Robotica (2015) volume 33, pp. 1835–1849. © Cambridge University Press 2014
doi:10.1017/S0263574714001076

Matching monocular lightweight features using *N*-gram techniques for topological location identification

Jaime Boal*, Álvaro Sánchez-Miralles and Manuel Alvar

Institute for Research in Technology (IIT), ICAI School of Engineering, Comillas Pontifical University, Madrid, Spain

(Accepted March 31, 2014. First published online: May 1, 2014)

SUMMARY

In SLAM (simultaneous localization and mapping), the topological paradigm provides a more natural and compact solution that scales better with the size of the environment. Computer vision has always been regarded as the ideal sensor technology for topological feature extraction and description and several methods have been proposed in the literature, but they are either time-consuming, require plenty of different sensors, or are very sensitive to perceptual aliasing, all of which limit their application scope.

This paper presents a fast-to-compute collection of features extracted from monocular images, and an adaptive matching procedure for location identification in structured indoor environments inspired by the natural language processing field. Although only dominant vertical lines, color histograms, and a reduced number of keypoints are employed in this paper, the matching framework introduced allows for the incorporation of almost any other type of feature. The results of the experiments carried out in a home and an office environment suggest that the proposed method could be used for real-time topological scene recognition even if the environment changes moderately over time. Due to the combination of complementary features, high precision can be achieved within reasonable computation time by using weaker but faster descriptors.

KEYWORDS: Computer vision; Mobile robots; Robot localization; SLAM; Topological modeling of robots.

1. Introduction

The rising interest on metric simultaneous localization and mapping (SLAM) over the past few years has led to the emergence of a less widespread research trend that has drawn its focus on trying to solve the SLAM problem through a topological approach.[2,10,28,31,34] The topological conception of robotics has been around for a long time, even though metric maps are undeniably more accurate. However, on closer examination, there is nothing surprising about it. Apart from providing significant storage savings, as it results in a more compact representation of space that is more in accordance with the size of the environment,[2] the absence of metric and geometric information, which is replaced by notions of proximity and order, eliminates dead-reckoning error issues. Moreover, the topological paradigm has been proven effective for a great deal of situations in Nature, as it is the behavior employed by a variety of different animal species, including human beings.

Back in 1990, Brooks stated that we do not need to answer the question "Where am I?" in millimeters and degrees in order to safely move through the environment and cope efficiently with uncertainty.[7] On the contrary, instead of using coordinates to localize, human beings rely on an abstract notion of distance and, in spite of this, we are still capable of recognizing where we are

* Corresponding author. E-mail: jaime.boal@iit.upcomillas.es

in space.[25] By means of vision, which is our primary source of information, we identify relevant or distinctive aspects of the environment which we use as landmarks for localization. Hence, why should a robot, equipped with a camera, not be able to do exactly the same?

The very first step in any topological SLAM implementation is to define what is going to be considered a landmark in the environment and choose the convenient sensing technologies to perceive them or, conversely, select one or several types of sensors and determine which cues can be extracted from the data they provide. Surprisingly, although an inappropriate decision at this stage complicates the subsequent steps of the algorithm, it is often disregarded and makes it even more difficult to overcome the *perceptual aliasing* problem (i.e. two totally distinct locations appear identical to the robot's sensors), not to mention the added complexity for the already challenging *correspondence problem* (i.e. attempt to determine if sensor measurements taken at different times correspond to the same physical location) in dynamic environments.

According to Stankiewicz and Kalia,[29] by using landmarks, three properties are being implicitly assumed: saliency, persistence, and informativeness. This means that the features should stand out, that they should still be present when the robot revisits the location, and that they have to provide enough information about the robot's position or the behavior it should adopt when perceiving them.

Unfortunately, it is more than somewhat unlikely to find a single type of cue that combines all of the previous properties. This could be explained by the fact that the so-called *structural* landmarks (e.g. intersections, entrances), which encode geometric properties of the environment, are permanent but fairly uninformative. By contrast, *object* landmarks, which are elements independent of the structure, tend to be unique but temporary like, for instance, a red coat hung among many black coats.[29]

Most of the developments in topological SLAM in the literature are based on visual techniques inherited from the object detection world, due to the overwhelming popularity of the successful keypoint detectors which provide highly distinctive features; thus, SIFT (scale-invariant feature transform) features[19] can be considered a sort of standard in this field, owing to their persistence and robustness. Angeli *et al.*[2] adopt standard SIFT features; Sabatta[28] employs persistent color SIFT features and determines the current location using the Mahalanobis distance to find the best matches; and Booij *et al.*[5] build appearance-based maps from SIFT features extracted from unwrapped panoramas. The latter computes a similarity measure for location recognition as the ratio of the number of point correspondences that satisfy the epipolar constraints to the lowest number of features found in the database and query images. In addition, Fraundorfer *et al.*[11] suggest storing all the captures acquired in a database while constructing a link graph to keep track of adjacency. Image matching is then performed by means of MSER (maximally stable extremal regions) features[22] described with SIFT to obtain the closest location. Finally, it is worth mentioning the work by Cummins and Newman[10], where SURF (speeded-up robust feature) features[3] are used for scene recognition using an offline-built vocabulary tree. Notwithstanding, their high computational burden, especially with large images, leaves very little time for other tasks like map fusion, motion planning, or obstacle avoidance, as these techniques were originally designed to run in batch, as opposed to real-time applications, which is precisely one of the essential requirements of robotics.

An alternative approach has its origin in an article by Lamon *et al.*,[15] where they coined the term *fingerprint* of places to refer to a circular list of complementary simple features (color patches and vertical edges), obtained from omnidirectional images, whose order matches their relative position around the robot. This idea led to the publication of a series of papers that further developed the concept of fingerprint. Of special relevance is that of Tapus and Siegwart[31] where, thanks to the information provided by two 180° laser range scanners, corners and empty areas are additionally detected. They employ a modified version of the *global alignment* algorithm[23]—used to compare DNA (deoxyribonucleic acid) sequences—capable of dealing with uncertainty to obtain a matching probability of the features. Unfortunately, this approach requires expensive sensors and gathers such a huge amount of information at each location that makes it highly difficult to handle.

More recently, Liu *et al.*[17] proposed a much simpler fingerprint procedure, exclusively based on panoramic images, which extracts vertical edges under the belief that the prevailing lines naturally segment a structured environment into meaningful areas, and encode the distance among those lines and the mean U-V chrominance of the defined regions—in a LUV color space—in a lightweight descriptor called FACT, which was later granted with statistical meaning and renamed DP-FACT.[18] Although using the U-V chrominance is an interesting option due to the fact that the difference

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

between colors can be computed applying the Euclidean distance, the average approach always has the risk that two completely different regions result in a very similar value.

