distances between all pairs of image columns from SS and CV, resulting in a $w \times w$ matrix called a *scale plane*. Summing the values along each diagonal of the scale plane gives the total dissimilarity value for each possible shift between the images (see Figure 1).

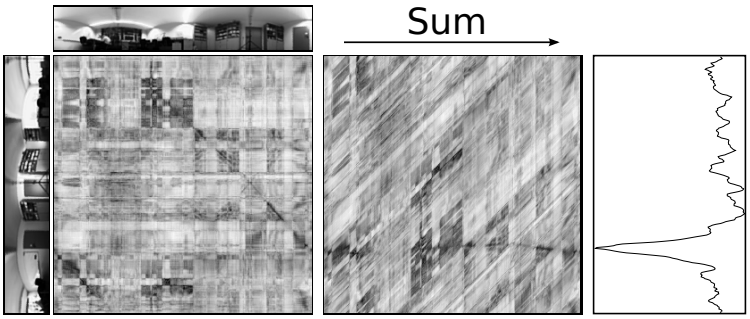

Figure 1. Illustration of the visual compass. **Left:** All pairwise column distances between two images are computed (here with the normalized sum of absolute differences (NSAD) distance measure) and stored in a scale plane. Small distances (good matches) are shown as dark pixels, large distances are bright. This scale plane can also be the minimum over multiple scale planes obtained from pre-scaled images (Figure 2). **Middle:** The scale plane is reordered so each diagonal is stored as a row, corresponding to a shift between the images. Summing along these rows gives the total image dissimilarity for each shift. **Right:** The rotational image dissimilarity function (R-IDF) shows a minimum close to the actual shift between the two input images. The location of the minimum can be used as a visual compass, while the value gives the image dissimilarity value for the image pair.

To account for vertical distortions that are caused when the distance to an object changes, the images can be scaled around the horizon using a discrete set of scale factors σ_i (to retain a feasible amount of computations) [11]. For each scale factor σ_i, its inverse $1/\sigma_i$ is commonly used as well to get a symmetric set of scale factors. The images are only magnified around the horizon, as unavailable information above the upper and below the lower image borders would be required to scale down an image (instead of scaling down one image, the other one is scaled up by an appropriate factor). Additional scale planes are then computed by pairing each scaled image with an unscaled one, resulting in a so-called scale-plane stack (see Figure 2). For example, using three scale factors results in three scale planes (unscaled SS and unscaled CV, scaled SS with unscaled CV, and unscaled SS with scaled CV). The minimum of corresponding pixels over all scale planes is determined and in the resulting image, summation along the diagonals yields the R-IDF. By computing the minimum, scaled image columns that result in the best match are selected. Using only unscaled images (one scale plane) corresponds to the R-IDF methods used in [13–15]. To the best of our knowledge, this scale-tolerant visual compass has only been used to accelerate warping [11], but not as a stand-alone image dissimilarity measure.

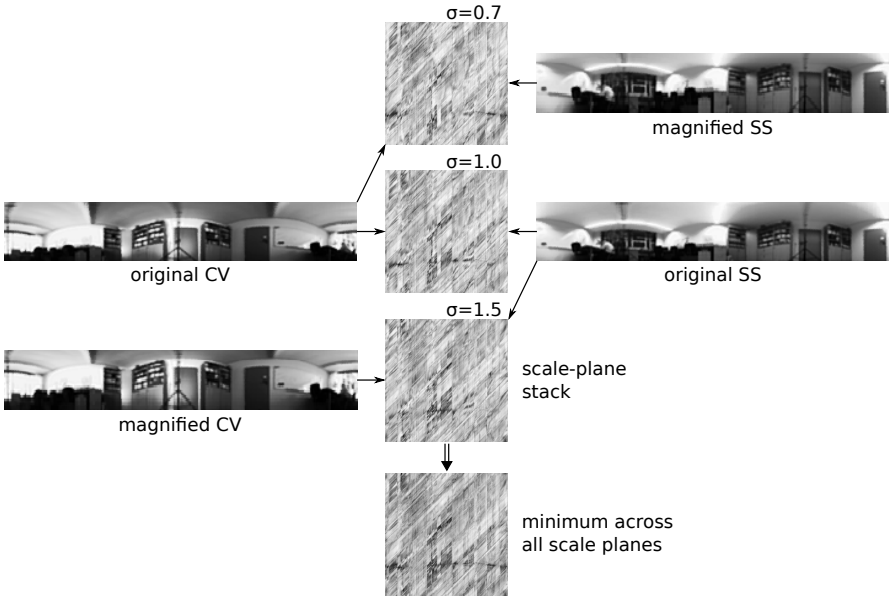

Figure 2. Computation of the scale-plane stack. For each scale factor σ, either the snapshot or current view is magnified around the image horizon (bottom row in this case). All distances between the columns of both images are computed and stored in a scale plane (compare Figure 1). For the visual compass, the minimum of each entry across all scale planes is determined and stored. Warping uses the individual scale planes.

To consider horizontal distortions, the image columns have to be matched independently, removing their fixed order. Warping (Section 2.1.2) accomplishes this task in its second phase, using the full scale-plane stack described here.

Different distance measures can be used as dissimilarity measures. We use the notation $J(\mathbf{a}, \mathbf{b})$ for the distance between two columns $\mathbf{a}$ and $\mathbf{b}$ (original or edge-filtered; possibly scaled), with entries a_i and b_i. The following three measures between image columns are used here:

- sum of squared distances (SSD):

$$J_{\mathrm{SSD}}(\mathbf{a}, \mathbf{b}) = \sum_i (a_i - b_i)^2 \tag{1}$$

- sum of absolute differences (SAD) or Manhattan distance:

$$J_{\mathrm{SAD}}(\mathbf{a}, \mathbf{b}) = \sum_i |a_i - b_i| \tag{2}$$

- normalized sum of absolute differences (NSAD) [41,42]:

$$J_{\mathrm{NSAD}}(\mathbf{a}, \mathbf{b}) = \frac{\sum_i |a_i - b_i|}{\sum_i |a_i| + \sum_i |b_i|} \tag{3}$$

A scale plane D_k is computed as

$$D_k(SS, CV, i_x, i_\psi) = J(SS[k](i_x), CV[k](i_x + i_\psi)), \tag{4}$$

where i_ψ is the relative orientation, or shift, in pixels, between 0 and $w - 1$, $SS[k](i_x)$ refers to column i_x (possibly edge-filtered) of the snapshot for scale k ($CV[k](x)$ accordingly), and J is one of the column-distance measures listed above. The indexing based on column index and shift corresponds to the scale plane shown in the middle part of Figure 1 (with i_x as horizontal coordinate).

The total distance between shifted images is then computed as the sum along a row of the minimum of all scale planes:

$$d(SS, CV, i_\psi) = \sum_{i_x} \min_k D_k(SS, CV, i_x, i_\psi). \quad (5)$$

The final dissimilarity value is obtained as the minimum of the R-IDF over all shifts i_ψ:

$$d = \min_{i_\psi} d(SS, CV, i_\psi). \quad (6)$$

Edge-filtered versions of the measures (here only used for NSAD) use a simple first-order vertical edge filter on the columns before any distance computation.

Throughout this work, compass methods are named `i|e-<measure>-<scaleplanes>[-<ratio>]`. The prefix `i` stands for *intensity-based*, `e` for *edge-filtered*. `<measure>` is a shorthand for different distance measures, for example `ssd` for sum of squared distances, or `sad` for sum of absolute differences. `<scaleplanes>` denotes the number of scale planes used for the measure. `<ratio>` is a further processing step (see Section 2.4), and is omitted if not used.

The methods `e-nsad`, `i-nsad`, `i-sad`, and `i-ssd` are used here (with the appropriate additional suffixes).

2.1.2. Warping

Min-Warping is a holistic local visual homing method, matching images based on simulated movement (see [11] for details). It computes the home vector and compass between a current view (CV) and a reference snapshot (SS).

The first phase of the Min-Warping algorithm computes the scale-plane stack containing all pairwise column distances between these two images, as described in Section 2.1.1.

The second (search) phase of the Min-Warping algorithm iterates through different movement hypotheses. Each movement causes image columns from the snapshot to appear in a different location and scale in the current view. The region, and scale, in the current view where each column may reappear is restricted by such a movement hypothesis. The minimal dissimilarities for each column in the snapshot and a column from within the corresponding region in the current view are obtained from the scale-plane stack and summed. The hypothesis resulting in the minimum sum gives the estimate for home vector direction and relative orientation. In contrast to the other holistic methods presented here where the order of columns is kept, the columns can be matched independently to account for relative horizontal movement of objects. Thus, their order may change in the matching process.

We use an optimized implementation utilizing a template-based single instruction, multiple data (SIMD) library [41,43], which is available online. Only the NSAD measure on edge-filtered images using nine scale planes is used for this study. The method is referred to as `warping`, using the ratio suffix as described at the end of Section 2.1.1.

2.1.3. Comparison of Visual Compass and Warping

Figure 3 shows R-IDFs generated using `warping`, `e-nsad-1`, and `e-nsad-3`. The red curves show values obtained from images captured at the same position, while the images used for the blue curves were taken about 34 cm apart. The relative orientations of the image pairs differed by 180°, so the plots can be distinguished more easily.

The curves for the same position R-IDFs show a sharp minimum at the actual shift between the images, the values of the different measures are very similar. As images captured at the same position with different orientations are used, the dissimilarity is not exactly zero (due to image noise and small scene changes). For increasing orientation differences the values of all three measures increase, stronger for `e-nsad-3` and `e-nsad-1`. The R-IDFs for the displaced image pair still show a clear minimum, albeit at higher values due to more changes in the image content.

Figure 4 shows the minimum of the R-IDFs for `warping`, `e-nsad-1`, and `e-nsad-3` along a cross section of a lab grid database for an image captured in the center. Again, all methods show a minimum at the capture position, with increasing values towards further away images. The dissimilarity values for `warping` are smallest, with higher values for `e-nsad-3` and `e-nsad-1`. This reflects the capabilities of the different methods to account for distortions in the images: `e-nsad-1` only considers orientations, resulting in large dissimilarities for displaced images. `e-nsad-3` can handle vertical scale changes, leading to slightly smaller values. By also changing the relative order of columns in the images, `warping` can tolerate more changes in the image, resulting in the smallest dissimilarities of the three methods.

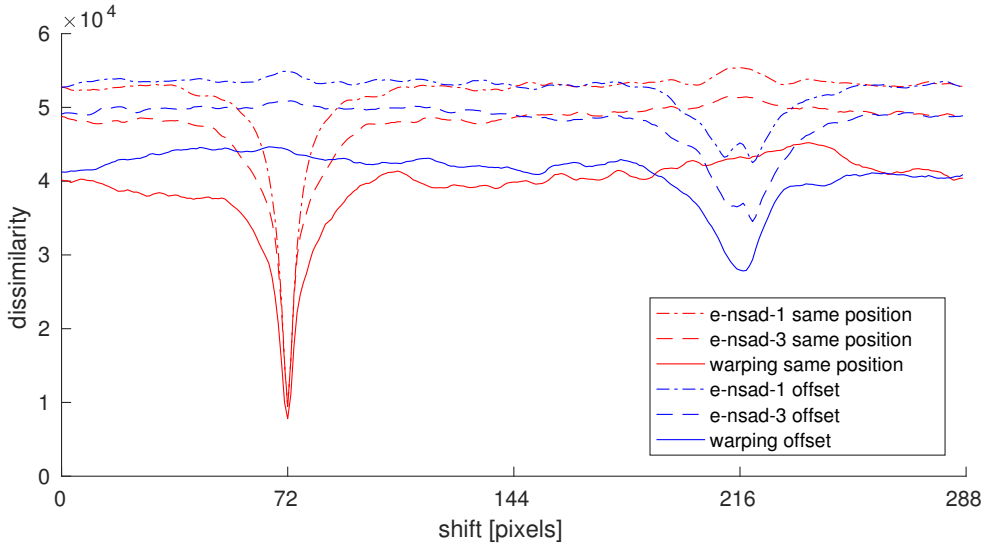

Figure 3. Rotational image dissimilarity functions (R-IDF) obtained from `e-nsad-1`, `e-nsad-3`, and `warping` for two image pairs. The first pair of images was captured at the same position, but with different orientations (red curves). The second image pair was captured about 34 cm apart, with a different relative orientation (blue curves). All curves show a minimum at the actual image shift, but level off at different values. The minima of the red curves are nearly identical, as no vertical or horizontal distortions appear in the images. The blue curve for `e-nsad-1` has large values at the minimum, as a good image match is harder to obtain by just rotationally aligning the images from different places. The additional scale planes used for `e-nsad-3` provide a tolerance to vertical distortions, leading to smaller values. `warping` results in even smaller values than `e-nsad-3`, due to its additional tolerance to horizontal distortions.

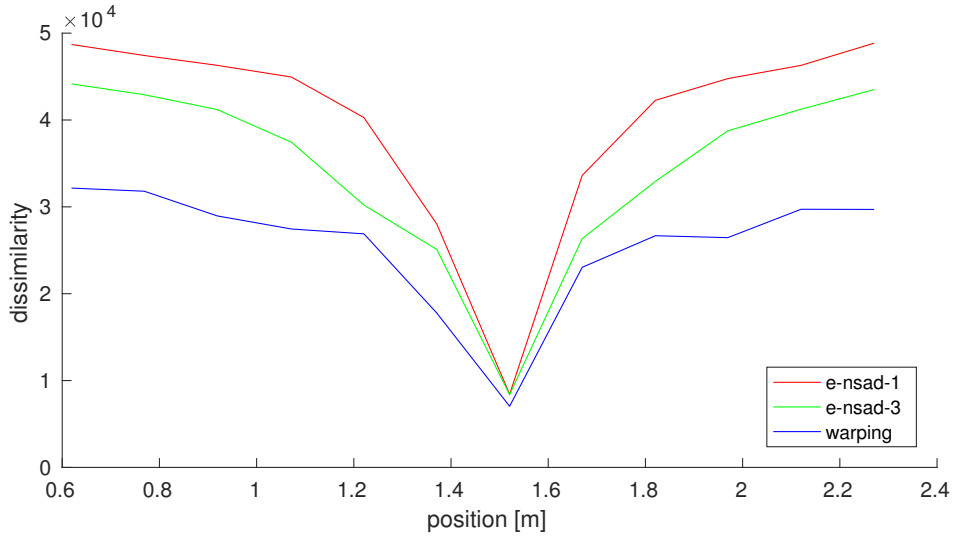

Figure 4. Dissimilarity values between images captured along a cross section of a lab grid database with an image (with different orientation) at its center obtained from `e-nsad-1`, `e-nsad-3`, and `warping`. All curves show a minimum with a similar value at the center image position, but level off at different values. `e-nsad-1` results in the largest dissimilarities for large spatial distances, as it can only compensate for rotations. The tolerance for vertical distortions provided by the additional scale planes in `e-nsad-3` leads to slightly smaller values. `warping` additionally tolerates horizontal distortions and a reorder of columns in the image, resulting in the smallest dissimilarity values.

2.2. Feature-Based Method: FabMap

We used OpenFABMAP as a method for feature-based place recognition. It requires a training phase before the actual localization can be done. Keypoints are detected on the training images and the corresponding descriptors are extracted. The descriptors are clustered to obtain a "vocabulary" (also called *bag of words*) of desired size, each cluster center representing a "word" of this vocabulary. For SIFT or SURF features with their floating-point descriptors, simple k-means clustering can be used: First, cluster centers are selected randomly out of the descriptors. Second, each descriptor is associated to its nearest cluster center, based on Euclidean distance. Third, new cluster centers are computed by averaging all descriptors associated to each current center. The second and third step are repeated until the cluster centers converge. Binary feature descriptors require a modified clustering, as they only contain ones and zeros as coordinates and rely on fast comparisons using the Hamming distance. The averaging step would result in fractional coordinates, eliminating this advantage. A majority-based clustering (*k-majority*) has been proposed in [34], which counts the number of zeros and ones for each coordinate over all cluster members and selects the dominant value (0 or 1) as the new value of the corresponding cluster center coordinate.

An image descriptor is computed for each image, based on the vocabulary: Each feature descriptor is associated with a visual word (based on Euclidean or Hamming distance). A histogram containing the frequencies of the occurrences of all visual words forms the resulting image descriptor. For further processing, only the presence or absence of visual words is considered (eventually treating the descriptor as a binary vector).

For each visual word, the likelihood that it appears in the scene is computed from these training image descriptors. These values are later used during the image matching process to determine the likelihood that two images were captured at the same place. Treating all visual words independently would discard a lot of information, as certain features often appear together (e.g., corners in a window frame). To include some of this information, the feature pairs with the highest mutual information are determined and stored in a Chow–Liu tree [44]. Features that often appear together in images of a place should also appear together in new images of this place.

