Attentional Landmarks and Active Gaze Control for Visual SLAM

Simone Frintrop and Patric Jensfelt

***Abstract*—This paper is centered around landmark detection, tracking and matching for visual SLAM (Simultaneous Localization And Mapping) using a monocular vision system with active gaze control. We present a system specialized in creating and maintaining a sparse set of landmarks based on a biologically motivated feature selection strategy. A visual attention system detects salient features which are highly discriminative, ideal candidates for visual landmarks which are easy to redetect. Features are tracked over several frames to determine stable landmarks and to estimate their 3D position in the environment. Matching of current landmarks to database entries enables loop closing. Active gaze control allows us to overcome some of the limitations of using a monocular vision system with a relatively small field of view. It supports (i) the tracking of landmarks which enable a better pose estimation, (ii) the exploration of regions without landmarks to obtain a better distribution of landmarks in the environment, and (iii) the active redetection of landmarks to enable loop closing in situations in which a fixed camera fails to close the loop. Several real-world experiments show that accurate pose estimation is obtained with the presented system and that active camera control outperforms the passive approach.**

***Index Terms*—Mobile robotics, visual SLAM, landmark selection, visual attention, saliency, active camera control**

I. Introduction

WHAT do I see? This is one of the most important questions for a robot that navigates and localizes itself based on camera data. What is "seen" or "perceived" at a certain moment in time is firstly determined by the images acquired by the camera and secondly by the information extracted from the images. The first aspect is usually determined by the hardware, but if a steerable camera is available, it is possible to actively direct the camera to obtain useful data. "Useful" refers here to data which supports improving the current task, e.g. localization and map building. The second aspect is especially important in tasks based on visual data since the large amount of image data together with real-time constraints make it impossible to process everything. Selecting the most important data is one of the most challenging tasks in this field.

SLAM is the task of simultaneously estimating a *model* or *map* of the environment and the robot's position in this map. The map is not necessarily a 3D reconstruction of the world, it is a representation that allows the robot to localize itself. Based on range sensors such as laser scanners, SLAM has reached a rather mature level [1], [2], [3], [4], [5]. Visual SLAM instead attempts to solve the problem with cameras as external sensors [6], [7], [8], [9], [10], [11]. This is desirable because cameras are low-cost, low-power and lightweight sensors which may be used in many applications where laser scanners are too expensive or too heavy. In addition, the rich visual information allows the use of more complex feature models for position estimation and recognition. On the other hand, visual SLAM is considerably harder, for example for the reasons given above.

A key competence in visual SLAM is to choose useful landmarks which are easy to track, stable over several frames, and easily re-detectable when returning to a previously visited location. This *loop closing* is important in SLAM since it decreases accumulated errors by distributing information from areas with lower uncertainty to those with higher. Furthermore, the number of landmarks should be kept under control since the complexity of SLAM typically is a function of the number of landmarks in the map. Landmarks should also be well distributed over the environment. Here, we suggest the application of a biologically motivated attention system [12] to find salient regions in images. Attention systems are designed to favor regions with a high uniqueness such as a red fire extinguisher on a white wall. Such regions are especially useful for visual SLAM because they are discriminative by definition and easy to track and redetect. We show that salient regions have a considerably higher repeatability than Harris-Laplacians and SIFT keypoints.

Another important part of our system is the gaze control module. The strategy to steer the camera consists of three behaviours: a *tracking* behaviour identifies the most promising landmarks and prevents them from leaving the field of view. A *redetection* behaviour actively searches for expected landmarks to support loop-closing. Finally, an *exploration* behaviour investigates regions with no landmarks, leading to a more uniform distribution of landmarks. The advantage of the active gaze control is to obtain more informative landmarks (e.g. with a better baseline), a faster loop closing, and a better distribution of landmarks in the environment.

The contributions of this paper are first, a landmark selection scheme which allows a reliable pose estimation with a sparse set of especially discriminative landmarks, second, a precision-based loop-closing procedure based on SIFT descriptors, and finally, an active gaze control strategy to obtain a better baseline for landmark estimations, a faster loop closing, and a more uniform distribution of landmarks in the environment. Experimental results are presented to show the performance of the system. This paper builds on our previous work [8], [13], [14] and combines all this knowledge into one system.

S. Frintrop is with the Institute of Computer Science III, Rheinische Friedrich-Wilhems-Universität, 53117 Bonn, Germany e-mail: frintrop@iai.uni-bonn.de

P. Jensfelt is with the Centre for Autonomous Systems (CAS), Royal Institute of Technology, 10044 Stockholm, Sweden patric@csc.kth.se

In the following, we first give an overview over related work (sec. II), then we introduce the SLAM architecture (sec. III). Sec. IV, V, and VI describe the landmark selection and matching processes and VII introduces the active camera control. Sec. VIII shows the performance of the SLAM system in several real-world scenarios and illustrates the advantages of active camera control. Finally, we finish with a conclusion.

II. Related Work

As mentioned in the introduction, there has been large interest in solving the visual SLAM problem during the last years [6], [7], [8], [9], [10], [11]. One of the most important issues in this field are landmark selection and matching. These mechanisms directly affect the ability of the system to reliably track and redetect regions in a scene and to build a consistent representation of the environment. Especially in loop closing situations, matching of regions has to be largely invariant to viewpoint and illumination changes.

The simplest kind of landmarks are artificial landmarks like red squares or white circles on floor or walls [15], [16]. They have the advantage that their appearance is known in advance and the re-detection is easy. While a simple solution if the main research focus is not on the visual processing, this approach has several obvious drawbacks. First, the environment has to be prepared before the system is started. Apart from the effort this requires, this is often not desired, especially since visual landmarks are also visible for humans. Second, landmarks with uniform appearance are difficult to tell apart which makes loop closing hard. Another approach is to detect frequently occurring objects like ceiling lights [17]. While this approach does not require a preparation of the environment, it is still dependent on the occurrence of this object.

Because of these drawbacks, current systems determine landmarks which are based on ubiquitous features like lines, corners, or blobs. Frequently used is the *Harris corner detector* [18] which detects corner-like regions with a significant signal change in two orthogonal directions. An extension to make the detector scale-invariant, the *Harris-Laplacian detector* [19], was used by Jensfelt et al. for visual SLAM [8]. Davison and Murray [6] find regions with a version of the Harris detector to large image patches (9×9 to 15×15) as suggested by Shi and Tomasi [20]. Newman and Ho [21] used *maximally stable extremal regions (MSERs)* [22] and in newer work [9] *Harris affine regions* [23]. In previous work, we used a combination of attention regions with Harris-Laplacian corners [13].

Here, we show that attention regions alone can be used as landmarks which simplifies and speeds up the system. Many attention systems have been developed during the last two decades [24], [25], [12]. They are all based on principles of visual attention in the human visual system and adopt many of their ideas from psychophysical and neuro-biological theories [26], [27], [28]. Here, we use the attention system VOCUS [12], which is capable to operate in real-time [29].

Attention methods are well suited for selecting landmark candidates since they favor especially discriminative regions in a scene, nevertheless, their application to landmark selection has rarely been studied. Nickerson et al. detect landmarks in hand-coded maps [30], Ouerhani et al. built a topological map based on attentional landmarks [31], and Siagian and Itti use attentional landmarks in combination with the gist of a scene for outdoor Monte-Carlo Localization [32]. The only approach we are aware of which uses an approach similar to a visual attention system for landmark detection for SLAM, is presented in [33]. They use a saliency measure based on entropy to define important regions in the environment primarily for the loop closing detection in SLAM. However, the map itself is built using a laser scanner.

