Attentional Robot Localization and Mapping

Simone Frintrop* and Patric Jensfelt** and Henrik Christensen***

* Comp. Science III, University of Bonn, Germany, frintrop@iai.uni-bonn.de
** CSC, KTH, Stockholm, Sweden, patric@csc.kth.se
*** GeorgiaTec, Atlanta, USA hic@cc.gatech.edu

Abstract. In this paper, we introduce an application of visual attention in the field of robotics: attentional visual SLAM (Simultaneous Localization and Mapping). A biologically motivated attention system finds regions of interest which serve as visual landmarks for the robot. The regions are tracked and matched over consecutive frames to build stable landmarks and to estimate the 3D position of the landmarks in the environment. Furthermore, matching of current landmarks to database entries enables loop closing and global localization. Additionally, the system is equipped with an active camera control, which supports the system with a tracking, a re-detection, and an exploration behaviour.

1 Introduction

In the field of robotics, *visual SLAM* has recently been a topic of much research [2, 7, 9]. The task is to build a map of the environment and to simultaneously stay localized within the map. In contrast to common laser-based approaches, visual SLAM aims at solving the problem only based on camera data.

A key competence in visual SLAM is to choose useful visual 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 most important problems in SLAM since it decreases accumulated errors. Furthermore, there should be a limited number of landmarks since the complexity of SLAM typically is a function of the number of landmarks in the map. On the other hand, landmarks should be distributed over the environment.

Often, the landmarks are selected by a human expert or the kind of landmark is determined in advance, e.g., ceiling lights or Harris corners. As pointed out by [15], there is a need for methods which enable a robot to choose landmarks autonomously. A good method should pick the landmarks which are best suitable for the current situation. An adequate method to find landmarks autonomously depending on the current surrounding are visual attention systems [16, 8, 4]. They select regions that "pop out" in a scene due to strong contrasts and uniqueness, as the famous black sheep in a white herd. The advantage of these methods is that they determine globally which regions in the image discriminate instead of locally detecting predefined properties.

In this paper, we present a visual SLAM system based on an attentional landmark detector. Regions of interest (ROIs) are detected by the attention system VOCUS [4], and are tracked and matched over consecutive frames to build stable landmarks. The 3D position of the landmarks in the environment is estimated by structure from motion and the landmarks are integrated into the map. When the robot returns to an already visited location, this *loop closing* is detected by matching current landmarks to database entries. This enables the updating of the current robot position as well as the other landmark entries in the map. Additionally, active camera control improves the quality and distribution of detected landmarks with three behaviours: a *redetection* behaviour actively searches for expected landmarks to support loop-closing. A *tracking* behaviour identifies the most promising landmarks and prevents them from moving out of the field of view. Finally, an *exploration* behaviour investigates regions with no landmarks, leading to a more uniform landmark distribution.

Although attention methods are well suited for selecting landmark candidates, the application of attention systems to landmark selection has rarely been studied. Two existing approaches are [13], in which landmarks are detected in hand-coded maps, and [14], in which a topological map is built. The only approach we are aware of which uses an approach similar to a visual attention system for SLAM, is presented in [12]. 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, so the approach belongs not to the category of visual SLAM.

The idea of active sensing is not new [1], but in the field of visual SLAM, it has almost not been investigated yet. Davison & Murray presented a first, interesting approach to active camera control [3]. They use artificial visual landmarks to control robot and camera motion manually. Also the selection of features to integrate into the map is done manually. In recent work [17], landmarks with the highest mutual information are chosen to optimize localization of the sensor and the features in the map. Since they use a hand-held camera, active movements are done not automatically but by the user, according to instructions from user-interface. To our knowledge, our system represents the first approach of active gaze control for visual SLAM which works in an un-prepared environment and in which feature selection and camera motion are done autonomously without intervention of a user. The attentional landmark selection is one of the most important components of the system.