Finally, it is worth mentioning the work by Ulrich and Nourbakhsh,[33] where a global image matching is performed on the basis of computing histograms in the red–green–blue (RGB) and hue–saturation–lightness (HSL) color spaces on omnidirectional images, and comparing them using the Jeffrey divergence. Localization is then carried out following a unanimous voting scheme.

It is important to emphasize that the vast majority of the aforementioned methods require an omnidirectional camera. This is easily explained by the fact that omnidirectional cameras are the only ones which guarantee *rotational invariance* (i.e. no matter what orientation a robot has in a given location, the image captured is always the same). This is a very desirable property but has the main disadvantage that, conversely to other types of cameras like stereo, this sensor alone is not appropriate for navigation as the distance to obstacles cannot be estimated. Furthermore, all the methods presented above have either one or several of the following drawbacks. They require plenty of different, and often costly, sensors, are sensitive to perceptual aliasing, or are computationally expensive to an extent that makes it fairly difficult to use them for real-time applications.

Bearing these shortcomings in mind, this paper proposes a lightweight vision-only monocular feature extraction procedure based on the notion of *fingerprint* and a matching algorithm adapted from the natural language processing (NLP) world, aimed at topological navigation—or, in general, at topological SLAM—for structured indoor environments. The usage of monocular images allows this method to be applied to any type of robot, as opposed to omnidirectional cameras that require a more complicated installation. Moreover, it is compatible with stereo and, therefore, permits to perform obstacle avoidance with a single camera.

The rest of the paper is structured as follows. First, Section 2 details the necessary steps to extract the proposed collection of landmarks, followed by an explanation of the matching technique employed to find correspondences between fingerprints in Section 3. Subsequently, Section 4 comments on the results obtained for different image sets and finally, the most relevant conclusions are drawn in Section 5.

2. Proposed Fingerprint

A monocular camera has been chosen as the unique sensor to extract a collection of complementary features to derive a fingerprint. First, the image is segmented into different regions of interest (ROIs) by extracting structural vertical lines. As a consequence, the subsequent features can be computed in parallel in each of the resulting subimages and moreover, they are granted with spatial meaning (i.e. they become ordered) without the need for complex calculations.

Any type of feature can be computed in these ROIs, but we propose a combination of color histograms and keypoints. These features compensate for each other's drawbacks as histograms operate on a global basis—or semi-global, given that they are computed in subimages—whereas keypoints are local. For instance, keypoints have trouble with homogeneous regions that can be told apart using color information. This fingerprint has been coined VPACK (Visually Perceivable Adjacent Color histograms and Keypoints). The details regarding the feature extraction process are put forward below.

2.1. Vertical lines for segmentation

Just like Liu *et al.*,[17] we support the hypothesis that vertical lines naturally divide structured environments into informatively distinct regions. Realize, however, that this assertion turns out to be valid only if the focal plane of the camera is perfectly parallel to the planes containing the vertical lines. For structured indoor environments this happens to be generally true as the floor is normally perpendicular to building walls.

With a view to taking advantage of this segmentation property, dominant lines are first extracted from the image. The initial step is to compute the second-order derivative of the Sobel operator over the x direction to enhance the vertical responses of the image. Afterwards, in order to compensate for slight deviations of the camera from the perpendicular plane a three by one dilation is performed (Fig. 1a). This allows to rejoin accidentally cut vertical edges due to horizontal displacement. Some authors like Liu *et al.*[17] may argue that after this step it is necessary to apply a median filter to