Landmarks can only be detected and re-detected if they are in the field of view of the robot's sensor. By actively controlling the viewing direction of the sensors much can be gained. The idea of actively controlling the sensors is not new. Control of sensors in general is a mature discipline that dates back several decades. In vision, the concept was first introduced by Bajcsy [34], and made popular by Active Vision [35] and Active Perception [36]. In terms of sensing for active localization, Maximum Information Systems are an early demonstration of sensing and localization [37]. Active motion to increase recognition performance and active exploration was introduced in [38]. More recent work has demonstrated the use of similar methods for exploration and mapping [39]. Active exploration by moving the robot to cover space was presented in [40] and in [41] the uncertainty of the robot pose and feature locations were also taken into account. In [42] an approach for active sensing with ultrasound sensors and laser-range finders in a localization context is presented. When cameras are used as sensors, the matching problem becomes more difficult but includes also a higher information content. In the field of object recognition, [43] show how to improve the recognition results by moving the camera actively to regions which maximize discriminability.

In the field of visual SLAM, most approaches use cameras mounted statically on a robot. Probably the most advanced work in the field of active camera control for visual SLAM is presented by Davison and colleagues. In [6], they present a robotic system which chooses landmarks for tracking which best improve the position knowledge of the system. In more recent work [44], [11], they apply their visual SLAM approach to a hand-held camera. Active movements are done by the user, according to instructions from a user-interface [44], or they use the active approach to choose the best landmarks from the current scene without controlling the camera [11].

III. System Overview

This paper describes a system for visual SLAM using an attention-based landmark selection scheme and an active gaze control strategy. This section gives an overview of the components in the system. The visual SLAM architecture is displayed in Fig. 1. Main components are a *robot* which provides camera images and odometry information, a *feature detector* which finds regions of interest (ROIs) in the images, a *feature tracker* which tracks ROIs over several frames and builds landmarks, a *triangulator* which identifies useful landmarks, a *database* in which triangulated landmarks are stored, a *SLAM module* which builds a map of the environment, a *loop closer* which

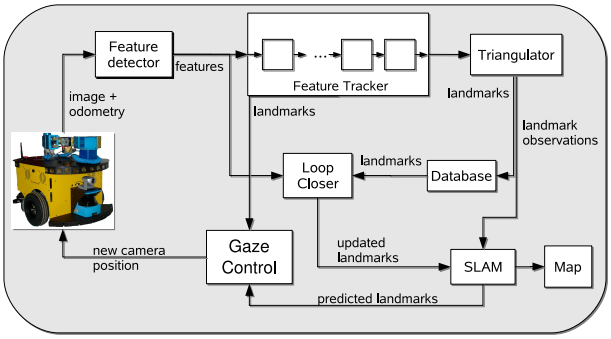


Fig. 1. The active visual SLAM system estimates a map of the environment from image data and odometry.

matches current ROIs to the database and a *gaze control module* which determines where to direct the camera to. The robot used in the experiments is an ActivMedia PowerBot equipped with a Canon VC-C4 pan/tilt/zoom camera mounted in the front of the robot at a height of about 0.35m above the floor. The ability to zoom is not used in this work.

When a new frame from the camera is available, it is provided to the *feature detector*, which finds ROIs based on a visual attention system. Next, the features are provided to the *feature tracker* which stores the last n frames, performs matching of ROIs in these frames and creates landmarks. The purpose of this buffer is to identify features which are stable over several frames and have enough parallax information for 3D initialization. These computations are performed by the *triangulator*. Selected landmarks are stored in a *database* and provided to the EKF-based *SLAM module* which computes an estimate of the position of landmarks and integrates the position estimate into the map. Details about the robot and the SLAM architecture can be found in [8]. Notice that the inverse depth representation for landmarks [45] would have allowed for an undelayed initialization of the landmarks. However the main purpose of the buffer in this paper is for selecting what landmarks are suitable for inclusion in the map and it would thus still be used had another SLAM technique been applied.

The task of the *loop closer* is to detect if a scene has been seen before. Therefore, the features from the current frame are compared with the landmarks in the database. The *gaze control module* actively controls the camera. It decides whether to track currently seen landmarks, to actively look for predicted landmarks, or to explore unseen areas. It computes a new camera position which is provided to the robot.

IV. Features and Landmarks

As mentioned before, landmark selection and matching belong to the most important issues in visual SLAM. A *landmark* is a region in the world. It has a 3D location and an appearance. A *feature* on the other hand is a region in an image. It has only a 2D location in the image and an appearance. The distance to the feature is initially not known since we use a monocular vision system. To build landmarks, features are detected in each frame, tracked over several frames and finally, the 3D position of the landmark is estimated by triangulation.

Feature selection is performed with a *detector* and the matching with a *descriptor*. While these two mechanisms are often not distinguished in the literature (people talk e.g. about "SIFT-features"), it is important to distinguish between them. A stable detector is necessary to redetect the same regions in different views of a scene. In applications like visual SLAM with time and memory constraints, it is also favorable to restrict the amount of detected regions. A powerful descriptor on the other hand has to capture the image properties at the detected region of interest and enable a stable matching of two regions with a high detection and low false detection rate. It has to be able to cope with viewpoint variations as well as with illumination changes. In this section, first the feature detection is introduced which finds ROIs in images (IV-A), then the descriptors which describe ROIs (IV-B), and finally the strategy to match two ROIs based on the descriptors (IV-C).

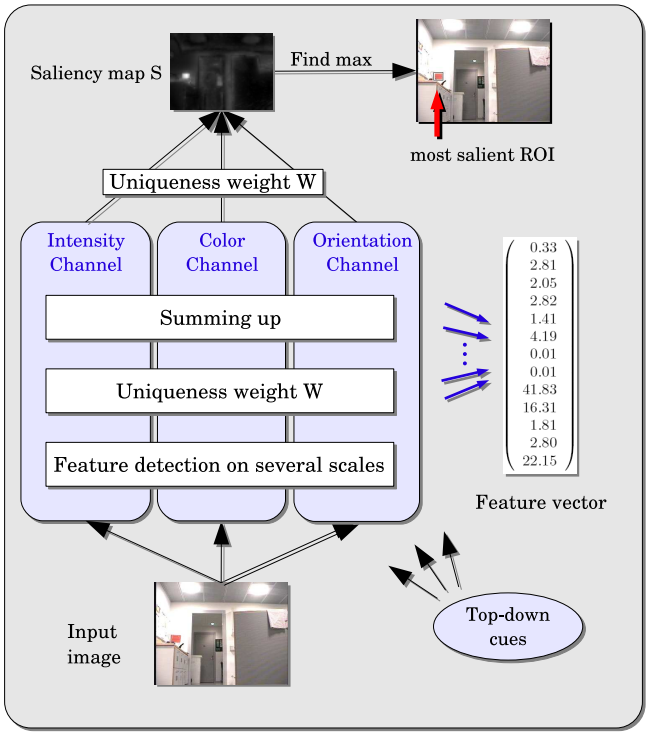
A. Attentional Feature Detection

An ideal candidate for selecting a few, discriminative regions in an image is a visual attention system. Computational attention systems select features motivated from mechanisms of the human visual system: several feature channels are considered independently and strong contrasts and the uniqueness of features determine their overall saliency. The resulting regions of interest have the advantage that they are highly discriminative, since repeated structure is assigned low saliency automatically. Another advantage is that there are usually only few regions detected per image (on average between 5 to 20), reducing the amount of features to be stored and matched considerably.

The attention system we use is VOCUS (Visual Object detection with a CompUtational attention System) [12]. VOCUS consists of a bottom-up part which computes saliency purely based on the content of the current image and a top-down part which considers pre-knowledge and target information to perform visual search. Here, we consider only the bottom-up part of VOCUS, however, top-down search can be used additionally if a target is specified.[1] For the approach presented here, any real-time capable attention system which computes a feature vector for each region of interest could be used.