2 System Overview

The visual SLAM architecture (Fig. 1) consists of 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 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*. This module finds ROIs based on the visual attention system VOCUS and Harris corners inside the ROIs. Next, the features are provided to the *feature tracker* which stores the last n frames, performs matching of ROIs and Harris

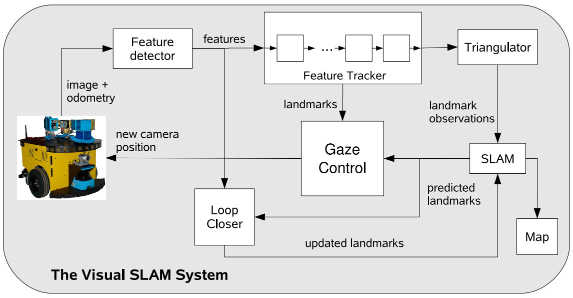


Fig. 1. The visual SLAM system builds a map based on image data and odometry

corners 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 *SLAM module* which computes an estimate of the position of landmarks and integrates the position estimate into the *map* (details to SLAM module in [9]).

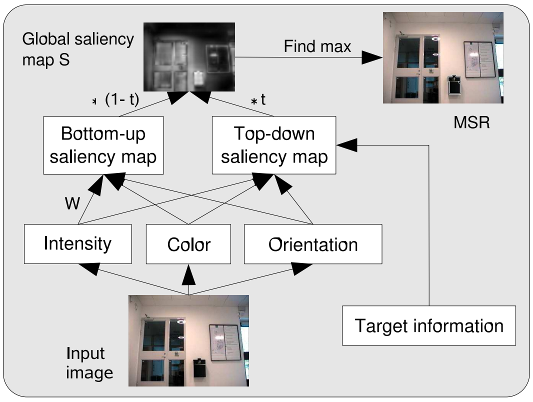
The task of the *loop closer* is to detect if a scene has been seen before. The features from the current frame are compared with the features from the landmarks in the database. To narrow down the search space, the SLAM module provides the loop closer with expected landmark positions. Only landmarks that should be currently visible are considered for matching. Finally, the *gaze control module* controls the camera actively. It decides whether to actively look for predicted landmarks, to track currently seen landmarks, or to explore unseen areas. It computes a new camera position which is provided to the robot.

3 Feature Selection

The feature selection is based on two different kinds of features: attentional ROIs and Harris corners. In [6] we have shown that this combination is useful, since it combines the advantages of both approaches: the attentional ROIs focus the processing on salient image regions which are thereby well redetectable. Harris corners on the other hand provide well localized points as required for precise depth estimation for structure from motion with a small baseline.

ROI Detection: Regions of interest (ROIs) are detected with the attention system VOCUS (Visual Object detection with a CompUtational attention System) [4] (Fig. 2). It consists of a bottom-up part similar to [8], and a top-down part enabling goal-directed search; global saliency is determined from both cues.

The bottom-up part detects salient image regions by computing image contrasts and uniqueness of a feature. The feature 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*; on-off and off-on contrasts are computed separately. After summing up the scales, this yields 2 intensity maps. Similarly, 4 orientation maps $(0^\circ, 45^\circ, 90^\circ, 135^\circ)$



Feature	vector $\boldsymbol{v}$
intensity on/off	0.16
intensity off/on	6.54
orientation 0°	2.06
orientation 45°	3.17
orientation 90°	3.06
orientation 135°	2.95
color green	3.14
color blue	1.35
color red	1.36
color yellow	0.93
conspicuity I	3.30
conspicuity O	2.60
conspicuity C	1.70

Fig. 2. Left: the visual attention system VOCUS. Right: feature vector for the MSR

are computed by Gabor filters and 4 color maps (green, blue, red, yellow) which highlight salient regions of a certain color. Each feature map i is weighted with a uniqueness weight $\mathcal{W}(i) = i/\sqrt{m}$, where m is the number of local maxima that exceed a threshold. This promotes pop-out features. The maps are summed up to 3 conspicuity maps I (intensity), O (orientation) and C (color) and combined to form the *bottom-up saliency map* $S_{bu} = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C)$.

If no top-down information is available, S_{bu} corresponds to the global saliency map S. In S, the *most salient regions (MSRs)* are determined: first the local maxima in S (seeds) are found and second all neighboring pixels over a saliency threshold (here: 25% of the seed) are detected recursively with *region growing.* A *region of interest (ROI)* is defined as *height* ∗ *width* of the MSR. For each *MSR*, a feature vector $\boldsymbol{v}$ with $(2 + 4 + 4 + 3 = 13)$ entries (one for each feature and conspicuity map) is determined. The feature value v_i for map i is the ratio of the mean saliency in the target region $m_{(MSR)}$ and in the background $m_{(image-MSR)}$: $v_i = m_{(MSR)}/m_{(image-MSR)}$. This computation does not only consider which features are the strongest in the target region, it also regards which features separate the region best from the rest of the image. Fig. 2 right shows a feature vector which corresponds the MSR of the image on the left. It tells us, e.g., that the region is dark on a bright background (off-on intensity).