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

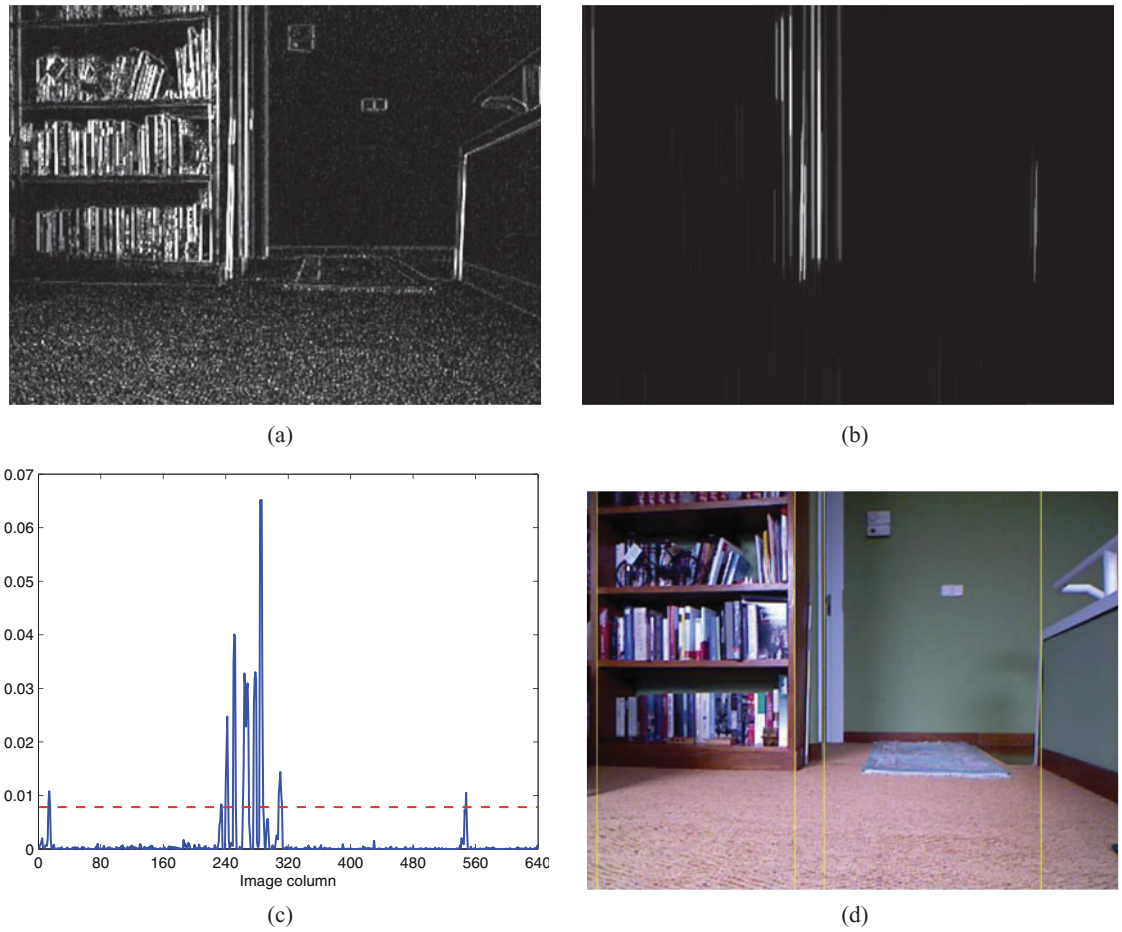


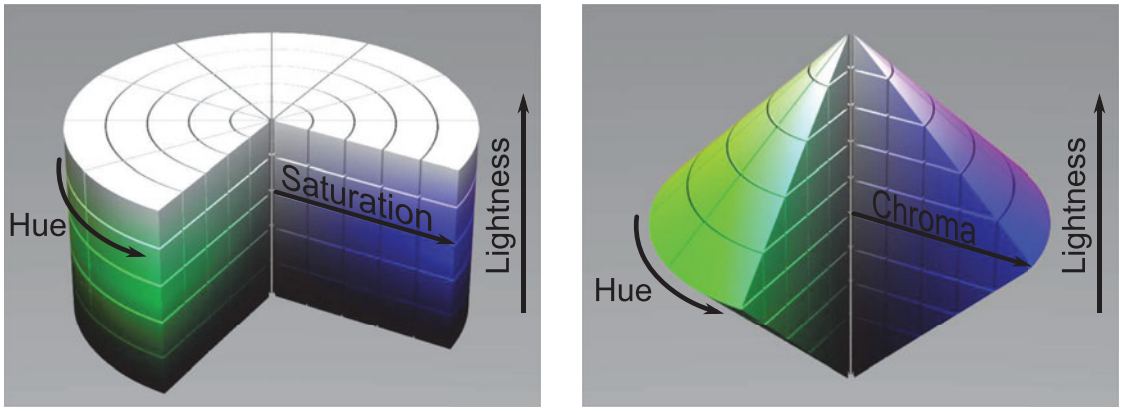
Fig. 1. During line extraction, the input color image is converted to grayscale in order to apply the second-order Sobel operator over the x direction. A dilation operation is then performed to enhance vertical responses (a), followed by a morphological opening to identify relatively long line segments (b). Subsequently, a normalized histogram of the edge responses is computed and the values lower than a given threshold are filtered out (c). Finally, non-maximum suppression is applied to preserve the strongest response over a fixed neighborhood. The extracted vertical lines are drawn on the input image for illustrative purposes in (d).

remove the additional noise introduced. However, on closer examination one realizes that this step is dispensable at the sight of the subsequent operations.

A vertical morphological opening (i.e. an erosion followed by a dilation) is then carried out to join together small segments that are vertically aligned and, at the same time, remove weak responses, which have been defined as those segments shorter than an experimentally adjusted threshold length (Fig. 1b). Given that lines are being used as a segmentation characteristic of the environment, the recommended threshold is rather high, around 1/5 of the image height, with the aim of capturing structural lines, like wall corners or furniture edges, while staying immune to noise coming from less permanent sources such as, for instance, books on a shelf. Furthermore, this operation disposes of the noise added in the previous step.

Afterwards, a normalized graph of the overall vertical response of each pixel column is computed (Fig. 1c). The objective is twofold: to reduce the dimensionality of the problem and to identify the predominant vertical lines. Instead of filtering out those values lower than the mean plus a standard deviation, as suggested by Tapus,[30] the values higher than five times the mean are preserved. The output happens to be almost identical in the typical cases. However, the threshold proposed overcomes a minor pitfall the mean plus standard deviation approach presents. Imagine a rare case where an image does not contain any outstanding vertical lines. Therefore, all the values are bound to be close to the mean and the standard deviation is expected to be low. Consequently, any noisy value that departs marginally from this threshold would be classified as a line. By contrast, with the proposed method no vertical edges would be identified.

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press



(a) (b)

Fig. 2. Hue–saturation–lightness or HSL (a); and hue–chroma–lightness or HCL (b) color spaces. *Source:* Figure created using the ShapeGrid macro by Michael Horvath.[12]

Notice that this last stage is completely different from applying a more restrictive condition to the opening operator. In fact, it is a way of preventing the removal of partially occluded structural lines. Let's consider someone resting his hand against a door frame. If the opening threshold were larger, the two segments in which the frame is divided will be probably suppressed. By contrast, with the proposed solution, they would end up counting for the same vertical response.

Finally, a non-maximum suppression (NMS) algorithm is applied to establish a minimum separation between lines (Fig. 1d). As the objective of the extracted lines is not directly to serve as visual cues but to divide the image into different regions to permit a more local computation of features, it seems clear that these regions must possess a minimum width (e.g. 5% of the image width). To this end, an adapted version of the efficient one-dimensional (1D) $(2n + 1)$ neighborhood NMS algorithm developed by Neubeck and van Gool[24] is applied. The modification introduced makes the algorithm consider maxima close to image edges.

Before concluding this section, it is important to remark the main assets and limitations this type of feature presents. On the side of the advantages, apart from the fact that in structured environments lines are present almost everywhere (e.g. corners, door frames, shelves. . .), one should not forget to mention *rotational invariance*. Assuming the aforesaid conditions required to extract vertical lines are met, no matter the orientation the robot has, vertical lines always appear the same, exception made of occlusions as, under certain viewpoints, lines disappear behind other objects. Another drawback derives from the definition of the segmentation lines. Remember that a minimum distance between lines is forced in order to obtain relevant information in each of the intervals. Hence, as the robot moves closer to a group of lines, their distance in the image increases and additional vertical edges are bound to be identified.

2.2. *Color histograms*

It is undeniable that color is a very informative visual source of information. However, capturing color with a camera is more challenging than it seems at first sight because it is very sensitive to illumination changes. For this reason, a color space like hue–saturation–value (HSV) or HSL which is supposed to split the color (hue) from the brightness information appears to be a reasonable starting point for color extraction.[15,32,33] Between the two aforementioned color spaces, HSL has been chosen over HSV because it is symmetrical to lightness and darkness.[4]

In spite of the color component being separated from saturation and lightness in the HSL color space, these components should not be overlooked in color extraction as hue is meaningless if colors are too bright, dark, or desaturated—they appear to be white, black, or gray, respectively—and can consequently lead to misclassification if they are not taken into account. Following this reasoning, color can be separated into *chromatic* and *achromatic* regions depending on the saturation and luminance components. Tseng and Chang[32] put forward different thresholds to define the achromatic area. Very dark and bright pixels are filtered out using only lightness to discriminate, whereas saturation is also used for mid-range intensity values. However, the values proposed have been

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

proven excessively restrictive and the fact that the color information is richer near the mid-plane of the HSL cylinder, as shown in Fig. 2a, forces to apply different saturation thresholds depending on the value of the lightness channel.

An alternative and more intuitive approach is to employ the hue–chroma–lightness (HCL) bicone model instead (Fig. 2b), where the saturation component is replaced by a combination of lightness and saturation known as *chroma*. In this case, it is sufficient to remove the central cylinder (using the chroma channel) to tell the color information apart. The pixels whose chroma values are lower or equal to 12.5% are classified as gray.