Images are compared based on their image descriptors and the Chow–Liu tree. A matching score is computed between an image and all snapshots based on the presence or absence of feature pairs, where a high score indicates a good match. A score is also returned for the case that an image is not contained in the set of snapshots but corresponds to a new place by determining the average match likelihood with all training images.

The score corresponds to a log-likelihood value (albeit without a proper normalization), which can also be converted to a (normalized) matching probability for a set of images. The normalized probabilities often show only a single prominent peak, while lower scores are strongly suppressed. Depending on the application, it may be beneficial to find multiple well-matching images. The log-likelihood values allow a better selection of multiple matches (see Figure 5). The range of values for the log-likelihood may vary for different image pairs (due to the missing normalization step). We invert the sign of the score, so good matches correspond to low scores, as in the other methods. We also normalize the resulting scores to the range $[0..1]$ (based on the extrema of all scores for a set of image comparisons). A similar normalization is done by the authors of ABLE [37]. While this does not affect the minimum selection for raw values, it influences the computation of ratios of dissimilarities (see Section 2.4).

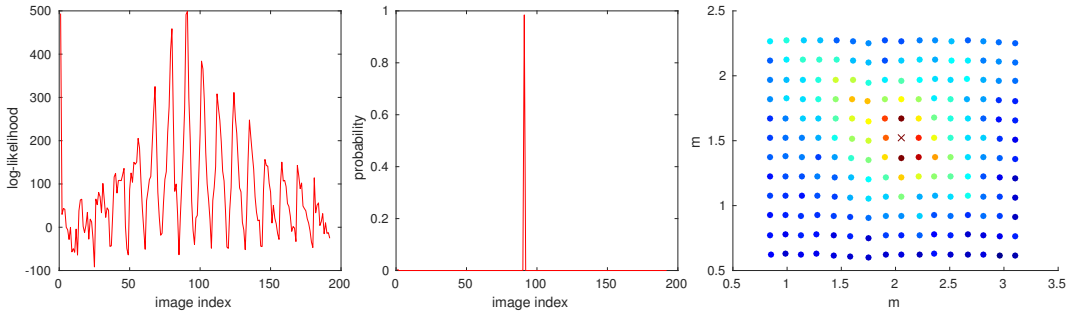

Figure 5. Comparison of (unnormalized) log-likelihoods (**left**) and normalized probabilities (**middle**) for an image match using FabMap with the ORB descriptor on a lab day grid database (see Section 3.1.1 for details). The horizontal axis shows the index of the image pair (column by column through the grid). The current view image was not included in the computation, so scores for 191 image pairs are shown, together with the score expressing that the current image corresponds to a new place (leftmost value in the graphs). The log-likelihood shows multiple peaks, indicating a number of similar images close to the selected one, while the final matching probability shows only a single peak. Even the probability for a new match is close to zero, while its log-likelihood is only slightly smaller than for the best matching image. The right image shows the nodes on the grid with color-coded log-likelihoods (rainbow colors for increasing values from blue to red). The position of the current view is marked with a cross. While the different peaks of the log-likelihood correspond to images close to the current view, a selection based on the normalized probability would only select the image directly below the cross as the best match.

Instead of the full FabMap SLAM method that also considers previous image matches along a route, we only use the image comparison function of OpenFABMAP. This allows the pairwise matching of two sets of images, in our case comparing a single current view to multiple snapshots in a database. The OpenCV implementations of SIFT [21], ORB [45] and BRISK [27] were used to obtain the feature descriptors. These descriptors have also been used in a recent comparison of holistic and feature-based methods for relative pose estimation, also done in our research group [46]. SIFT is relatively slow, but generally achieves better results than the fast binary methods. While ORB is the faster binary descriptor, BRISK should be more tolerant to illumination changes. The bag-of-words for ORB and BRISK was built using k-majority clustering [34]. The Hamming distance was used as distance measure. SIFT uses k-means clustering with the L2 norm.

The method is referred to as `fabmap-<desc>[-<ratio>]`, where `<desc>` is `SIFT`, `ORB`, or `BRISK`. `<ratio>` is a further processing step (see Section 2.4).

2.3. Signature-Based Methods

Two signature-based methods have been used in this study: a signature based on Fourier coefficients, and a signature based on a feature-descriptor.

2.3.1. Fourier Signatures

Place recognition based on image signatures has been evaluated in [35,36]. The method we use is a modification of [47], which has shown promising results [35] and whose parameters and choice of distance measure were adopted here. Figure 6 illustrates the computation of the image signature: The image is split into eight horizontal rings. Each ring is averaged vertically to obtain a vector of pixel intensities. The absolute values of the first twelve Fourier coefficients of each averaged ring are stacked into a 96-dimensional signature for each image (signature function $s_{\texttt{afc}}$ in [35]). The maximum norm $d = \max_i |\mathbf{s}_i - \mathbf{c}_i|$ is used to compute the dissimilarity between two signature vectors $\mathbf{s}$ and $\mathbf{c}$. As absolute values of the coefficients are compared, this method is invariant to azimuthal rotations. It is called `sigafc` in this evaluation, with the `ratio` suffix added if appropriate.

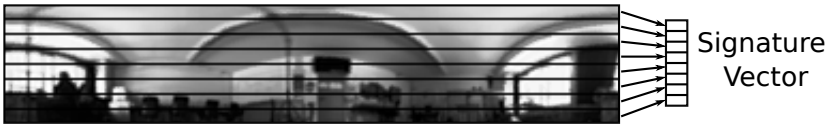

Figure 6. Calculation of a signature vector using absolute Fourier coefficients. The image is split into multiple horizontal rings (here 8). Each ring is averaged vertically and the 1D Fourier transformation is computed. The absolute values of the first (here 12) coefficients of all rings are stored in a signature vector.

2.3.2. ABLE

ABLE was used as a signature method using feature descriptors. Figure 7 illustrates the steps of ABLE. The panoramic images are split into one or multiple regions. Each region is downsized to a patch of fixed size (e.g., 64×64) pixels, and a feature descriptor is computed for each patch (the keypoint is set to the center of the patch). We use the binary LDB and BRISK descriptors. LDB has been identified as the best method in the original paper [37], while BRISK showed a good trade-off between speed and accuracy in one of our own studies about visual homing [46]. To compare two images, the Hamming distance for all pairs of descriptors is computed, and the minimum is stored as dissimilarity value. Using multiple regions in each image provides a simple rotation invariance, but requires a certain translational tolerance of the feature descriptor for arbitrary shifts. The method has originally been developed for routes traveled by car, where intersections often occur in perpendicular directions. Thus, numbers of regions that are multiples of four would be most suitable for those cases.

Feature descriptors can also be computed for multiple keypoints placed on a regular grid within each region to obtain a more detailed signature (e.g., a 2×2 grid). The resulting descriptors are stacked to get a single image signature. The signatures are compared without pairing descriptors of different grid cells.

We used the open source toolbox OpenABLE [40] for feature description. The toolbox is designed for images from a single database taken along a route, but we also want to include comparisons where SS and CV come from different databases. Thus, we used our own implementation to calculate the dissimilarity between images. We also normalize the resulting dissimilarity values for a database test to the range $[0..1]$ (based on the extrema of the dissimilarities for the database test). This normalization is also done by the authors of ABLE [37].

The method is referred to as `able<desc><numberOfRegions>[-<ratio>]`, where `<desc>` is `LDB`, or `BRISK`. `<ratio>` is a further processing step (see Section 2.4).

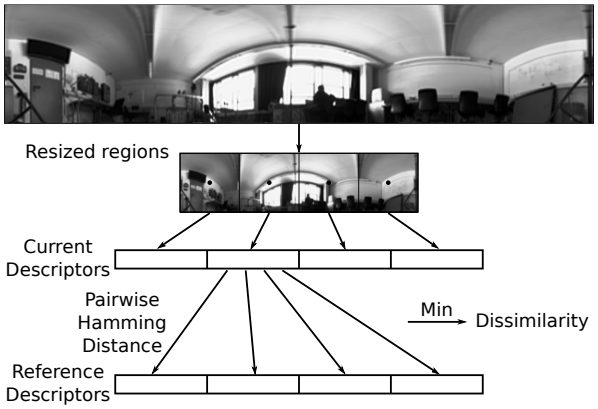

Figure 7. Computation steps for ABLE signatures. The original panoramic image (**top**) is split into multiple regions, which are each resized to a patch of size 64×64 pixels (**middle**). A feature descriptor is computed for each patch using its center (black circles) as a fixed keypoint. All descriptors together form the image signature. To compare two signatures, the Hamming distance between all pairs of descriptors is computed (**bottom**, here illustrated for just one descriptor of the current image). The minimum over all pairs gives the final dissimilarity value.

2.4. Further Processing: Neighborhood Ratio

The dissimilarity values obtained using the methods presented in the previous sections can be processed further to compute derived dissimilarity measures. We investigate a method using the ratio of dissimilarity values. The range of values obtained by the different dissimilarity measures presented above usually depends on the environment and conditions in which the images were captured. Images with similar illumination capture conditions tend to have smaller dissimilarities than images captured under changing illumination (see Figure 8). Computing ratios of dissimilarity values should normalize the values to a more common range. Images with different capture conditions can appear in a single map if a robot returns to a previously visited location after a relatively long time. Changing weather conditions during the mapping process (e.g., clouds vs. direct sunlight) can also have an impact on the illumination conditions. Thus, the map can contain a mixture of images captured under different conditions, which may result in a wide range of dissimilarity values even among neighboring snapshots.

We compute the dissimilarity values between the reference image and all its neighbors (selected by a threshold in the metric distance which is known from pose estimates of connected nodes) and store the minimum. Next, we compute the dissimilarity values between the current view and again the same neighbors of the reference image and store the minimum. The ratio between these two minima is used as the final dissimilarity value between the current view and reference image. If the two images are spatially close, this ratio should be close to or below one. For larger spatial distances, the minimum obtained with the current view should be much larger than the minimum from the reference image, resulting in a higher ratio. Besides the minimum, we also tried using the mean of the dissimilarity values of close neighbors. While the minimum should always come from a neighbor close to the reference image, the mean depends on the dissimilarity values of *all* images within the selection radius. Thus, a larger selection radius is expected to cause larger mean values, with negligible effect on the minimum values.

Figure 8 illustrates the image dissimilarity for the `e-nsad` measure for different metric distances to a node near the center of a lab grid database, for constant (red) and changing (green) illumination conditions. While the values for constant illumination show a clear minimum and increase with larger offsets, the other profile is much more shallow. The corresponding values of mean and min

ratio postprocessing are shown for the same image pairs. The mean ratio for close images shows a wide range of values, both above and below one, especially for constant illumination. This is problematic for thresholding, as false positives will be introduced when a larger threshold is selected to include these values. The min ratio, however, shows values consistently below one for close images. This should allow a more reliable place recognition when using a fixed threshold.

The ratios for the constant illumination image pairs are spread wider than for changing illumination, because the original dissimilarity values already span a wide range. This results in a wide range of ratios. For changing illumination, the original values are much closer together, so the ratio will also be close to 1.0. This compression of values may pose a challenge for a threshold-based classification.

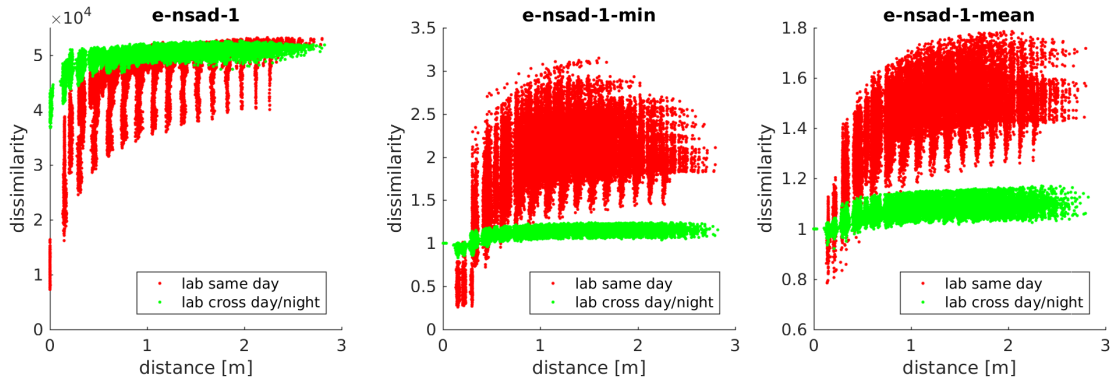

Figure 8. Dissimilarity values obtained from image pairs taken under constant (red) and changing (green) illumination conditions in a lab database. The vertical axis of the plots shows the dissimilarity value between a current image at the center of a grid and all remaining images in the grid. The horizontal axis shows the actual metric distance between the capture locations of the image pairs. **Left:** Dissimilarity values from the `e-nsad-1` measure. While the values obtained under constant illumination conditions show a deep pronounced minimum for short distances, the changing illumination causes much larger values with only a shallow minimum. Using a fixed threshold to classify both datasets is prone to fail. **Middle:** Values of the min ratio for the same image pairs (see text for details). A correct classification using the same threshold looks more promising, at least for very close images. **Right:** Values for the mean ratio. The ratio for nearby images under constant illumination is spread wider than for the min ratio (especially attaining values larger than 1.0 also for very close image pairs). Both min and mean ratios for changing illumination are centered much closer around 1.0, as the original values from which these are computed are already very similar. The wider range of dissimilarities obtained under constant illumination results in a wider range of ratios.

We use the suffix `-<ratio>` for these methods, where `ratio` is one of `min` and `mean`. If no ratio postprocessing is used, the entire suffix is omitted.

3. Experiment Description

This section describes the datasets used to evaluate the methods presented above, the experimental setup, and the evaluation criteria used in the results section.

3.1. Datasets

We used our own grid databases [46] (http://www.ti.uni-bielefeld.de/html/people/dfleer/fisheyedb.html), the Quorum V database [48] (http://arvc.umh.es/db/images/quorumv/), and a map built by our cleaning robot in the CITEC (Cognitive Interaction Technology (Center of Excellence) at Bielefeld University) research apartment (https://www.cit-ec.de/en/central-lab-facilities/research-apartment-0). We also included a small outdoor database as a preliminary test for outdoor use of the presented methods.

3.1.1. Grid Databases

The grid databases [46] were captured with a cleaning robot prototype in an *office* and in a *lab* environment, both in the early afternoon with natural lighting ("day") and in the evening with artificial lighting ("night"). Images were captured on a regular grid, size 12 × 16, with a resolution of 0.15 m, exact positions were determined using an overhead tracking system with a resolution of a few millimeters. At every position, images were taken in four different azimuthal orientations. Thus, each single database contains 192 images per orientation. For the experiments, the current view and reference snapshots were taken from fixed perpendicular azimuthal orientations (CV 180° and SS 270°).

The robot was equipped with a UI-1246LE-M-HQ (IDS Imaging Development Systems, Obersulm, Germany) monochrome camera and a DSL215 fisheye lens (Sunex, Greenville, SC, USA). The lens has an opening angle of 185°. The images were captured at the camera's full resolution of 1280 × 1024 pixels and scaled down by 2 × 2 binning to 640 × 512 (the resolution actually used when running our cleaning robot prototype). Preprocessing includes a third-order Butterworth low-pass filter, unfolding to a panoramic image of size 288 × 48 pixels (field of view 360° × 75°; horizon at row 47), and histogram equalization (see also [46] for details). The exposure time of the camera was controlled to maintain an average pixel intensity of 0.5 (value range [0..1]) on the unfolded image (before histogram equalization). We use the lab day database to study the effect of the relative cut-off frequency of the Butterworth filter, and use a relative cut–off frequency of 0.20 unless noted otherwise. We also investigate the influence of the vertical opening angle for the image mapping using the two lab databases.

Figure 9 shows an example fisheye and unfolded image for all four databases.