In top-down mode, VOCUS aims to detect a target, i.e., input to the system is the image and some target information, provided as feature vector $\boldsymbol{v}$. In *search mode*, VOCUS multiplies the feature and conspicuity maps with the weights of $\boldsymbol{v}$. The resulting maps are summed up, yielding the *top-down saliency map* S_{td}. Finally, S_{bu} and S_{td} are combined by: $S = (1-t)*S_{bu}+t*S_{td}$, where t determines the contributions of bottom-up and top-down (details in [4]).

Harris corners: To detect features with high position stability inside the ROIs, we used the Harris-Laplace feature detector [11] – an extension of the Harris corner detector to Laplacian pyramids which enables scale invariance. This resulted in a few (average 1.6) points per ROI (cf. Fig. 3 bottom right). To allow matching of points, a SIFT descriptor is computed for each detected corner [10].

4 Matching and Tracking of Features

Feature matching is performed in the feature tracker (for creating landmarks) and in the loop closer (to detect if this landmark has been seen before). The matching is based on two criteria: proximity and similarity. First, the features in the new frame have to be close enough to the predicted position. Secondly, the similarity of the features is determined. This is done differently for attentional ROIs and for Harris corners: the matching of Harris corners is based on the SIFT descriptor by determining the Euclidean distance between the descriptors. When the distance is below a threshold, the points match.

For the attentional ROIs, we consider the size of the ROIs and the similarity of the feature values. We set the allowed deviation in width and height of the ROI to 10 pixels to allow some variations. This is required, because the ROIs might differ slightly in shape depending on image noise and illumination variations.

The similarity of two feature vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ is determined by:

$$d = \sqrt{\frac{\boldsymbol{v}_{11}\boldsymbol{w}_{11} \sum\limits_{i=1,2} (\boldsymbol{v}_i - \boldsymbol{w}_i)^2 + \boldsymbol{v}_{12}\boldsymbol{w}_{12} \sum\limits_{i=3,..,6} (\boldsymbol{v}_i - \boldsymbol{w}_i)^2 + \boldsymbol{v}_{13}\boldsymbol{w}_{13} \sum\limits_{i=7,..,10} (\boldsymbol{v}_i - \boldsymbol{w}_i)^2}{\boldsymbol{v}_{11}\boldsymbol{w}_{11} + \boldsymbol{v}_{12}\boldsymbol{w}_{12} + \boldsymbol{v}_{13}\boldsymbol{w}_{13}}}$$

The computation is similar to the Euclidean distance of the vectors, but it treats the feature map values ($\boldsymbol{v_1}$,..,$\boldsymbol{v_{10}}$) differently than the conspicuity map values ($\boldsymbol{v_{11}}$,..., $\boldsymbol{v_{13}}$). The reason is as follows: the conspicuity values provide information about how important the respective feature maps are. For example, a low value for the color conspicuity map $\boldsymbol{v_{13}}$ means the values of the color feature maps ($\boldsymbol{v_7}$,...,$\boldsymbol{v_{10}}$) are not discriminative and should be assigned less weight than the other values. Therefore, we use the conspicuity values to weight the feature values. We found out in several experiments that this matching procedure outperforms the simple Euclidean distance of the feature vectors considerably.

If the distance d is below a certain threshold δ, the ROIs match. We use different values for tracking ($\delta = 3.0$) and loop closing ($\delta = 1.7$). When tracking, the estimated position from odometry is usually accurate, and we can afford a more relaxed threshold than for loop closing where the position estimation is less accurate. In [5], we investigated the choice of the threshold in detail.

In the feature tracker, the features are tracked over several frames. We store the last n frames in a buffer (here: $n = 30$). This buffer provides a way to determine which landmarks are stable over time and thus good candidates to use in 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 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*.