Moving to the actual algorithm, for each of the ROIs defined by the vertical lines extracted in Section 2.1, chromatic pixels are first identified, using the chroma threshold proposed above, with the aim of building eight-bin hue histograms as, according to Tapus[30], eight distinct values are enough to encode colorimetric information. Nonetheless, building histograms directly, adding up the pixels that correspond to each bin, has an important drawback. It would not be uncommon to find locations where many pixels lie along the border between two adjacent bins and that in two consecutive images those pixels fall into different bins, giving rise to completely dissimilar histograms.

In order to address this issue, a fuzzy voting scheme with triangular membership functions, inspired by the one proposed in Lamon *et al.*,[15] has been implemented. Rather than counting for a single bin, each pixel belongs to two histogram columns simultaneously, in an inversely proportional manner to its distance to both bin centers. Realize that as the hue component is circular, the histogram ends are contiguous. To perform this computation, a histogram with a large number of bins is obtained first. Then, the membership degrees are calculated on this histogram.

However, there exist locations where white, gray, black are the prevailing colors, and therefore the hue component is not sufficiently meaningful. For this reason, the pixels from the achromatic region are used to obtain a five-bin histogram to accommodate different tones of gray following a procedure that is analogous to the one applied to the chromatic area except for the fact that low and high values are not mixed when performing the voting. If due to the illumination conditions of the environment the images are prone to over- or under-exposure, the apexes of the HCL bicone can be removed in order to mitigate this issue. The trade-off for this decision is that pure black and white would no longer be identified.

Finally, both the color and grayscale histograms are stacked together and then normalized to obtain a single 13-bin histogram.

2.3. Keypoints

Provided that keypoints (i.e. points in an image that stand out from its surroundings with respect to some particular property like brightness) have been proven effective for rather difficult object recognition tasks, it seems appropriate to include this type of visual cue in the fingerprint. There exist many different alternatives and the choice of the keypoint detector and descriptor depends on many factors like the type of environment, the computing power available, and the size of the images analyzed, among others. In Section 4 the performance of the method is analyzed for three qualitatively distinct algorithms: SIFT,[19] which is robust but slow; Star features—a variant of CenSurE[1]—described with upright SURF,[3] which are faster but less robust; and ORB (Oriented FAST and Rotated BRIEF)[26] that is much faster due to the fact that it employs a less distinctive binary descriptor.

In any case, no matter which type of keypoints is chosen, no more than 100 robust keypoints, distributed among the different ROIs, are extracted in total. These keypoints are obtained by iteratively adjusting the response threshold. However, if for whatever reason more points are still present after a reduced number of iterations, they are all preserved. Realize that this is a totally opposite approach to that of the image recognition field, where several hundreds or even thousands of features are extracted from each image in order to be able to perform matching with relatively low uncertainty.

2.4. Final descriptor

To sum up, the resulting fingerprint (Fig. 3) consists of two sorted sets of $n + 1$ elements, where n is the number of structural vertical lines identified in the image. One set contains 13-bin histograms (eight bins represent color and five model grayscale values) whereas the elements of the other are collections of keypoints extracted from each of the subimages defined by the vertical lines.

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

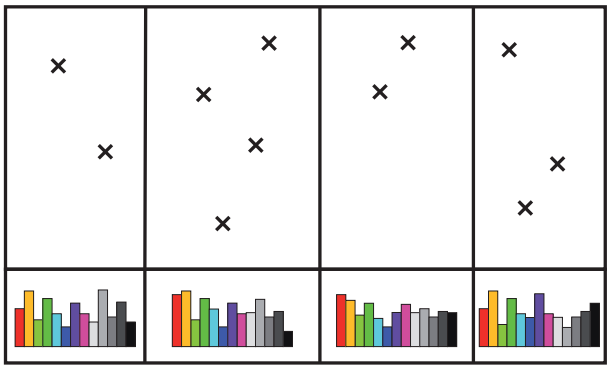


Fig. 3. Final descriptor extracted. From each of the regions defined by structural vertical lines, keypoints and color histograms (which include both chromatic and achromatic pixels) are obtained.

3. *N*-gram Matching

Once the features of an image have been extracted and described, they have to be compared with those previously gathered in order to evaluate their similarity and decide on the current location. This section introduces a matching procedure based on *N*-grams, with a view to taking adjacency between features into account. The term *N*-gram is employed in NLP to refer to a subsequence of *N* consecutive elements (i.e. letters or words) within a larger sequence.[13] However, this concept can be easily extended to different contexts. In this paper, each of the subimages defined by the structural vertical lines extracted in Section 2.1 is considered a "letter." As usual in NLP, only *N*-grams of up to three items are evaluated. When *N* is equal to one, they are called "unigrams"; for *N* equal to two, "bigrams;" and three-element grams are referred to as "trigrams."

3.1. From features to N-grams

In order to be able to apply the *N*-gram framework, corresponding "letters" between the query and the reference images must be first identified. To begin with, individual keypoints and histograms are matched.

The similarity between two histograms is determined using the normalized version (i.e. the range of possible values is scaled to the [0, 1] interval) of the commonly used *Hellinger distance* (1)[8,16], which has been chosen because it satisfies the metric axioms (i.e. non-negativity, reflexivity, symmetry, and triangle inequality).[9] It is important to point out that this metric is sometimes confused in the literature with the *Bhattacharyya distance* because both make use of the *Bhattacharyya coefficient* (2)—also known as *Hellinger affinity*—which is nothing more than the sum of the geometric means of each i-bin pair. The actual Bhattacharyya distance is similar but violates the triangle inequality property.[14] For every histogram in the reference image R, only the best match in the query image Q is kept given that its normalized Hellinger distance is no larger than 0.3. This threshold has been experimentally adjusted.

$$d_H(Q, R) = \sqrt{1 - \rho(Q, R)}, \tag{1}$$

$$\rho(Q, R) = \sum_{i=1}^{N} \sqrt{Q_i \cdot R_i}\ , \text{ where } \sum_{i=1}^{N} Q_i = \sum_{i=1}^{N} R_i = 1. \tag{2}$$

By contrast, the best match for each query keypoint is retrieved from the keypoints of all the reference images using the Euclidean distance, which is the one generally used in the literature for these types of features, except for the case of binary descriptors (like the ones in ORB), where the Hamming distance is employed instead.

