2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Active Gaze Control for Attentional Visual SLAM

Simone Frintrop and Patric Jensfelt

Abstract— **In this paper, we introduce an approach to active camera control for visual SLAM. Features, detected by a biologically motivated attention system, are tracked over several frames to determine stable landmarks. Matching of features to database entries enables global loop closing. The focus of this paper is the active camera control module, which supports the system with three behaviours: i) A tracking behaviour tracks promising landmarks and prevents them from leaving the field of view. ii) A redetection behaviour directs the camera actively to regions where landmarks are expected and thus supports loop closing. iii) Finally, an exploration behaviour investigates regions without landmarks and enables a more uniform distribution of landmarks. Several real-world experiments show that the active camera control outperforms the passive system considerably.**

I. Introduction

SLAM (Simultaneous localization and mapping) has been a topic of significant interest in the robotic community over the last decade [1], [2], [3]. While being considered widely solved for small indoor environments based on laser range finders, current topics of active research include *visual SLAM*, based only on camera data [4], [5], [6], [7], [8].

The use of cameras holds advantages as well as challenges and difficulties: on the one hand, 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, the high amount of data in images challenges real-time processing: choosing the relevant data for processing and storing is crucial. Second, depth estimation is difficult when performing bearing-only SLAM with a single camera without manual initialization. And third, different appearances of the same scene under illumination and viewpoint changes make tracking and matching a challenge.

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 one of the important problems in SLAM since it decreases accumulated errors. Furthermore, there should be a limited amount of landmarks since the complexity of SLAM typically is a function of the number of landmarks in the map. Additionally, landmarks should be well distributed over the environment.

Current approaches for landmark selection include artificial landmarks [9], Harris corners [5], maximally stable extremal regions (MSERs) [10], or a combination of attention regions with Harris corners [11]. In this paper we show that attention regions alone can be used as landmarks which simplifies and speeds up the system.

The focus of this paper is the extension of the SLAM system to active camera control. The strategy 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 landmark distribution. The advantage of the active gaze control is to obtain more informative landmarks with a better baseline, a faster loop closing, and a better distribution of landmarks in the environment.

The idea of active sensing 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 [12], and made popular by Active Vision [13] and Active Perception [14]. In terms of sensing for active localization, Maximum Information Systems are an early demonstration of sensing and localization [15]. Active motion to increase recognition performance and active exploration was introduced in [16]. More recent work has demonstrated the use of similar methods for exploration and mapping [17]. Active exploration by moving the robot to cover space was presented in [18] and in [19] the uncertainty of the robot pose and feature locations were also taken into account.

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 [20], [21], they present a robotic system which chooses landmarks for tracking which best improve the position knowledge of the system. In more recent work [7], [22], they apply their visual SLAM approach to a hand-held camera. Active movements are done by the user, according to instructions from user-interface [7], or they use the active approach to choose the best landmarks from the current scene without controlling the camera [22].

The contributions of this paper are first, presenting a landmark selection scheme based on a biologically motivated attention system, second, a precision-based matching procedure, and finally, an active gaze control strategy to obtain a better baseline for landmark estimations, a faster loop

This work was partially sponsored by the SSF through its Centre for Autonomous Systems (CAS) and the EU as part of the project CoSy FP6-004250-IP. The research of S. Frintrop was funded by Prof. A.B. Cremers. The support is gratefully acknowledged.

S. Frintrop is with the Institute of Computer Science III, Rheinische Friedrich-Wilhems-Universität, 53111 Bonn, Germany 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

