# Attentional Landmark Selection for Visual SLAM

Simone Frintrop, Patric Jensfelt and Henrik I. Christensen
Computational Vision and Active Perception Laboratory (CVAP), Computer Science and Communication (CSC)
Kungliga Tekniska Högskolan (KTH)
SE-100 44 Stockholm, Sweden
Email: {frintrop/patric/hic}@nada.kth.se

***Abstract*— In this paper, we introduce a new method to automatically detect useful landmarks for visual SLAM. A biologically motivated attention system detects regions of interest which "pop-out" automatically due to strong contrasts and the uniqueness of features. This property makes the regions easily redetectable and thus they are useful candidates for visual landmarks. Matching based on scene prediction and feature similarity allows not only short-term tracking of the regions, but also redetection in loop closing situations. The paper demonstrates how regions are determined and how they are matched reliably. Various experimental results on real-world data show that the landmarks are useful with respect to be tracked in consecutive frames and to enable closing loops.**

## I. Introduction

Self localization is a key competence for an autonomous mobile robot and has been a topic for much research within the robotics community over the years. Two main paradigms can be identified, metric [1], [2], [3] and topological [4], [5], [6] localization approaches. In the metric methods, the robot position is represented by coordinates in some coordinate system whereas in the topological methods the robot position is defined to be at some place in the topological graph. Such a place can be for example a room or some functional unit such as a copying machine.

An active area of research in mobile robotics is *simultaneous localization and mapping (SLAM)*, where a map is autonomously constructed of the environment. Obviously, SLAM is closely related to localization and has been approached using both topological and metric methods. SLAM for indoor settings based on laser scanners is today considered a mature technology. However, the laser scanner is much too expensive for many applications. Therefore and because of the high amount of information offered by camera images, the focus within the community has shifted towards using visual information instead [7], [8], [9], [10]. One of the key problems in SLAM is *loop closing*, i.e. the ability to detect when the robot is revisiting some area that has been mapped earlier. This is similar to the problem of place recognition faced in the topological approaches to localization.

To perform SLAM, landmarks have to be detected in the environment. A key characteristic for a good landmark is that it can be reliably detected, this means that the robot is able to detect it over several frames. Additionally, it should be redetectable when the same area is visited again, this means the detection has to be stable under viewpoint changes. Often, the landmarks are selected by a human expert or the kind of landmark is determined in advance. Examples include localization based on ceiling lights [11]. As pointed out by [12], 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.

In this paper, we turn the attention to computational visual attention systems [13], [14], [15]. They select regions that "pop out" in a scene due to strong contrasts and uniqueness, e.g., 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 like corners. Some systems additionally integrate previous knowledge (top-down information) into the computations [16], [15]. This enables a better redetection of landmarks when presuming to revisit a known location. Applications of visual attention systems are found in object recognition [17], image segmentation [18], and active vision [19].

Attention methods are well suited for selecting unique regions which are good landmark candidates, yet the application of attention systems to landmark selection has rarely been studied. Two existing approaches are [20], in which landmarks are detected in hand-coded maps, and [21], in which a topological map is build. The only approach we are aware of which uses an approach similar to a visual attention system for SLAM, is presented in [9]. They use a saliency measure based on entropy to define important regions in the environment primarily for the loop closing detection in SLAM. The map itself is built using a laser scanner though.

In this paper, we introduce a new method to detect useful landmarks for visual SLAM based on the visual attention system VOCUS [15]. The focus of the paper is on the detection of landmarks, on tracking of landmarks over frames, and on the ability to detect loop closing situations. We evaluate the usefulness of the attentional regions of interest (ROIs) and show that ROIs are easily tracked over many frames even without using position information and are well suited to be redetected in loop closing situations.

## II. System Overview

In this section, we give an overview over the architecture in which the landmark detection is embedded (see Fig. 1). The main components are the *robot* which provides camera images and odometry information, a *landmark detector* to create landmarks, a *database* to store the landmarks, and a *SLAM module* to build a map of the environment.