Once individual corresponding features have been identified, the matching score between "letters" can be computed as follows. Remember that each "letter" is represented by a single histogram and several keypoints. For histograms, the matching score is calculated as $1 - d_H(Q, R)$, whereas for

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

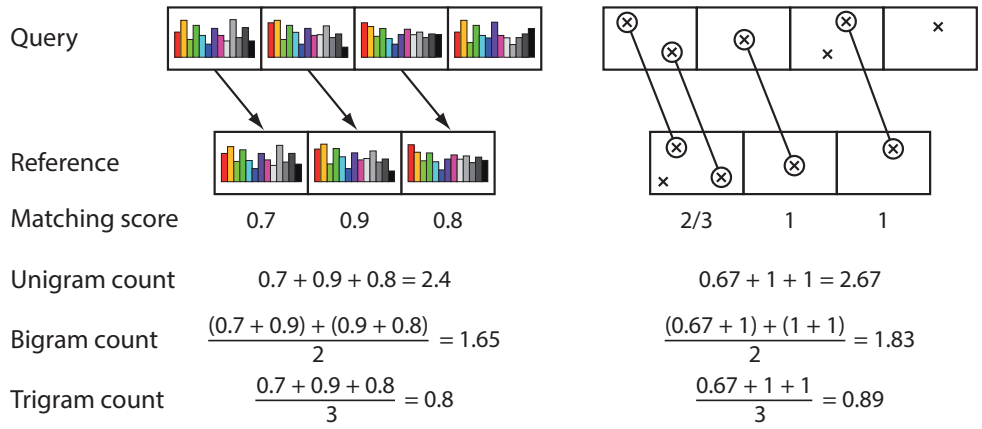


Fig. 4. Example of the N-gram counts computation. The image depicts a query and a reference fingerprint. Lines and arrows indicate corresponding histograms and keypoints. The matching score for individual subimages is one minus the Hellinger distance for histograms and the number of matched keypoints over the total number of keypoints in the reference subimage. Three unigrams, two bigrams, and one trigram have been matched for both type of features.

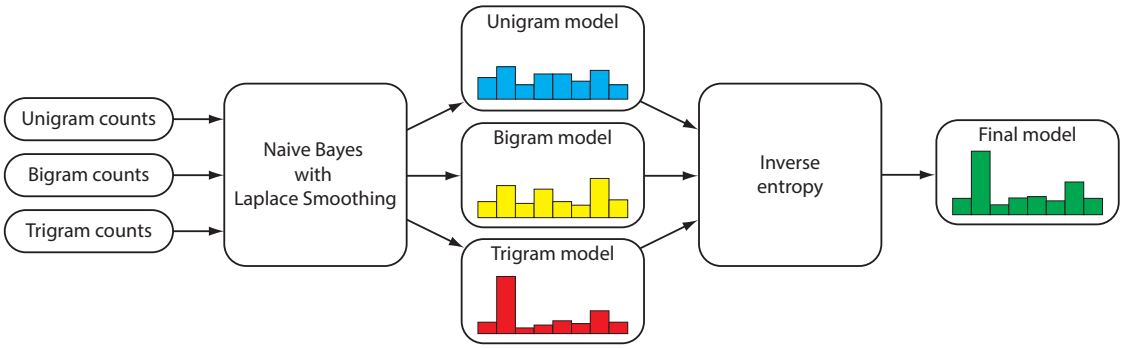


Fig. 5. Matching procedure. The unigram, bigram, and trigram counts are employed to build three discrete probability distributions for the current location using Naive Bayes with Laplace smoothing. These distributions are then combined using inverse entropy, which gives more weight to the one that has more information (i.e. is more confident about its prediction) to obtain the probabilities of being in each encoded location.

keypoints it is the ratio between the number of keypoints matched and the total number of keypoints in the reference "letter."

Finally, the number of unigrams, bigrams, and trigrams that have been matched are computed for each type of feature separately (i.e. three N-gram counts for histograms and three for keypoints are obtained). This implies looking for N consecutive subimages both in the query and representative images that correspond to each other. Two subimages are considered to match if they have at least one feature in common. N-gram counts are then computed as the sum of the average matching scores of its "letters." See Fig. 4 for an example.

3.2. Matching algorithm

With the identified N-grams, unigram, bigram, and trigram models are built separately for color histograms and keypoints. The process, which is analogous for both types of features, is explained below (Fig. 5).

For each of the N-gram models, the probability of being at a previously visited location is estimated employing a naive Bayes approach with Laplace smoothing[27]—to avoid having zero probabilities. Let $n \in \{1, \ldots, N\}$ be the different models, L_k denote one of the K encoded locations, and O stand for the features in the query image. The probability of being at location L_k given O for N-gram model

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

n can be therefore computed as

$$P_n\left(L_k|O\right) = \frac{x_{n,k} + \alpha_n}{\sum_{k=1}^{K} x_{n,k} + \alpha_n \cdot K}, \qquad \forall n, \tag{3}$$

where $x_{n,k}$ is the number of grams of size n that have been matched in L_k (i.e. unigram, bigram, and trigram counts obtained in Section 3.1) and $\alpha_n > 0$ is the smoothing parameter, which can be regarded as a measure of the confidence in the observations; the more observations, the less it affects the probability. If an add-one smoothing is chosen (i.e. $\alpha_n = 1$), one assumes that every seen or unseen event occurred once more than it did in the training data and, as a consequence, it moves a lot of probability mass from seen to unseen events. In practice, much lower values are used, mainly to prevent having zero probabilities. For this particular application, $\alpha_n = 0.01\ \forall n$ was selected. Nonetheless, the value of α_n need not necessarily be the same for low- and high-order models (e.g. unigram and trigram models respectively).

After this process, there are three different discrete probability distributions for the current location (i.e. for unigram, bigram, and trigram models) that have to be combined somehow. Two distinct approaches exist in NLP: interpolation and backoff. The main difference between them is that the former considers all models whereas the latter relies solely on the most complex (trigrams in this case) and only "backs off" to lower-order N-grams if there is no evidence in the higher-order model (i.e. no trigrams have been observed).[13] For the particular task of scene recognition, backoff is too optimistic as, contrary to NLP, observations are noisy and one cannot blindly trust the most complex model. For this reason, an interpolation approach has been chosen.

The problem that arises is how to determine the weighting coefficients of each of the models in the absence of training data to adjust them. The solution that has been adopted is to estimate the coefficients every time using inverse entropy.[21] The aim is to assign larger weights to those models that are more confident about the current location. The entropy H_n of a probability distribution is defined as

$$H_n = \sum_{k=1}^{K} -P_n\left(L_k|O\right) \cdot \log\left(P_n\left(L_k|O\right)\right), \qquad \forall n. \tag{4}$$

The weight ω_n for each model can then be computed with the following expression:

$$\omega_n = \frac{\frac{1}{H_n}}{\sum_{n=1}^{N} \frac{1}{H_n}}, \qquad \forall n. \tag{5}$$

The resulting probability distribution is calculated as

$$P\left(L_k|O\right) = \sum_{n=1}^{N} \omega_n \cdot P_n\left(L_k|O\right), \qquad \forall k. \tag{6}$$

This interpolation procedure based on inverse entropy helps to deal with those situations where high-order N-grams do not exist. For instance, if no trigrams are found for any reference image either because the query image is not similar enough or because the reference images do not have at least three ROIs, the resulting trigram model will be a discrete uniform distribution. As it is the maximum entropy probability distribution, its weight will be small compared to other lower-order models, provided that they have enough information.

