Adaptive image processing methods for outdoor autonomous vehicles

Lucie Halodová[1], Eliška Dvořáková[1], Filip Majer[1], Jiří Ulrich[1], Tomáš Vintr[1], Keerthy Kusumam[2], Tomáš Krajník[1] *

[1] Artificial Intelligence Center, Faculty of Electrical Engineering, Czech Technical University
halodluc@fel.cvut.cz
[2] University of Nottingham

Abstract. This paper concerns with adaptive image processing for visual teach-and-repeat navigation systems of autonomous vehicles operating outdoors. The robustness and accuracy of these systems relies on their ability to extract relevant information from the on-board camera images, which is then used for autonomous navigation and map building. In this paper, we present methods that allow an image-based navigation system to adapt to varying appearance of outdoor environments caused by dynamic illumination conditions and naturally occurring environment changes. In the experiments performed, we demonstrate that adaptive and learning methods for camera parameter control, image feature extraction and environment map refinement allow autonomous vehicles to operate in real, changing world for extended periods of time.

1 Introduction

Digital cameras are gradually becoming one of the most important sensors used in mobile robots that are deployed in natural environments. Their popularity can be attributed to the low-price, small size and the fact that they can provide large amounts of data in real time. However, while a typical camera generates a significant amount of data over relatively short time periods, most of this data are not relevant to the tasks assigned to a robot. Thus, a crucial problem of vision-based systems is to tackle the extraction, storage and management of relevant information contained in the image streams. Ideally, a vision-guided robot should not only be able to extract information rich enough to perform the tasks at hand, but also to build upon and gradually improve its knowledge of the environment in order to perform its tasks efficiently.

To achieve that, the images acquired by the camera should contain a sufficient number of identifiable, salient elements that are pertinent to the task at hand. These elements have not only to be extracted but, most importantly, correctly interpreted and utilised for the required task. Interpretation of images becomes more efficient and reliable if the robot can anticipate the information it

* The work has been supported by the project 17-27006Y.

searches for and adapt the aforementioned phases, extraction and interpretation, accordingly. This requires the use of a knowledge base, which provides the robot with the description of salient image elements that the robot might encounter within different spatial and temporal contexts. A truly intelligent system should be able to build and refine this knowledge base during the course of its routine operation.

In the context of visual navigation, a robot must ensure that the images have sufficient contrast to identify an adequate number of environmental features necessary for position estimation of the robot relative to its goal. This not only means that the robot has to adapt the camera settings and feature extraction parameters properly, but also to carefully choose which of the obtained features qualify to store for later use and when to retrieve or discard them from its memory. In other words, the robot not only needs to make sure that it perceives information relevant to its environment model but also that it can update the environment model based on the perceived information. The ability to adapt the settings of the robot perception subsystems, as well as its knowledge bases, is especially crucial in environments which exhibit appearance and structural changes. The causes for these changes include short-term factors like varying illumination and unpredictable weather and long-term, partially predictable seasonal processes.

This paper, hence concerns with adaptive methods, which are specifically aimed at ensuring a reliable long-term operation of visually-navigated robots in outdoor environments. In particular, we will discuss methods which *(i)* control camera parameters to ensure sufficient quality of images obtained, *(ii)* adapt image preprocessing modules to extract a suitable number of salient image elements for self-localisation and mapping, *(iii)* gradually adapt the environment map, so that it stays up-to-date with the environment variations and *(iv)* interpret the temporal behaviour of the changes in the map and predict which elements of the map will be visible at a particular time and location. Apart from a detailed description of the adaptive methods, we demonstrate that their integration into a visual navigation pipeline significantly improves the efficiency of long-term mobile robot operation in outdoor, unstructured environments.

2 Related work

Visual navigation systems can be divided according to their working principle to map-less, map-based and map-building based [1]. Map-less systems such as [2] assume that the environment contains traversable structures such as highway lanes, pathways or roads. Thus, these systems attempt to identify the specific structures and control the autonomous vehicle in a way to keep it on these structures. Map-based systems do not assume that traversable structures in the robots environments will be easy to recognise. Instead, these systems navigate and localise the robots based on maps, which are known a priori [3]. Map-building-based systems are able to build these maps themselves typically by utilising the Simultaneous Localisation and Mapping (SLAM) [4–6] technology. These metric

Adaptive image processing methods for outdoor autonomous vehicles 3

maps are then used to determine the position of the robot relative to its goal, so that the robot can be driven along the intended trajectory. An alternative line of map-building approaches does not perform metric position estimation of the robot within the created maps. Instead, these systems use principles of visual servoing, which allow the robot to repeat the path it was taught by a human operator during a teleoperated drive [7–11]. While these 'teach-and-repeat' systems are somewhat less versatile, because the robot can only reach positions it was driven to before, they were reported to be successfully deployed for long periods of time [12–14]. Regardless of their principle, most of the aforementioned systems are based on a similar processing pipeline: they capture images, process them to extract salient features, match those features to environment representations they created beforehand and determine how to steer the robot so that it stays on the intended path. Some of these systems are able to gradually adapt their environment representations [13] or perception systems [15] to the changes they observe as they repeat the taught paths.

The first stage of the visual processing pipeline is the image acquisition, which determines if the images contain information relevant for further processing. This is mainly influenced by the robot camera exposure settings, which have to be adapted to varying illumination conditions. A fully autonomous system needs to adapt the camera exposure constantly to cope with varying illumination. However, typical built-in auto-exposure methods are not aimed for visual navigation and thus fail to set the exposure properly [16, 17]. Furthermore, while the impact of settings of subsequent processing stages can be analysed offline based on the images gathered, compensating for incorrect exposure setting is difficult.

Because of the aforementioned issues, several researchers proposed different methods of exposure setting. For example, Lu et al. [18] measure colour image quality by information-theoretic methods and propose to set the exposure to maximise the Shannon entropy of the RGB image and optimise the camera parameters by achieving similar values of exposure time and camera gain. Neves et al. [19] build a histogram of pixel intensities and set the camera exposure based on the mean sample value of that histogram. Additional to that, they take into account the size of underexposed and overexposed areas in the image. Shim et al. [16] base their exposure setting on the sum of gradients of individual pixel intensities in the image, which they aim to maximise. Zhang et al. [17] compare four gradient-based metrics that indicate the quality of images and conclude that Shim's method could be improved by the use of more advanced statistics and the photometric response function compensation [20]. To assess the quality of the images, Zhang uses the number of extracted FAST [21] features and tries to obtain as many FAST keypoints as possible. As the FAST features are used in many visual navigation systems, including the one which we use in this work [14], Zhang's method is highly relevant for the work presented herein.