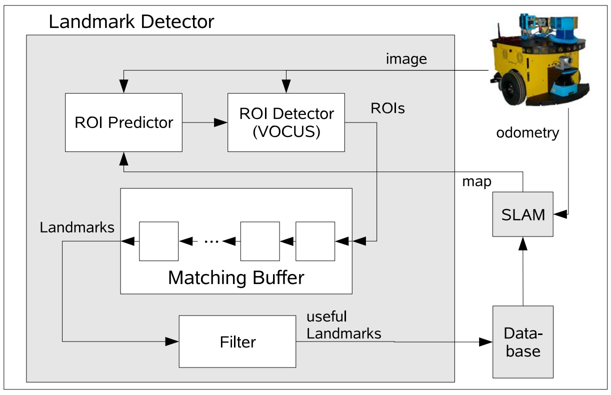


Fig. 1. Attentional landmark selection for visual SLAM

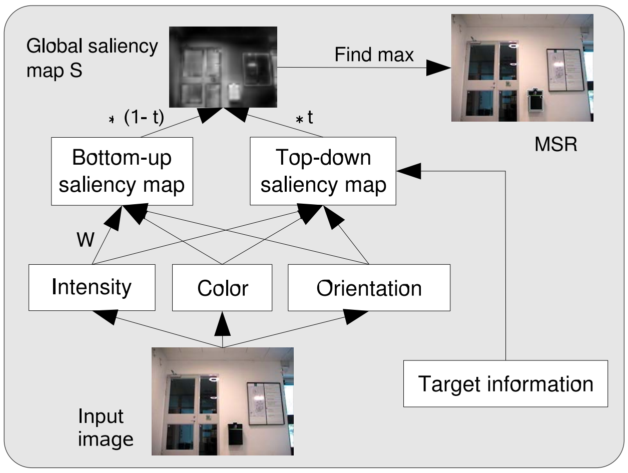


Fig. 2. The visual attention system VOCUS

When a new frame from the camera is available, the motion of the camera since the last frame is first checked. If there was not enough movement (here more than $3cm$ translation or $1°$ rotation), the frame is ignored. Otherwise, the frame is provided to the landmark detector. The reason for ignoring frames with too little motion in between is that we want to assess the quality of landmarks and thus want to analyze images taken from different view points.

The landmark detector consists of a *ROI predictor* which predicts the position and appearance of landmarks depending on previous frames and on position estimations from the SLAM module, a *ROI detector* which redetects predicted ROIs and finds new ROIs based on the visual attention system VOCUS, a *matching buffer* which stores the last $n$ frames, performs matching of the ROIs in these frames and creates landmarks from matched ROIs, and a *filter module* which filters out low quality landmarks.

The landmarks which pass the filter are stored in the database and provided to the SLAM module which performs the estimate of the position of landmarks and integrates the position into the environmental map. To detect old landmarks, the landmark position estimates from the SLAM module are used to narrow down the search space in the database.

The focus in this paper is on the detection and tracking of landmarks, therefore the landmark detector is explained most detailed. Special emphasis is on the detection of ROIs (ROI detector) and on the creation of landmarks (matching buffer). Details about the robot and the SLAM architecture can be found in [10].

## III. ROI Detection with VOCUS

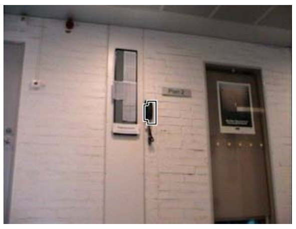
The detection of regions of interest (ROIs) is performed with VOCUS (Visual Object detection with a CompUtational attention System) [22], [15]. Motivated from the human visual system, it detects salient regions in images. VOCUS differs from most other systems by the ability to consider target knowledge (top-down information) to enable goal-directed search. It consists of a bottom-up part and a top-down part; global saliency is determined from both cues (cf. Fig. 2).

The bottom-up part detects salient image regions by computing image contrasts and uniqueness of a feature, e.g., a red ball on green grass. 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* [22]; on-off and off-on contrasts are computed separately. After summing up the scales, this yields 2 intensity maps. Similarly, 4 orientation maps ($0°, 45°, 90°, 135°$) are computed by Gabor filters and 4 color maps (green, blue, red, yellow) which highlight salient regions of a certain color (details in [22]). Each feature map X is weighted with the uniqueness weight $\mathcal{W}(X) = X/\sqrt{m}$, where $m$ is the number of local maxima that exceed a threshold. This weighting is essential since it emphasizes important maps with few peaks, enabling the detection of *pop-outs*. After weighting, the maps are summed up first to 3 conspicuity maps $I$ (intensity), $O$ (orientation) and $C$ (color) and finally, after again weighting for uniqueness, to the *bottom-up saliency map* $S_{bu} = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C)$ (cf. Fig. 4, top left).