The aforementioned algorithm is computed twice in order to obtain two different probability distributions, one according to the color histograms and another to the keypoints. These features can be assumed to be independent, as histograms are global features—or semi-global in this case—while keypoints are local, so both models are simply multiplied and normalized to compute the final probabilities. As a result, if both distributions agree on the current location, they reinforce each other and provide a significantly higher matching probability.

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

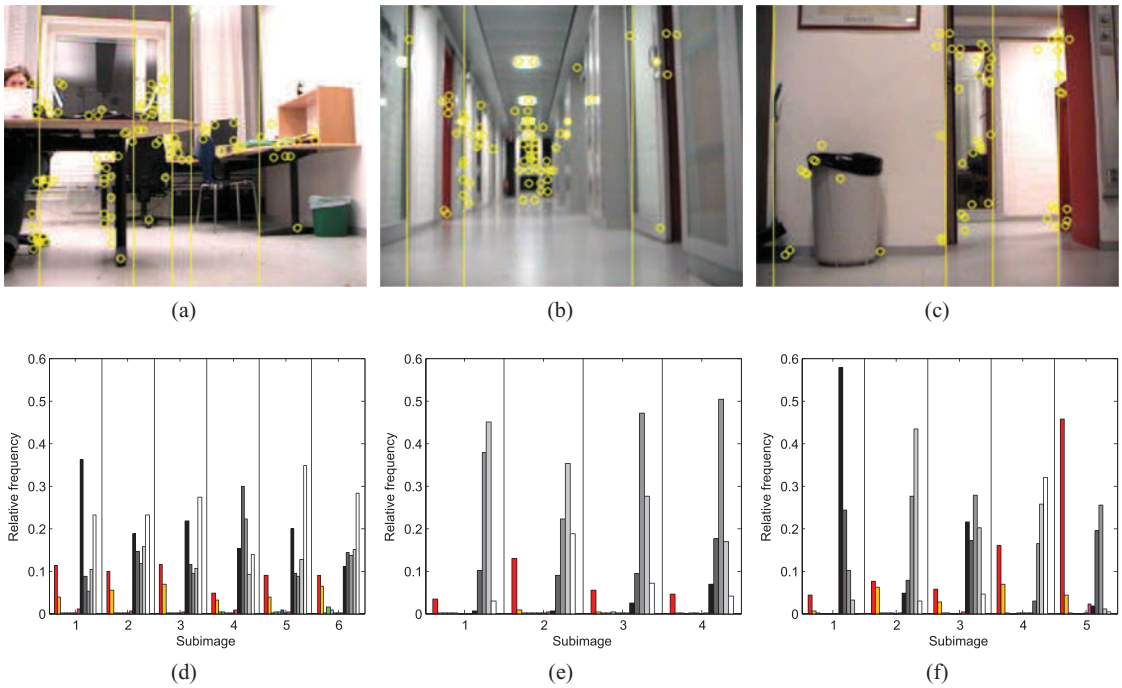


Fig. 6. Sample reference images of the room for two people (a), the corridor (b), and the kitchen (c) extracted from the Dumbo night 1 dataset with the identified vertical lines and Star keypoints superimposed. The corresponding color histograms (d)–(f) extracted from each image are presented underneath. In each of the histograms, the first eight bins represent the color components, whereas the last five are grayscale values.

To sum up, all the steps taken to perform feature extraction and matching are succinctly put forward in Algorithm 1.

4. Experimental Results and Discussion

The proposed fingerprint and matching procedure, programmed in C++ using the OpenCV library,[6] has been tested in two different environments: an office and a house. The former corresponds to the publicly available KTH-IDOL2 database.[20] The image set chosen was that captured in night illumination conditions with the PowerBot Dumbo robot, whose camera is mounted 36 cm above the floor. The robot was manually driven through the environment to acquire 309 × 240 images at 5 fps. By contrast, the home environment dataset was custom-made as, to our knowledge, there are no house image datasets that use monocular vision and guarantee that the camera's focal plane parallel to the building walls available online. The camera was installed at a height of approximately 30 cm and 640 × 480 images were acquired at 1 fps with a remotely controlled robot.

In both experiments, the environment was first modeled by selecting a few representative images from each of the different locations. For instance, if the robot was intended to explore a room, four or five captures that cover more or less 360° were picked. These captures were taken from different datasets than those used for testing. For every query frame, only the best representative from each location is employed to compute the final probabilities.

As mentioned in Section 2.3, SIFT,[19] Star features[1] described with upright SURF,[3] and ORB[26] have been tested and compared as keypoint detectors and descriptors for the VPACK.

In the KTH-IDOL2 dataset, there exist four different image collections for night illumination conditions. They will be referred to as Dumbo night 1 to 4 from now on. The four datasets are different from each other because there are people walking around and objects being used and moved. From Dumbo night 1, 22 reference images were selected to model the five locations present in the environment: printer area, corridor, room for two people, room for one person, and kitchen. Some sample images are shown in Fig. 6. Dumbo night 2 and 3 were used for testing. The Dumbo night 2 dataset consists of 952 images and was acquired the same day as Dumbo night 1, whereas

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

Algorithm 1: Fingerprint generation and N-gram matching procedure.

foreach *query image* **do**

Fingerprint generation (section 2)

Identify structural vertical lines to split the capture in subimages (Section 2.1).

1: Compute the 2nd order Sobel operator over the x direction.
2: Perform a 3 by 1 morphological dilation.
3: Apply a vertical morphological opening of size 1/5 of the image height.
4: Sum each column and compute a normalized graph of the vertical response.
5: Filter out those values lower than 5 times the mean.
6: Apply a NMS algorithm to ensure a minimum separation between lines (e.g. 5% of the image width).

foreach *subimage* **do**

Compute a 13-bin histogram that encodes color and grayscale information (Section 2.2).

1: Using the HCL bicone model, classify pixels as chromatic (chroma $>$ 12.5%) or achromatic (chroma $\leq$ 12.5%).
2: Build a 8-bin color histogram and a 5-bin grayscale histogram using a fuzzy voting scheme.
3: Stack both histograms together and normalize.