An overview of VOCUS is shown in Fig. 2. The bottom-up part detects salient image regions by computing image contrasts and the uniqueness of a feature. The computations for the features intensity, orientation, and color are performed on 3 different scales with image pyramids. The feature intensity is computed by *center-surround mechanisms*; in contrast to most other attention systems [24], [31], on-off and off-on contrasts are computed separately. After summing up the scales, this yields 2 intensity maps. Similarly, 4 orientation maps ($0\,^\circ, 45\,^\circ, 90\,^\circ, 135\,^\circ$) are computed by Gabor filters and 4 color maps (green, blue, red, yellow) which highlight salient

[1] In [46] we found that in tracking situations, bottom-up matching outperforms top-down search, for loop-closing, top-down search is preferable. But since using the top-down mechanism requires a target, rather precise expectations about expected landmarks are necessary. If the system searches for many expected landmarks in each frame this slows down the system considerably since the top-down search has to be applied for each expectation.



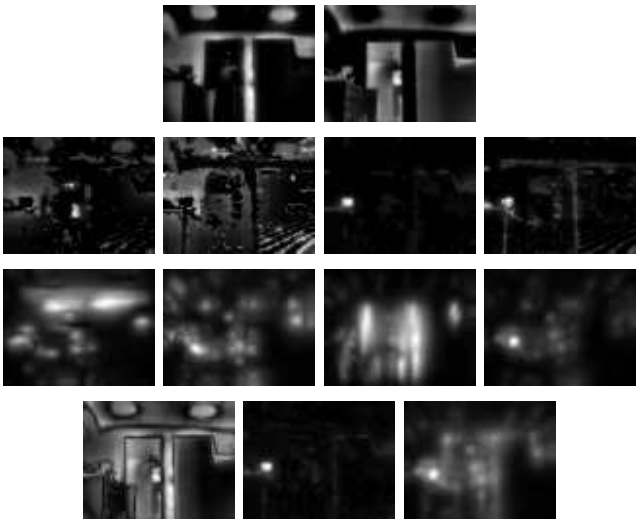


Fig. 2. Left: the visual attention system VOCUS detects regions of interest (ROIs) in images based on the features intensity, orientation, and color. For each ROI, it computes a feature vector which describes the contribution of the features to the ROI. Right: The feature and conspicuity maps for the image on the left. Top-left to bottom-right: intensity on-off, intensity off-on, color maps green, blue, red, yellow, orientation maps 0°, 45°, 90°, 135° and conspicuity maps I, C, O. Since the red region sticks out as a unique peak in the feature map *red*, this map is weighted strongly by the uniqueness weight function and the corresponding region becomes the brightest in the saliency map (left, top).

regions of a certain color. Before the features are fused, they are weighted according to their *uniqueness*: a feature which occurs seldomly in a scene is assigned a higher saliency than a frequently occurring feature. This is a mechanism which enables humans to instantly detect outliers like a black sheep in a white herd [26], [27]. The uniqueness $\mathcal{W}$ of map X is defined as

$$\mathcal{W}(X) = X/\sqrt{m}, \tag{1}$$

where m is the number of local maxima that exceed a threshold and '$/$' is here the point-wise division of an image with a scalar. The maps are summed up to 3 conspicuity maps I (intensity), O (orientation) and C (color) and combined to form the *saliency map*:

$$S = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C) \tag{2}$$

From the saliency map, the brightest regions are extracted as *regions of interest (ROIs)*. This is done by first determining the maxima (brightest points) in the map and then finding for each maximum a surrounding region with *seeded region growing*. This method finds recursively all neighbors with sufficient saliency. For simpler storing of ROIs, we approximate the region here by a rectangle.

The output of VOCUS for one image is a list of ROIs, each defined by 2D location, size and a feature vector (see next section). The feature and conspicuity maps for one example image are displayed in Fig. 2, right.

Discussion on Feature Detection: The most common feature detectors for visual SLAM are corner-like features as SIFT keypoints [47] or Harris-Laplacian points [19]. These approaches are usually based on the idea that many features are extracted and a few of them show to be useful for tracking and matching.[2] Matching these features between frames to find stable ones, matching to existing landmarks, storing landmarks in the database, and matching current features to the database requires considerable time. With intelligent database management based on search trees, it is possible to store and access a large amount of features in real-time [8], [48], [49]. Nevertheless, solving the task equally well with less features is favorable and enables to use computational power and storage for other processes. To enable the system to use only few features, it is necessary to have a detector which computes discriminative features and is able to prioritize them.

We claim that an attention system is especially well suited to detect discriminative features and that the repeatability of salient regions is higher than the repeatability of non-salient regions and of features detected by standard detectors. The *repeatability* is defined as the percentage of regions which are redetected in a subsequent frame (cf. [23]). While an exhaustive analysis is beyond the scope of this paper, a

[2]We obtained in average 400 – 500 Harris-Laplace features per frame. Computing these features together with a SIFT descriptor required 250 ms per frame.

few experiments shall illustrate this.[3] The precondition for the following experiments is that one or a few object(s) or region(s) in the scene are salient (a *salient* region differs from the rest of the scene in at least one feature type).

In the experiment in Fig. 3, we compare an image sequence showing many white and one green object. For humans, the green object visually pops out of the scene, so it does for VOCUS. We compared the performance of VOCUS with two other detectors: Harris-Laplace corners and SIFT keypoints, i.e. extrema in DoG scale space, since these are the most commonly used detectors in visual SLAM scenarios.[4] To make the approaches comparable, we reduced the number of points by sorting them according to their response value and using only the points with the strongest response. We compared whether this response can be used to obtain a similar result as with salient regions.

We determined the repeatability of regions over 10 frames for different amounts of detected features.[5] The result of the comparison is shown in Fig. 3. The highest repeatability is naturally obtained for the most salient region: it is detected in each frame. The strongest Harris-Laplace feature and the strongest SIFT keypoint on the other hand are in a subsequent frame only detected at the same position in 20% of the images. We compared the repeatability up to 11 features per frame since this is the average number of features detected by the attention system in our experiments. It shows that the repeatability of attentional ROIs is consistently higher than the one of the other detectors. It remains to mention that the repeatability of Harris-Laplace features and SIFT points goes up when computing more features, repeatability rates of about 60% have been reported for Harris-Laplacians in [23]. Note that our point here is that with attentional ROIs it is possible to select very few discriminative features with high repeatability, which is not possible with the other, locally operating detectors.

To show that the results in these simple experiments also extend to longer image sequences and to more natural settings, some videos showing qualitative results can be found on http://www.informatik.uni-bonn.de/∼frintrop/research/saliency.html. While these experiments illustrate the advantages of salient regions for visual SLAM, more detailed experiments will be necessary to investigate the differences of the different detectors in different settings.

Another aspect to mention is the accuracy of the detectors. The Harris-Laplace detector is known to be very precise and to obtain sub-pixel accuracy. Attention regions on the other hand are not as precise, their position varies sometimes a few pixels from frame to frame. This is partially due to

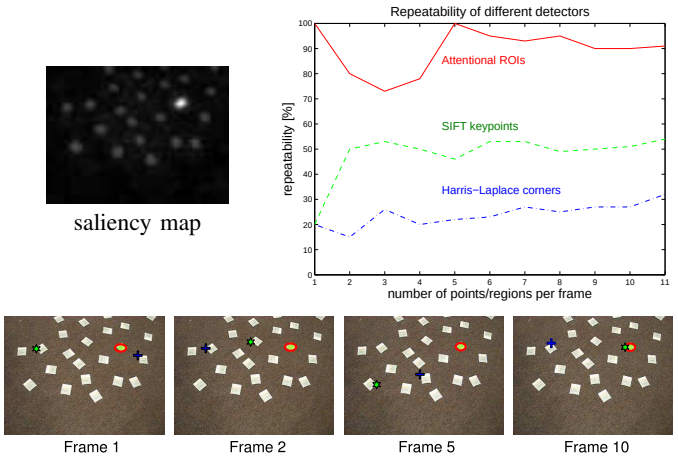


Fig. 3. Comparison of the repeatability of attentional ROIs (red ellipses), Harris-Laplace corners (blue crosses), and SIFT keypoints (green stars) on 10 frames of a sequence with a visually salient object (bottom: some example frames with detected features. top left: saliency map of 1st frame). The most salient attention region is detected in all frames (100% repeatability), the strongest point of the other detectors reaches only 20% (see also videos on http://www.informatik.uni-bonn.de/∼frintrop/research/saliency.html).

the segmentation process which determines the region. In previous work, we therefore combined Harris-Laplace corners and attention regions [13]. Tracking of landmarks with this approach was accurate and the matching process based on two descriptors resulted in a very low false detection rate. A problem however was that the detection rate also was very low: both detectors had to detect a feature in the same area and both descriptors had to agree on the high reliability of a match.