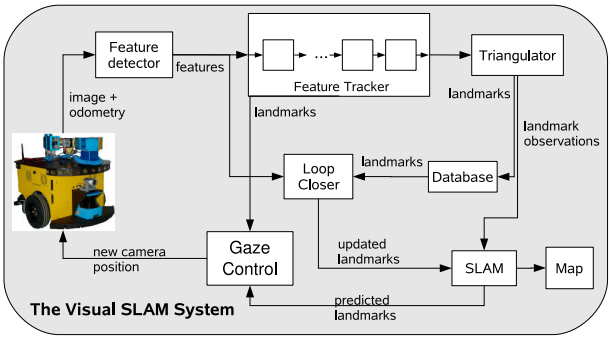


Fig. 1. The active visual SLAM system

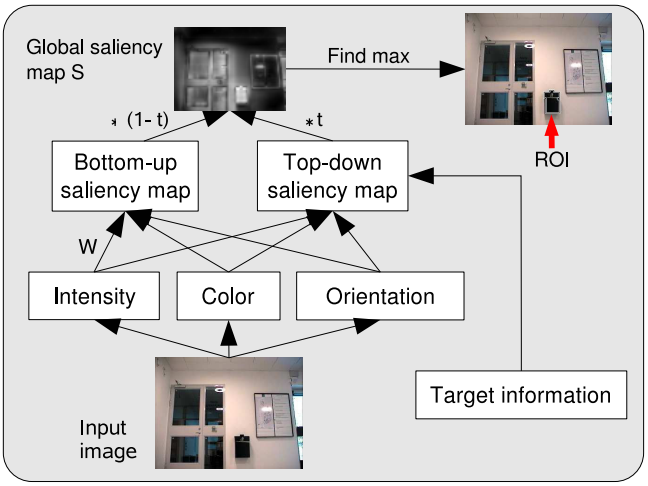


Fig. 2. The visual attention system VOCUS detects regions of interest (ROIs) in images based on the features intensity, orientation, and color.

closing, and a more uniform distribution of landmarks in the environment. Experimental results are presented to show the performance of the system.

In the following, we first give an overview over the whole SLAM architecture (sec. II), then we describe the modules of the system in detail (sec. III–sec. VII). Finally, we illustrate in sec. VIII the operation of the method on a real robot and show the advantages of active camera control.

II. System Overview

The visual SLAM architecture is displayed in Fig. 1. The 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 *SLAM module* which builds a map of the environment, a *loop closer* which matches current ROIs to the database and, as main part of the current paper, a *gaze control module* which determines where to direct the camera to.

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 [5].

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 features from 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.

III. Feature Detection

The detection of regions of interest (ROIs) is performed with the attention system VOCUS (Visual Object detection with a CompUtational attention System) [23], [24]. VOCUS is based on concepts of the human visual system, namely on the ability to quickly focus on salient regions of interest. It is grounded on psychological work like the feature integration theory [25] and neurobiological findings [26]. The system 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, a first approach for integrating top-down processes into the SLAM system is described in [27].

The saliency is computed for 3 features: intensity, color, and orientations. For each feature, the contrast of a region to its background is computed by *center-surround mechanisms* [23]. For each feature, several *feature types* are determined, e.g. bright-dark (on-off) as well as dark-bright (off-on) contrasts for the feature intensity. Before the features are fused into a single saliency map, 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. From the saliency map, the brightest regions are extracted as *regions of interest (ROIs)*.

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. 3). The last three entries describe the combination of the feature types, i.e., the value for *intensity* determines the combination of on-off and off-on intensities (cf. [23]).

Additionally to $\vec{v}$, a SIFT descriptor is determined for each ROI [28]. It 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

Feature	vector $\vec{v}$
intensity on-off	0.11
intensity off-on	7.92
orientation 0°	2.36
orientation 45°	6.82
orientation 90°	7.32
orientation 135°	8.48
color green	5.32
color blue	2.97
color red	0.73
color yellow	0.19
intensity	4.99
orientation	5.70
color	2.52

Fig. 3. Left: image with region of interest (ROI). Right: feature vector $\vec{v}$ for ROI. The values of $\vec{v}$ show that the region is dark on a bright background (intensity off-on), that the vertical orientation is stronger than the horizontal one, and that generally intensity and orientation are more important than color.

each of the grid cells. Usually, SIFT descriptors are computed for corner features such as Harris corners [29] or intensity extrema in scale space [28]. Here, we calculate one 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 surround but should also be not too large to stay within the image borders. We chose a grid size of 1.5 times the maximum of width and height of the ROI.