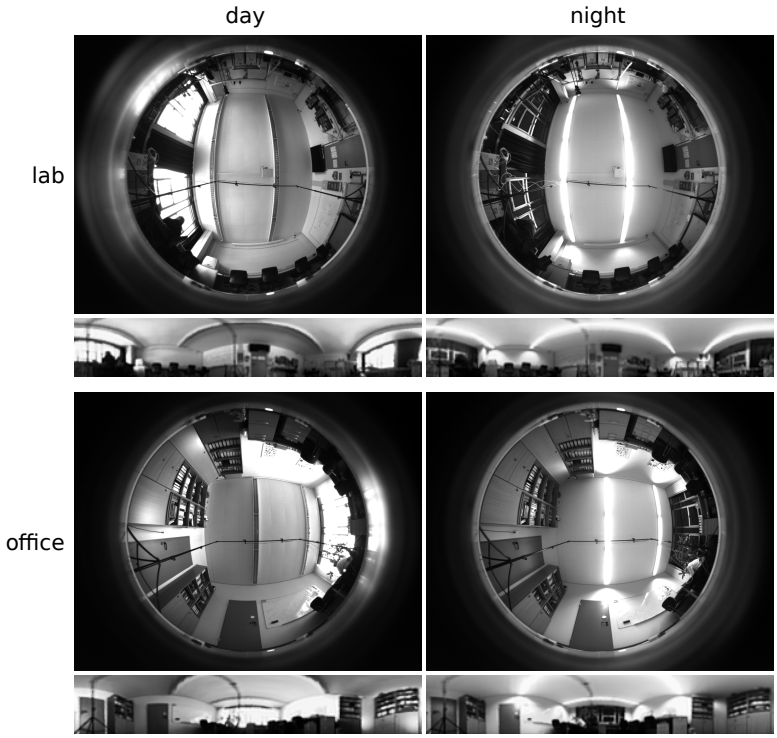

Figure 9. Example images (not to scale) from the lab and office day and night databases. The original camera images (1280 × 1024 pixels) are shown together with the corresponding preprocessed images (288 × 48 pixels, vertical elevation range 0°–75°).

3.1.2. Quorum V Database

The Quorum V database [48] was captured in a corridor and five adjacent rooms (Figure 10) with a camera looking at a hyperbolic mirror. A total of 872 images were captured on a regularly spaced grid with a resolution of 0.4 m. The camera images have a resolution of 402 × 402 pixels. Unfolded

panoramic images with size 512 × 128 were already provided on the website. Without any information on the exact mapping, the location of the horizon in the images was estimated at row 58. They were converted to gray scale, but no further preprocessing was applied.

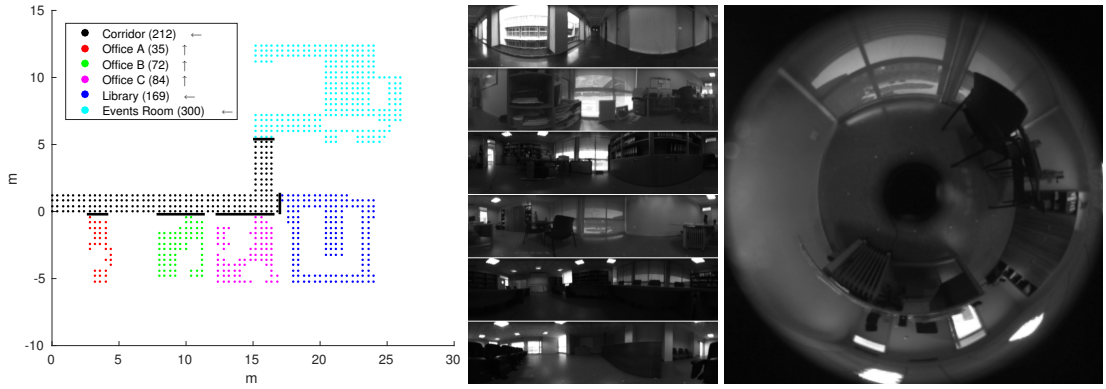

Figure 10. Layout and example images of the Quorum V dataset. The spacing of the image grid is 0.4 m. Each room is shown in a different color, walls containing doorways between rooms are highlighted with thick lines. The number of images and their orientation is also given in the legend: Up for the three offices, left for the remaining rooms. The order of the images in the middle is the same as in the legend. An original camera image with the hyperbolic mirror in the center is shown to the right for the image from *Office C*.

3.1.3. CITEC Database

We included a cleaning-robot run in the CITEC research apartment to have a realistic dataset in addition to the grid databases. The robot was controlled by our coverage strategy, which is an extension of [10], and built a topometric map of the environment. The area was covered by multiple parts with parallel meandering lanes in 0.3 m lane distance; the nodes (each containing a panoramic image) along each lane are about 0.1 m apart. While the final map is not metrically accurate, the general layout of the rooms is captured well enough to allow the evaluation on this dataset by taking estimated positions as ground truth (Figure 11). 750 images were captured during the cleaning run using the same equipment as described in Section 3.1.1. Binning to 640 × 512 pixels was done by the camera itself, preprocessing included unfolding to panoramic images (320 × 48 pixels), a binomial filter, and histogram equalization. The vertical elevation angle is 0°–75°, the horizon is at row 47.

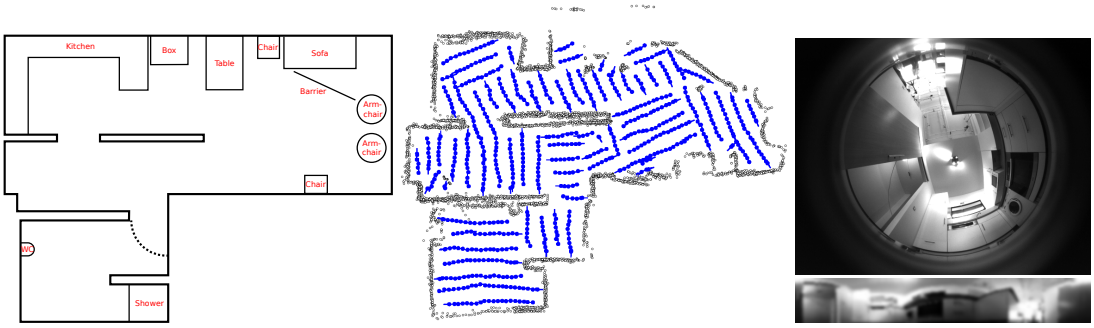

Figure 11. Layout of the CITEC research apartment (**left**), map built by the robot (**middle**), and example image from the kitchen (**right**). The blue markers in the map indicate nodes of the topometric map where images were captured. Obstacle measurements are shown as small black circles. The fisheye image is of size 640 × 512 pixels, the unfolded panoramic image is 320 × 48 pixels, vertical elevation angle 0°–75°.

3.1.4. Outdoor Database

A small outdoor database with 84 images was captured next to the main building of Bielefeld University using an UI-1240LE-C-HQ color camera (IDS Imaging Development Systems, Obersulm, Germany) and a Lensagon CSF5M1414 fisheye lens (Lensation, Karlsruhe, Germany) with opening angle of 182° [49]. Images with a resolution of 1280 × 1024 pixels were captured on a regular grid of 4 × 21 positions, about 1.5 m apart. Usually strong illumination changes in different image regions are present in outdoor images (for example, direct sunlight vs. shadows). To compensate for this, we computed high dynamic range (HDR) images using seven exposure times per grid position [50]. Image preprocessing on the HDR images includes conversion to gray scale, low-pass filtering using a third order Butterworth filter with relative cut-off frequency of 0.1, image unfolding to a spherical panoramic image (288 × 73 pixels; vertical opening angle −1°–90°; horizon in row 71), and histogram equalization.

An *inertial measurement unit (IMU)* (MTi 30 AHRS (Xsens, Enschede, Netherlands)) recorded the orientation of the robot, as uneven ground is more common in outdoor settings than indoor rooms. The visual compass methods presented in this paper assume all images to have the same horizontal plane, without any tilt. Thus, roll and pitch of the robot—measured by the IMU—were compensated during the image unfolding. The mapping generated to untilt an image may reference pixels in the fisheye image without actual image content. A mask can be generated to identify these pixels in the unfolded image (see Figure 12). The unfolded image contains black pixels at these positions.

While these masks should be considered in the image matching process, we currently ignore them and just compare the whole images. Using edge-filtered images may introduce a strong edge at the bottom of an image, but as only a small number of pixels is affected in an image pair, we expect no strong influence. A more elaborate evaluation on outdoor images should take these effects into account.

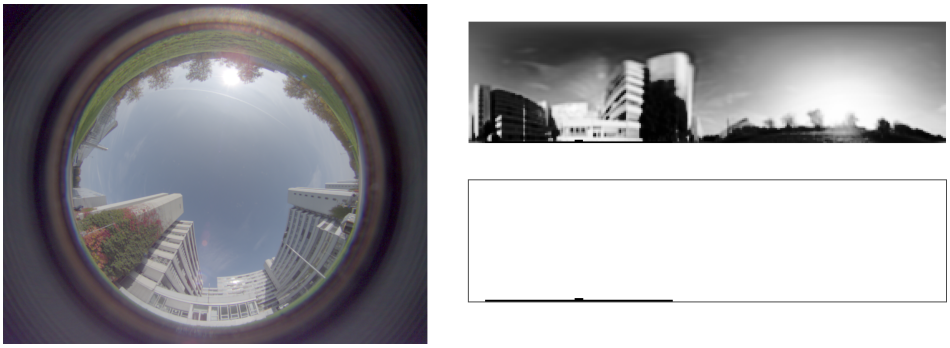

Figure 12. Example high dynamic range (HDR) image (1280 × 1024 pixels) of the outdoor database captured next to the main building of Bielefeld University, and preprocessed unfolded image (288 × 73 pixels). The bottom right image shows a mask with invalid pixel values (black) due to tilt correction. The corresponding pixels in the unfolded image are also black.

For this study, we used the same image comparisons as for indoor scenes. Different measures, for example color-based edge filtering, may be beneficial, but are not investigated in this paper.

3.2. Experimental Setups

In this section, we describe the database pairings and image selection. The training and recall phases of FabMap are also explained.

3.2.1. Database Tests

We used three kinds of tests to evaluate the dissimilarity measures described in Section 2: First, images within a single database are compared with each other (*same-database tests*). All four grid databases, as well as the Quorum V, CITEC, and outdoor databases, are used for

this evaluation. Dissimilarity values for the grid databases are combined into one large dataset. Second, we include *cross-database tests*, where image pairs are taken from databases captured in the same environment, but under different illumination conditions. We use the grid day and night database pairs for this test. We include both day/night and night/day image pairs, because we saw varying results when exchanging the images in a study about visual homing [46]. Third, we also present results of *mixed-database tests* obtained from combining the dissimilarity values for same-database and cross-database tests of the four grid databases to gain insights on the general applicability of the methods. This should show whether a common threshold can be used for successful place recognition both in same-database and cross-database tests. Images captured under different illumination conditions can appear within a single map in long-term applications.

3.2.2. Image Selection

To verify rotation invariance of methods for place recognition, images captured with a single orientation could be shifted randomly in horizontal direction. However, physically rotating the robot instead of simulating the shift has certain advantages: The resulting images are more realistic, as effects like possible small tilts of the camera and image noise appear in the final images. Shifting assumes an ideally calibrated system with the optical axis perpendicular to the ground plane to correspond to an actual rotation of the camera. Within the four grid databases, all images with orientation 270° (as snapshots) were paired with all images of orientation 180° (as current view), images at the same grid position were excluded in the same-database tests. For the Quorum V and CITEC datasets, each image is compared to all other images in the database.

All images from a single room have the same orientation in the Quorum V dataset, so the relative orientations of the image pairs can only vary when images come from different rooms. For the CITEC map, images on a single lane have about the same orientation, while images taken from adjacent lanes have opposite orientations. Additional parts of meandering lanes were usually attached perpendicular to previous parts. Thus, four main orientations (horizontally and vertically, bidirectional) can be identified due to the map building process, providing a variety of relative image orientations.

For the ABLE signature method on the grid databases, the images were unfolded to a size of 640×128 pixels instead of 288×48 pixels. This was done to avoid upscaling of image regions due to the chosen patch size of 64×64 pixels, which may have unknown effects on the results. The images originally used by the authors were much larger than our low-resolution panoramic images (between 640×256 and 1920×1080 pixels according to [39]). We tested 1, 4, 6, and 8 regions per image. As our images are rather small, we omit the subdivision of the patches into a regular grid, but use only a single descriptor per region.

3.2.3. FabMap Training and Recall

For FabMap we trained the bag-of-words vocabulary and the Chow–Liu tree on the four grid databases, the CITEC database, and the Quorum V database. The original fisheye or hyperbolic camera images were used, together with a mask selecting the regions containing actual visual information (removing the outer black region, and the mirror in the center of the Quorum V images). The day grid databases were only used with an orientation of 270°, the night databases with 180° to reduce the number of training images. 40% of the images of each database were randomly selected for training (952 of 2390 images in total), the vocabulary size was chosen as 10,000. The evaluation was done for each of these databases separately.

The small outdoor database was not included, as it is only used as a preliminary study. Including the outdoor images into the FabMap training seemed inappropriate for this reason.

3.2.4. Parameter Settings for Methods

The ratio postprocessing methods require a set of neighbors to be selected. We chose a metric distance for this: 0.3 m for the grid databases (3 to 12 neighbors, up to two grid positions away); 0.85 m

for Quorum V (2 to 12 neighbors, up to two grid positions away); 0.4 m for CITEC (7 to 20 neighbors, depending on the graph structure); 2.5 m for outdoor (8 direct neighbors). If the method is applied on a robot that builds a topological or topometric map, neighbors could also be selected based on connectivity of nodes (for example, up to two links in a topometric map).

We use use 1, 3, and 5 scale planes for the visual compass, 9 for `warping`. ABLE was tested with 1, 4, 6, and 8 regions.

See Appendix A for additional parameter settings of the different methods.

3.3. Evaluation Criteria

Together with their metric position, all dissimilarity values obtained for a single image form a *dissimilarity profile*. As exact metric positions may not be available for many applications, the goal is to detect images captured at nearby places based on thresholding the dissimilarity profile of an image. Best matches should be detected for image pairs that are metrically close. The actual metric distance that is considered "close" (*acceptance radius*) depends on the experimental conditions. For example, our grid databases have a spacing of 0.15 m, while the images in the outdoor database were captured at 1.5 m intervals. We chose 0.5 m for the grid (spacing: 0.15 m) and CITEC (spacing 0.1 m along lanes, 0.3 m between lanes) databases. The spacing between image capture locations in the Quorum V database is 0.4 m, so we chose 0.6 m to accept diagonal neighbors as image matches. Similarly, we chose 2.5 m for the outdoor database (spacing about 1.5 m). Depending on the task at hand, it may be necessary to detect only the best match (e.g., loop closure) or multiple matches (e.g., reference nodes for triangulation).

We use *ROC analysis (receiver-operator characteristics)* and *area-under-curve values (AUC)* [51] to determine if thresholding can be applied for place recognition. We also determine the metric distance between the current image location and the position of the best-matching reference image. As long as the measure increases for larger distances between image pairs, as observed in previous studies [13], the distance to the best-matching image should be small. Large distances to best-matching images indicate methods that are not suitable for place recognition.

3.3.1. ROC and AUC

As low dissimilarity values should correspond to close image pairs, a higher threshold increases the number of true positive matches, but may also lead to a larger number of false positives. The final choice of a threshold is usually a trade-off to achieve high true-positive rates with only few false positives. The number of acceptable false positives depends on the requirements of the actual application: Early SLAM systems, for example, relied on the absence of false positives (which could irreversibly break the whole map), while newer methods can tolerate or reject them [6–8]. The identified image matches may also be used in further processing stages, for example to triangulate a position estimate or to insert links into a topological map. Mismatches may also be rejected in such a later stage.

We use ROC analysis to evaluate the different dissimilarity measures according to their classification ability. This includes plots of the true-positive rate against the false-positive rate for varying classification thresholds (ROC curve) and AUC values. Images within the acceptance radius (based on known ground-truth positions) are considered as actual positive matches to evaluate the classifier performance. The AUC is the area between the ROC curve and the horizontal axis and can be used as a performance measure for the classifier. For a perfect classifier it would be 1.0, a random guess (based on the sizes of the classes) would result in 0.5. Classifiers with AUC values lower than 0.5 could be inverted to get a better classifier. ROC curves can eventually be used to select a threshold for the image dissimilarity values that gives a satisfying classification (acceptably low false-positive rate for sufficiently high true-positive rate).

Precision-recall curves are often used in place recognition papers [12], but they depend on the number of true and false place matches. The number of true matches is usually small compared

to non-matching locations. ROC curves are independent of the absolute number of samples, as no measure depending on both true and false samples is involved.

3.3.2. Distance to Minimum

We also investigate whether the best-matching reference image is actually a neighbor of the current image within the acceptance radius. If this is not the case, the assumption that the image dissimilarity function increases with larger distances is violated. We plot the median and 95th percentile of the metric distance between each current image location and the corresponding best matching image location (MINDIST). Outliers larger than the acceptance radius are shown individually.