Using only attention regions with reasonable accuracy is possible with an improved outlier rejection mechanism during the triangulation process (cf. sec. V); this made the system considerably simpler and about 8 times faster.

B. The Descriptors

To compare if two image regions belong to the same part in the world, each region has to have a description vector. The most simple vector is a vector consisting of the pixel values of the region and possibly some surrounding. The similarity of two vectors can then be computed by cross-correlation. However, this results in high-dimensional vectors and matching does not perform well under image transformations.

An evaluation of more powerful descriptors is provided in [50]. The best performance was obtained for the SIFT descriptor (scale invariant feature transform [47]) and the GLOH descriptor (gradient location-orientation histogram) – an extension of the SIFT descriptor. The SIFT descriptor is also probably the most used descriptor in visual tasks for mobile robots [51], [7], [8], [10].

In this work, we use two kinds of descriptors: first, we determine an attentional descriptor for tracking ROIs between consecutive frames. The attentional descriptor can be obtained almost without cost from the feature maps of VOCUS. Since it is only an 13-element vector, matching is faster than with the

[3]We did not compare the detectors on standard datasets as in [23] because these have been designed for tasks like object recognition and do not contain especially salient regions. Therefore, the advantages of salient regions cannot be shown there.

[4]We used the publically available PYRA real-time vision library for both detectors (http://www.csc.kth.se/∼celle/).

[5]For this comparison, VOCUS was adapted to compute all local maxima from the saliency map to make it comparable to the Harris detector. In normal usage it determines only regions which have a saliency of at least 50% of the most salient region.

SIFT descriptor. It is less powerful, but in tracking situations sufficient. Second, we use the SIFT descriptor to match ROIs in loop closing situations.

The *attentional descriptor* is determined from the values of the 10 feature and 3 conspicuity maps of VOCUS. For each ROI, a feature vector $\vec{v}$ with 13 entries is determined, which describes how much each feature contributes to the ROI (cf. Fig. 2). The value v_i for map X_i is the ratio of the mean saliency in the target region $m_{(ROI)}$ and in the background $m_{(image-ROI)}$:

$$v_i = m_{(ROI)}/m_{(image-ROI)}. \tag{3}$$

This computation does not only consider which features are the strongest in the target region but also which features separate the region best from the rest of the image (details in [12]).

The *SIFT descriptor* is a $4 \times 4 \times 8 = 128$ dimensional descriptor vector which results from placing a 4×4 grid on a point and calculating a pixel gradient magnitude at $45°$ intervals for each of the grid cells. Usually, SIFT descriptors are computed at intensity extrema in scale space [47] or at Harris-Laplacians [19]. Here, we calculate one SIFT descriptor for each ROI. The center of the ROI provides the position and the size of the ROI determines the size of the descriptor grid. The grid should be larger than the ROI to allow catching information about the surrounding but should also not include too much background and stay within the image borders.[6]

C. Feature Matching

Feature matching is performed in two of the visual SLAM modules: in the feature tracker and in the loop closer.

In the tracker, we apply simple matching based on attentional descriptors. Two vectors $\vec{v}$ and $\vec{w}$ are matched by calculating the similarity $d(\vec{v}, \vec{w})$ according to a distance similar to the Euclidean distance [13]. This simple matching is sufficient for the comparably easy matching task in tracking situations.

In the loop closer, SIFT matching is applied to achieve a higher matching stability. Usual approaches to perform matching based on SIFT descriptors are *threshold-based matching*, *nearest neighbor-based matching* and *nearest neighbor distance ratio matching* [50]. For each ROI in the image, we use threshold-based matching to find a fitting ROI in the database. Then, we apply nearest neighbor matching in the other direction to verify this match.[7]

The distance d_S of two SIFT descriptors is calculated as the sum of squared differences (SSD) of the descriptor vectors. Thresholding on the distance between two descriptors is a bit tricky. Small changes on the threshold might have unexpected effects on the detection quality since the dependence of distance and matching precision is not linear (cf. Fig. 4).

Therefore, we suggest a slightly modified thresholding approach. By learning the dependence of distance and matching precision from training data, it is possible to set directly a threshold for the precision from which the corresponding distance threshold is determined.

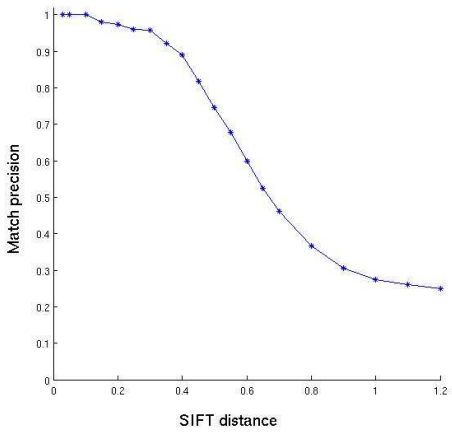


Fig. 4. The dependence of the distance of two SIFT descriptors and their matching precision (cf. eq. 4) determined from training data.

This is done as follows: for a large amount of image data, we gathered statistics regarding the distribution of correct and false matches. 698 correct matches and 2253 false matches were classified manually to obtain ground truth. We used data from two different environments, one was the office environment shown in Fig. 11, the other a different environment not used in the experiments. The training data for the office environment was obtained one year earlier than the test data for the current experiments.[8] Since the d_S are real values, we discretized the domain of d_S into $t = 20$ values. For the t distinct distance threshold values θ_j, we compute the *precision* as

$$p(\theta_j) = \frac{c(\theta_j)}{c(\theta_j) + f(\theta_j)}, \qquad \forall\, j \in \{1..t\} \tag{4}$$

where $c(\theta_j)$ and $f(\theta_j)$ denote the number of correct and false matches. The resulting distribution is displayed in Fig. 4.

To finally determine if two ROIs match, the distance of the SIFT descriptors is computed and the corresponding matching precision is determined according to the distribution in Fig. 4. If the precision is above a threshold, the ROIs match.[9]

Discussion on Feature Matching: The precision-based matching has several advantages over the usual thresholding. First, it is possible to choose an intuitive threshold like "98% matching precision".[10] Second, linear changes on the threshold result in linear changes on the matching precision. Finally,

[6]We chose a grid size of 1.5 times the maximum of width and height of the ROI.

[7]Mikolajczyk and Schmid show that the nearest neighbor and nearest neighbor distance ratio matching are more powerful than threshold-based matching but also point out that they are difficult to apply when searching in large databases [50].

[8]Correct matches are naturally much more difficult to obtain than false matches since there is a extremely large amount of possible false matches. To enable a reasonable amount of correct matches, we considered only distances below 1.2. As can be seen in Fig. 4, this does not affect the final matching mechanism as long as a precision of at least 0.3 is desired.

[9]For our system, we chose a threshold of 0.98. We chose a high threshold because an EKF SLAM system is sensitive to outliers.

[10]Note however that the precision value refers to the training data, so in test data the obtained precision might be lower than the specified threshold. However, the threshold gives a reasonable approximation of the precision on test data.

for every match a precision value is obtained. This value can be directly used by other components of the system to treat a match according to the precision that it is correct. For example, a SLAM subsystem which can deal with more uncertain associations could use these values.