IV. Feature Matching

Feature matching is performed in two of the visual SLAM modules: in the feature tracker and in the loop closer. In the feature tracker, features are matched between consecutive frames to build landmarks and to enable structure from motion computations. In the loop closer, matching is performed between features from the current frame and features from the database to detect if this scene has been seen before.

Matching of interest regions is usually based on a similarity comparison depending on the distance $d(\xi_1, \xi_2)$ between two descriptors ξ_i (different descriptor types may be used, or a combination of them. This will be discussed later). If d is below a threshold, the regions are considered to match. However, thresholding on a distance is a bit tricky. Setting the threshold is unintuitive and requires experience with the system. Furthermore, small changes on the threshold might have unexpected effects on the detection quality since the dependence of distance and matching precision is not linear. Therefore, we suggest a slightly modified thresholding approach. We learn from training data how the matching precision depends on the descriptor distance threshold. This enables to directly set a threshold for the matching precision and let the system calculate the required corresponding distance threshold automatically.

For a large amount of image data, we gathered statistics regarding the distribution of the matching precision depending on the descriptor threshold. For t distinct distance threshold values, we compute the *precision* p as

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

where $c(\theta_j)$ and $f(\theta_j)$ denote the number of correct and false matches for a given descriptor distance threshold θ_j. Hereby, the correct and false matches are classified manually to obtain ground truth. The distribution is one-dimensional if a single descriptor type is used and multi-dimensional for several different descriptor types.

Matching is now performed depending on a threshold on the precision instead directly on the descriptor distance. Here, we use a precision threshold of 0.98: if the estimated precision is above the threshold, the ROIs are considered to match. We chose a high threshold because an EKF SLAM system is sensitive to outliers.

The presented approach has several advantages over the usual thresholding. First, it is possible to choose an intuitive threshold like "98% matching precision". Second, linear changes on the threshold result in linear changes on the matching precision. Finally, for every match a precision value is obtained. Since this corresponds to a probability estimation, this value can be directly used by other components of the system to treat a match according to the probability estimate that it is correct. For example, a SLAM subsystem which can deal with more uncertain associations could use these values. We consider the exploitation of this value for future work.

As mentioned above, different descriptor types can be used. We investigated two approaches. The first uses a combination of an attentional descriptor and the SIFT descriptor. The attentional descriptor is the previously introduced vector $\vec{v}$. The distance $d_A(\vec{v}_1, \vec{v}_2)$ between two attention vectors is calculated according to an equation similar to the Euclidean distance, details in [11]. The distance d_S of two SIFT descriptors is calculated as their Euclidean distance. To determine the two-dimensional distribution of matching precision depending on d_A and d_S, 378 correct matches and 535 false matches were classified manually. The experiments in this paper were based on this method.

Recently, we investigated a second method: matching based on only the SIFT descriptor. This resulted even in slightly better matching results, i.e., for the same amount of false detections more correct matches were found. While surprising at first, this can be explained as follows: a region may be described by two descriptor types, the perfect descriptor δ_1 and the weaker descriptor δ_2. δ_1 detects all correct matches and rejects all possible false matches. Combining δ_1 with δ_2 cannot improve the process, it can only reduce the detection rate by rejecting correct matches. Corresponding experiments will be published in [30].

V. Feature Tracking

In the feature tracker, *landmarks* are built from ROIs by tracking the ROIs over several frames. That means, a landmarks is a list of tracked ROIs and 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.

The matching of ROIs is performed not only between consecutive frames, but allows for gaps of several (here: 2) frames where a ROI is not found. We call frames which are at most 3 frames behind the current frame *close frames*. Since a scene usually does not change strongly between such close frames, it is possible to determine the approximate position of a feature in the current frame from its position in the last frame and the motion of the robot. This position estimation makes the tracking more stable.

The procedure to create landmarks is the following: when a new frame comes into the buffer, each of its ROIs is matched to all existing landmarks of close frames. If the matching is successful, the new ROI is appended to the end of the best matching landmark. Additionally, the ROIs that did not match any existing landmark are matched to the unmatched ROIs of the previous frame. If two ROIs match, a new landmark is created consisting of these two ROIs. At the end of the buffer, we consider the length of the resulting landmarks and filter out too short ones (here ≤ 5).