3.3.3. Time Measurements

We compare the computation times for the different dissimilarity measures. Times were measured on a high-speed i7 CPU (i7-4790K, 4 GHz, (Intel, Santa Clara, CA, USA)), as well as an embedded Atom CPU (N2600, 1.6 GHz, (Intel, Santa Clara, CA, USA)). The Atom processor is the same model as the one used on our robot platform.

We used an optimized streaming SIMD extensions (SSE) implementation using integer computations for Min-Warping and the holistic compass measures. Integer multiplications in general increase the bit width of the result. This case is not covered by our SSE implementation, so we use floating-point values for the SSD measure to avoid additional data type conversions. We use a configuration of Min-Warping giving the most exact results, but it can be accelerated using heuristics, e.g., a compass estimate based on the visual compass (see [11] for details).

The grid and Quorum V databases were used for timing experiments to include a range of different image sizes. For holistic methods and `sigafc`, the unfolded image sizes were 288×48 (grid), and 512×128 (Quorum V). FabMap used original camera images with size 1280×1024 (grid) and 402×402 (Quorum V). Unfolded images for ABLE were 640×128 (grid), and 512×128 (Quorum V). The CITEC and outdoor images have intermediate sizes.

The time to preprocess the images (including low pass filter and unfolding) is not measured, as we assume these images are available on the robot (e.g., for other navigation tasks).

4. Experimental Results

We present the results of same-database tests for all four grid databases, the Quorum V dataset, the CITEC map, and the small outdoor database. Cross-database tests to examine tolerance against strong illumination changes are provided for the grid databases only. Finally, we show the results of mixed-database tests using grid same and cross databases combined. We present the *Holistic Methods* (visual compass and `warping`) together. ABLE signatures with varying number of regions, Fourier signatures, and FabMap are combined as *Feature and Signature Methods*.

4.1. Grid Same-Database Experiments

4.1.1. Holistic Methods

Figure 8 (left) in Section 2.4 shows the dissimilarity values obtained on the lab day same-database test (red) and lab day/night cross-database test (green, see Section 4.5 for details) using `e-nsad-1`, plotted against the metric distance of the corresponding image pairs. The results for lab day show a clear minimum for close image pairs, indicating this is a suitable dissimilarity measure for place recognition. The profiles for the other measures and databases look similar, as is to be expected from previous studies on the visual compass.

Figure 13 shows the results for all four grid databases combined into one dataset. The columns show the ROC curves, AUC values, and MINDIST values; the rows correspond to *raw values*, *mean ratio*, and *min ratio* postprocessing. All methods achieve high AUC values, so most image pairs can be classified correctly. The minimum of the spatial dissimilarity profile is also always close

to the current image, as can be seen by the plot of the distances (the grid spacing is 0.15 m, so no smaller values are possible). Using multiple scale planes improves the result, with three being slightly better than five on this database. The best matching image is always found close to the current view for all methods. With an AUC value of 0.9 and the lowest ROC curve, `warping` is worse than the compass methods. The ratio postprocessing methods further improve the ROC curves and AUC values. The ROC curves of the compass methods improve more from the min ratio postprocessing, while the other measures benefit more from the mean. Although the MINDIST values are higher, most best matches are still in close vicinity to the current image location. None of these methods produces any outlier for MINDIST beyond the acceptance radius.

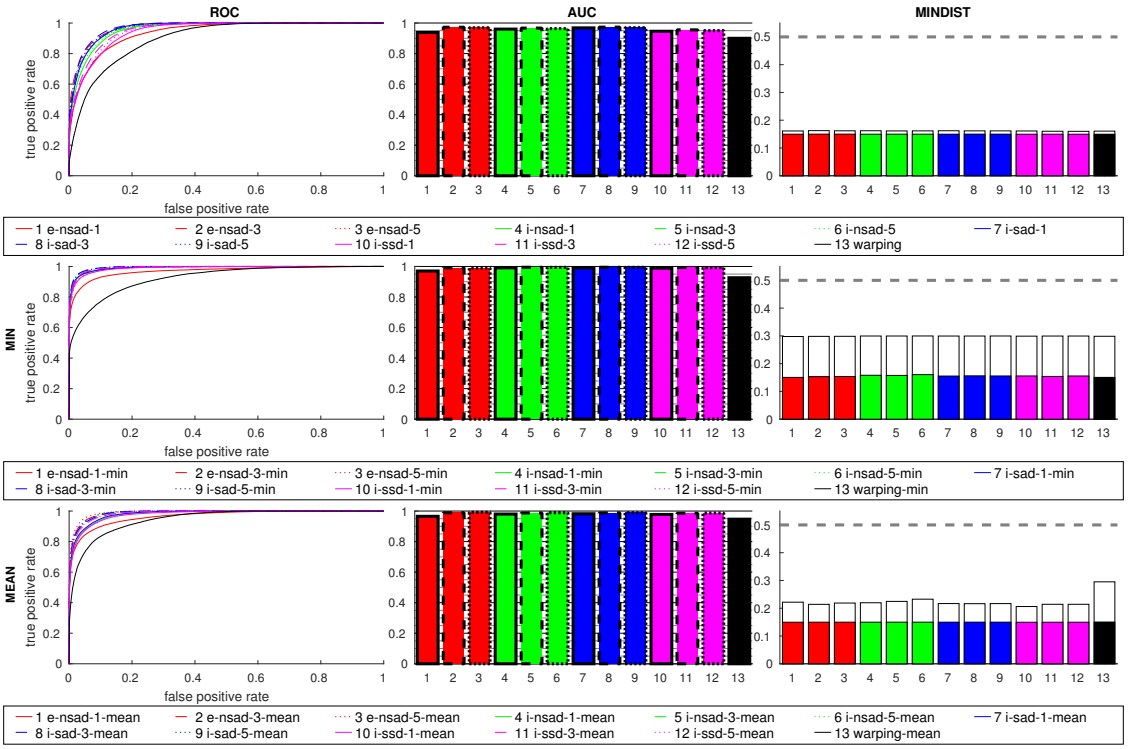

Figure 13. Results for all grid same-database tests using the visual compass method and `warping`. Each distance measure is presented in a different color, the line styles indicate the number of scale planes used (except black for `warping`, which always uses 9 scale planes). The top row shows the results for the dissimilarity values without further postprocessing, the middle row shows the min ratio postprocessing, the bottom row the mean ratio postprocessing method. The first column shows ROC curves, plotting the true-positive rate against the false-positive rate for varying classification thresholds. Curves near the top left corner of the plot indicate better methods, a curve near the diagonal performs like a classifier based on random guessing. The second column shows bar plots of the AUC values for the corresponding ROC curves. Values closer to 1.0 are better. The right column shows the metric distance to the best matching reference image for each current view (lower is better). The top of the colored box shows the median, the white box extends up to the 95th percentile of the values. Individual outliers beyond the acceptance radius (dashed horizontal line) are shown as well (but only appear in later figures).

4.1.2. Influence of the Low-Pass Filter

We also investigated the influence of the low-pass filter using the lab grid same-database test. We tested values of 0.04, 0.08, 0.10, 0.15, 0.20, and 0.25 for the Butterworth cut-off frequency, and no filter at all. We only computed the raw dissimilarity values, without the ratio postprocessing. The AUC values are always close to 1.0 for all holistic measures, as well as for `sigafc` (data not shown). So

for same-database tests, the cut-off has no significant influence on the results. This supports our decision to omit further preprocessing on the Quorum V images.

4.1.3. Feature and Signature Methods

Figure 14 shows the results using features and signature methods. FabMap used the SIFT, ORB and BRISK descriptors, while ABLE used LDB and BRISK. The number of regions for ABLE included 1, 4, 6, and 8. As the relative orientation between the images is 90°, the best results can be expected for 4 or 8 regions. This is clearly shown in the plots, where 1 and 6 regions perform poorly. This indicates that a suitable number of regions has to be selected for the prevalent rotations. The BRISK descriptor performs worse than LDB.

All FabMap variants perform similar to the two best ABLE-LDB results, `sigafc` is slightly worse. All best-matching images for these six methods, which all have high AUC values, fall within the acceptance radius, while the remaining methods are considerably worse.

Using the ratio postprocessing shows only small improvements for 4 and 8 regions with LDB, while BRISK benefits more: For example, for `ableBRISK4` the AUC value rises from 0.72 to 0.85 with the min ratio. The MINDIST values again rise for the ratio postprocessing methods, the overall effect on FabMap and `sigafc` is small, introducing some outliers.

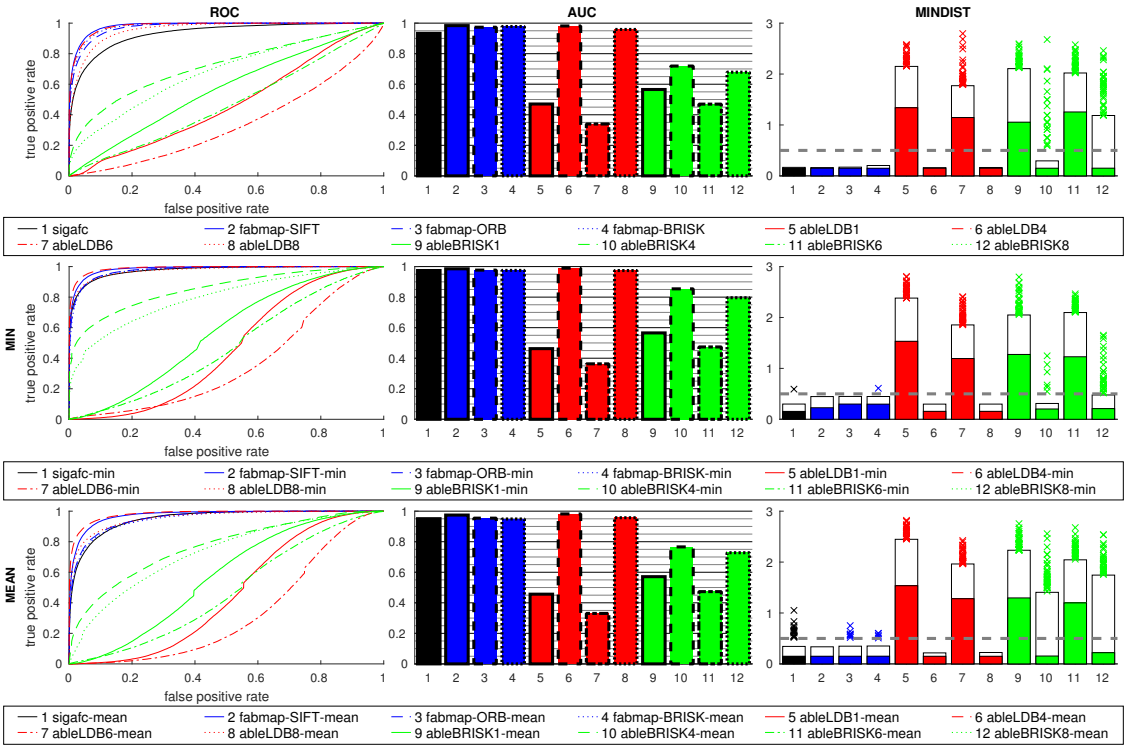

Figure 14. Results for all grid same-database tests using `sigafc` (black), FabMap (blue), and ABLE signatures with LDB (red) and BRISK (green) descriptors. 1, 4, 6, and 8 regions were used for ABLE (indicated by line style). See Figure 13 for details on the plots.

4.2. Quorum V Database

4.2.1. Holistic Methods

Figure 15 shows the results for the Quorum V database. All methods have high AUC values, but the MINDIST measure shows many large distance outliers. For example, about 83.9% of all 872 images can be matched successfully using `e-nsad-1`. Most mismatches occur in the corridor part of the dataset, which has a very repetitive structure: Large window fronts and a wall with doors lead to

similar image content in different places. Figure 16 (left) shows two example images from the corridor that were captured 2.4 m apart and are nearly indiscernible. The `e-nsad-1` image dissimilarity profile for the corridor (Figure 17) clearly indicates multiple similar locations; the intensity-based methods have the same problem. Omitting the corridor from the dataset allows to match 98.6% of the remaining 660 images, again with `e-nsad-1`. The largest remaining mismatch using this method occurs at the doors of offices A and C; the corresponding images are shown in Figure 16 (right). The results for all methods on the reduced dataset without the corridor are shown in Figure 18. All methods improve when omitting the corridor and the number of outliers in MINDIST is clearly reduced.

Applying the ratio postprocessing increases the AUC values, but some larger outliers are introduced. Especially for `i-ssd` using the mean postprocessing increases the median MINDIST in the reduced dataset. On the full Quorum V database, larger MINDIST values appear for all methods.

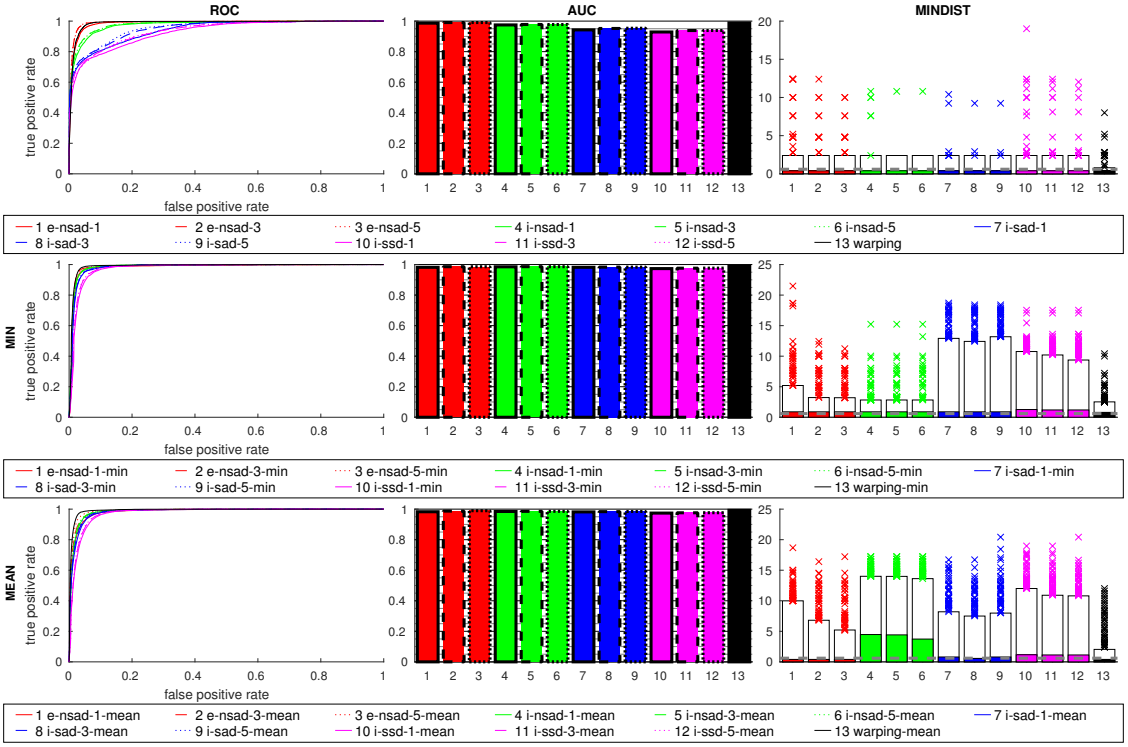

Figure 15. Results for the full Quorum V database using the visual compass method and `warping`. See Figure 13 for details.

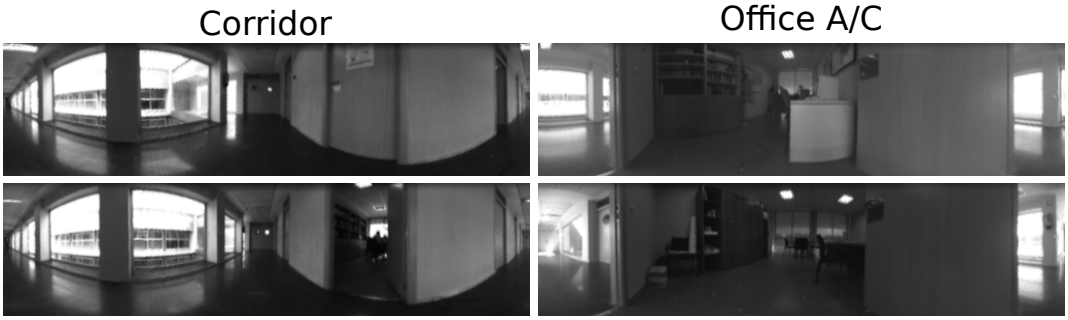

Figure 16. Problematic images in the Quorum V dataset. **(Left):** the corridor has a very repetitive structure, with doors and large window fronts. The two images were actually captured about 2.4 m apart. **(Right):** Images captured near the doorways of Office A (**top**) and Office C (**bottom**). These places are incorrectly identified as the same place using the `e-nsad-1` method.