Once an image is acquired successfully, the positions of its salient features (or keypoints) are extracted in a process called keypoint detection. A typical keypoint detection method, such as [21–23], assigns each pixel a given measure of saliency and reports pixels with saliencies which are locally maximal and exceed

a pre-set threshold. For example, a SURF [23] keypoint detection is based on a Hessian matrix determinant, which retrieves points in the image with sufficient contrast that makes them easy to localise and track. Since the aforementioned threshold is set prior to the keypoint detection process, it is hard to predict how many keypoints are going to be detected in the processed image. However, the number of detected features strongly influences the ability of the robot to successfully establish the relation of the current image to its environment model and thus, to navigate reliably. As shown in [24], setting the threshold too high and acquiring a small number of relevant features negatively impacts navigation accuracy. On the other hand, too low threshold produces an excessive number of features which do not contribute to the quality of navigation but hamper the ability of the system to match them to the map in real time. Thus, similarly to the camera exposure time, the saliency threshold needs to be continuously adapted to the images that are being processed.

Once the salient keypoints are detected, another set of methods, called feature descriptors, are applied to the vicinity of these keypoints. The keypoints' descriptors, which are typically engineered [22, 23, 25], are generally meant to be invariant to contrast, scale, rotation and viewpoint changes. However, these invariances are often not needed for autonomous navigation and other properties such as robustness to illumination and seasonal changes, are desired. This led to research of learning methods, which can generate feature description algorithms robust to illumination and seasonal changes [15, 26]. In their latest work, [27] even demonstrated that adaptation of the feature descriptors during routine autonomous operation of the robots [27] improves their ability for long-term operation. However, the feature description adaptation is planned to be included in our system in the future and is not subject to the evaluation in this work.

The extracted features, along with their positions and descriptions, are used either to build an environment map (teaching phase), or they are matched to the map to determine the robot's position relative to the intended path (repeat phase). However, as the environment changes, the features stored in the initially-taught map disappear and other features become visible. After a sufficiently long time, an environment might change its appearance so much, that the map becomes completely obsolete and irrelevant. Thus, long-term operation requires that the map is adapted to the changes during the robot operation. One of the most popular frameworks for map maintenance is based on 'experiences' [28], which allow representing the same location with several appearances or 'experiences', which depend on particular environmental conditions such as day/night or weather. During navigation, a robot retrieves several 'experiences' tied to a given location and tries to associate them with its current view. The failed association indicates that the appearance of the particular location changed drastically and the perceived appearance is added to the set of experiences associated with a given location. The experience-based approach allows not only smart management of the environment maps [29, 30], but it was successfully integrated into teach-and-repeat systems [13]. Another popular approach, inspired by the interplay of long- and short-term memory, is described in [31, 32], who gradually

add newly detected features and discard features which are no longer visible. A similar approach, called Summary Map [33], ranks all the map features based on their past visibility and uses the rank to remove or add the elements to the current map. Inspired by [31–33], we implemented a similar scheme, which gradually adapts the map during the routing operation of the robot. Unlike in [31–33], where the experimental evaluation is performed off-line, our navigation system updates the maps during its operation.

Once the navigation system can cope with the environment changes it observes during routine operation, its ability to operate for longer time periods is greatly improved. Robust, long-term operation opens the possibility of repeated re-observation of the same locations, which capture the long-term temporal behaviour of the environment variations. These observations allow the robot to create not only spatial but also spatiotemporal models of its operational environment. These models can predict, how a given location will look like at the time of robot operation. For example, Lowry et al. [34] use consecutive observations to distinguish between time variable and time-invariant image components. Other works [35–37] attempt to predict how a winter scene will look based on its appearance captured during summer and vice versa. Visibility of image features in a period of time is based on a temporal model, which is recomputed while repeatedly watching the same or similar places. Rosen et al. [38] assume that persistence of features is limited and they employ the survivability theory to create environmental models that predict, which elements are not likely to be visible anymore. Similarly, Krajník et al. [39] propose to use spectral analysis to capture the periodic behaviour of feature visibility caused by day/night and seasonal cycles. Our navigation system employs the FreMEn method proposed in [39, 40], to model the temporal behaviour of each feature.

Since the problem of long-term, reliable operation gradually comes into focus of the robotics research community, the aforementioned list of papers is by no means exhaustive. Thus, we refer to a comprehensive review of approaches for long-term visual localisation in [41] and a general review of AI methods for long-term autonomy in [42].

3 Adaptive navigation system

The navigation paradigm which we base our system on belongs to the group of teach-and-replay navigation systems. These systems allow mobile robots to autonomously traverse paths, through which they have been previously driven by human operators. In the teaching phase, a person guides the robot along the intended path and the robot creates an environment map. Once the map is created, the robot can navigate along the path autonomously, which is referred to as 'replay'. The underlying principle of the autonomous replay depends on the robot's sensory and computational equipment and on the way it represents the environment.

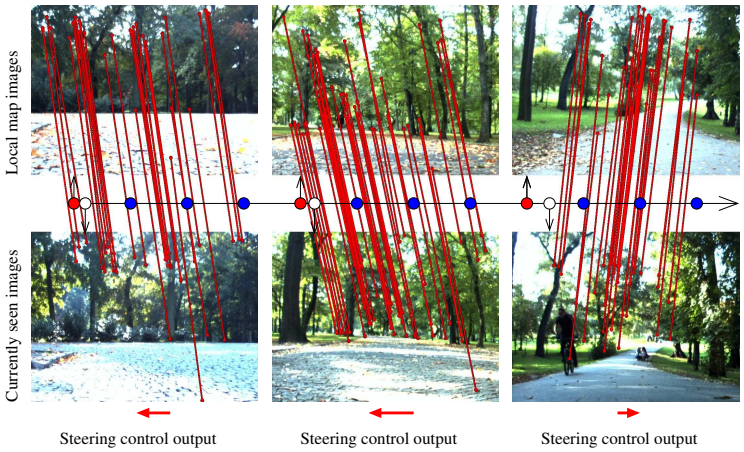

Fig. 1. Navigation method overview: During a teleoperated drive or 'teaching', a robot creates a sequence of local maps at regular intervals (blue circles). Each map contains the image captured at the given spot and the features extracted from the image. During the replay phase, a robot at a given distance from the path start (white circle) selects the closest map (red circle) and establishes correspondences between the visible and mapped features. Difference between the horizontal coordinates of the feature pairs, which corresponds to the horizontal shift of the images, determines the robot steering velocity (shown as red arrows at the bottom).