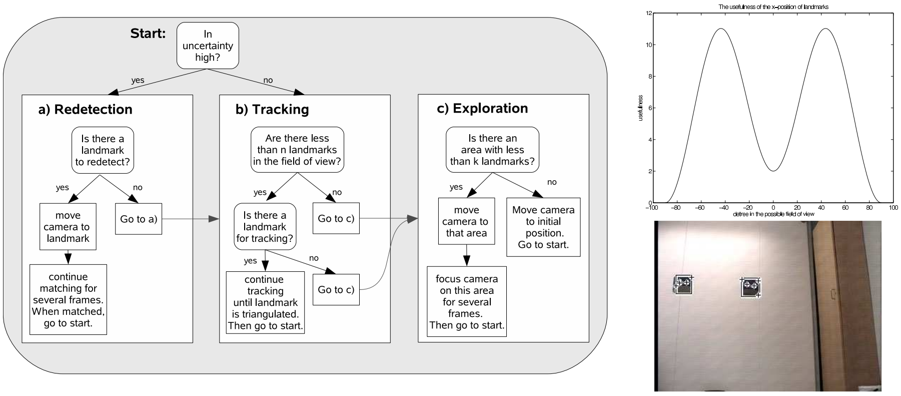

Start:
In uncertainty high?
yes
no
a) Redetection
Is there a landmark to redetect?
yes
no
move camera to landmark
Go to a)
continue matching for several frames. When matched, go to start.
b) Tracking
Are there less than n landmarks in the field of view?
yes
no
Is there a landmark for tracking?
Go to c)
yes
no
continue tracking until landmark is triangulated. Then go to start.
Go to c)
c) Exploration
Is there an area with less than k landmarks?
yes
no
move camera to that area
Move camera to initial position. Go to start.
focus camera on this area for several frames. Then go to start.
The usefulness of the x–position of landmarks
usefulness
0
2
4
6
8
10
12
−100
−80
−60
−40
−20
0
20
40
60
80
100
detree in the possible field of view

Fig. 3. Left: the three camera behaviours. Right top: usefulness function U. Bottom: example image with two ROI-landmarks and several Harris-landmarks. The landmarks of the left ROI are more useful, since they are not in the center of the field of view.

Creating Landmarks: A *landmark* is a list of tracked features. Features can be ROIs (ROI-landmark) or Harris corners (Harris-landmark). The *length* of a landmark is the number of elements in the list, which is equivalent to the number of frames the feature was detected in. 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 landmarks 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).

5 Active Gaze Control

The active gaze control is divided into three behaviours: a) redetection of landmarks to close loops, b) tracking of landmarks, and c) exploration of unknown areas. The strategy to decide which behaviour to choose is as follows (Fig. 3): Redetection has the highest priority, but it is only chosen if the position uncertainty is over a certain value. If the uncertainty is low or if there is no expected landmark for redetection, the *tracking* behaviour is activated. Tracking is only performed if there are not yet enough landmarks in this area. As soon as a certain amount of landmarks is obtained in the field of view, the *exploration* behaviour takes over. It moves the camera to an area with no detected landmarks.

Redetection: The redetection of landmarks is performed if the current robot pose uncertainty is high and there are old landmarks that are or could be made visible through active camera control. This information is provided by the SLAM module. If there is an expected landmark and the robot pose uncertainty is high, the camera is moved to focus on the expected landmark. If we have more than one expected landmark, we have to choose the potentially most useful landmark for redetection. Here, we consider only the length of the current ROI-landmark: the longer this landmark, the better. The new camera position is maintained until a match is performed or until a waiting threshold is exceeded.

Tracking: 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. 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 ROI- and Harris-landmarks are important factors for the usefulness of a landmark, 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 result often in poor triangulation results. 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, we determine the usefulness of a landmark by first considering the length of the ROI-landmark, second the angle of the landmark in the potential field of view, and third the length of the Harris-landmark. The length of the ROI-landmarks is considered by sorting out landmarks below a certain size (here: 5). The usefulness of the angle of a ROI is determined by the following function:

$$w = (k_1\,(1.0 + \cos(4\,(\alpha - \pi))) + k2\,(1.0 + \cos(2\alpha))) \tag{1}$$

where α is the angle and $k_1 = 5$ and $k_2 = 1$. The function is displayed in Fig. 3 (top 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.

The usefulness U of a Harris-landmark is then determined by: $U = w\sqrt{l}$, where l is the length of the landmark. In Fig. 3, we demonstrate the effect of U. The bottom-right image shows two identical regions on the wall, both are detected by VOCUS and have several Harris corners which were detected inside the ROI. The main difference between the landmarks is that one of them is almost in the center of the image and the other one at the border (the camera points straight ahead). The values w and U are higher for the landmark on the left. This leads to choosing the left landmark for tracking since it is likely that it provides a better baseline for triangulation.