In top-down mode, the system aims to detect a target, i.e., input to the system is the image and some target information, provided as a feature vector $\vec{v}$. This vector is learned from a region which is provided manually or automatically; in this application it is determined automatically from a *most salient region (MSR)* of $S_{bu}$ (see below). In *search mode*, the system multiplies the feature and conspicuity maps with the weights of $\vec{v}$. The resulting maps are summed up, yielding the *top-down saliency map* $S_{td}$ (cf. Fig. 4, bottom left). 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.

Fig. 4 shows the differences of bottom-up and top-down saliency: the bottom-up saliency map highlights all regions which might be of interest, regardless of a certain target. It can be used to redetect a region without actively searching for it. The top-down map shows much higher values for the target regions. It is especially well suited to actively search for a target.

In both bottom-up and top-down mode, the MSRs in $S$ are determined the same way: first the local maxima in $S$ (seeds)



| 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 |
| conspicuity I | 4.99 |
| conspicuity O | 5.70 |
| conspicuity C | 2.52 |

Fig. 3. Left: image with most salient region (MSR). Right: feature vector for the MSR.

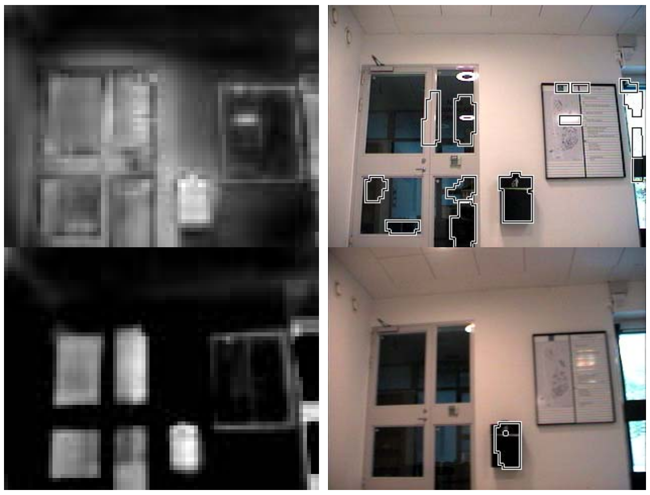


Fig. 4. Saliency maps (top, left: $S_{bu}$, bottom, left: $S_{td}$) and MSRs in a loop closing example: Top: scene at beginning of sequence. Bottom: revisited scene, 592 frames later, searching for the black waste bin with top-down attention.

are found and second all neighboring pixels over a saliency threshold (here: 25% of the seed) are detected recursively with *region growing* (Fig. 4, col. 2). For simplicity, we use the rectangle determined by height and width of the MSR as *region of interest (ROI)* (cf. Fig. 7). Furthermore, we ignore MSRs on the borders of images, since these are not stable when the viewpoint changes.

For each *MSR*, a feature vector $\vec{v}$ with $(2+4+4+3 = 13)$ entries (one for each feature and conspicuity map) is determined. It describes the values of a map in this region compared with the values in the rest of the image. 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. 3 shows an MSR and the corresponding feature vector $\vec{v}$. 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 (conspicuity weights I, O, C). Although the target is black on a white background, there are values for the color maps. This results from the camera, which produces slightly reddish images. Therefore, the background wall appears slightly red too, resulting in higher values for green and blue and lower for red and yellow for a black target.

The feature computations are efficiently performed on *integral images* [23]. After once creating an integral image in linear time with respect to the number of pixels, a rectangular feature value of arbitrary size is computed with only 4 references. This results in a fast computation (50ms for $400 \times 300$ pixel image, 2.8GHz) that enables real-time performance.