3.1 Navigation system core

In our case, the robot is equipped with a monocular camera and odometry. During the teaching phase, the robot uses the odometry to measure the travelled distance and once every certain distance it saves the latest captured image along with its features into a local map. Thus, at the end of the teaching phase, the taught path is represented as a sequence of local maps indexed by their distance from path start, see Figure 1. During the autonomous navigation or 'replay', the robot retrieves the local map according to its distance from the path start and matches the features extracted from its current camera image to the ones from the local map. Then, it subtracts the horizontal image coordinates of the corresponding pairs and uses a histogram voting scheme to determine the most prevalent difference δ. This process, referred to as image registration, aims to recover the horizontal shift δ between the image, which is currently seen and the image stored in the local map. The value of δ indicates not only the difference of the robot's current heading from the heading it had during the teaching but also its lateral displacement from the path. Thus, using a simple regulator to steer the robot in a way, which would keep the difference between the horizontal coordinates of the corresponding features close to zero, causes the robot to follow the taught path. The works [14, 12, 43] show that if a robot traverses a closed path repeatedly, the aforementioned steering correction scheme efficiently suppresses

both lateral and longitudinal errors of the robot position and keeps the robot on the taught path. The aforementioned navigation process is illustrated in Figure 1, and also in several videos available from `http://github.com/gestom/stroll_bearnav`.

The processing stages of the navigation pipeline along with the information flows are shown in Figure 2. A classic, non-adaptive visual navigation pipeline would compose only of the modules drawn in black, which acquire an image, extract its features, establish their correspondences with the map and calculate the robot steering. However, outdoor visual navigation requires that the navigation deals with changing illumination and environment variations. Thus, our system extends the classic pipeline by several components responsible for adaptation with the aforementioned variations – these modules are shown in Figure 2 in blue colour. These modules *control camera exposure* to ensure sufficient quality of captured images, *adapt feature extraction* to obtain a suitable number of image features, gradually *adapt maps* of the environment by removing obsolete features and adding new ones, and *predict* which features are going to be visible at a particular time and location. Since this paper is aimed at the evaluation

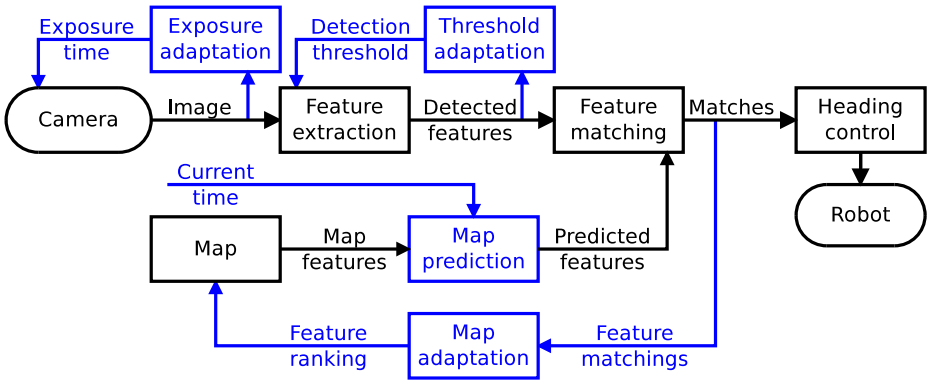

Fig. 2. Navigation system modules: The standard, core processing modules are drawn in black, and the modules responsible for adaptation to the environmental conditions are shown in blue.

of the adaptative methods on the quality of visual navigation, we provided only a coarse overview of the navigation method. For further details, please refer to [14, 43, 44]. In the following sections, we will explain how the individual modules (shown in blue in Figure 2) handle the adaptation of the core modules of the visual navigation.

3.2 Camera exposure control

The quality of the image features directly depends on the quality of the input image stream, which, in outdoor environments, is affected by varying illumination.

Therefore, we developed a simple method, which compensates unstable illumination by adaptively setting exposure of the robot on-board camera, which, in our case, does not provide us with a built-in automatic exposure setting. To do so, we had to edit the driver of e-Con TARA camera, which is used on most of our robots. Since the robot on-board camera is aimed forwards, the bottom half of the images typically contains the ground, which does not provide information particularly useful for robot navigation. Thus, our method attempts to control the camera exposure to keep the mean brightness of the top half of the images at a certain value.

Each time an i^{th} image is captured, we calculate the mean brightness b_i of its top half. Then we compare the desired brightness b_d to the actual one (b_i) and calculate the next exposure setting e_{i+1} according to:

$$e_{i+1} = e_i + c_e\, e_i\, (\frac{b_d}{b_i} - 1), \tag{1}$$

where e_i is current exposure setting, $c_e \in < 0, 1 >$ is the exposure control gain, b_d is the desired brightness and b_i is current brightness of the image top half. Since the exposure setting of the camera takes some time to take effect, we perform this calculation once per five frames obtained.

The equation 1 has two parameters which have to be set: the desired image brightness b_d and the exposure control gain c_e. To ensure quick response to the changes in illumination, while avoiding possible oscillations of the image brightness, we can set c_e to or just below the value of one. As the paper [45] showed that in challenging lighting conditions, slightly underexposed images are more likely to be registered correctly, we set the desired mean brightness b_d in the range of 0.4 - 0.5. While this exposure setting scheme is rather simple, the experiments shown in Section 4.2 indicate that the images obtained are more suitable than if the exposure was controlled automatically in the traditional way, i.e. according to the brightness of the entire image.

3.3 Feature detection adaptation

The accuracy and reliability of the visual navigation is affected by the number of features the detector extracted from the on-board camera image. However, the relation between the number of features and the quality of the navigation is not straightforward. While the higher number of extracted features typically results in more accurate registration, exceeding a certain amount of elements does not bring significant improvement [15]. Moreover, a high number of features takes more time to extract and match, which slows down the response of the robot to its position perturbances. Furthermore, a high number of features increases the chance of obtaining incorrect correspondences, which also negatively impacts the navigation accuracy. From the experiments performed in [15], it seems that the optimal number of features depends on the particular environment. However, the results in [15, 46] show that it's better to process the images in a way, which produces a certain, fixed number of features per image.