After determining the most useful landmark for tracking, the camera is moved into the direction of the landmark. It is moved slowly (here 0.1 radians per step), since this turned out to be more successful than moving it quickly to center the landmark. This corresponds to the pursuit eye movements of humans when following a target. On the other hand, the quick camera motion for the redetection and exploration behaviour corresponds to saccades (quick eye movements) in human viewing behaviour, which are performed when searching for a target or exploring a scene. The tracking ends when the landmark is not visible any more

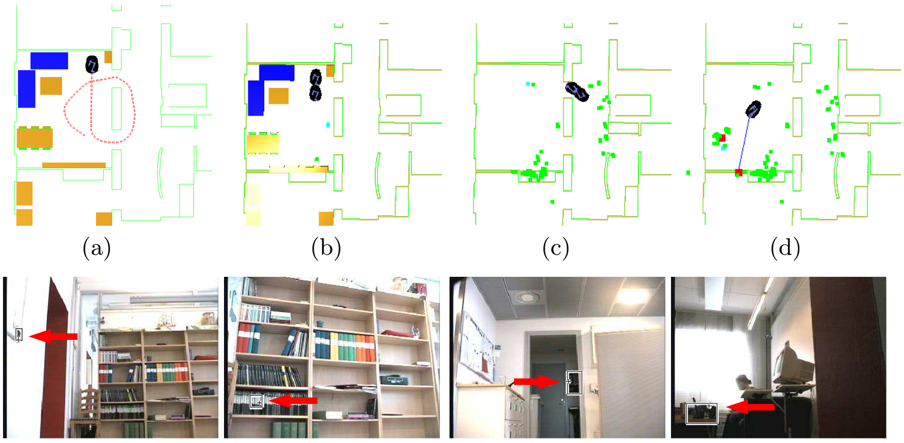


Fig. 4. Top: Snapshots of the robot sequence with active camera control. Two robots in one image correspond to the robot at the beginning and at the end of the buffer, i.e., the robot further ahead on the path is the real robot, the one behind is the virtual robot position 30 frames later. Currently visible landmarks are displayed as cyan dots, currently not visible landmarks in green. Landmarks matched to database entries are larger and displayed in red. When the robot tries to redetect a landmark, the estimated direction of the landmark is displayed as a blue line. a) robot trajectory. b) first landmarks detected). c) more landmarks detected. d) loop closing: a landmark is expected and matched successfully. **Bottom:** Some examples of ROIs and Harris points chosen for tracking. The red arrows point to these regions.

(because it left the field of view or because the matching failed) or when the landmark was successfully triangulated.

Exploration: In exploration mode, the camera is moved to an area in the possible field of view where the map contains no landmarks. To avoid too many camera movements and to enable building of landmarks over several frames, the camera focuses one region for a while (here 10 frames). As soon as a landmark for tracking is found, the system switches automatically the behaviour.

6 Experiments and Results

To illustrate the robot behaviour in our framework, the robot drove in our office environment according to the path displayed in Fig. 4 top a). During the sequence, 642 images were processed and 61 landmarks were determined which were tracked on average over 10 frames. In Fig. 4 top, we show some snapshots of the environment and the landmarks which were determined during the sequence. Some of the landmarks chosen for tracking are displayed in Fig. 4 bottom. When the robot starts driving, it first keeps the camera in its initial position for 15 frames to build some landmarks. Then, it starts to choose behaviours. Since the uncertainty is low in the beginning, it does not consider the redetection behaviour and switches over to the tracking. The first landmark which is considered for tracking is displayed in Fig. 4, bottom left. It is chosen because it pops out of the image, which makes it stable over consecutive frames, and because it is at the border of the image, which promises a good baseline for triangulation. The landmark is now tracked until it is triangulated. Its estimated position is displayed as the cyan dot in Fig. 4 top b) (near the wall on the right).

After this landmark is triangulated, the tracking switches over to track a new landmark (Fig. 4 bottom b)). This is continued until there is no landmark found for tracking or until there are enough (> 5) landmarks in the field of view. As soon as there were enough landmarks in the field of view, the exploration behaviour was started. It picked the door region since there were no landmarks found yet. It found several landmarks for tracking there, e.g. the one in Fig. 4 bottom c). While driving through the hallway, the tracking and exploration behaviour alternated, resulting in the landmarks displayed in Fig. 4 top c). Some of the landmarks in the middle of the hallway correspond to points in the ceiling.