The SIFT descriptor is currently one of the most powerful descriptors, however, people have worked on improving the performance, e.g. by combining it with other descriptors. While intuitively a good idea, we suggest to be careful with this approach. In previous work, we matched ROIs based on the attentional and the SIFT descriptor [14]. While obtaining good matching results, we found out that using only the SIFT descriptor results in a higher detection rate for the same amount of false detections. While surprising at first, this might be explained as follows: a region may be described by two descriptors, the perfect descriptor d_1 and the weaker descriptor d_2. d_1 detects all correct matches and rejects all possible false matches. Combining d_1 with d_2 cannot improve the process, it can only reduce the detection rate by rejecting correct matches.

V. The Feature Tracker

In the feature tracker, *landmarks* are built from ROIs by tracking the ROIs over several frames. The *length* of a landmark is the number of elements in the list, which is equivalent to the number of frames the ROI was detected in.

To compute the landmarks, we store the last n frames in a buffer (here: $n = 30$). This buffer enables to determine which landmarks are stable over time and therefore good candidates for the map. The output from the buffer is thus delayed by n frames but in return quality assessment can be utilized before using the data. New ROIs are matched with their attentional feature vector to previously detected landmarks and to ROIs from the previous frame to build new landmarks (details in [14]). At the end of the buffer, we consider the length of the resulting landmarks and filter out too short ones (here ≤ 3). Finally, the triangulator attempts to find an estimate for the location of the landmark. In this process, also outliers, i.e. bearings that fall far away from the estimated landmark location, are detected and removed from the landmark. These could be the result of mismatches or a poorly localized landmark.

VI. Loop Closing

In the loop closing module, it is detected if the robot has returned to an area where it has been before. This is essential to update the estimations of landmark and robot positions in the map. *Loop closing* is done by matching the ROIs from the current frame to landmarks from the database. It is possible to use position prediction of landmarks to determine which landmarks could be visible and thus prune the search space, but since this prediction is usually not precise when uncertainty grows after driving for a while, we perform "global loop closing" instead without using the SLAM pose estimate, as in [33]. That means, we match to all landmarks from the database. For the environments in our test it is possible to search the whole database in each iteration. However, for larger environments it would be necessary to use e.g. a tree-structure to organize the database, perform global loop closing less frequently or distribute the search over several iterations.

Fig. 6. Falsely matched ROIs (rectangles): in both cases, lamps are matched to a different lamp. Top: current frame. Bottom: frame from the database.

ROIs are matched to the landmarks from the database with the precision matching based on SIFT descriptors described in sec. IV-C. When a match is detected, the coordinates of the matched ROI in the current frame are provided to the SLAM system, to update the coordinates of the corresponding landmark. Additionally, the ROI is appended to the landmark in the database. Some examples of correct matches in loop closing situations are displayed in Fig. 5. False matches occur seldomly with this approach. If they do, the ROIs usually correspond to almost identical objects. Two examples are shown in Fig. 6.

VII. Active Gaze Control

The active gaze control module controls the camera according to three behaviours:

- Redetection of landmarks to close loops
- Tracking of landmarks
- Exploration of unknown areas

The strategy to decide which behaviour to choose is as follows: Redetection has the highest priority, but it is only chosen if there is an expected landmark in the possible field of view (def. see below). If there is no expected landmark for redetection, the *tracking* behaviour is activated. Tracking should only be performed if more landmarks are desired in this area. As soon as a certain amount of landmarks is obtained in the field of view, the *exploration* behaviour is activated. In this behaviour, the camera is moved to an area without landmarks. Most times, the system alternates between tracking and exploration, the redetection behaviour is only activated every once in a while (see sec. VII-A and Fig. 8). An overview over the decision process is displayed in Fig. 7. In the following, we describe the respective behaviours in more detail.

Fig. 5. Some examples of correctly matched ROIs, displayed as rectangles. Top: current frame. Bottom: frame from the database.

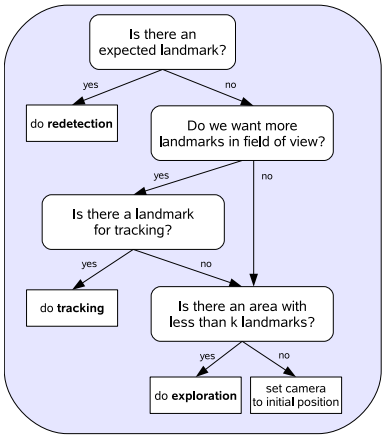


Fig. 7. The three camera behaviours *Redetection*, *Tracking*, *Exploration*.

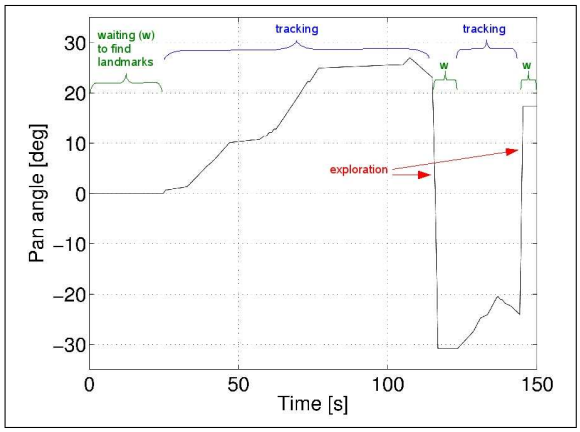


Fig. 8. The camera pan angle as a function of time. The camera behaviour alternates here between tracking and exploration.

A. *Redetection of Landmarks*

In redetection mode, the camera is directed to expected landmarks. *Expected landmarks*

(a) are in the potential field of view of the camera,[11]
(b) have low-enough uncertainty in the expected positions relative to the camera,[12]
(c) have not been seen recently, [13]
(d) had no matching attempt recently.

If there are several expected landmarks, the most promising one is chosen. Currently, we use a simple approach: the longest landmark is chosen because a landmark which has been observed frequently is more likely to be redetected than a seldomly observed one. In future work, we consider integrating information theory to choose the landmark that will result in the largest information gain, as e.g. in [44].

When a landmark has been chosen, the camera is moved to focus it and pointed there for several (here 8) frames, until it is matched. Note, that redetection and matching are two independent mechanisms: active redetection only controls the camera, matching is permanently done in the loop closer, also if there is no expected landmark.

If no match is found after 8 frames, the system blocks this landmark and chooses the next expected landmark or continues with tracking or exploration.

B. *Tracking of Landmarks*

Tracking a landmark means to follow it with the camera so that it stays longer within the field of view. This enables better triangulation results. This behaviour is activated if the preconditions for redetection do not apply.

First, one of the ROIs in the current frame has to be chosen for tracking. There are several aspects which make a landmark useful for tracking. First, the length of a landmark is an important factor for its usefulness since longer landmarks are more likely to be triangulated soon. Second, an important factor is the horizontal angle of the landmark: points in the direction of motion result in a very small baseline over several

[11] The potential field of view of the camera is set to $\pm 90°$ horizontally and $7m$ distance. This prevents considering landmarks which are too far away, since these are probably not visible although they are in the right direction.

[12] The uncertainty is considered as too high if it exceeds the image size, i.e. if the uncertainty of the landmark in pan-direction, projected to the image plane, is larger than the width of the image. Note, that these are actually the most useful landmarks to redetect, but on the other hand the matching is likely to fail. Passive matching attempts for these landmarks are permanently done in the loop closer, only the active redetection is prevented.

[13] The redetection behaviour focuses on landmarks which have not been visible for a while (here: 30 frames) to prevent switching the camera position constantly. The longer a landmark had not been visible, the more useful is usually its redetection.