Extract and describe a reduced amount of robust keypoints (Section 2.3).

end

***N*-gram matching (section 3)**

for *histograms and keypoints independently* **do**

Obtain matching N-grams in the reference images used to represent different locations (Section 3.1).

if *histograms* **then**

foreach *representative image* **do**

1: For every histogram in the representative image, find the best match in the query image using the Hellinger distance (1). Keep only those that satisfy $d_H(Q, R) \leq 0.3$.
2: Compute the score for each match as $1 - d_H(Q, R)$.

end

else if *keypoints* **then**

1: Find the best match for each keypoint in the query image among the keypoints of all representative images using the Euclidean distance (or the Hamming distance for binary descriptors). A threshold on the distance can be optionally applied.
2: Compute the matching score for each reference subimage as

$$\frac{\#\text{ matched keypoints}}{\#\text{ keypoints in the reference subimage}}$$

end

3: Compute N-gram counts for each location $x_{n,k}$ using the matching scores (Fig. 4).

Build N-gram models using naive Bayes with Laplace smoothing from the N-gram counts (3).

$\Rightarrow$ **Output:** N histogram and N keypoint models (for unigrams, bigrams, and trigrams if $N = 3$).

Employ inverse entropy to combine the N-gram models.

1: Compute the entropy of every N-gram model (4).
2: Calculate the weighting factor of each model using inverse entropy (5) .
3: Obtain the weighted sum of the three models (6).

$\Rightarrow$ **Output:** One histogram and one keypoint model.

end

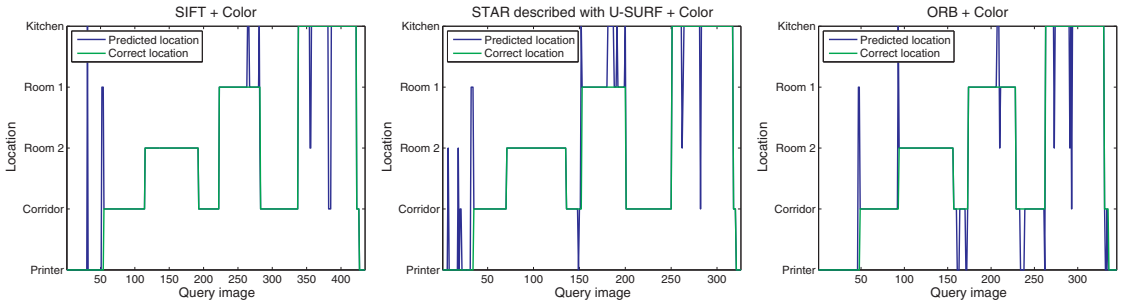
Multiply the histogram and keypoint probability distributions to obtain the final probabilities.

end

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

Table I. Results for Dumbo night 2 and 3 datasets for VPACK descriptor with different types of keypoints and thresholds. Both the number of images that are above the threshold and the percentage of correctly classified images are shown.

	Dumbo night 2				Dumbo night 3		
	Threshold	Images	Precision		Threshold	Images	Precision
SIFT	0.7	538	93.87%	SIFT	0.7	492	86.18%
	0.8	435	96.55%		0.8	332	93.98%
	0.9	327	99.08%		0.9	197	97.93%
Star	0.7	452	89.82%	Star	0.7	416	79.09%
	0.8	326	93.56%		0.8	271	86.35%
	0.9	206	97.09%		0.9	145	93.79%
ORB	0.7	429	86.01%	ORB	0.7	373	79.09%
	0.8	345	93.04%		0.8	235	86.38%
	0.9	239	97.91%		0.9	139	93.84%



(a) (b) (c)

Fig. 7. Comparison of the predicted and the ground-truth location with a threshold of 0.8 using SIFT (a), Star features described with upright SURF (b), and ORB (c).

Dumbo night 3, which contains 1034 images, was recorded four months later. As a consequence, there are noticeable differences between the latter and the reference images.

Table I presents the correctly classified instances for each of type of keypoint with different probability thresholds. This means that a prediction is considered valid if its probability is higher than this value. According to these results, it seems clear that the method performs well for all types of keypoints and that it is robust to changes that may occur in the environment over time, although SIFT is slightly ahead and helps to identify more images within the sequence.

A detailed analysis of the predictions obtained for Dumbo night 2 with a threshold of 0.8 is shown in Fig. 7. The sequence of visited places can be distinguished no matter which keypoint descriptor is used. The robot starts in the printer area, moves into the corridor, explores the room for two people, goes back to the corridor, enters the room for a single person, returns to the corridor, goes into the kitchen and ends in the printer area after going through the corridor briefly. Realize that most of the errors occur near the transition areas which are difficult to encode correctly. The other mistakes could be explained by the fact that the representative images do not cover the environment perfectly. They could be easily removed using a median filter.

For the home environment test, 12 images were used to represent six locations: kitchen, entrance hall, living room, terrace, grass area, and pergola. The results for the test dataset, which contains 372 captures, are shown in Table II. In this case, the precision exhibited is even better than in the KTH-IDOL2 dataset. This may be due to the fact that, conversely to an office, where all walls tend to be of the same color, and rooms are normally similar to each other, a house is much more distinctive.

Overall, the performance using SIFT seems to be somewhat better, but its main drawback is its high computational burden. The tests presented above were carried out on a second-generation Intel i5^{TM} CPU at 1.6 GHz. The average computation times are presented in Table III. If there is not much computational power available, either Star or ORB are fairly good alternatives to make VPACK lighter.

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

Table II. Classification results for the home environment dataset with different probability thresholds.

	Home environment		
	Threshold	Images	Precision
SIFT	0.7	308	97.40%
	0.8	290	98.62%
	0.9	255	99.61%
Star	0.7	291	94.16%
	0.8	267	95.51%
	0.9	271	98.62%
ORB	0.7	287	94.77%
	0.8	264	96.21%
	0.9	209	99.52%

Table III. Average computation times for both datasets.

	Computation times	
	Office	Home
SIFT	399 ms	792 ms
Star	174 ms	217 ms
ORB	70 ms	134 ms

5. Conclusion and Future Work

The new vision-only monocular VPACK fingerprint based on vertical lines, color histograms, and keypoints, along with the *N*-gram-based matching procedure presented in this paper have been proven effective for topological scene recognition in structured indoor environments. In addition, they are fairly robust to small changes that may occur over time and the combination of complementary features permits to employ weaker and faster descriptors in order to keep computing times under control without compromising precision. Even though it has been designed for and tested with monocular images, this method should be applicable to unwrapped panoramas too. Moreover, speed could be significantly improved if the algorithm is parallelized by taking advantage of the fact that vertical lines split the image into independent regions.

This method could be a starting point for the implementation a full topological SLAM system. To this end, an automatic node selection procedure should be developed in the first place. A way to determine when a new node is necessary could be to apply a change-point detection algorithm to VPACK features.