Adaptive image processing methods for outdoor autonomous vehicles 9

In the case of the SURF detector, which is used in our experiments, the number of the extracted features depends on the local image contrast and the value of the Hessian matrix threshold. Since the images the robot perceives along the path differ in contrast, setting a fixed threshold would produce a different number of features for each image. This often results in the deficiency of the features required for accurate image registration or in detection slowdown due to the excessive number of extracted features. Thus, one needs to adapt the Hessian threshold on the fly during the robot routine operation.

To achieve that, we adapt the Hessian threshold according to the number of features obtained in the last image. Let us assume that we want to obtain f_d features for further processing. Since we can always erase features that are in excess, but too many detected features would slow down the detection, we attempt to set the Hessian threshold to obtain a slightly higher number of features f'_d, where $f'_d = f_d\,(1 + o)$, where o stands for an 'overshoot' factor, which we set to 0.3 in our experiments. Assume that the last, i^{th} image with a Hessian threshold of t_i provided us with f_i features and we need to calculate the next threshold t_{i+1}. To do so, we order the detected features according to their response, i.e. the value of their Hessian matrix determinant, obtaining a sequence $\mathcal{H}$, where $h(1)$ is the Hessian response of the most prominent feature and $h(f_i)$ is the Hessian threshold of the least prominent detected feature. If the number of features f_i is higher than f'_d, then we simply set the t_{i+1} to the response of the feature, which is at position $(f'_d + 1)$ in the aforementioned set, i.e. $t_{i+1} = h(f'_d + 1)$. In case that $f_i < f'_d$, we take the 10 last values of $\mathcal{H}$ and use linear extrapolation to determine t_{i+1}. In particular, we use the values of $h(f_i - 10), h(f_i - 9) \ldots h(f_i)$ to estimated a linear function $h'(n)$ and then set t_{i+1} to $h'(f'_d)$.

As mentioned in the previous section, features that are most useful for heading correction typically appear in the upper half of the image. Thus, the detector is set up to extract the features from this upper half. The impact of the aforementioned Hessian threshold adaptation on the quality of visual navigation is evaluated in Section 4.4, which shows that extracting the features from the upper half of the image results in more accurate image registration.

3.4 Map adaptation

Outdoor, natural environment is subject to gradual, but perpetual change, which affects both its structure and appearance. As the environment changes, the map which was created during the teaching phase becomes gradually obsolete. Thus, to achieve long-term operation, a mobile robot should be able to adapt its environment representation to the changes it observes. Ideally, the adaptation of the map should be performed automatically during the routine robot operation, and it should not require human intervention.

Inspired by the methods presented in [31–33], we implemented a method, which allows to refine and update the environment maps during the routine robot navigation. The core idea is to evaluate the utility of the elements in the maps, keep the features which are useful, remove the ones which are often matched incorrectly, and add features which were not visible before. To do so,

we developed a system, which ranks the features according to their usefulness for navigation by continuously monitoring if they provide correct information about the robot heading. As mentioned in Section 3.1, our navigation system continuously retrieves the image features from the local maps in the robot vicinity, matches these features to the ones extracted from the camera image, and uses a histogram voting method to determine the most frequent difference in the horizontal coordinate of the feature pairs.

Thus, the system can assess the correctness of the established correspondences by comparing the horizontal difference of the feature pairs to the result provided by the histogram voting method. The Figure 3 illustrates the results of this assessment – the feature pairs, whose horizontal displacement is in consensus (which is established by the histogram voting), and are considered to be correctly established, are drawn in green. The feature pairs, whose horizontal coordinates differ from the consensus, are considered to be incorrectly established (these are drawn in red colour in Figure 3). Whenever the system establishes the correspondences and divides them into correct and incorrect, it increases the rating of the map features, which were correctly matched and decreases the rating of the map features, which produced incorrect correspondences. The score of unmatched features is not changed. In this way, features that contribute to the correct estimation of the robot heading gradually improve their rating, while map features that are often mismatched end up with the low rating.

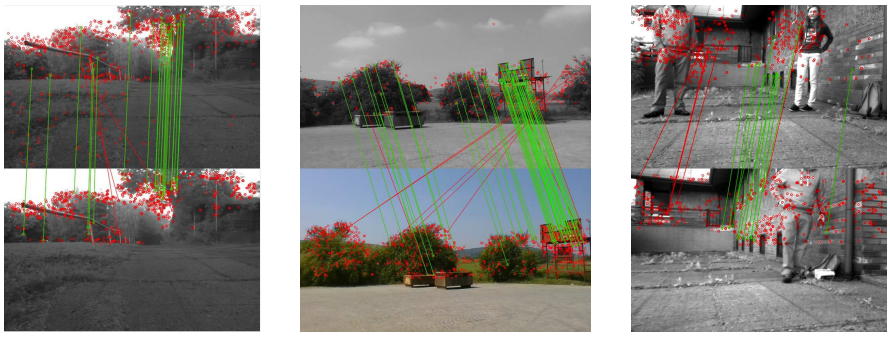

Fig. 3. Feature-based image registration results: the green correspondences are consistent with the results of the histogram voting, and the map features that belong to them will have their ranking increased. On the other hand, the correspondences in red are considered to be incorrect, and ranking of their map features will be decreased. Features with low ranking will be eventually removed from the map.

The system also rates unmatched 'view' features, i.e. features extracted from the current camera image, according to the distance of their descriptors to the features of the map. In particular, the rank of each view feature equals to its distance (in descriptor space) to the nearest feature in the local map. Such a rating corresponds to the feature saliency or uniqueness. Highly-rated view

features, which are unlikely to be mismatched and paired incorrectly, are good candidates to be added to the map.

Thus, to adapt the map, the system selects n map features with the lowest ranking (i.e. the ones which are often mismatched) and substitutes them with n view features with the highest ranking. One of the important questions is the choice of n, which defines how quickly the map adapts to the changes. If the n is low, the map cannot adapt to fast environmental changes, but it's robust to occasional glitches of the image registration, which might result in wrong features being added to the map. An extreme case is $n = 0$, which indicates that the map does not adapt to the environment change at all. If the n is high, the map can adapt to rapid changes. However, since the navigation algorithm does not establish the robot heading with perfect accuracy, the positions of the new features tend to drift, and the map gradually deteriorates. Moreover, any failure of the image registration populates the map with features at wrong positions. An extreme case is $n = f_i$, which means that the system discards all features and creates a completely new local map. The thesis [47] demonstrates that both extremes of $n = 0$ and $n = f_i$ are not suitable for long term navigation.