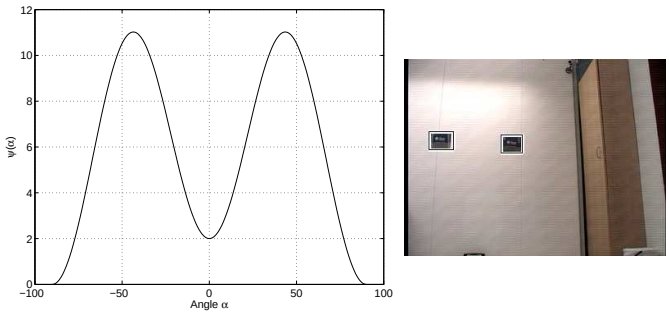


Fig. 9. Left: function $\psi(\alpha)$ with $k_1 = 5$ and $k_2 = 1$. Right: One test image with two (almost) identical ROIs, differing only by their position in the image. The center ROI has the angle $\alpha_1 = 0.04$ resulting in $\psi(\alpha_1) = 2.06$. The left ROI has a larger angle $\alpha_2 = 0.3$ resulting in $\psi(\alpha_2) = 5.09$ ($> \psi(\alpha_1)$). The tracking behaviour selects the left ROI for tracking and prevents it from moving out of the image.

frames and hence often in poor triangulations. Points at the side usually give much better triangulation results, but on the other hand they are more likely to move outside the image borders soon so that tracking is lost.

We define a usefulness function capturing the length l of the landmark and the angle α of the landmark in the potential field of view as

$$U(L) = \psi(\alpha)\sqrt{l} \tag{5}$$

where

$$\psi(\alpha) = k_1\,(1.0 + \cos(4\alpha - 180)) + k2\,(1.0 + \cos(2\alpha)). \tag{6}$$

The function is displayed in Fig. 9, left, and an example is shown in Fig. 9, right. Like in redetection mode, integrating the information gain could improve this estimation. After determining the most useful landmark for tracking, the camera is directed into the direction of the landmark.[14] The tracking stops when the landmark is not visible any more or when it was successfully triangulated.

C. Exploration of Unknown Areas

As soon as there are enough (here more than 5) landmarks in the field of view, the exploration behaviour is started, i.e., the camera is directed to an area within the possible field of view without landmarks. We favor regions with no landmarks over regions with few landmarks since few landmarks are a hint that we already looked there and did not find more landmarks.

We look for a region which corresponds to the size of the field of view. If the camera is currently pointing to the right, we start by investigating the field directly on the left of the camera and vice versa. We continue the search in that direction, in steps corresponding to the field of view. If there is no landmark, the camera is moved there. Otherwise we switch to the opposite side and investigate the regions there. If no area without landmarks is found, the camera is set to the initial position.

[14]The camera is moved slowly (here 0.1 radians per step), since this changes the appearance of the ROI less than large camera movements. This results in a higher matching rate and prevents to loose other currently visible landmarks.

To enable building of landmarks over several frames, we let the camera focus one region for a while (here 10 frames). As soon as a landmark for tracking is found, the system will automatically switch behaviour and start tracking it (cf. Fig. 8).

VIII. Experiments and Results

We tested the system in two different environments: an office environment and an atrium area at the Royal Institute of Technology (KTH) in Stockholm. In both environments, several test runs were performed, some at day, some at night to test differing lighting conditions. Test runs were performed during normal work days, therefore they include normal occlusions like people moving around. The matching examples in Fig. 5 show that loop closing is possible anyway.

For each run, the same parameter set was used. During each test run, between 1200 and 1800 images with 320×240 pixels were processed. In the office environment, the robot drove the same loop several times. This has the advantage that there are many occasions in which loop closing can take place. Therefore, this is a good setting to investigate the matching capability of the system. On the other hand, the advantage of the active camera control is not obvious here since loop closing is already easy in passive mode. To test the advantages of the active camera mode, the atrium sequence fits especially well. Here, the robot drove an "eight", making loop closing difficult in passive mode because the robot approaches the same area from three different directions. Active camera motion makes it possible to close the loop even in such difficult settings.

The current system allows real-time performance. Currently, it runs on average at ~ 90 ms/frame on a Pentium IV 2 GHz machine. Since the code is not yet optimized, a higher frame rate should be easily achievable by standard optimizations. Although VOCUS is relatively fast with ~ 50 ms/frame since it is based on integral images [29], this part requires about half of the processing time. If a faster system is required, a GPU implementation of VOCUS is possible, as realized in [52].

The experiments section has two parts. First, we investigate the quality of the attentional landmarks. Second, we compare active and passive camera control.

A. Visual SLAM with Attentional Landmarks

In this section, we investigate the quality of landmark detection, of data association in loop closing situations, and the effect on the resulting maps and robot trajectories. We show that we obtain a high performance with a low number of landmarks. Loop closing is obtained easily even if only few landmarks are visible and if they are seen from very different viewpoints.

In the first experiment, the same trajectory was driven three times in the office environment. Fig. 10 shows the robot trajectory which was determined from pure odometry (left) and from the SLAM process (right). Although the environment is small compared to other scenarios of the literature, it is well visible that the odometry estimation becomes wrong quickly. The estimated end position differs considerably from the real end position. The SLAM estimate on the other hand (right), is much more accurate. During this run, the robot acquired 17

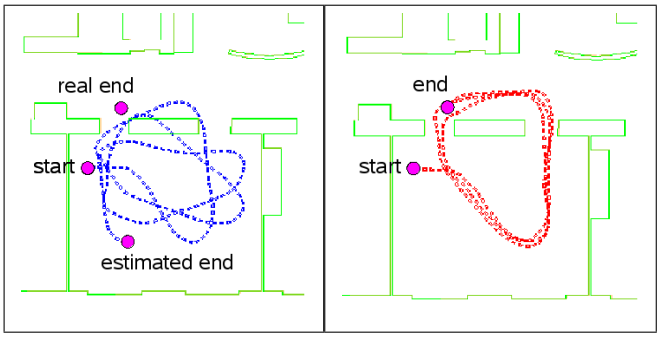


Fig. 10. A test run in the office environment. The robot trajectory was estimated once from only odometry (left) and once from the SLAM system (right).

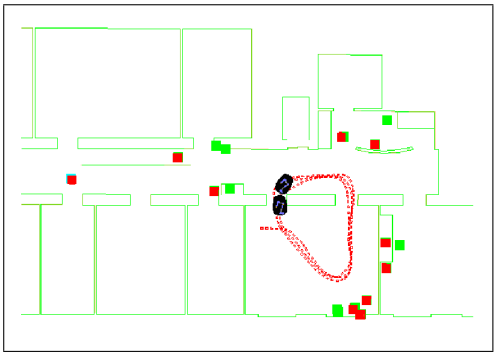


Fig. 11. Estimated robot trajectory with final robot position (the "first" robot is the real robot, whereas the robot behind visualizes the robot position at the end of the buffer. The latter is used for trajectory and landmark estimation). Green dots are landmarks, red dots are landmarks which were redetected in loop-closing situations.

environment	camera control	# landmarks	# correct matches	# false matches
office	passive	17	21	0
office	active	36	31	2
office	passive	18	23	1
office	passive	21	21	0
office	active	34	16	1
atrium	active	57	14	1
atrium	active	61	15	3
atrium	active	50	8	2
atrium	passive	19	1	1

TABLE I

MATCHING QUALITY FOR DIFFERENT TEST RUNS IN TWO ENVIRONMENTS. 2ND COLUMN: PASSIVE/ACTIVE CAMERA CONTROL. 3RD COLUMN: THE NUMBER OF MAPPED LANDMARKS. 4TH/5TH COLUMN: THE NUMBER OF TIMES A CURRENT LANDMARK WAS MATCHED TO AN ENTRY IN THE DATABASE. MATCHES ARE ONLY COUNTED, IF THE CORRESPONDING LANDMARK HAD NOT BEEN SEEN FOR AT LEAST 30 FRAMES. NOTE THAT A LANDMARK CAN ALSO BE MATCHED SEVERAL TIMES.