References

1. M. Agrawal, K. Konolige and M. R. Blas, "CenSurE: Center surround Extremas for Realtime Feature Detection and Matching," **In**: *European Conference on Computer Vision Lecture Notes in Computer Science* , vol. 5305. (D. Forsyth, P. Torr and A. Zisserman, eds.) (Springer, 2008) pp. 102–115.
2. A. Angeli, S. Doncieux, J.-A. Meyer and D. Filliat, "Incremental Vision-Based Topological SLAM," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* Nice, France (Sep. 22–26, 2008) pp. 1031–036.
3. H. Bay, A. Ess, T. Tuytelaars and L. van Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008).
4. G. Bologna, B. Deville and T. Pun, "Blind Navigation Along a Sinuous Path by Means of the See ColOr Interface," **In**: *Proceedings of the International Work–Conference on the Interplay Between Natural and Artificial Computation. Part II: Bioinspired Applications in Artificial and*

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press

Natural Computation (J. Mira, J. R. Álvarez, F. de la Paz and J. M. Ferrández, eds.) (Springer-Verlag, Santiago de Compostela, Spain, 2009) pp. 235–243.
5. O. Booij, B. Terwijn, Z. Zivkovic and B. Kröse, "Navigation Using an Appearance Based Topological Map," *Proceedings of the IEEE International Conference on Robotics and Automation,* Rome, Italy (Apr. 10–14, 2007) pp. 3927–3932.
6. G. Bradski, "The OpenCV library," (2000). `http://opencv.willowgarage.com`.
7. R. A. Brooks, "Elephants don't play chess," *Robot. Auton. Syst.* **6**, 3–15 (1990).
8. S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *Int. J. Math. Models Methods Appl. Sci.* **1**(4), 300–307 (2007).
9. D. Comaniciu, V. Ramesh and P. Meer, "Kernel-based object tracking," *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5), 564–577 (2003).
10. M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.* **27**(6), 647–665 (2008).
11. F. Fraundorfer, C. Engels and D. Nistér, "Topological mapping, localization and navigation using image collections," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* San Diego, USA (Oct. 29–Nov. 2, 2007) pp. 3872–3877.
12. M. Horvath, "ShapeGrid macro—Isometricland," (2008). `http://isometricland.net/povray/povray.php`; last accessed: Nov. 21, 2013.
13. D. Jurafsky and J. H. Martin, *Speech and Language Processing* (Prentice Hall, New Jersey 2009).
14. T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE Trans. Commun. Technol.* **15**(1), 52–60 (1967).
15. P. Lamon, I. Nourbakhsh, B. Jensen and R. Siegwart, "Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization," *Proceedings of the IEEE International Conference on Robotics and Automation,* Seoul, Korea (May 21–26, 2001) pp. 1609–1614.
16. L. LeCam, *Asymptotic Methods in Statistical Decision Theory* (Springer-Verlag, New York, 1986).
17. M. Liu, D. Scaramuzza, C. Pradalier, R. Siegwart and Q. Chen, "Scene Recognition with Omnidirectional Vision for Topological Map Using Lightweight Adaptive Descriptors." *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* St. Louis, USA (Oct. 11–15, 2009) pp. 116–121.
18. M. Liu and R. Siegwart, "DP-FACT: Towards Topological Mapping and Scene Recognition with Color for Omnidirectional Camera," *IEEE International Conference on Robotics and Automation,* Saint Paul, USA (May 14–18, 2012) pp. 3503–3508.
19. D. G. Lowe. "Distinctive image features from scale-invariant keypoints." *International Journal of Computer Vision*, **60**(2), 91–110 (2004).
20. J. Luo, A. Pronobis, B. Caputo and P. Jensfelt, "Incremental Learning for Place Recognition in Dynamic Environments," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* San Diego, USA (Oct. 29–Nov. 2, 2007) pp. 721–728.
21. M. Magimai-Doss, D. Hakkani-Tür, Ö. Çetin, E. Shriberg, J. Fung and N. Mirghafori, "Entropy Based Classifier Combination for Sentence Segmentation," *IEEE International Conference on Acoustics, Speech and Signal Processing*, Honolulu, HI, USA vol. 4 (Apr. 15–20, 2007) pp. IV-189–IV-192.
22. J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions," *Proceedings of the British Machine Vision Conference*, Cardiff, UK, vol. 1 (Sep. 2–5, 2002) pp. 384–393.
23. S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J. Mol. Biol.* **48**, 443–453 (1970).
24. A. Neubeck and L. van Gool, "Efficient Non-Maximum Suppression," *Proceedings of the International Conference on Pattern Recognition*, Hong Kong, China, vol. 3 (Aug. 20–24, 2006) pp. 850–855.
25. F. T. Ramos, B. Upcroft, S. Kumar and H. F. Durrant-Whyte. "A Bayesian Approach for Place Recognition." *Proceedings of the IJCAI Workshop Reasoning with Uncertainty in Robotics,* Edinburgh, Scotland (Jul. 30, 2005).
26. E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," *Proceedings of the IEEE International Conference on Computer Vision,* Barcelona, Spain (2011) pp. 2564–2571.

27. S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. (Pearson Education, New Jersey, 2010).
28. D. G. Sabatta, "Vision-Based Topological Map Building and Localisation Using Persistent Features," *Robotics and Mechatronics Symposium,* Bloemfontein, South Africa (Nov. 11, 2008) pp. 1–6.
29. B. J. Stankiewicz and A. A. Kalia, "Acquisition of structural versus object landmark knowledge," *J. Exp. Psychol.: Human Perception and Performance* **33**(2), 378–390 (2007).
30. A. Tapus, Topological SLAM – Simultaneous Localization and Mapping with Fingerprints of Places *PhD thesis* (Lausanne, Switzerland: École Polytechnique Fédérale de Lausanne, 2005).
31. A. Tapus and R. Siegwart, "Incremental Robot Mapping with Fingerprints of Places," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, AB, Canada (Aug. 2–6, 2005) pp. 2429–2434.
32. D.-C. Tseng and C.-H. Chang, "Color Segmentation Using Perceptual Attributes," *Proceedings of the IAPR International Conference on Pattern Recognition*, The Hage, Netherlands, vol. 3 (Aug. 30–Sep. 3, 1992) pp. 228–231.
33. I. Ulrich and I. Nourbakhsh, "Appearance-Based Place Recognition for Topological Localization," *Proceedings of the IEEE International Conference on Robotics and Automation,* San Francisco, USA (2000) pp. 1023–1029.
34. F. Werner, J. Sitte and F. Maire, "Topological map induction using neighbourhood information of places," *Auton. Robots* **32**(4), 405–418 (2012).

https://doi.org/10.1017/S0263574714001076 Published online by Cambridge University Press