In our experiments, we set n to the half of the number of correctly-established correspondences, i.e. to the half of the value of the highest bin of the histogram voting. This value ensures that the map gradually adapts to the changes observed. Since n is much lower than the number of features that constitute the highest histogram bin, it makes the map update also prone to occasional failures of the navigation. If the navigation method fails and the features are added to the wrong positions, these wrong features do not cause the method to fail in the subsequent navigation run. Rather, the system will start to decrease their rank, because their positions will not conform to the results of the histogram voting, and these features will be eventually removed from the map. To ensure that the map always adapts to the changes, we set the minimal value of n to 10.

3.5 Map prediction

The previously presented map adaptation is capable of dealing with gradual environment changes. If the appearance change is significant and abrupt because the robot observes a given location after a large hiatus, no reliable correspondences will be established and the navigation method will fail. However, the visibilities of the features are not random, but they are strongly influenced by processes that exhibit certain temporal properties. For example, day-night, seasonal and vegetation cycles cause certain features to disappear and appear again with known periodicities, and tree growth rate decides the feature persistence. Since the adaptive map already provides the robot with the ability to operate for long time periods, the navigation system has the opportunity to learn about the influence of time on the visibility of the image features in its maps, i.e. to find the aforementioned cycles and map decay rates. In other words, the adaptive map provides sufficient information to create a spatiotemporal representation of the operational environment which allows predicting which features will be visible at which time. Thus, a robot can predict the environment appearance at

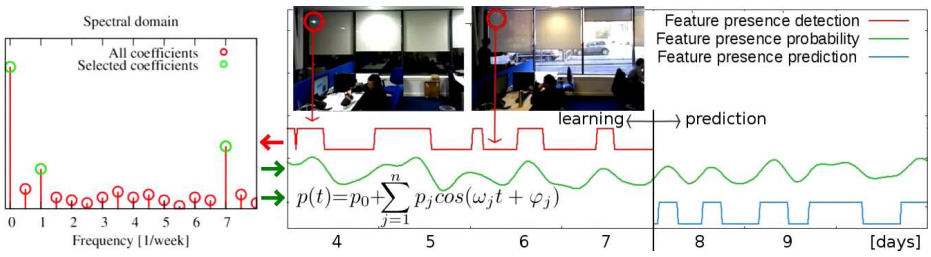

Fig. 4. FreMEn feature map for visual localization and navigation: The observations of image feature visibility (centre, red) are transferred to the spectral domain (left). The most prominent components of the model (left, green) constitute an analytic expression (centre, bottom) that represents the probability of the feature being visible at a given time (green). This allows to predict the feature visibility at a time when the robot performs visual navigation (blue). Courtesy of [40].

night even if it would radically differ from the appearance observed during the last robot operation.

To build such a predictive model, we capture and store the visibility information of each map feature along with the time it was (un-)detected. Once we have enough information, we can start to reason about the feature's persistence (i.e. how likely the feature will be visible in the future) [38], or periodicity [39, 40]. In particular, our system employs the concept of the Frequency Map Enhancement (FreMEn) [39], which retrieves the periodicities of the feature visibility through a Fourier transformation-based spectral analysis. The resulting model can predict the likelihood of each feature visibility in the future, which is especially beneficial if the robot visits a given area after a long time [48, 39]. The core principle of the FreMEn method is illustrated on the picture 4 and on the FreMEn project webpage `www.fremen.uk`.

Thus, every time our robot loads the environment maps for navigation, it creates temporal models of all features and calculates the likelihood of their visibility at the time of its operation. The most likely-to-be-seen features (selected by a greedy or a Monte-Carlo scheme), then constitute the map for a given navigation trial. As explained in [48], the Monte-Carlo scheme is applied to avoid situations, where some features are not detected simply because the system does not place them in the environment map. The main benefit of the FreMEn-based predictions comes from the fact that the FreMEn model captures the cyclic behaviour of the feature visibility caused by the changing position of the main outdoor illuminant, the Sun.

4 Experimental evaluation

To evaluate how the proposed adaptive methods affect the performance of the visual navigation, we tested them on a CAMELEON ECA tracked robot for difficult terrain. The robot was modified by installing an aluminium superstructure,

which contains several mounts for cameras, control laptop and other equipment. In the experiments presented here, we used images from the left camera of the e-Con TARA stereo device. For night experiments, we installed the Fenix 4000 lumen torch. The robot configuration used in our experiments is shown on Figure 5. The data gathering took place at Hostibejk Hill in Kralupy nad Vltavou,

Fig. 5. Cameleon robot configuration during the experimental trials.

Czechia, which is a small forest park with one building, see Figure 5. Near this building, the robot was taught a closed trajectory, and then it traversed the taught path autonomously more than 100 times over the course of one month in various environmental conditions ranging from cloudless days, overcast, light rain, sunset and night. Each time the robot used a local map to determine its steering, it saved its current image and associated it with the given local map. Since the taught path is represented by ∼80 local maps, and the robot traversed the path autonomously more than 100 times, the resulting dataset is composed of more than 8000 images. Approximately one-third of the images contain a small building otherwise, the images associated with the local maps contain trees, shrubs and other close and distant structures.

4.1 Evaluation metrics

The purpose of the image processing in our system is to register the images from the local maps to the images captured by the robot camera. The horizontal shift between the mapped and perceived images, which is recovered by establishing correspondences between these image features, is used to correct the robot heading. Thus, the more accurate and robust the image registration is, the more precise and reliable visual-based navigation would be. Therefore, the primary criterion to evaluate the visual navigation pipeline is the accuracy of the image registration.

To do so, we use the dataset images captured during the autonomous drive to simulate the robot movement by 'replaying' them to the robot navigation system.

The navigation system uses the histogram voting to calculate the horizontal shift between the dataset images and the images in the local maps in the same way as during normal operation. After that, we compare the calculated shift to the ground truth, obtained by manual annotation. The difference between the shift calculated by the system and the human-generated one is the measure of the method accuracy. Since during each simulated drive, the system registers several hundreds of images, we denote the result of i^{th} image registration as δ_i (see Section 3.1). By comparing the calculated δ_i to the values obtained by human annotation γ_i, we obtain a sequence $\epsilon_i = |\delta_i - \gamma_i|$ which corresponds to the accuracy of the image registration for i^{th} image pair.