The final quality check for a tentative landmark that is long enough but has not yet been added to the map data is made by the triangulator. It attempts to find an estimate for the location of the landmark. In the triangulation process, also outliers are detected and removed from the landmark. By outlier we mean bearings that fall far away from the estimated landmark location. 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 seen the current scene before. This 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 detect loop closing without using the SLAM pose estimate as in [31]. That means, we match to all landmarks from the database. Since our system usually focuses on few landmarks (e.g. 57 for a 162 m^2 environment) it is possible to search the whole database in each iteration. However, for larger environments it would be necessary to perform global loop closing less frequently and distribute the search over several iterations.

A ROI r_1 is said to match to a landmark L, if there are at least j (here: j = 3) ROIs $r_i, i \in 1..j$ in L for which (i) the size difference of r_1 and r_i is small enough, (ii) the probability for a match (based on the attention vector and SIFT descriptor similarities) is $> 98\%$ and (iii) if there is no other ROI from the current frame with a higher matching probability to r_i. To prune the search space, the feature vectors of r_1 and r_i have to pass a similarity threshold before the match probability is computed in (ii).

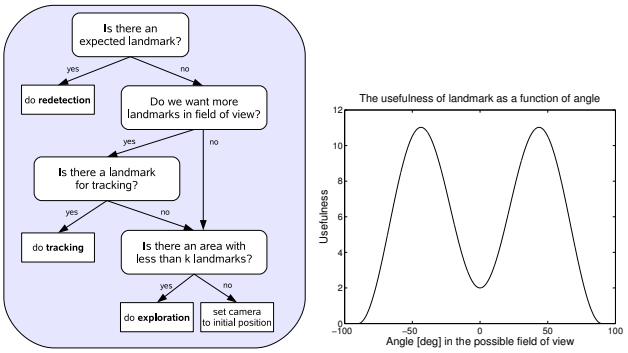


Fig. 4. Left: The three camera behaviours *Redetection*, *Tracking*, *Exploration*. Right: The usefulness function $\varphi(\alpha)$.

When a match is detected, the coordinates of the matched ROI in the current frame are fed to the SLAM system, to update the coordinates of the corresponding landmark. Additionally, the ROI is appended to the landmark in the database.

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 cf. Fig. 5). An overview over the decision process is displayed in Fig. 4. In the following, we describe the respective behaviours in detail.

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,
(b) have low-enough uncertainties in the expected positions relative to the camera,
(c) have not been seen recently,
(d) had no matching attempt recently.

To (a): 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

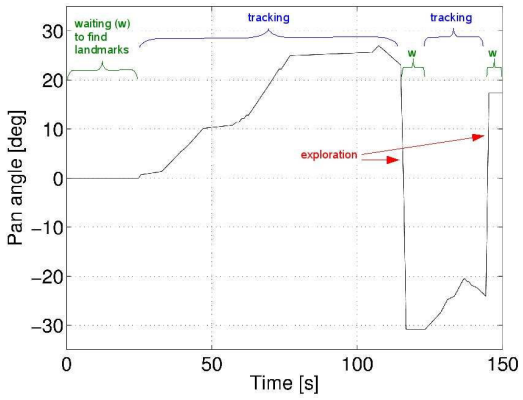


Fig. 5. The pan angle as a function of time. The camera behaviour alternates between tracking and exploration.

probably not visible although they are in the right direction: obstacles like walls are likely to block the view. In the current implementation, there is no way to know whether the landmarks are in the same room, therefore landmarks from different rooms might be considered. Of course, the restriction to a certain distance is only a rough estimate which is also dependent on the current environment. This model causes problems primarily in environments where the robot is actually able to detect landmarks that are further away than 7m which means that not all available information is used. In smaller areas there is a slight increase in computational cost as more landmarks than necessary are considered.

To (b): Landmarks with a high pose uncertainty in pan- or tilt-direction relative to the camera are not considered as expected landmarks, because matching is likely to fail when directing the camera there. 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, the landmark is too uncertain. 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.