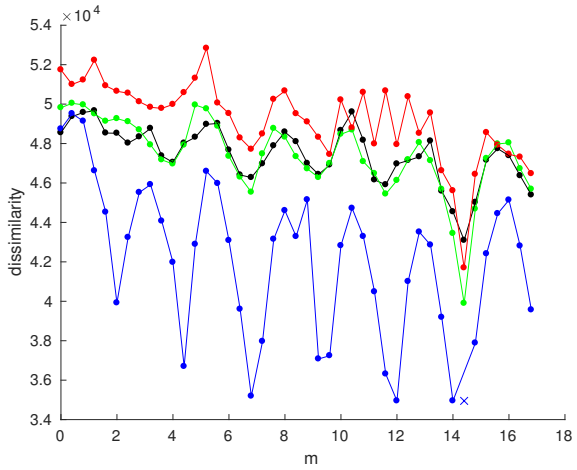

Figure 17. Side view of the `e-nsad-1` dissimilarity profile for an image in the long part of the corridor of the Quorum V dataset. The four parallel "lanes" are shown in different colors (black, green, blue, red for increasing vertical coordinates in Figure 10). Multiple local minima are clearly visible, due to the repetitive structure of the environment. The position of the current view is marked with a cross.

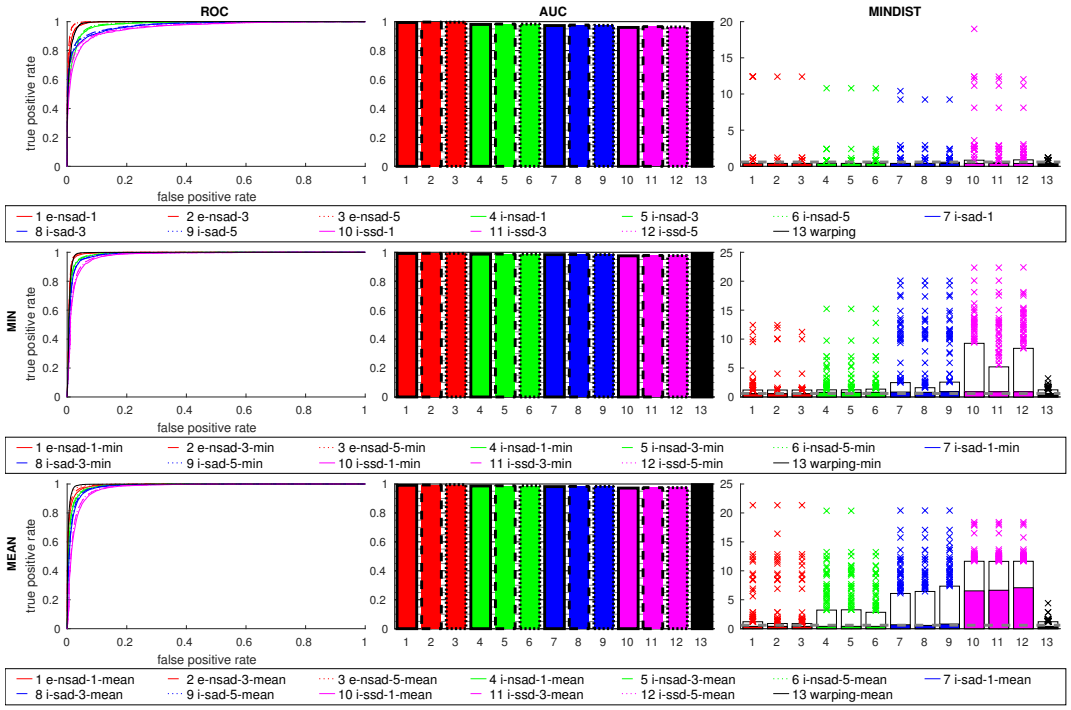

Figure 18. Results for the rooms of the Quorum V database, without the corridor, using the visual compass method and `warping`. See Figure 13 for details.

4.2.2. Feature and Signature Methods

Figure 19 shows the results for feature and signature methods on the full Quorum V database. ABLE using LDB performs very well, independent of the number of regions, with one region showing the least MINDIST values. Orientation mismatches between close image pairs only occur at the doorways, as all images within a single room have the same orientation, so rotation invariance of the method is not as relevant as for the grid databases. Omitting the corridor has only a small impact on the results (data not shown). Applying the ratio postprocessing slightly reduces the performance of the methods and again increases the number of outliers in MINDIST. BRISK is again worse than LDB; even more so when using multiple image regions.

FabMap using SIFT and ORB is again very good, while BRISK results are slightly worse, comparable to `sigafc`. For most methods, MINDIST values are higher than using holistic methods. However, FabMap with SIFT and ORB without the ratio postprocessing have lower 95th percentiles.

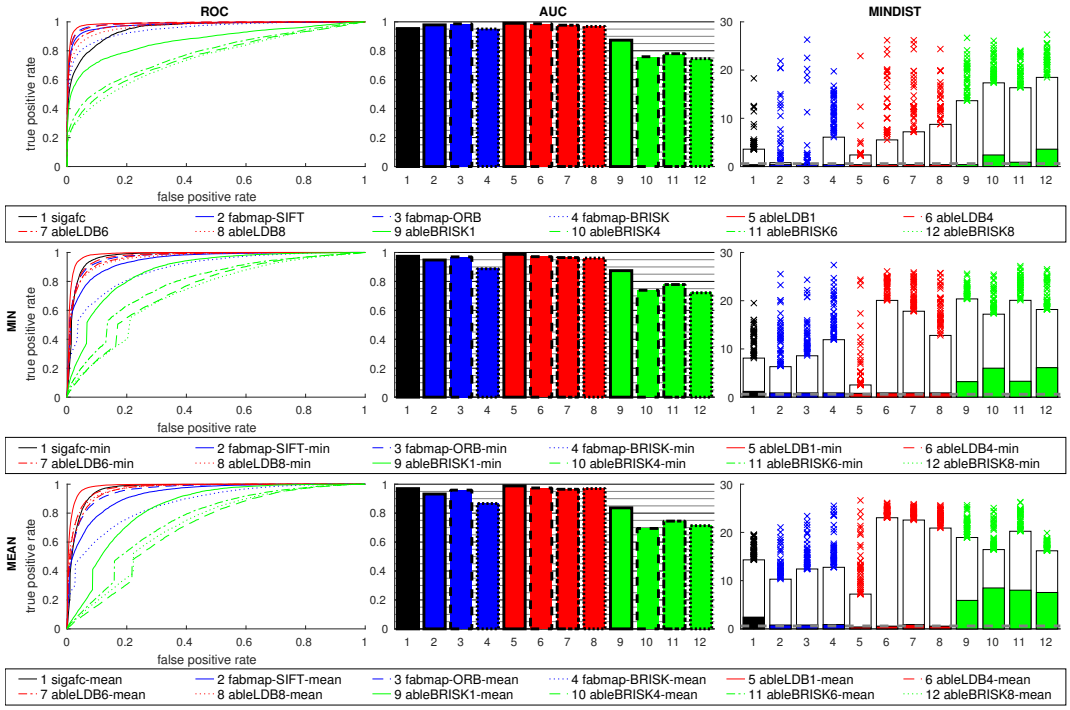

Figure 19. Results for the full Quorum V database using signatures and features. See Figures 13 and 14 for details.

4.3. CITEC Database

4.3.1. Holistic Methods

Figure 20 shows the results for the CITEC database. All methods perform well, like with the previous datasets (the minimal distance between images is 0.1 m on this database). Using the mean ratio postprocessing introduces many outliers in MINDIST, while the min ratio only adds some of them for the intensity-based methods. The ROC curves for the min ratio improve, but the overall results of the mean ratio are worse.

4.3.2. Feature and Signature Methods

Figure 21 shows the results for the CITEC database, using feature and signature methods. Using multiple regions for ABLE with LDB is again better than a single one, with four giving the best results. Although the orientations of the images are more random throughout the map, neighboring images often are oriented in the same or opposite direction, due to the meandering lane structure of the map. The min ratio greatly improves the AUC results for LDB for all tested numbers of regions, while the mean ratio mostly benefits the single region descriptor. BRISK as usual performs worse than LDB. While the ratio postprocessing increases AUC values, especially for `ableBRISK1-min`, MINDIST values increase as well.

FabMap is better than ABLE for this database, and it shows no outliers in MINDIST with the SIFT or ORB descriptor. The ratio postprocessing methods impact the results by increasing the outliers for BRISK (both min and mean) and SIFT (especially mean). `sigafc` also performs well, but using the ratio postprocessing method clearly increases the median of MINDIST.

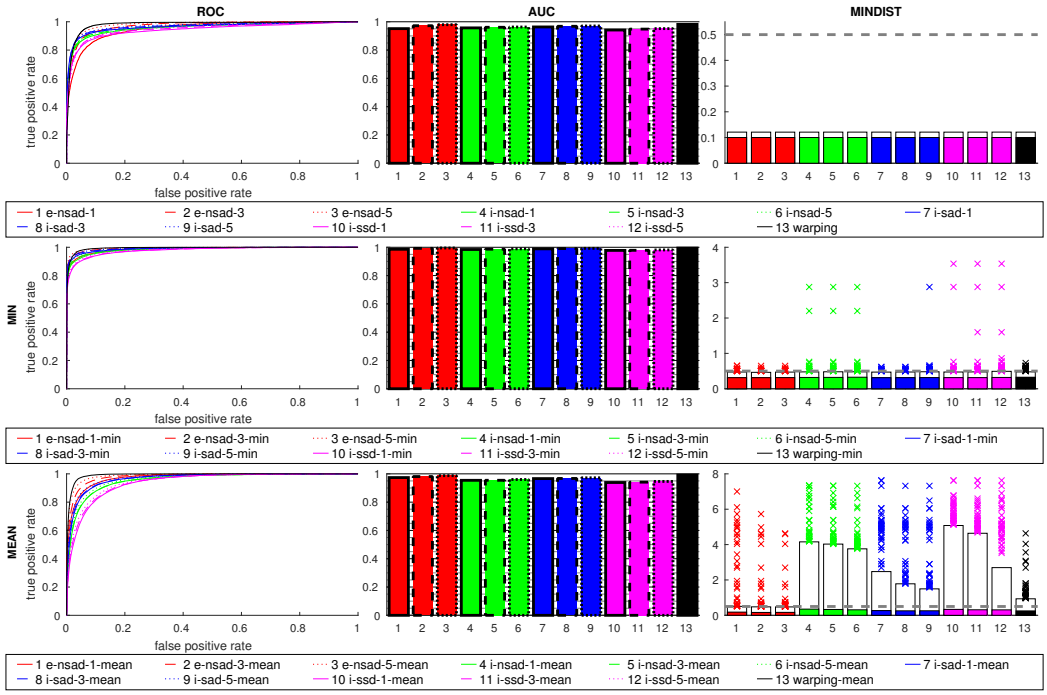

Figure 20. Results for the CITEC database using the visual compass method and `warping`. See Figure 13 for details.

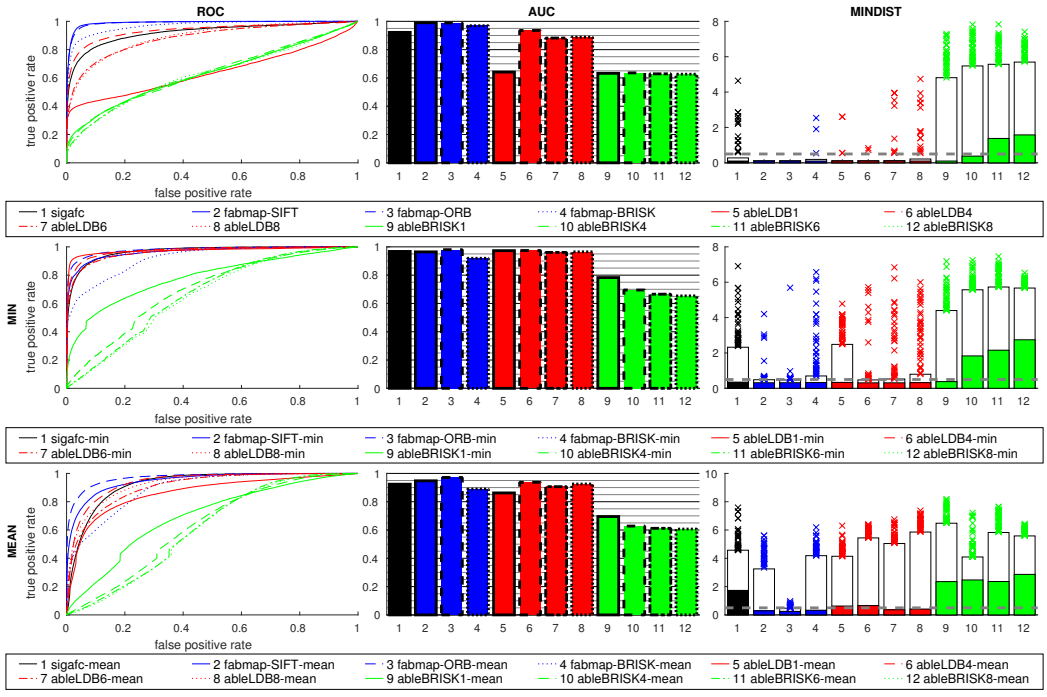

Figure 21. Results for the CITEC database using signatures and features. See Figures 13 and 14 for details.

4.4. Outdoor Database

4.4.1. Holistic Methods

Figure 22 shows the results for the outdoor database. All methods perform very well, with high AUC values and only few outliers in MINDIST. The ratio postprocessing again causes more outliers to appear, with only small impacts on AUC values.

4.4.2. Signature Methods

Figure 23 shows the results for ABLE and `sigafc` on the outdoor database. FabMap was not included, as we only used the indoor databases for training and the outdoor database is only used as a preliminary study. While LDB and `sigafc` perform similarly well as the holistic methods, BRISK results are again worse. The number of regions has no considerable impact on LDB, as all images have about the same orientation. However, results for BRISK differ, with one region giving the best result.

Computing the ratio postprocessing methods shows similar effects as in the previous datasets.

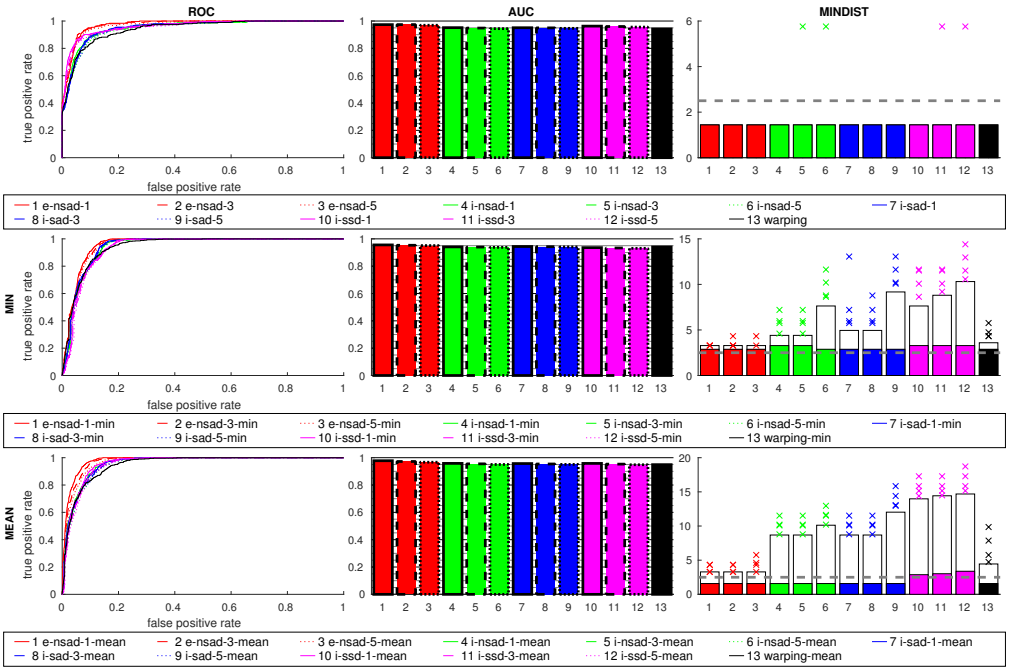

Figure 22. Results for the outdoor database using the visual compass method and `warping`. See Figure 13 for details.