After entering the room again, the robot detected several more landmarks (Fig. 4 top d)). Some of them the middle of the room correspond to the furniture visible in Fig. 4 top a). Finally, the conditions for redetection are fulfilled: the position uncertainty exceeds the threshold and several landmarks are expected to be visible. The system picks the one with the longest ROI-landmark and moves the camera to focus this region. The direction of the expected landmark is indicated by the blue line in Fig. 4 top d). Finally, this landmark is successfully matched to an entry in the database as indicated by the red dot. This first match is displayed in Fig. 5. After the match, the system chooses the next behaviour.

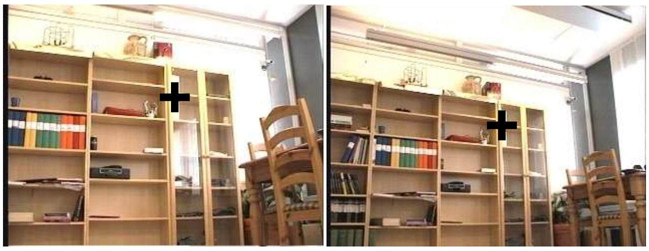


Fig. 5. A match of Harris points between a current frame (left) and a scene from the database (right). It corresponds to the red dot at the end of the blue line in Fig. 4,d).

7 Conclusion

In this paper, we presented a visual SLAM system based on an attentional landmark detector. The attentional regions are especially useful landmarks for tracking and redetection. Three behaviours for active camera control help to handle some of the problems of visual SLAM: landmarks with a better baseline are preferred and a better distribution of landmarks is achieved.

Needless to say, a lot has still to be done. Currently, VOCUS works only in bottom-up mode and the ROIs are matched according to their bottom-up appearance. Including top-down behaviour would improve both the tracking and the redetection of landmarks. Interesting would be also the combination of the method with other visual loop-closing techniques, for example by considering not only one expected landmark for matching, but all in the current field of view.

8 ACKNOWLEDGMENTS

The present research has been sponsored by the European Commission through the project NEUROBOTICS (EC Contract FP6-IST-001917) and the Swedish Foundation for Strategic Research through its Center for Autonomous Systems. The support is gratefully acknowledged.

References

1. J. Aloimonos, I. Weiss, and A. Bandyopadhyay. Active vision. *Intl. Jour. of Computer Vision*, 1(4):333–356, Jan. 1988.
2. A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. of the ICCV*, oct 2003.
3. A. J. Davison and D. W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Trans. PAMI*, 2002.
4. S. Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. PhD thesis, 2005. Published 2006 in LNAI, Vol. 3899, Springer.
5. S. Frintrop, P. Jensfelt, and H. Christensen. Attentional Landmark Selection for Visual SLAM. In *Proc. Int'l Conf. on Intelligent Robots and Systems*, 2006.
6. S. Frintrop, P. Jensfelt, and H. Christensen. Pay attention when selecting features. In *Proc. Int'l Conf. on Pattern Recognition (ICPR 2006)*, 2006.
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 ICRA*, pages 44–49, apr 2005.
8. L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11), 1998.
9. P. Jensfelt, D. Kragic, J. Folkesson, and M. Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. of ICRA'06*, Orlando, FL, May 2006.
10. D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. of ICCV*, pages 1150–57, 1999.
11. K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. of ICCV*, pages 525–531, 2001.
12. P. Newman and K. Ho. SLAM- loop closing with visually salient features. In *Proc. Int'l Conf. on Robotics and Automation, (ICRA 2005)*, 2005.
13. 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. *J. on RAS*, 25(1-2):83–104, 1998.
14. N. Ouerhani, A. Bur, and H. Hügli. Visual attention-based robot self-localization. In *Proc. of European Conf. on Mobile Robotics (ECMR 2005)*, 2005.
15. S. Thrun. Finding landmarks for mobile robot navigation. In *Proc. of ICRA*, 1998.
16. J. K. Tsotsos, S. M. Culhane, W. Y. K. Wai, Y. Lai, N. Davis, and F. Nuflo. Modeling visual attention via selective tuning. *AI*, 78(1-2), 1995.
17. T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D. W. Murray. Active control for single camera SLAM. In *Proc. of ICRA 2006*, 2006.