Thus, evaluation of each method or its setting produces another error sequence of ϵ_i. To determine which method or which settings produce more accurate navigation, we compare the values ϵ_i produced by the aforementioned procedure statistically and qualitatively. For statistical evaluation, we apply Student's paired sample test, which is able to determine if an error sequence of ϵ_i^A generated by method A is statistically significantly smaller than another error sequence ϵ_i^B, generated by method B. For qualitative evaluation, we use the values of ϵ_i to calculate a function $p(\epsilon_t)$, which indicates the probability that the registration error was lower than a given threshold, i.e. $p(\epsilon) = P(\epsilon_i \leq \epsilon_t)$. While the first test clearly indicates if one method performs better than the other, displaying $p(\epsilon_t)$ of two different methods inside of the same graph provides one with a better insight how much these methods actually differ.

4.2 Camera exposure control

To evaluate the impact of camera exposure adaptation, described in Section 3.2, we let the robot traverse the taught path several times with three different exposure settings. To provide a fair experimental setting, we performed the trial during the late afternoon, when the illumination was quite stable. The first, 'standard' setting, used in our system by default, controls the camera exposure to keep the brightness of the upper image half at 0.5. The second, 'full' setting, controls the camera exposure to keep the brightness of the entire image at 0.5. The third, 'fixed' setting, does not perform adaptation. Rather, the operator sets the camera exposure at the beginning of the autonomous drive to a value, which causes the overall contrast of the image to be equal to 0.5. During each traversal, the robot collected 658 images, which were subsequently used for evaluation as described in 4.1. For examples of the images gathered, see Figure 6.

While the graph shown in Figure 7 seems to indicate only the small difference in the performance of the methods, the statistical tests confirmed that the 'standard' setting achieves statistically significantly lower registration errors compared to the 'full' and 'fixed' exposure settings. Later on, we performed the same test during sunset. Here, the robot was not able to navigate properly neither with 'full' nor 'fixed' exposure setting, but it worked with the 'standard' one. In this case, the navigation performance itself proved the superiority of the 'standard' exposure adaptation scheme, and it was not necessary to perform the statistical or qualitative evaluation.

Fig. 6. Different exposure settings. The first image is captured with fixed exposure, the second with exposure control applied to the complete image and the last image was captures when camera exposure was controlled according to the top half part of the images.

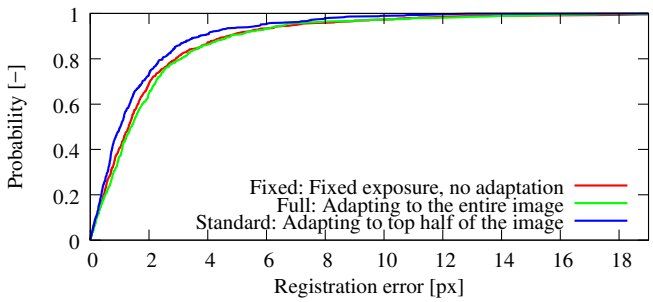

Fig. 7. Probability of registration error being smaller than a given number of pixels for different exposure adaptation schemes.

4.3 Feature detection adaptation

To evaluate the impact of feature number adaptation, proposed in Section 3.3 we used the dataset gathered with the 'standard' exposure setting. We replayed the dataset images through the navigation system as described in Section 4.1 using three different schemes of feature detection. The first, 'standard' setting, used in our system by default, controls the Hessian threshold as described in Section 3.3 in a way to extract 500 image features from the top image half. The second, 'full' setting, controls the Hessian threshold as described in Section 3.3 in a way to extract 500 image features from the entire image. The third, 'fixed' setting, does not adapt the Hessian threshold. Rather, the Hessian threshold is set to hand at the beginning of the simulated autonomous drive to obtain approximately 500 features from the first image. Examples of images with different keypoint detection strategy is shown in Figure 8.

While the graph, shown in Figure 9, seems to indicate only a small difference in the performance of the methods, the statistical tests confirmed that the 'standard' Hessian adaptation achieves statistically significantly lower registration errors compared to the 'full' case where the Hessian threshold is adapted to

Fig. 8. Different setting of the keypoints detection: In the first image, 500 features are detected only from the top half. In the second image, the same amount is extracted from the whole image, and the last image has a fixed Hessian threshold, which results in feature deficiency.

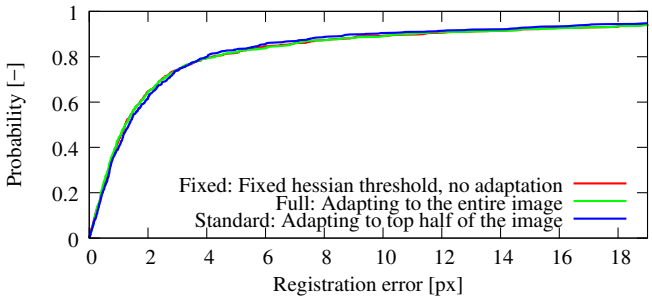

Fig. 9. Probability of registration error being smaller than a given number of pixels for different Hessian threshold adaptation schemes.

extract features from the entire image as well as to the 'fixed' case, where the Hessian threshold does not adapt but is fixed.

4.4 Map adaptation

To evaluate the effect of map adaptation, we the same dataset as in Section 4.3, i.e. the dataset gathered with the 'standard' exposure setting. Again, we replayed the dataset images through the navigation system as described in Section 4.1 using three different map adaptation schemes. The first, Adaptive map adaptation scheme, used in our system by default, gradually exchanges the map features as described in Section 3.4. The second, Plastic scheme, discards all features from the map and saves the ones from the current view. This scheme corresponds to setting the n parameter described in Section 3.4 to a very high value. The third, Static map, does not adapt the map during autonomous traversals, which corresponds the n parameter being set to 0. The ability of feature matching using Adaptive map and Static map is also shown in Figure 11. This time, the graph shown in Figure 10 clearly indicates that the Adaptive map achieves lower registration errors compared to the Plastic adaptation scheme, which is caused by

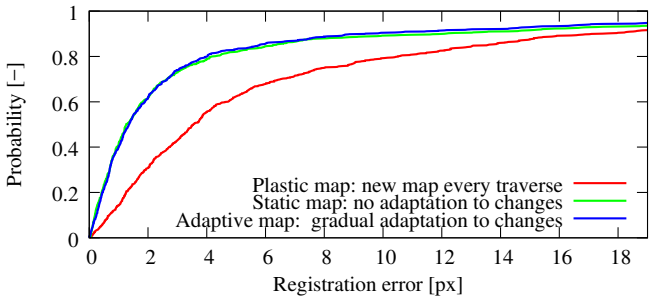

Fig. 10. Probability of registration error being smaller than a given number of pixels for different map adaptation schemes in environment without significant appearance changes.

the feature position drift, see [47] for details. Since the testing dataset does not exhibit any significant appearance changes, the performance of the Static map is comparable to the performance of the Adaptive one. This was also confirmed by the statistical tests.