To (c): 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 its redetection.

To (d): If an expected landmark has been focused for some frames and is still not redetected, it is likely that it will not be redetectable in the near future. Therefore, the redetection of these landmarks is blocked for a while (here: 30 frames). This behaviour prevents the system from repeatedly directing the camera at undetectable landmarks and allows the system to continue with tracking and exploration, once it checked all expected landmarks in the possible field of view.

If there are several expected landmarks, the longest landmark is chosen because the probability for a match is high. Then, the camera is moved to focus this landmark 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 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.

Therefore, the usefulness of a landmark is determined by first considering the length of the landmark and, second, the angle of the landmark in the potential field of view. The length of the landmarks is considered by sorting out landmarks below a certain size (here: 5). The usefulness of the angle α of a ROI is determined by the following function:

$$\varphi(\alpha) = (k_1\,(1.0+\cos(4\alpha-\pi))) + k2\,(1.0+\cos(2\alpha))) \quad (2)$$

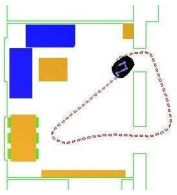
where $k_1 = 5$ and $k_2 = 1$. The function is displayed in Fig. 4 (right). It has the highest weight for points at $\alpha = 45°$ and $\alpha = -45°$ and has minima at $\alpha = 0°$ and $\alpha = \pm 90°$. Since points which are at the border of the field of view are likely to move out of view very soon, they are considered even worse than points in the center. Notice that we cannot actively control the robot motion, only the camera's, which would otherwise allow us to make sure that points on the border stay in the image. The exact shape of the function is not crucial, functions with similar shape should do as well. The usefulness of a landmark L is determined by:

$$U(L) = \varphi(\alpha)\,\sqrt{l} \quad (3)$$

where l is the length of the landmark.

After determining the most useful landmark for tracking, the camera is directed into the direction of the landmark. The camera is moved slowly (here 0.1 radians per step), since this changes the appearance of the ROI less than large camera movements resulting in a higher matching rate and prevents to loose other currently visible landmarks.

The tracking ends when the landmark is not visible any more (because it left the field of view or because the



	# LMs mapped		# correct matches		# false matches	
	pass.	act.	pass.	act.	pass.	act.
experiment 1 a (after 1st loop)	9	21	0	5	0	0
experiment 1 a (after 2nd loop)	15	27	5	16	1	0
experiment 1 b (after 1st loop)	10	22	1	11	0	0
experiment 1 b (after 2nd loop)	16	28	8	18	1	0
experiment 2	26	57	0	21	0	4

TABLE I

LEFT: ROBOT PATH FOR EXPERIMENT 1. RIGHT: COMPARISON OF NUMBER OF MAPPED LANDMARKS AND OF CORRECT AND FALSE MATCHES FOR PASSIVE AND FOR ACTIVE CAMERA MODE.

matching failed) or when the landmark was successfully triangulated. If there is no other useful landmark to track or there are already enough landmarks detected in this region, the exploration behaviour is activated.

C. Exploration of unknown areas

As soon as there are enough (≥ 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 proceed as follows: the possible field of view is divided in two parts, one on each side of the current field of view. Each of these regions is divided into parts which correspond to the size of the field of view. Then one field after the other is checked until one without landmarks is found. The order in which fields are checked is as follows: 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. This enables a broader distribution of detected landmarks in the environment. If there is no landmark, the camera is moved there. Otherwise we switch to the opposite side and investigate the areas there. If no area without landmarks is found, the camera is set to the initial position.

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. 5).