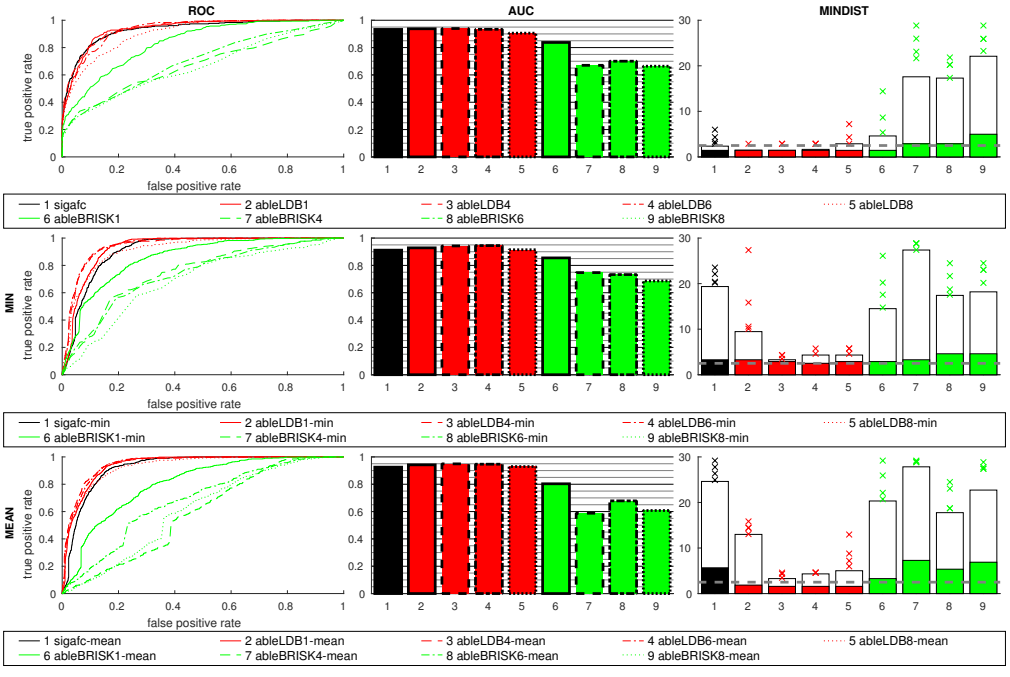

Figure 23. Results for the outdoor database using signatures. See Figures 13 and 14 for details.

4.5. Grid Cross-Database Experiments

4.5.1. Holistic Methods

The dissimilarity values for the lab day/night database (green in Figure 8 (left)) show a much more shallow minimum than for the same-database test. The more indistinct minimum poses a challenge to a threshold-based classification which is confirmed by the results.

Figure 24 shows the results for the cross-database experiments. The `e-nsad` measure is clearly better than all other methods, which perform worse than in the same-database tests. While a small improvement occurs for `e-nsad` using more scale planes, this does not generally hold for the intensity-based measures. The median MINDIST for `i-nsad` and `i-sad` are small, but the 95th percentile is very high. `warping` lies between `e-nsad` and the intensity-based methods. This indicates the advantage of using edge-filtered images for illumination-tolerant image matching. The mean ratio causes worse results for all methods, while the min ratio slightly improves the AUC values. The MINDIST values are again increased for most methods, but outliers are reduced for `warping` using the min ratio postprocessing.

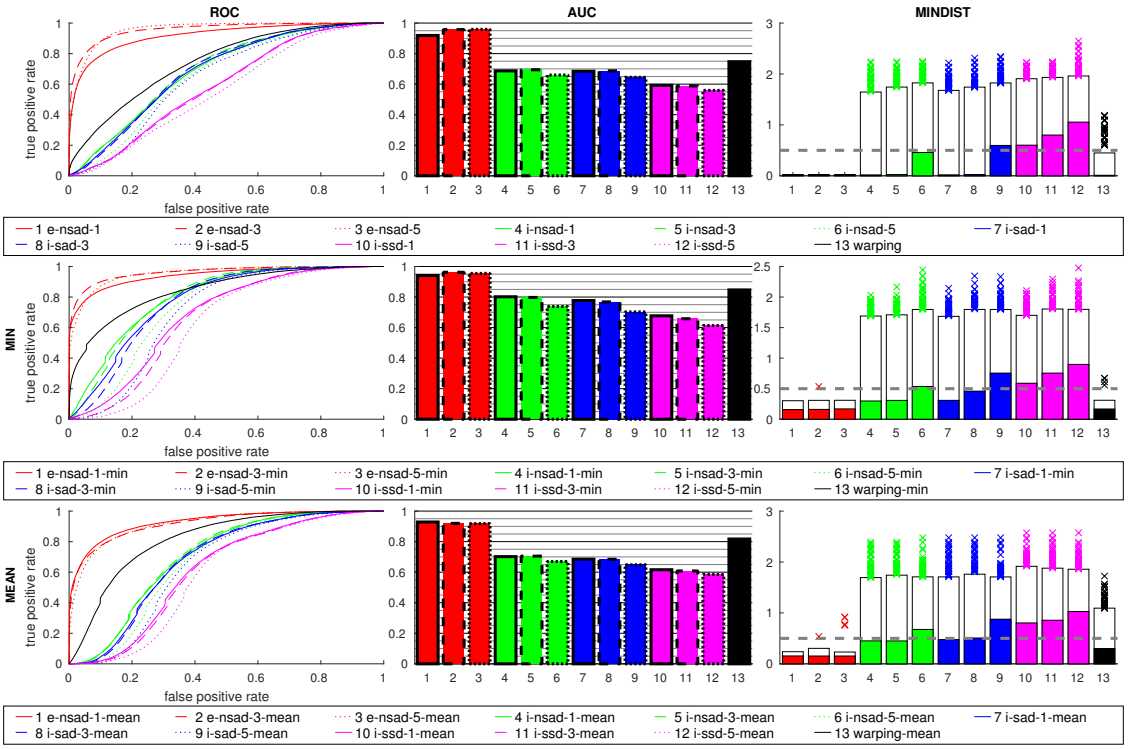

Figure 24. Results for the cross-database tests using the visual compass method and `warping`. See Figure 13 for details.

Figure 25 shows AUC and MINDIST values for the individual cross databases. An influence of the day-night/night-day pairing can be seen especially for the intensity-based methods. The snapshots were always taken from orientation 270° and current views from orientation 180° (effectively changing the orientations within the image pairs). Intensity-based methods perform especially poorly on the lab cross-database, while the results for `e-nsad-1` are very good. The ceiling lamps in the night images feature prominently (Figure 26) and appear very different in the day images. Such strong changes pose a challenge for intensity-based methods. Reducing the elevation angle of the image mapping from 75° to 48° improves the performance of these methods (Figure 27).

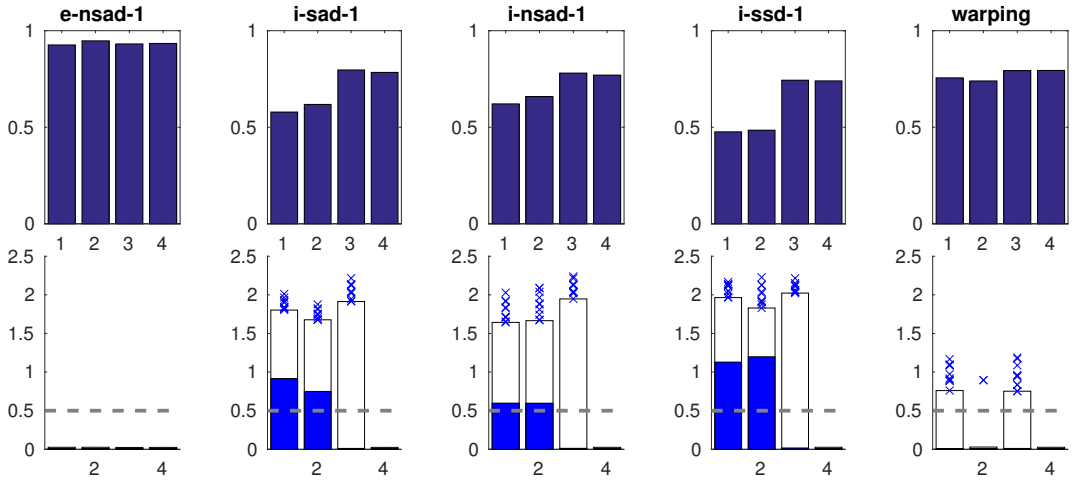

Figure 25. Plots of the AUC values (top row) and MINDIST (bottom row) for the individual cross-database tests (1: lab day/night; 2: lab night/day; 3: office day/night; 4: office night/day). The results on the lab database are noticeably worse than on the office databases, especially using intensity-based measures. An influence of the day/night or night/day pairing can also be seen.

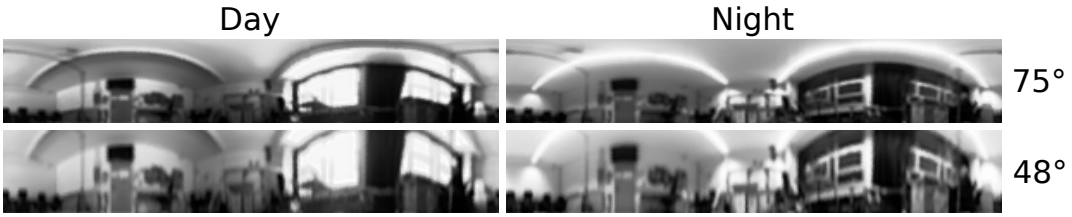

Figure 26. Example images from the lab database at day and night for maximum elevation angles of the image mapping of 75° and 48°. While the ceiling lamps feature prominently in the high-elevation images, they are less pronounced with the reduced elevation angle.

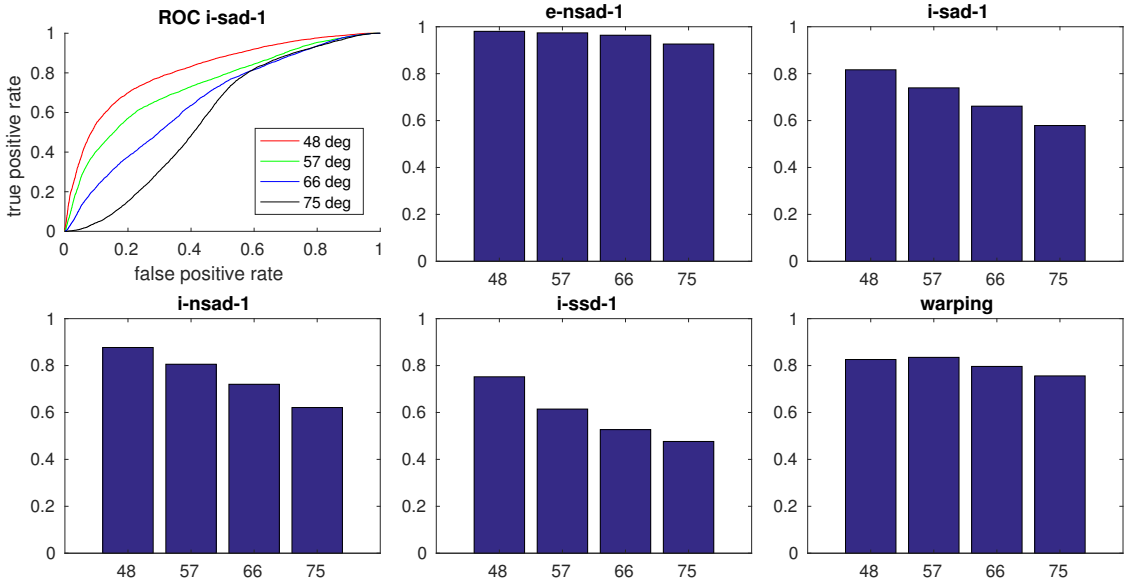

Figure 27. Exemplary ROC curve for `i-sad-1` and AUC values for all holistic methods with different elevation angles on the lab day/night database.

4.5.2. Influence of the Low-Pass Filter

Larger relative cut-off frequencies for the Butterworth filter (weaker filtering) show small improvements on cross-databases. The AUC values lie between 0.4 and 0.65 for most intensity-based measures. The values for `e-nsad-1` are consistently above 0.85, increasing with higher frequencies, only dropping down to 0.6 for a very strong low-pass filter with relative cut-off at 0.04 (data not shown).

4.5.3. Feature and Signature Methods

Figure 28 shows the results for cross-database tests using feature and signature methods. Results are close to chance level for most methods, the choice of the descriptor for ABLE has little influence, the results are similarly poor. FabMap using SIFT is better than the other methods, and improves again using the ratio postprocessing. The ratios for the remaining methods show no considerable improvement. Results for `sigafc` are similar.

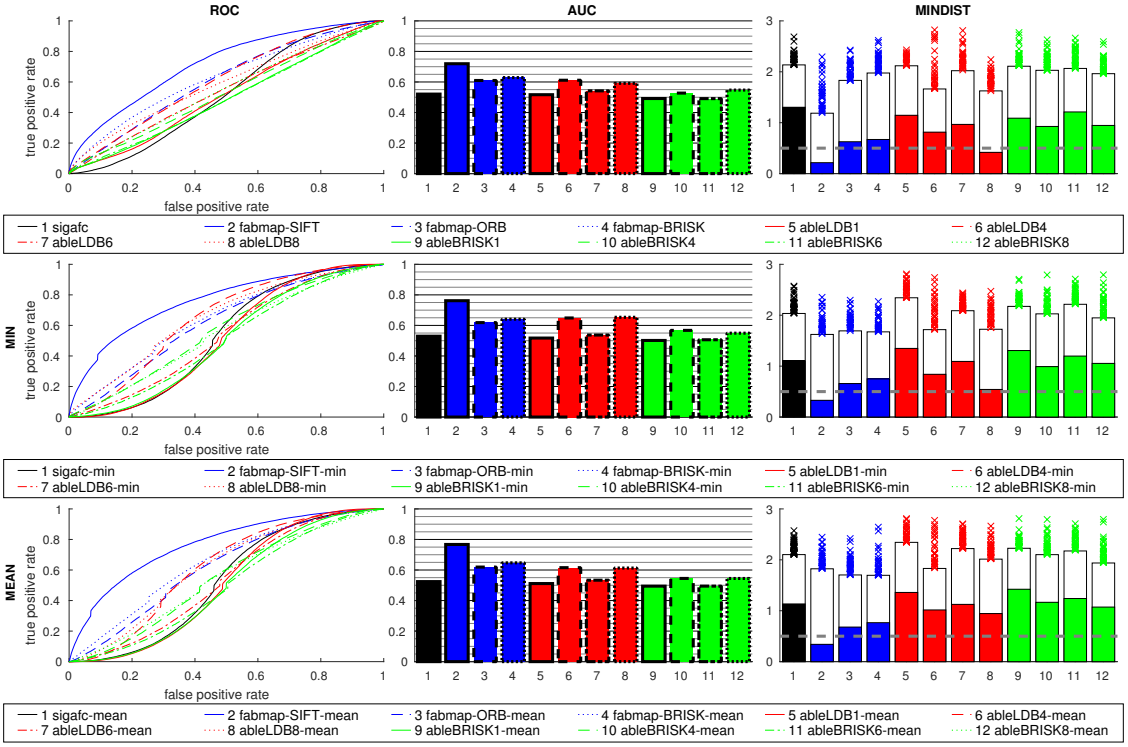

Figure 28. Results for the cross-database tests using signatures and features. See Figures 13 and 14 for details.

4.6. Mixed-Database Experiments

4.6.1. Holistic Methods

Figure 29 shows the result for all four same and cross grid databases combined. The initial increase of image matches in the ROC curve comes from the same-database tests, as the dissimilarity values tend to be smaller than for cross-database image pairs (compare Figure 8 (left)). So a lower threshold is sufficient to select neighbors. The higher threshold required to distinguish image pairs captured under different illumination conditions leads to an increase in false positives detected on the same-database tests. The `e-nsad` measures again clearly outperform all other methods, similar to the cross-database tests. Using the ratio postprocessing methods generally improves the ROC curves. AUC values for `i-nsad` and `i-sad` are comparable to cross-database tests, `i-ssd` performs better.