To further test the ability of the adaptive mapping to deal with the changing environment appearance, we performed 12 autonomous traversals during sunset, where the robot started its traversals during daylight and ended at full dark. To allow the robot to autonomously navigate at night, we switched on its 4000lm torch, see Figure 5. As shown in Figure 11, the Adaptive map provides more rele-

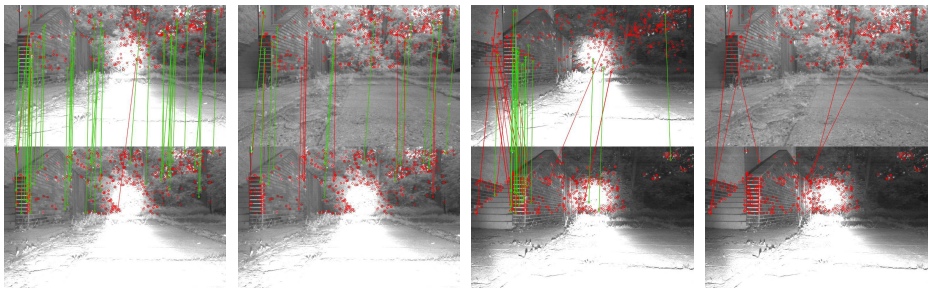

Fig. 11. Feature matching between the map and the current robot view: The first and the third image shows that the use of Adaptive map results in more correspondences compared to the Static map use, shown in the second and fourth image. The second image pair shows a situation, where the light changed from full day to full night, and the Static map, created during the day, becomes obsolete. The Adaptive map has enough correct correspondences due to using information from Static map augmented with features from later traversals.

vant features to the navigation system compared to the Static map. As indicated

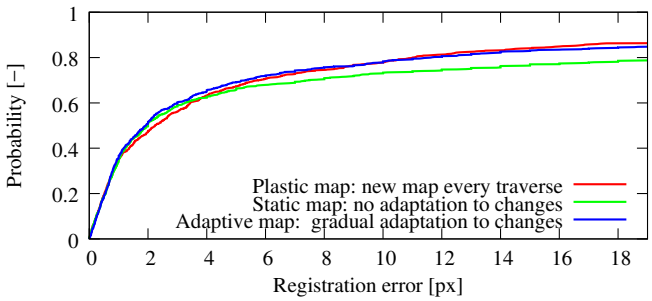

Fig. 12. Probability of registration error being smaller than a given number of pixels for different map adaptation schemes with significant appearance changes.

by the Figure 12 and the statistical tests performed, the Adaptive and Plastic maps also outperform the Static one in terms of registration accuracy. Note, that the effects of the Plastic map drift, observed in the previous experiment, are insigificant compared to the effects caused by the changes.

4.5 Map prediction

To evaluate the effect of map prediction, we processed data from 87 autonomous traversals. We divided the dataset into a training set of length 57, which is used to build temporal models of the features and a testing set, which we use for evaluation as described in Section 4.1. To predict the features which are going to be visible, we used the FreMEn [39] temporal model to predict 500 most likely visible features as specified in Section 3.5. Then, we simulated the robot drive on the testing dataset, where the maps were gathered both during the day and night.

The difference between the registration error of the system using and not using map prediction is shown in Figure 13. Statistical testing, performed as specified in Section 4.1 confirmed that the map predicted by the FreMEn and Monte Carlo scheme achieved lower registration error.

In comparison with the previous results, a higher error is more probable which was caused by insufficient illumination of late testing drives. The lack of light caused the camera to set longer exposure which caused significant blur in places where the robot had to turn. Figure 14 demonstrate the gradual deterioration of image quality in the testing set used. The images are taken from the same part of the robot path at a different time. Despite the blurry images, the map prediction method improved the directional correction.

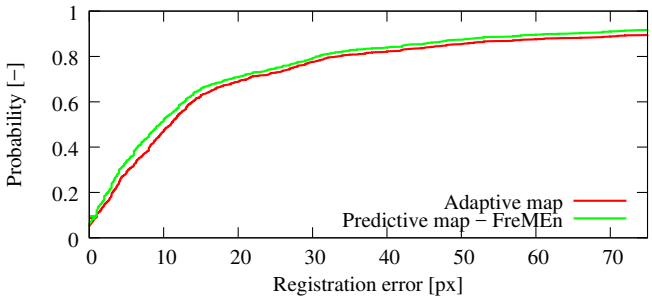

Fig. 13. Probability of registration error being smaller than a given number of pixels with standard and predicted maps.

Fig. 14. Image blur caused by the lack of light at a sharp turn. The top left image is from the beginning of testing dataset, and the bottom right image was captured at the end of data collection.

5 Conclusion

We have presented adaptative methods for a visual navigation system, intended for robots that are supposed to operate in outdoor environments for extended periods of time. The purpose of these methods is to deal with the appearance changes commonly occurring outdoors due to lighting variations and natural processes which affect the environment structure. In a series of experiments, performed on a real robot over the course of several weeks, we demonstrate that adaptation of camera exposure, feature extraction and map representation has a positive impact on the ability of the robots to autonomously navigate outdoors. Finally, we demonstrated that a robot operating for extended periods of time could acquire enough observations to understand how the environment changes over time and use this knowledge to predict the future environment appearance, which further improves the efficiency and reliability of its operation. To ensure the reproducibility of the research presented here research, we provide the system's source codes as well as access to the datasets used for evaluation at `https://github.com/gestom/stroll_bearnav/wiki`.