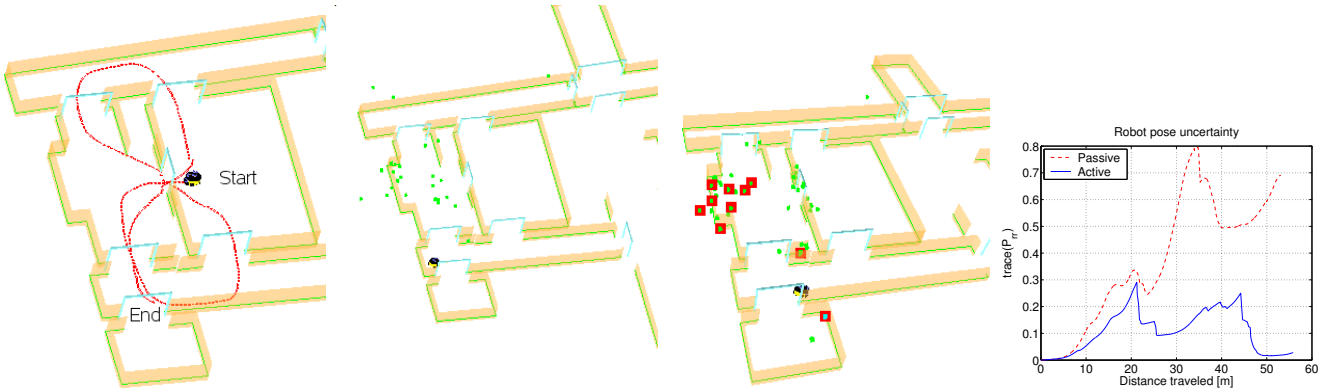
VIII. EXPERIMENTS AND RESULTS

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 and more database matches are obtained (experiment 1). Finally, we show a case in which a loop closing is not detected in passive mode but is in active mode (experiment 2).

In **experiment 1**, the robot drove two loops on the path displayed in Tab. I, left. To show the repeatability of the results, the experiment was carried out twice: experiment 1a was performed during the day and experiment 1b during night, with different lightning conditions. Each sequence consists of $\sim$1200 images (320×240). We monitored the number of landmarks which were mapped and the number of correct and false matches after 1 and after 2 loops. The results are shown in Tab. I. Col. 2 and 3 show that in active mode, considerably more landmarks are mapped than in passive mode, usually about twice as many. This results from the exploration mode: areas are investigated in active mode which are not visible to the camera in passive mode. Thus, a better distribution of landmarks can be achieved. Col. 4–7 show the number of matches in loop closing situations. We count only matches which appeared at most 30 frames after the landmark had been visible for the last time. Matches to landmarks which have been visible more recently are also used to update the map data, but are not counted here since we want to focus on matches with a higher impact on uncertainty reduction. The table shows that the number of matches also increases considerably in active mode. This is due to first, having more landmarks in the database, second, actively directing the camera to expected landmarks (redetection), and third, directing the camera by chance to previously visible landmarks (exploration).

The result of experiment 1 is that by active camera control, more landmarks are mapped with a better distribution in the environment and more landmark matches. However, in this experiment, the robot pose uncertainty is similar in both cases. It drops slightly earlier in active mode if the camera is directed to an expected landmark while the loop is not yet closed completely, but since exactly the same path is repeated, the system is also able to close the loops in passive mode.

In **experiment 2**, we show a case where loop closing is not possible in passive but in active mode. Here, the robot drove the path of an eight, as displayed in Fig. 6, once in passive and once in active mode. 1803 (passive) resp. 1788 (active) images were processed during the path. Although the first door is passed three times, the robot does not face exactly the same area in these three cases and is not able to close a loop in passive mode (in the last part of the path, no landmarks where detected during the first run, so no matching is possible). In Fig. 6 (b), the resulting map is displayed. It can be seen that the final robot pose is wrong by about 3m since the robot was not able to correct its pose by loop closing. On the other hand, in active mode the camera is directed to regions which had been seen before and the robot closes loops first after the first circle and again after the second circle (cf. Fig. 8, left). Fig. 6 (c) shows the resulting map, with matches displayed as larger, red dots. The number



(a) Robot path in experiment 2 (b) Resulting map, passive (c) Resulting map, active (d) Robot pose uncertainty

Fig. 6. Experiment 2: comparison of passive and active camera control. Green, small dots are landmarks (b,c), red, large dots are database matches (c). (d): the robot pose uncertainty computed as the trace of P_{rr} (covariance of robot pose) for passive and active camera mode. A video showing the trajectory of the robot in active camera mode is available on the CD proceedings of ICRA 2008.