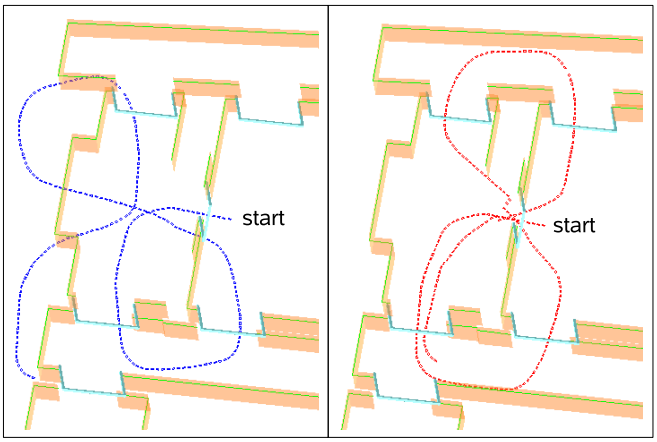


Fig. 12. A test run in the atrium area. The robot trajectory was estimated once from only odometry (left) and once from the SLAM system (right).

landmarks, found 21 matches to the database (one landmark can be detected several times) and all of the matches were correct (cf. Tab. I, row 1). The estimated landmark positions and the matches are displayed in Fig. 11. Notice that more than half of the landmarks are redetected when revisiting an area. More results from the office environment are shown in row 2–5 of Tab. I. The three occurring false matches belong always to the same object in the world: the lamp in Fig. 6 left.

More experiments were performed in the atrium environment. A comparison between the estimated robot trajectory from odometry data and from the SLAM system is visualized in Fig. 12. In this example, the system operated in active camera mode (cf. sec. VIII-B). Also here, the big difference in accuracy of the robot trajectory is visible. The corresponding number of landmark detections and matches is shown in Tab. I, row 6. Results from additional runs are shown in rows 7-9. Note that the percentage of matches with respect to the number of all landmarks is smaller in the atrium area than in the office environment since a loop can be only closed at a few places. Also in this environment, all the false matches belong to identical lamps (cf. Fig. 6 right).

In the presented examples, the few false matches did not lead to problems, the trajectory was estimated correctly anyway. Only the falsely matched landmarks are assigned a wrong position. But note that more false matches might cause problems for the SLAM process. The detection quality could be improved by collecting evidence for a match from several landmarks.

B. *Passive versus Active Camera Control*

In this section, we compare the passive and the active camera mode of the visual SLAM system. We show that with active camera control, more landmarks are mapped with a better distribution in the environment, more database matches are obtained, and that loop closing occurs earlier and even in situations where no loop closing is possible in passive mode.

From Tab. I, it can be seen that the test runs with active camera control result in more mapped landmarks than the runs with passive camera. Although this is not necessarily an advantage — we claim actually that the sparseness of the map is an advantage — it is favorable if the larger number results from a better distribution of landmarks. That this is the case here can be seen e.g. in the example in Fig. 13: landmarks show up in active mode (right), where there are no landmarks in passive mode (left).

Loop closing occurs usually earlier in active mode. For example in Fig. 11, the robot is already able to close the loop when it enters the doorway (position of front robot in figure)

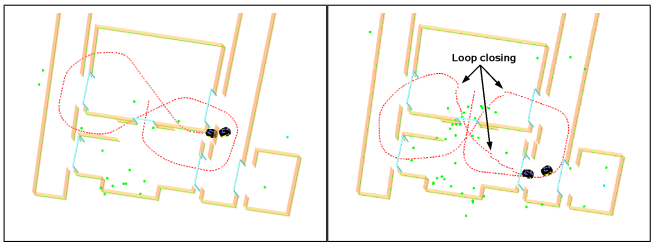


Fig. 13. Atrium environment: the estimated robot trajectory in passive (left, cf. Tab. I row 9) and active (right, cf. Tab. I row 8) camera mode (the 1st robot is the real robot, the 2nd a virtual robot at the end of the buffer). Landmarks are displayed as green dots. In passive mode, the robot is not able to close the loop. In active mode, loop closing is clearly visible and results in an accurate pose estimation (see also videos on http://www.informatik.uni-bonn.de/∼frintrop/research/aslam.html).

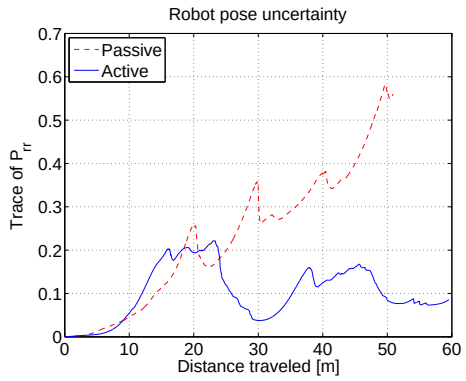


Fig. 14. The robot pose uncertainty computed as the trace of P_{rr} (covariance of robot pose) for passive and active camera mode.

by directing the camera to the landmark area on its left. In passive mode, loop closing only occurs when the robot itself moved to face this area. An earlier loop closing leads to an earlier correction of measurements and provides time to earlier go back to other behaviours like exploration.

In active mode, the robot closed a loop several times in the atrium. This is visible from the small jumps in the estimated trajectory in Fig. 13 right. The final pose estimate is much more accurate here than in passive mode. Fig. 14 displays a comparison of the robot pose uncertainty in passive and active mode, computed as the trace of P_{rr} (covariance of robot pose). The two loop closing situations in active mode around meter 30 and 50 reduce the pose uncertainty considerably, resulting at the end of the sequence in a value which is much lower than the uncertainty in passive mode.

IX. Discussion and Conclusion

In this paper, we have presented a complete visual SLAM system, which includes feature detection, tracking, loop closing and active camera control. Landmarks are selected based on biological mechanisms which favor salient regions, an approach which enables focusing on a sparse landmark representation. We have shown that the repeatability of salient regions is considerably higher than the one of regions from standard detectors. Additionally, we presented a precision-based matching strategy, which enables to intuitively choose a matching threshold to obtain a preferred matching precision.

The active gaze control module presented here enabled to obtain a better distribution of landmarks in the map and to re-detect considerably more landmarks in loop closing situations than in passive camera mode. In some cases, loop closing is actually only possible by actively controlling the camera.

While we obtain a good pose estimation and a high matching rate, further improvements are always possible and planned for future work. For example, we plan to collect evidence for a match from several landmarks together with their spatial organization as already done in other systems. Also determining the salience of a landmark not only in the image but in the whole environment would help to focus on even more discriminative landmarks. Using the precision value of a match could be very helpful to improve the system performance too. Adapting the system to deal with really large environments could be achieved by removing landmarks which are not redetected to keep the number of landmarks low, by database management based on search trees, indexing [53], [49], and by using hierarchical maps as in [11]. Also testing the system in outdoor environments is an interesting challenge for future work.

Acknowledgment

The present research has been sponsored by SSF through the Centre for Autonomous Systems, VR (621-20 06-4520), the EU project “CoSy” (FP6-004150-IP), and the university of Bonn through Prof. A. B. Cremers. This support is gratefully acknowledged. We also want to thank Mårten Björkman for providing the PYRA real-time vision library.

References

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localization and map building (SLAM) problem,” *IEEE Trans. Robot. Automat.*, vol. 17, no. 3, pp. 229–241, 2001.

[2] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, “FastSLAM: A factored solution to the simultaneous localization and mapping problem,” in *Proc. of the National Conf. on Artificial Intelligence (AAAI)*, 2002.

[3] J. Folkesson and H. Christensen, “Graphical SLAM - a self-correcting map,” in *Proc. of the IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2004.

[4] F. Dellaert, “Square root SLAM: Simultaneous location and mapping via square root information smoothing,” in *Proc. of Robotics: Science and Systems (RSS)*, 2005.