References

1. G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002.
2. P. De Cristóforis, M. Nitsche, and T. Krajník, "Real-time image-based autonomous robot navigation method for unstructured outdoor roads," *Journal of Real Time Image Processing*, 2013.
3. A. Kosaka and A. C. Kak, "Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainties," *CVGIP: Image understanding*, vol. 56, no. 3, pp. 271–329, 1992.
4. S. Holmes, G. Klein, and D. W. Murray, "A Square Root Unscented Kalman Filter for visual monoSLAM," in *International Conference on Robotics and Automation (ICRA)*, 2008, pp. 3710–3716.
5. R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
6. J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular slam," in *Eur. Conf. on Computer Vision*, 2014.
7. G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
8. Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, Minneapolis, USA, 1996.
9. E. Royer, M. Lhuillier, M. Dhome, and J.-M. Lavest, "Monocular vision for mobile robot localization and autonomous navigation," *International Journal of Computer Vision*, 2007.
10. Z. Chen and S. T. Birchfield, "Qualitative vision-based path following," *IEEE Transactions on Robotics and Automation*, 2009.
11. S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette, "Large scale vision based navigation without an accurate global reconstruction," in *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.
12. T. Krajník, J. Faigl, V. Vonásek et al., "Simple, yet Stable Bearing-only Navigation," *Journal of Field Robotics*, 2010.
13. M. Paton, K. MacTavish, L.-P. Berczi, S. K. van Es, and T. D. Barfoot, "I can see for miles and miles: An extended field test of visual teach and repeat 2.0," in *Field and Service Robotics*. Springer, 2018, pp. 415–431.
14. T. Krajník, F. Majer, L. Halodová, and Tomáš, "Navigation without localisation: reliable teach and repeat based on the convergence theorem," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
15. T. Krajník, P. Cristóforis, K. Kusumam, P. Neubert, and T. Duckett, "Image features for visual teach-and-repeat navigation in changing environments," *Robotics and Autonomous Systems*, 2017.
16. I. Shim, J.-Y. Lee, and I. S. Kweon, "Auto-adjusting camera exposure for outdoor robotics using gradient information," in *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*. IEEE, 2014, pp. 1011–1017.
17. Z. Zhang, C. Forster, and D. Scaramuzza, "Active exposure control for robust visual odometry in hdr environments," in *ICRA*, no. EPFL-CONF-228466, 2017.
18. H. Lu, H. Zhang, S. Yang, and Z. Zheng, "Camera parameters auto-adjusting technique for robust robot vision," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. IEEE, 2010, pp. 1518–1523.

19. A. J. Neves, B. Cunha, A. J. Pinho, and I. Pinheiro, "Autonomous configuration of parameters in robotic digital cameras," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2009, pp. 80–87.
20. P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *ACM SIGGRAPH 2008 classes*. ACM, 2008, p. 31.
21. E. Mair, G. D. Hager, D. Burschka, M. Suppa, and G. Hirzinger, "Adaptive and generic corner detection based on the accelerated segment test," in *European Conference on Computer Vision*, 2010.
22. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
23. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer Vision and Image Understanding*, 2008.
24. T. Krajník, P. Cristoforis, K. Kusumam, P. Neubert, and T. Duckett, "Image features for visual teach-and-repeat navigation in changing environments," *Robotics and Autonomous Systems*, vol. 88, pp. 127–141, 2017.
25. M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: binary robust independent elementary features," in *Proceedings of the ICCV*, 2010.
26. N. Carlevaris-Bianco and R. M. Eustice, "Learning visual feature descriptors for dynamic lighting conditions," in *IEEE/RSJ Int. Conference on Intelligent Robots and Systems (IROS)*, 2014.
27. N. hang, M. Warren, and T. Barfoot, "Learning place-and-time-dependent binary descriptors for long-term visual localization," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016.
28. W. S. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *IJRR*, 2013.
29. M. Gadd and P. Newman, "Checkout my map: Version control for fleetwide visual localisation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, South Korea, October 2016, pp. 5729–5736.
30. C. Linegar, W. Churchill, and P. Newman, "Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 90–97.
31. F. Dayoub and T. Duckett, "An adaptive appearance-based map for long-term topological localization of mobile robots," in *Proc. of Int. Conference on Intelligent Robots and Systems (IROS)*, 2008.
32. P. Biber and T. Duckett, "Dynamic maps for long-term operation of mobile service robots," in *RSS*, 2005.
33. P. Mühlfellner *et al.*, "Summary maps for lifelong visual localization," *Journal of Field Robotics*, 2016.
34. S. Lowry and M. J. Milford, "Supervised and unsupervised linear learning techniques for visual place recognition in changing environments," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 600–613, 2016.
35. P. Neubert *et al.*, "Appearance change prediction for long-term navigation across seasons," in *European Conf. on Mobile Robotics*, 2013.
36. H. Porav, W. Maddern, and P. Newman, "Adversarial training for adverse conditions: Robust metric localisation using appearance transfer," in *ICRA*, 2018.
37. Y. Latif, R. Garg, M. Milford, and I. Reid, "Addressing challenging place recognition tasks using generative adversarial networks," in *ICRA*, 2018.
38. D. M. Rosen, J. Mason, and J. J. Leonard, "Towards lifelong feature-based mapping in semi-static environments," in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1063–1070.

39. T. Krajník, J. P. Fentanes, J. M. Santos, and T. Duckett, "Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments," *IEEE Transactions on Robotics*, 2017.
40. T. Krajník, J. P. Fentanes, O. M. Mozos, T. Duckett, J. Ekekrantz, and M. Hanheide, "Long-term topological localization for service robots in dynamic environments using spectral maps," in *IROS*, 2014.
41. S. Lowry, N. Sünderhauf, P. Newman, J. J. Leonard, D. Cox, P. Corke, and M. J. Milford, "Visual place recognition: A survey," *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 1–19, 2016.
42. L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajnik, "Artificial intelligence for long-term robot autonomy: A survey," *IEEE Robotics and Automation Letters*, pp. 1–1, 2018.
43. F. Majer *et al.*, "A versatile visual navigation system for outdoor autonomous vehicles," in *Modeling and Simulation for Autonomous Systems*, 2018, in review.
44. F. Majer, L. Halodová, and T. Krajník, "Source codes: Bearing-only navigation." [Online]. Available: https://github.com/gestom/stroll_bearnav
45. L. Halodová and T. Krajník, "Exposure setting for visual navigation of mobile robots," in *Student Conference on Planning in Artificial Intelligence and Robotics (PAIR)*, 2017.
46. T. Krajník, P. Cristóforis, M. Nitsche, K. Kusumam, and T. Duckett, "Image features and seasons revisited," in *European Conference on Mobile Robots (ECMR)*. IEEE, 2015, pp. 1–7.
47. L. Halodová, "Map management for long-term navigation of mobile robots," Bachelor thesis, Czech Technical University, May 2018.
48. E. Dvořáková, "Temporal models for mobile robot visual navigation," B.S. thesis, Czech Technical Univerzity in Prague., 2018.