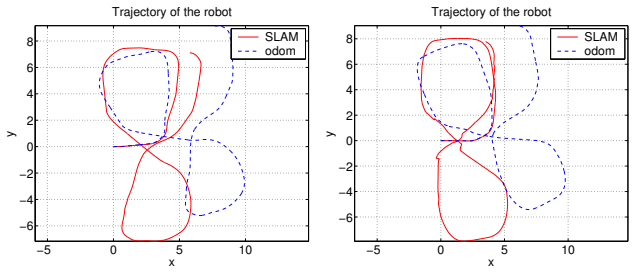


Fig. 7. Trajectory of robot path estimated from odometry (blue, dashed) and SLAM (red, solid) for passive (left) and active (right) camera mode.

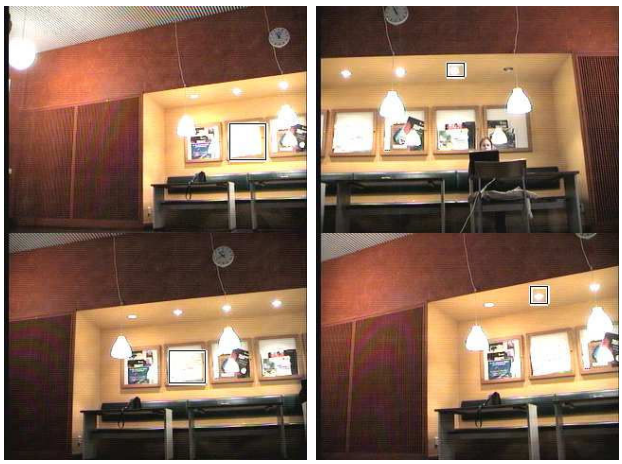


Fig. 8. Left: correct loop closing match. Right: false match.

of matches is shown in the last row of Tab. I: 21 correct and 4 false matches. Most false matches result from confusing some of the lamps with identical appearance (cf. Fig. 8, right). Considering the geometric arrangement of landmarks would help to prevent such false matches. Also visible from Fig. 6 (b) and (c) is that the final robot pose is much more accurate in active than in passive mode. This can also be seen in Fig. 7, in which the trajectory of the robot, estimated once directly from odometry and once from SLAM, is displayed for passive and for active camera mode. When comparing it with the path in Fig. 6 (a), it can be seen that first, the SLAM estimation is much more accurate than the odometry estimate and second, that the actively estimated SLAM path is more accurate than the passive one.

In Fig. 6 (d), the robot pose uncertainty, computed as the trace of P_{rr} (covariance of robot pose) is displayed for passive and for active mode. It shows clearly how the two loop closing situations in active mode reduce the pose uncertainty (at meter 21 and meter 44), resulting at the end of the sequence in a value which is about 80% lower than the uncertainty in passive mode.

IX. Conclusion

This paper presents an active visual SLAM system based on attentional landmarks. The attention regions provide useful landmarks for visual SLAM since they provide a way to immediately, that means already when the features are computed, determine which regions in an image are useful. This results in few landmarks compared to corner-like features what is helpful for an EKF-based SLAM system that scales with the number of landmarks. The precision-based matching procedure provides a powerful way to achieve a certain detection rate. Another advantage of this approach is that it directly provides a probability value that a match is correct. With a different SLAM subsystem than the current one, one that can deal with more uncertain associations, these matching probability could be used.

The system seems to generalize well to new environments: system development and all parameter tuning was performed in environment 1, testing the system in environment 2 in another building was only done after the system was complete. As shown, good performance was obtained here. However, it would be interesting to investigate how robust the system behaves in completely different environments such as outdoor environments. This is subject to future work.

The computation of the attention regions is relatively fast (∼50 ms/frame) since it is based on integral images [32]. The

rest of the 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.