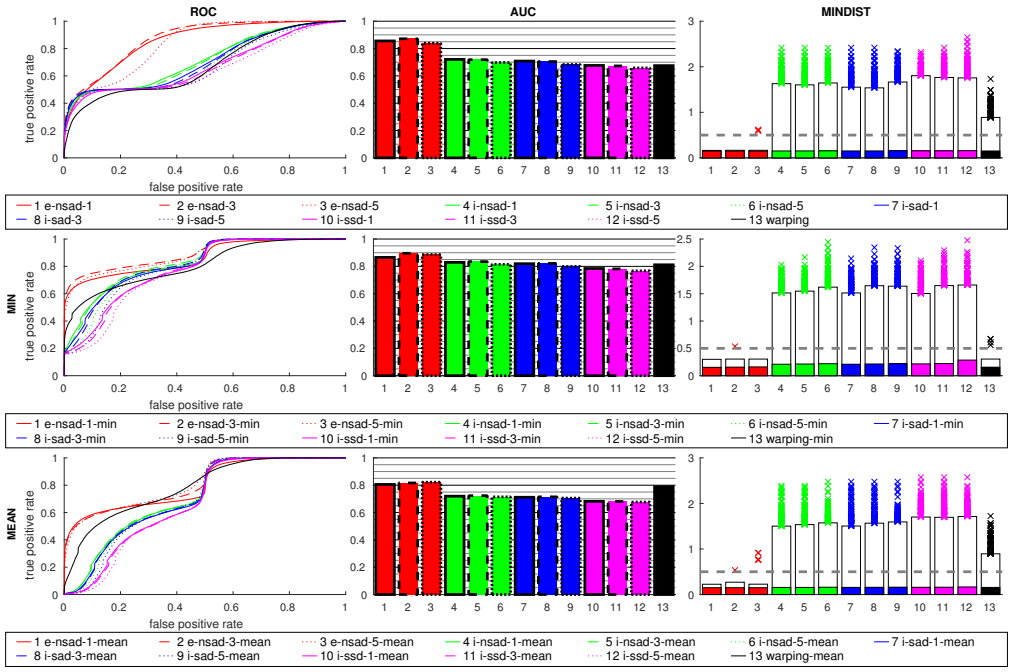

Figure 29. Results for the mixed-database tests using the visual compass method and `warping`. See Figure 13 for details.

4.6.2. Feature and Signature Methods

Figure 30 shows the results using feature and signature methods. The ROC curve of `sigafc` has a shape similar to most holistic methods shown in Figure 29. Using ABLE-signatures on both same and cross databases together also gives a mixed result. The true-positive rate increases up to about 0.4 before the false-positive rate starts to increase significantly for the LDB raw values. Using the ratios the false-positive rate starts to increase earlier, the mean ratio cannot identify any correct matches even for small thresholds. BRISK results are comparably poor.

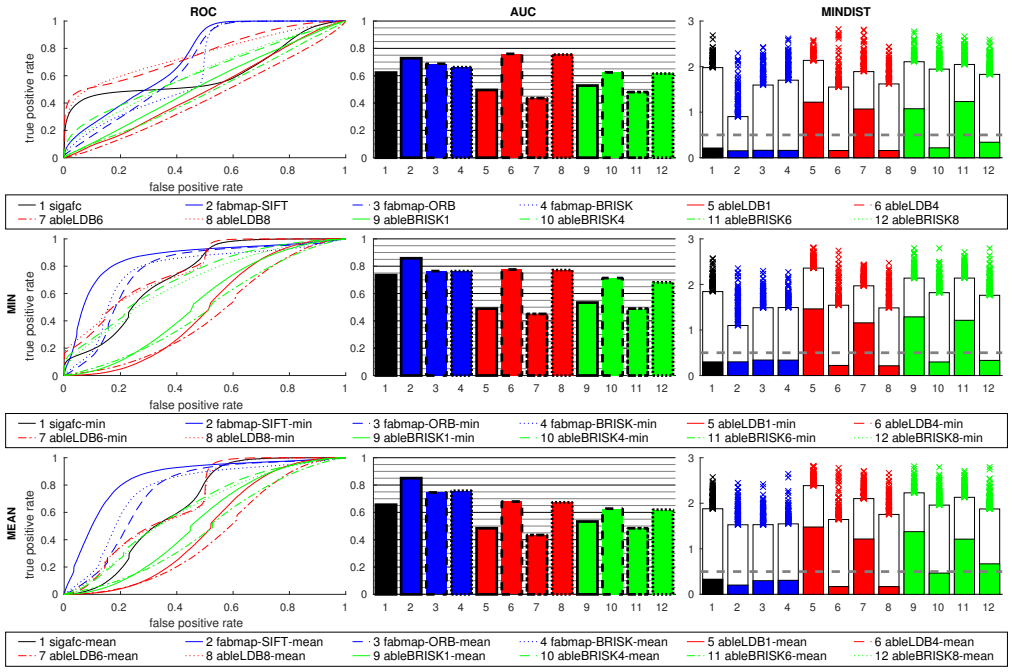

Figure 30. Results for the mixed-database tests using signatures and features. See Figures 13 and 14 for details.

FabMap with raw values is also not very good. Using the ratio postprocessing with ORB or BRISK first increases both the true-positive rate and the false-positive rate by similar amounts, but for larger thresholds the true-positive rate increases faster. SIFT again benefits more from the ratios, with min-ratio giving the best results on mixed-database tests with features. The AUC values for FabMap with SIFT are similar to the holistic `e-nsad` method.

While the median MINDIST is often small, the 95th percentile is always very high.

4.7. Speed

Tables 1–3 show the computation times (in milliseconds; with three significant digits, except for very small values) for holistic methods, FabMap, and signature-based methods, respectively. Each table shows the computation times on a high-end Intel i7 and an embedded Intel Atom processor, using the lab grid cross database and the Quorum V database. In all cases, the Atom CPU is considerably slower than the i7, by a factor of about 10 to 30.

Table 1. Average computation times per image pair (in ms) for the holistic methods on a grid and the Quorum V database on an Intel i7 and an Intel Atom CPU. Image sizes were 288 × 48 (grid) and 512 × 128 (Quorum V).

	Grid		**Quorum V**	
Method	**i7**	**Atom**	**i7**	**Atom**
e-nsad-1	0.258	4.91	2.33	35.7
e-nsad-3	0.727	14.1	6.04	93.6
e-nsad-5	1.19	23.4	9.72	151
i-nsad-1	0.261	4.87	2.32	33.4
i-nsad-3	0.737	14.2	6.03	92.0
i-nsad-5	1.21	23.6	9.72	150
i-sad-1	0.224	3.99	2.33	30.2
i-sad-3	0.629	11.4	6.04	82.7
i-sad-5	1.03	18.8	9.73	134
i-ssd-1	0.443	13.5	4.35	116
i-ssd-3	1.27	40.1	11.9	339
i-ssd-5	2.09	66.5	19.7	562
warping	13.3	317	39.5	967

Table 2. Average computation times (in ms) for FabMap using SIFT, ORB, and BRISK on a grid and the Quorum V database on an Intel i7 and an Intel Atom CPU. Feature extraction (Extract) and cluster association to the bag of words (Cluster) are preprocessing steps required once per image. Compare gives the average time for a single image match. Image sizes were 1280 × 1024 (grid) and 404 × 404 (Quorum V).

	Grid		**Quorum V**	
Method	**i7**	**Atom**	**i7**	**Atom**
SIFT Extract	268	3100	32.3	369
SIFT Cluster	286	3730	60.2	797
SIFT Compare	0.259	2.07	0.122	1.06
ORB Extract	10.2	128	2.46	30.3
ORB Cluster	32.6	428	21.0	291.
ORB Compare	0.235	1.88	0.178	1.457
BRISK Extract	11.7	111	2.38	17.1
BRISK Cluster	45.1	656	5.01	68.7
BRISK Compare	0.272	2.13	0.0795	0.774

Table 3. Average computation times (in ms) for signature-based methods. All methods require a one-time signature computation, which depends on the image size. The final distance computation (Compare) is independent of the image size, and shown in the last columns. Image sizes were 288 × 48 (sigafc) and 640 × 128 (ABLE) for grid, and 512 × 128 (Quorum V, all methods).

	Grid		**Quorum V**		**Compare**	
Method	**i7**	**Atom**	**i7**	**Atom**	**i7**	**Atom**
sigafc	0.239	2.46	0.535	6.10	0.0001	0.0015
ableLDB-1	0.057	0.773	0.056	0.744	0.0001	0.001
ableLDB-4	0.212	2.87	0.157	2.16	0.0020	0.019
ableLDB-6	0.317	4.28	0.313	4.27	0.0044	0.043
ableLDB-8	0.419	5.64	0.413	5.59	0.0078	0.077
ableBRISK-1	0.032	0.346	0.029	0.338	0.0001	0.0006
ableBRISK-4	0.109	1.29	0.052	0.578	0.0011	0.010
ableBRISK-6	0.161	1.93	0.156	1.94	0.0026	0.023
ableBRISK-8	0.212	2.54	0.208	2.49	0.0046	0.041

The times for the optimized compass measures (Table 1) are very similar, as the basic structure of the code is always the same. Only the actual dissimilarity measure between image columns changes. On the grid database `i-sad-1` is slightly faster than `e-nsad-1` and `i-nsad-1`. The `i-ssd-1` measure uses floating-point representation for intermediate values, resulting in a higher processing time. For the extended compass with multiple scale planes the computation time increases about linear with the number of scale planes. Due to some constant overhead, the average factor is 2.8 for three scale planes and 4.6 for five scale planes.

`Warping` is by far the slowest measure of these, but it also computes a home vector and refined compass estimate, adding a second phase with an exhaustive search to the visual compass calculation. Also, `warping` computes nine scale planes instead of just one. It is possible to speed-up this method using different heuristics, trading speed for accuracy of the results.

For the larger Quorum V images the general results are similar, but computation times are higher due to the larger image size.

Table 2 shows the time measurements for FabMap. SIFT feature extraction and clustering is considerably slower than ORB or BRISK, by a factor of 10 to 20. However, this step has to be done only once per image. The final image descriptors are independent of the feature descriptor, so the time for comparisons is about equal. The average time to compare two images in the grid database is similar to the computation time of a single-plane visual compass. As the Quorum V camera images are smaller than the grid camera images, the computation times are lower. BRISK feature extraction and clustering is also faster, as fewer features are detected in these images.

The computation of the Fourier signature of an image also takes about 0.24 ms (grid, i7), but the distance can be computed very fast, as only low-dimensional vectors (96 entries) have to be compared. ABLE signatures take similarly long to compute (depending on the number of regions), along with a very short distance computation time.

5. Discussion

The holistic methods presented in this paper all work very well on the same-database tests. The ROC curves for multiple scale planes tend to be slightly better than using only the simple visual compass with one scale plane. Results for `warping` are comparable to the visual compass methods, but the ROC curve is worse on the grid databases, as more changes in the image content can be tolerated by this method. Thus, images farther away get a lower dissimilarity value, possibly causing false positive matches in the thresholding process. However, depending on the application, it may be beneficial to find matches from a wider area: Computing home vectors to matched places allows to triangulate a relative position estimate. Using triangulations to multiple places allows to check for

consistency among estimates. The preliminary test on the outdoor database shows that the holistic compass measures are also suitable for outdoor environments. A more thorough investigation of the outdoor performance is needed, though, with more databases and cross-database experiments.

FabMap, `sigafc` and ABLE with the LDB descriptor also show a good performance on the same-database tests, while ABLE using BRISK generally performs worse than LDB. The number of regions used for ABLE has to be suitable for the orientation changes occurring in the dataset. As expected, using 1 or 6 regions fails on the grid databases (where the images have a relative orientation of about 90°), while 4 and 8 regions give good results. The images in the Quorum V and outdoor datasets have mostly a common orientation, so the number of regions has a smaller impact. Using multiple regions with BRISK in the Quorum V dataset, however, performs worse than only one region. The images in the CITEC dataset have basically a random orientation, but four main directions are present. This allows ABLE to find good matches for most image pairs using 4 or more regions. ABLE with LDB also performs well on the small outdoor dataset, while BRISK is worse, again more so with multiple regions, although the images have the same orientation. The choice of the feature descriptor for FabMap has no large influence in our study, with SIFT giving better results than the binary descriptors.

The ratio postprocessing has a mixed effect on all methods used for place recognition. The ROC curves often get better, especially for the holistic methods, but the distance to the best matching image, here presented as MINDIST, generally increases. While the increase is usually moderate, additional outliers are frequently introduced. Results for the mean ratio postprocessing are usually worse than for the min ratio.

The cross-database tests with image pairs captured under changing illumination conditions pose a challenge for all methods. The holistic compass using NSAD on edge-filtered images with 3 or 5 scale planes gives the best results in this case. The other holistic compass methods as well as `warping` perform much worse than for same-database tests. While the median of MINDIST is often small, the 95th percentile is very high, meaning more large distances are present. Edge-filtering the images shows an important advantage for these tests, as already shown in our experiments with visual homing [42,52]. While the ROC curve and AUC for `warping` (which also works on edge-filtered images) is comparable to the intensity-based methods, it still shows lower MINDIST values. The ROC curves for most holistic methods improve when using the min-ratio postprocessing. The ceiling lamps featuring prominently in the images also cause problems for the intensity-based methods, as they are switched on or off in the image pairs. Reducing the elevation angle so the images contain less information about the ceiling mitigates this problem. Thus, it is important to consider possible impacts that the environment may have on the applied method. However, such impacts may not be apparent in advance.

The signature and feature-based methods perform close to chance level on the cross-database tests. Even the BRISK descriptor, which compared well in our visual homing study [46], gives no usable results. SIFT is slightly better than the other descriptors, and the ROC curve using this descriptor improves using the ratio postprocessing.

The holistic compass using the NSAD measure on edge-filtered images also performs best on mixed-database tests, clearly outperforming the intensity-based measures. `warping` again performs similar to the intensity-based holistic compass methods, but has lower MINDIST values. Applying the min-ratio postprocessing further reduces these values and removes outliers. FabMap using the SIFT descriptor performs similar to `warping` on mixed-database test, with more outliers in MINDIST. Using the min-ratio postprocessing, SIFT results improve, achieving AUC values comparable to *e-nsad*. MINDIST values are still higher, though. The other feature and signature-based methods perform worse, without any improvements from the ratio postprocessing.

The time to compare two images using a visual compass or FabMap is relatively slow compared to the very fast comparison of low-dimensional signature vectors. ABLE, FabMap, and Fourier signatures require a one-time preprocessing computation per image. The holistic methods do not require this step.

Min-Warping, which also computes a home vector and includes an extensive search, is considerably slower than the compass methods.

Figure 31 shows an overview of computation time (per image pair; i7 CPU) and the mean distance between the current view and the best-matching snapshot on the lab day/night cross database. Signature-based methods are very fast, but the mean distance to the best-matching image always lies above the acceptance radius (dashed line). The visual compass on edge-filtered images (`e-nsad`) achieves the best results in terms of distance. `Warping`, while being the slowest method, also achieves a very low mean distance. Using multiple scale planes for the intensity-based visual compass methods increases the mean distances and computation times. Results for FabMap are comparable, with SIFT giving the lowest mean distances. However, it has a very long preprocessing time for feature detection compared to the binary descriptors.

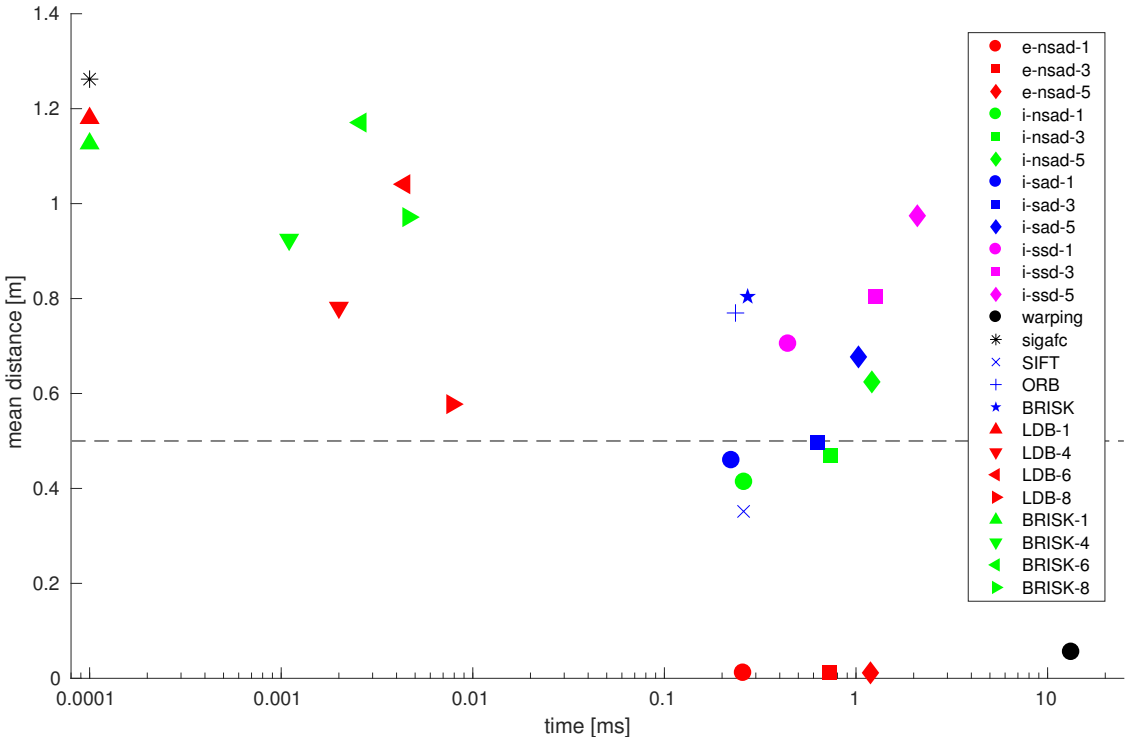

Figure 31. Average image comparison time vs. place recognition performance for all methods (without ratios) on the lab day/night cross database. The horizontal axis shows computation times on a logarithmic scale, the vertical axis the mean distance between a current view and the best-matching snapshot. The dashed line marks the acceptance radius for the grid databases. The colors match the ones used in Section 4. SIFT/ORB/BRISK (blue symbols) refer to FabMap, LDB-*/BRISK-* (triangle symbols) refer to ABLE.