[5] U. Frese and L. Schröder, “Closing a million-landmarks loop,” in *Proc. of the IEEE/RSJ Int’l Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[6] A. Davison and D. Murray, “Simultaneous localisation and map-building using active vision,” *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 24, no. 7, pp. 865–880, 2002.

[7] L. Goncavles, E. di Bernardo, D. Benson, M. Svedman, J. Ostrovski, N. Karlsson, and P. Pirjanian, “A visual front-end for simultaneous localization and mapping,” in *Proc. of the Int’l Conf. on Robotics and Automation, (ICRA)*, 2005.

[8] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, “A framework for vision based bearing only 3D SLAM,” in *Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2006.

[9] K. Ho and P. Newman, “Detecting loop closure with scene sequences,” *Int’l J. of Computer Vision and Int’l J. of Robotics Research. Joint issue on computer vision and robotics*, 2007.

[10] M. Cummins and P. Newman, “Probabilistic appearance based navigation and loop closing,” in *Proc. IEEE Int’l Conf. on Robotics and Automation (ICRA)*, 2007.

[11] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira, and J. D. Tardos, “Mapping large loops with a single hand-held camera,” in *Proc. of Robotics: Science and Systems (RSS)*, 2007.

[12] S. Frintrop, "VOCUS: A visual attention system for object detection and goal-directed search," Ph.D. dissertation, Universität Bonn, Germany, 2005, ser. LNAI. Springer, 2006, vol. 3899.

[13] S. Frintrop, P. Jensfelt, and H. Christensen, "Simultaneous robot localization and mapping based on a visual attention system," in *Attention in Cognitive Systems*, ser. LNAI. Springer, 2007, vol. 4840.

[14] S. Frintrop and P. Jensfelt, "Active gaze control for attentional visual SLAM," in *Proc. of the IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2008.

[15] P. Zhang, E. E. Milios, and J. Gu, "Underwater robot localization using artificial visual landmarks," in *Proc. of IEEE Int'l Conf. on Robotics and Biomimetics*, 2004.

[16] U. Frese, "Treemap: An O($log\, n$) algorithm for indoor simultaneous localization and mapping," *Autonomus Robots*, vol. 21, no. 2, pp. 103–122, 2006.

[17] F. Launay, A. Ohya, and S. Yuta, "A corridors lights based navigation system including path definition using a topologically corrected map for indoor mobile robots," in *Int'l Conf. on Robotics and Automation (ICRA)*, 2002.

[18] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey Vision Conference*, 1988.

[19] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2001.

[20] J. Shi and C. Tomasi, "Good features to track," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 1994.

[21] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2005.

[22] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *Proc. of the British Machine Vision Conference (BMVC)*, 2002.

[23] K. Mikolajczyk and C. Schmid, "A comparison of affine region detectors," *Int'l J. of Computer Vision (IJCV)*, vol. 65, no. 1-2, pp. 43–72, 2006.

[24] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 20, no. 11, pp. 1254–1259, 1998.

[25] J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial Intelligence*, vol. 78, no. 1-2, pp. 507–545, 1995.

[26] A. M. Treisman and G. Gelade, "A feature integration theory of attention," *Cognitive Psychology*, vol. 12, pp. 97–136, 1980.

[27] J. M. Wolfe, "Guided search 2.0: A revised model of visual search," *Psychonomic Bulletin and Review*, vol. 1, no. 2, pp. 202–238, 1994.

[28] M. Corbetta and G. L. Shulman, "Control of goal-directed and stimulus-driven attention in the brain," *Nature Reviews*, vol. 3, no. 3, pp. 201–215, 2002.

[29] S. Frintrop, M. Klodt, and E. Rome, "A real-time visual attention system using integral images," in *Proc. of the Int'l Conf. on Computer Vision Systems (ICVS)*, 2007.

[30] S. B. Nickerson, P. Jasiobedzki, D. Wilkes, M. Jenkin, E. Milios, J. K. Tsotsos, A. Jepson, and O. N. Bains, "The ARK project: Autonomous mobile robots for known industrial environments," *Robotics and Autonomous Systems*, vol. 25, no. 1-2, pp. 83–104, 1998.

[31] N. Ouerhani, A. Bur, and H. Hügli, "Visual attention-based robot self-localization," in *Proc. of Europ. Conf. on Mobile Robotics (ECMR)*, 2005.

[32] C. Siagian and L. Itti, "Biologically-inspired robotics vision monte-carlo localization in the outdoor environment," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[33] P. Newman and K. Ho, "SLAM- loop closing with visually salient features," in *Proc. of the Int'l Conf. on Robotics and Automation (ICRA)*, 2005.

[34] R. Bajcsy, "Active perception vs. passive perception," in *Proc. of Workshop on Computer Vision: Representation and Control*. IEEE Press, 1985.

[35] Y. Aloimonos, I. Weiss, and A. Bandopadhay, "Active vision," *Int'l J. of Computer Vision (IJCV)*, vol. 1, no. 4, pp. 333–356, 1988.

[36] R. Bajcsy, "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 996–1005, 1988.

[37] B. Grocholsky, H. F. Durrant-Whyte, and P. Gibbens, "An information-theoretic approach to decentralized control of multiple autonomous flight vehicles," in *Sensor Fusion and Decentralized Control in Robotic Systems III*, 2000.

[38] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 15, no. 5, pp. 417–433, 1993.

[39] R. Sim and J. J. Little, "Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters," in *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[40] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *In Proc. of the IEEE Int'l Symp. on Computational Intelligence in Robotics and Automation*, 1997.

[41] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, "An experiment in integrated exploration," in *Proc. of the IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2002.

[42] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.

[43] T. Arbel and F. P. Ferrie, "Entropy-based gaze planning," in *Proc. of IEEE Workshop on Perception for Mobile Agents*, 1999.

[44] T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D. W. Murray, "Active control for single camera SLAM," in *Proc. IEEE Int'l Conf. on Robotics and Automation (ICRA)*, 2006.

[45] J. M. M. Montiel, J. Civera, and A. J. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proc. of Robotics: Science and Systems (RSS)*, 2006.

[46] S. Frintrop and A. B. Cremers, "Top-down attention supports visual loop closing," in *Proc. of European Conference on Mobile Robotics (ECMR)*, 2007.

[47] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. of Computer Vision (IJCV)*, vol. 60, no. 2, pp. 91–110, 2004.

[48] S. Obdrzalek and J. Matas, "Sub-linear indexing for large scale object recognition," in *Proc. of the British Machine Vision Conference (BMVC)*, 2005.

[49] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[50] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. of Pattern Analysis and Machine intelligence (PAMI)*, vol. 27, no. 10, pp. 1615–1630, 2005.

[51] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *Int'l J. of Robotics Research*, vol. 21, no. 8, pp. 735–758, 2002.

[52] S. May, M. Klodt, E. Rome, and R. Breithaupt, "GPU-accelerated affordance cueing based on visual attention," in *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2007.

[53] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. of the Int'l Conf. on Computer Vision*, 2003.

Simone Frintrop Simone Frintop got her Ph.D. from the University of Bonn, 2005. She was a postdoctoral researcher at the Computer Vision and Active Perception lab (CVAP) at the School of Computer Science and Communications (CSC) at the Royal Institue of Technology (KTH), Stockholm, Sweden until 2006. She now works in the Intelligent Vision Systems Group at the Institute of Computer Science III, University of Bonn, Germany, where she is currently a Senior Scientific Assistant.

Patric Jensfelt Patric Jensfelt received his M.Sc. in Engineering Physics in 1996 and Ph.D. in Automatic Control in 2001, from the Royal Institute of Technology, Stockholm, Sweden. Between 2002 and 2004 he worked as a project leader in two industrial projects. He is currently an assistant professor with the Centre for Autonomous System (CAS) and the principal investigator of the European project CogX at CAS. His research interests include mapping and localiation and systems integration.