The main contribution of the paper is the active gaze control module with the behaviours tracking, redetection, and exploration. Experimental results showed that about twice as many landmarks are mapped in active camera mode and at least twice as many database matches are obtained, usually much more. In some cases, loop closing is only possible by actively controlling the camera.

Needless to say, much could be done to further improve the system. False detections could be eliminated by considering the spatial organization of several landmarks. Extending the system to larger environments could be achieved by removing landmarks which are not redetected to keep the number of landmarks low, and by using hierarchical maps as in [22], in which many local maps are built which do not exceed a certain size.

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] U. Frese, P. Larsson, and T. Duckett, "A multigrid algorithm for simultaneous localization and mapping," *IEEE Trans. Robot.*, vol. 21, no. 2, pp. 1–12, 2005.

[3] S. Thrun, Y. Liu, D. Koller, A. Ng, Y. Ghahramani, and H. Durrant-Whyte, "Simultaneous localization and mapping with sparse extended information filters," *IJRR*, vol. 23, no. 7-8, pp. 693–716, 2004.

[4] 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 ICRA*, apr 2005, pp. 44–49.

[5] P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman, "A framework for vision based bearing only 3D SLAM," in *Proc. of ICRA*, 2006.

[6] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "Probabilistic algorithms and the interactive museum tour-guide robot minerva," *Int'l J. of Rob. Research*, vol. 19, no. 11, 2000.

[7] T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D. W. Murray, "Active control for single camera SLAM," in *Proc. of ICRA*, 2006.

[8] K. Ho and P. Newman, "Detecting loop closure with scene sequences," *International Journal of Computer Vision and International Journal of Robotics Research. Joint issue on computer vision and robotics*, 2007.

[9] 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*, Shenyang, China, Aug. 2004, pp. 705–710.

[10] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," in *Proc. of ICRA'05*, 2005, pp. 644–651.

[11] 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.

[12] R. Bajcsy, "Active perception vs. passive perception," in *Proc. 3rd Workshop on Computer Vision: Representation and Control*. Washington, DC.: IEEE Press, October 1985, pp. 55–59.

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

[14] R. Bajcsy, "Active perception," *Proc. of IEEE*, vol. 76, no. 8, 1988.

[15] 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,*, Boston, Nov 2000.

[16] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE Trans. PAMI*, vol. 15, no. 5, pp. 417–433, 1993.

[17] R. Sim and J. J. Little, "Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters," in *Proc. of IROS*, 2006.

[18] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Int'l Symp. on Comp. Intelligence in Rob. and Automation*, 1997.

[19] A. Makarenko, S. Williams, F. Bourgault, and H. Durrant-Whyte, "An experiment in integrated exploration," in *Proc. of IROS'02*, 2002.

[20] A. Davison and D. Murray, "Mobile robot localisation using active vision," in *Proc. of ECCV*, May 1998.

[21] ——, "Simultaneous localisation and map-building using active vision," *IEEE Trans. PAMI*, 2002.

[22] 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.

[23] S. Frintrop, "VOCUS: A visual attention system for object detection and goal-directed search," Ph.D. dissertation, Germany, 2006, lNAI, Vol. 3899.

[24] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," in *Proc. of DAGM*, ser. LNCS. Springer, 2005, pp. 117–124.

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

[26] 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.

[27] S. Frintrop and A. B. Cremers, "Top-down attention supports visual loop closing," in *accepted for Proc. of European Conference on Mobile Robotics (ECMR 2005)*, 2007.

[28] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. of ICCV*, 1999, pp. 1150–57.

[29] K. Mikolajczyk and C. Schmid, "Indexing based on scale invariant interest points," in *Proc. of ICCV*, 2001, pp. 525–531.

[30] S. Frintrop and P. Jensfelt, "Attentional landmarks and active gaze control for visual SLAM," *submitted*, 2008.

[31] P. Newman and K. Ho, "SLAM-loop closing with visually salient features," in *Proc. of the International Conference on Robotics and Automation, (ICRA 2005)*, Barcelona, Spain, April 2005.

[32] S. Frintrop, M. Klodt, and E. Rome, "A real-time visual attention system using integral images," in *Proc. of ICVS*, 2007.