As mentioned earlier, the actual application at hand determines the required quality of place recognition results. Early SLAM systems required the absence of false-positive matches, but current methods can tolerate them more easily. One approach for complete coverage that is currently being developed in our research group uses place recognition results to calculate a relative pose estimate using triangulation with multiple image pairs. False image matches may be rejected based on inconsistent triangulations.

For the most efficient application of the presented methods for place recognition, a preselection of images using a fast signature method, followed by a more accurate holistic visual compass, seems advisable. Such a combined method still has to be investigated, though.

6. Conclusions

We compared different methods for visual place recognition in multiple indoor and outdoor databases, under constant and changing illumination conditions. Most methods perform very well in same-database tests, but challenging situations still occur. This includes places that are visually very similar, as in the corridor segment of the Quorum V database. Parameters of different methods also have to be chosen carefully: When using ABLE signatures, the number of regions used to describe an image has to be suitable for the expected orientation changes between images. FabMap requires a training phase before it can be applied for place recognition, and environments that differ strongly from the training data may be challenging. In cross- and mixed-database tests, the visual compass using NSAD distance measure on edge-filtered images (`e-nsad`) clearly outperforms all other methods.

Although the visual compass provides accurate matches, it is significantly slower than signatures when comparing images. Signatures, however, require a one-time precomputation step. A promising approach when comparing a large number of images within a tight time constraint should be a *preselection* of match candidates using signatures, with a more accurate validation using a visual compass. Regarding the applications mentioned at the beginning of the paper, the following suggestions can be made:

Exact matches are crucial for loop-closure detection, so only accurate methods should be used for place recognition in this case. However, the computations should be fast enough to check for matches frequently, e.g., for every node added to a topological map. Depending on the robot's navigation strategy, it may be necessary to check many images for matches, so the preselection using signature-based methods may be required for low computation times. Using a sequence of multiple images to determine a match (like the approach taken in [20]) should allow to improve the accuracy for this task.

Solving the kidnapped-robot problem should not have a tight time constraint, as the robot may remain stationary during this task. Still, preselection could be used to quickly exclude non-matching locations, before using a more exact method like the visual compass. Warping, which tends to accept images within a broader region, can also be used to identify promising matches using fewer images.

While traveling along a route in a topological map, the next target image is usually known in advance, and place recognition only has to detect when the reference image position has been reached. Thus, no large number of image comparisons is required and holistic methods can be applied directly. Identifying intermediate nodes on the route may be less important than exactly arriving at the goal, so matches from a broader region may be acceptable in this case. However, the target node should only be switched if the robot can reach the next node (using for example visual homing) from its current position. Thus, matches should not be accepted too early. A possible approach is to compare the image dissimilarity between the previous and next target node, and switch once the best match changes. Warping may be used both for visual homing and place recognition in this task, as it computes all information necessary to reach and match the next node along a route.

Acknowledgments: This work was supported by Deutsche Forschungsgemeinschaft (grant No. MO 1037/10-1). We acknowledge support for the Article Processing Charge by the Deutsche Forschungsgemeinschaft and the Open Access Publication Fund of Bielefeld University.

Author Contributions: Michael Horst and Ralf Möller conceived and designed the experiments; Michael Horst performed the experiments, analyzed the data and wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest. Deutsche Forschungsgemeinschaft had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Appendix A. Parameters

Table A1 lists the parameters used for the visual compass and `warping`. The `pixelScale` and `postScale` parameters used for conversion between floating-point and integer types in the SIMD library have been computed for each database individually. For the normalized measure NSAD and

`warping`, these scales only affect intermediate results directly, but not the final dissimilarity value (only through possible saturation or rounding of intermediate values). Scales for SSD were set to 1.0, as floating-point data types are used, so no saturation has to be considered. SAD used a pixelScale of 100 (conversion from floating-point values between 0 and 1 to integer) and a postScale of 0.1.

The vertical resolution in radians per pixel was computed from image height and vertical opening angle for our own image databases. As the opening angle and exact mapping (e.g., polar or hyperbolic) were not provided for the Quorum V database, we use nearest neighbor interpolation (vertical resolution argument of -1 in the Min-Warping code).

Table A2 lists the parameters for ABLE, Table A3 the ones for FabMap.

Table A1. Parameter settings for holistic methods.

Parameter	Value
scale planes (warping)	9
max. scale factor	2
max. threshold	2.5
interpolation	0
n_α (grid, outdoor)	96
n_ψ (grid, outdoor)	96
n_α (CITEC, Quorum V)	64
n_ψ (CITEC, Quorum V)	64
max. scale factor (3 scale planes)	1.1
max. threshold (3 scale planes)	1.4
max. scale factor (5 scale planes)	1.3
max. threshold (5 scale planes)	1.6
warping search mode	full double
bins for histogram equalization	64

Table A2. Parameter settings for ABLE (library defaults).

Parameter	Value
BRISK corner detection threshold	5
BRISK octaves	8
BRISK pattern scale	1.0

Table A3. Parameter settings for FabMap. Most values are library defaults.

Parameter	Value
SIFT number of features	500
SIFT octave layers	3
SIFT contrast threshold	0.04
SIFT edge threshold	10
SIFT sigma	1.6
ORB number of features	500
ORB pyramid scale factor	1.2
ORB pyramid levels	8
ORB edge threshold	31
ORB BRIEF pixel comparisons	2
ORB score	Harris
ORB patch size	31
ORB fast threshold	20
BRISK corner detection threshold	85
BRISK octaves	3
BRISK pattern scale	1
Vocabulary size	10,000
Detector model PzGe	0.39
Detector model PzGNe	0

References

1. Franz, M.O.; Schölkopf, B.; Mallot, H.A.; Bülthoff, H.H. Learning View Graphs for Robot Navigation. *Auton. Robots* **1998**, *5*, 111–125.
2. Ulrich, I.; Nourbakhsh, I. Appearance-Based Place Recognition for Topological Localization. In Proceedings of the ICRA 2000, San Francisco, CA, USA, 24–28 April 2000; Volume 2, pp. 1023–1029.
3. Gerstmayr-Hillen, L.; Röben, F.; Krzykawski, M.; Kreft, S.; Venjakob, D.; Möller, R. Dense Topological Maps and Partial Pose Estimation for Visual Control of an Autonomous Cleaning Robot. *Robot. Auton. Syst.* **2013**, *61*, 497–516.
4. Vardy, A. Long-Range Visual Homing. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Kunming, China, 17–20 December 2006; pp. 220–226.
5. Milford, M. Vision-Based Place Recognition: How Low Can You Go? *Int. J. Robot. Res.* **2013**, *32*, 766–789.
6. Sünderhauf, N.; Protzel, P. BRIEF-Gist—Closing the Loop by Simple Means. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 1234–1241.
7. Sünderhauf, N.; Protzel, P. Towards a Robust Back-End for Pose Graph SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 1254–1261.
8. Latif, Y.; Cadena, C.; Neira, J. Robust Loop Closing over Time for Pose Graph SLAM. *Int. J. Robot. Res.* **2013**, *32*, 1611–1626.
9. Menegatti, E.; Zoccarato, M.; Pagello, E.; Ishiguro, H. Image-Based Monte Carlo Localisation with Omnidirectional Images. *Robot. Auton. Syst.* **2004**, *48*, 17–30.
10. Möller, R.; Krzykawski, M.; Gerstmayr-Hillen, L.; Horst, M.; Fleer, D.; de Jong, J. Cleaning Robot Navigation Using Panoramic Views and Particle Clouds as Landmarks. *Robot. Auton. Syst.* **2013**, *61*, 1415–1439.
11. Möller, R.; Krzykawski, M.; Gerstmayr, L. Three 2D-Warping Schemes for Visual Robot Navigation. *Auton. Robots* **2010**, *29*, 253–291.
12. Lowry, S.; Sünderhauf, N.; Newman, P.; Leonard, J.J.; Cox, D.; Corke, P.; Milford, M.J. Visual Place Recognition: A Survey. *IEEE Trans. Robot.* **2016**, *32*, 1–19.
13. Zeil, J.; Hofmann, M.I.; Chahl, J.S. Catchment Areas of Panoramic Snapshots in Outdoor Scenes. *J. Opt. Soc. Am. A* **2003**, *20*, 450–469.
14. Labrosse, F. The Visual Compass: Performance and Limitations of an Appearance-Based Method. *J. Field Robot.* **2006**, *23*, 913–941.
15. Stürzl, W.; Zeil, J. Depth, Contrast and View-Based Homing in Outdoor Scenes. *Biol. Cybern.* **2007**, *96*, 519–531.
16. Kuglin, C.D.; Hines, D.C. The Phase Correlation Image Alignment Method. In Proceedings of the International Conference on Cybernetics and Society, San Francisco, CA, USA, 23–25 September 1975; pp. 163–165.
17. Burke, A.; Vardy, A. Visual Compass Methods for Robot Navigation. In Proceedings of the Newfoundland Conference on Electrical and Computer Engineering, St. Johns, NL, Canada, 9 November 2006.
18. Möller, R. Local Visual Homing by Warping of Two-Dimensional Images. *Robot. Auton. Syst.* **2009**, *57*, 87–101.
19. Milford, M.; Wyeth, G. SeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 1643–1649.
20. Mount, J.; Milford, M. 2D Visual Place Recognition for Domestic Service Robots at Night. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 4822–4829.
21. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
22. Bay, H.; Ess, A.; Tuytelaars, T.; Gool, L.V. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359.
23. Hamming, R.W. Error Detecting and Error Correcting Codes. *Bell Syst. Tech. J.* **1950**, *29*, 147–160.
24. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In Proceedings of the 9th European Conference on Computer Vision, Graz, Austria, 7–13 May 2006; Leonardis, A., Bischof, H., Pinz, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443.
25. Calonder, M.; Lepetit, V.; Ozuysal, M.; Trzcinski, T.; Strecha, C.; Fua, P. BRIEF: Computing a Local Binary Descriptor Very Fast. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1281–1298.

26. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An Efficient Alternative to SIFT or SURF. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
27. Leutenegger, S.; Chli, M.; Siegwart, R.Y. BRISK: Binary Robust Invariant Scalable Keypoints. In Proceedings of the International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2548–2555.
28. Yang, X.; Cheng, K.T. LDB: An Ultra-Fast Feature for Scalable Augmented Reality on Mobile Devices. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Atlanta, GA, USA, 5–8 November 2012; pp. 49–57.
29. Cummins, M.; Newman, P. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Int. J. Robot. Res.* **2008**, *27*, 647–665.
30. Cummins, M.; Newman, P. Accelerating FAB-MAP with Concentration Inequalities. *IEEE Trans. Robot.* **2010**, *26*, 1042–1050.
31. Cummins, M.; Newman, P. Appearance-Only SLAM at Large Scale with FAB-MAP 2.0. *Int. J. Robot. Res.* **2011**, *30*, 1100–1123.
32. Sivic, J.; Zisserman, A. Video Google: A Text Retrieval Approach to Object Matching in Videos. In Proceedings of the Ninth IEEE International Conference on Computer Vision, Nice, France, 13–16 October 2003; Volume 2, pp. 1470–1477.
33. Glover, A.; Maddern, W.; Warren, M.; Reid, S.; Milford, M.; Wyeth, G. OpenFABMAP: An Open Source Toolbox for Appearance-Based Loop Closure Detection. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), St. Paul, MN, USA, 14–18 May 2012; pp. 4730–4735.
34. Grana, C.; Borghesani, D.; Manfredi, M.; Cucchiara, R. A Fast Approach for Integrating ORB Descriptors in the Bag of Words Model. In Proceedings of the SPIE, Burlingame, CA, USA, 4–6 February 2013; Volume 8667, p. 866709.
35. Gerstmayr-Hillen, L.; Schlüter, O.; Krzykawski, M.; Möller, R. Parsimonious Loop-Closure Detection Based on Global Image-Descriptors of Panoramic Images. In Proceedings of the 15th International Conference on Advanced Robotics (ICAR), Tallinn, Estonia, 20–23 June 2011; pp. 576–581.
36. Gerstmayr-Hillen, L. From Local Visual Homing towards Navigation of Autonomous Cleaning Robots. Ph.D. Thesis, Bielefeld University, Bielefeld, Germany, 2013.
37. Arroyo, R.; Alcantarilla, P.F.; Bergasa, L.M.; Yebes, J.J.; Gámez, S. Bidirectional Loop Closure Detection on Panoramas for Visual Navigation. In Proceedings of the IEEE Intelligent Vehicles Symposium Proceedings, Ypsilanti, MI, USA, 8–11 June 2014; pp. 1378–1383.
38. Arroyo, R.; Alcantarilla, P.F.; Bergasa, L.M.; Yebes, J.J.; Bronte, S. Fast and Effective Visual Place Recognition Using Binary Codes and Disparity Information. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3089–3094.
39. Arroyo, R.; Alcantarilla, P.F.; Bergasa, L.M.; Romera, E. Towards Life-Long Visual Localization Using an Efficient Matching of Binary Sequences from Images. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 6328–6335.
40. Arroyo, R.; Alcantarilla, P.F.; Bergasa, L.M.; Romera, E. OpenABLE: An Open-Source Toolbox for Application in Life-Long Visual Localization of Autonomous Vehicles. In Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 965–970.
41. Möller, R. *A SIMD Implementation of the MinWarping Method for Local Visual Homing*; Computer Engineering Group, Bielefeld University: Bielefeld, Germany, 2016.
42. Möller, R. *Column Distance Measures and Their Effect on Illumination Tolerance in MinWarping*; Computer Engineering Group, Bielefeld University: Bielefeld, Germany, 2016.
43. Möller, R. *Design of a Low-Level C++ Template SIMD Library*; Computer Engineering Group, Bielefeld University: Bielefeld, Germany, 2016.
44. Chow, C.; Liu, C. Approximating Discrete Probability Distributions with Dependence Trees. *IEEE Trans. Inf. Theory* **1968**, *14*, 462–467.
45. Bradski, G. The OpenCV Library. *Dr. Dobb's J. Softw. Tools* **2000**, *25*, 120–126.
46. Fleer, D.; Möller, R. Comparing Holistic and Feature-Based Visual Methods for Estimating the Relative Pose of Mobile Robots. *Robot. Auton. Syst.* **2017**, *89*, 51–74.
47. Menegatti, E.; Maeda, T.; Ishiguro, H. Image-Based Memory for Robot Navigation Using Properties of Omnidirectional Images. *Robot. Auton. Syst.* **2004**, *47*, 251–267.

48. Payá, L.; Amorós, F.; Fernández, L.; Reinoso, O. Performance of Global-Appearance Descriptors in Map Building and Localization Using Omnidirectional Vision. *Sensors* **2014**, *14*, 3033–3064.
49. Viertel, P. Improvements and Analysis of Warping for Outdoor Robots: Illumination Invariance, Tilt Tolerance and Overall Robustness. Bachelor's Thesis, FH Bielefeld, University of Applied Sciences, Bielefeld, Germany, 2016.
50. Debevec, P.E.; Malik, J. Recovering High Dynamic Range Radiance Maps from Photographs. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, Los Angeles, CA, USA, 3–8 August 1997; pp. 369–378.
51. Fawcett, T. An Introduction to ROC Analysis. *Pattern Recognit. Lett.* **2006**, *27*, 861–874.
52. Möller, R.; Horst, M.; Fleer, D. Illumination Tolerance for Visual Navigation with the Holistic Min-Warping Method. *Robotics* **2014**, *3*, 22–67.