In [24], we show an extension of the ROI detection: we compute additionally Harris corners [25] inside the ROIs. Harris corners are very stable concerning position which is useful for estimating the depth of a landmark from several frames. This approach combines the advantages of the ROIs (complexity reduction, good redetectability) with the advantages of the Harris corners (high position stability).

## IV. The Matching Buffer

The matching buffer is a structure which stores the last $n$ frames (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 matching 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. This is necessary, because ROIs are sometimes not detected in each frame. We call frames within this distance *close frames*.

Fig. 5 shows the matching buffer. New frames enter the buffer on the right and each frame contains the ROIs detected by VOCUS. The matching buffer matches ROIs between frames, appends them to existing landmarks, or creates new landmarks when the ROIs do not fit to existing ones. The figure shows three different examples of matching: on the top, a ROI is matched to the last ROI of an existing landmark. In the middle, a ROI is matched to a single ROI from the previous frame and a new landmark is created. The lowest ROI was not detected in the previous frame (frame $n-1$), but in the frame before (frame $n-2$); it belonged to an existing landmark. The gap is ignored and the ROI is appended to the landmark.

### A. Matching ROIs

The matching is based on two criteria: proximity and similarity. First, the ROI in the new frame has to be within a close area around the predicted point position from the old frame. Secondly, the similarity of the ROIs is determined considering the size of the ROIs and the similarity of the feature values.

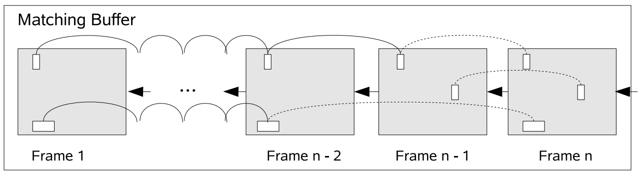


Fig. 5. The matching buffer contains the last $n$ frames (here: $n = 30$). New frames enter the buffer on the right. Each frame contains several ROIs (rectangles inside the frames). If the buffer matches new ROIs successfully to old ones, landmarks are created (curved lines between ROIs).

The position of a ROI in a new frame is predicted by considering the relative motion of the camera between the old and the new frame. Provided we have an estimate of how the camera has moved between frames, we can predict where ROIs seen before should be now.

Next, the similarity of the ROIs is determined depending on the size of the ROI and on the similarity of the feature vector. We set the allowed deviation in width and height of the ROI to 10 pixels each to allow some variations. This is useful, because the ROIs differ sometimes slightly in shape depending on image noise and illumination variations. Additionally, the Euclidean distance between two feature vectors $\vec{v_1}$ and $\vec{v_2}$ is determined: $d = ||\vec{v_1}, \vec{v_2}|| = \sqrt{(v_{1,1} - v_{2,1})^2 + ... + (v_{1,13} - v_{2,13})^2}$. If $d$ is below a certain threshold $\delta$, the ROIs are said to match.

To summarize, if a ROI in a new frame is close to the predicted position of an old ROI and the ROIs are similar concerning size and feature values, the ROIs are said to match.

## B. Creating Landmarks

A landmark is a list of ROIs which were matched successfully (cf. Fig. 5). The procedure to create landmarks is the following: if a new frame comes into the buffer, each of its ROIs is matched to all existing landmarks corresponding to close frames. This means, from each landmark the most recently appended ROI is taken, it is checked if it belongs to a close frame, and then it is matched to the new ROI. If the matching is successful, the new ROI is appended to the end of this landmark. Additionally, the new 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. In a final step, we consider the length of the resulting landmarks and filter out those which consist of too few ROIs (here less than 5).

## V. Experiments and Results

The experiments were performed on a sequence of 658 images, obtained in a hallway by a robot driving as shown in Fig. 6, and consist of two parts: first, we show how the matching of ROIs can be used for tracking over consecutive frames and investigate the matching quality with and without position prediction. Second, we show how the matching of ROIs can be used in loop closing situations.

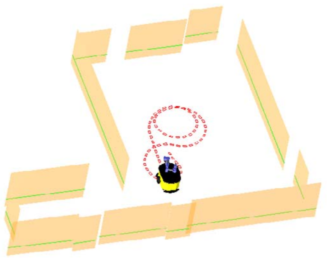


Fig. 6. The robot environment and the driven trajectory.

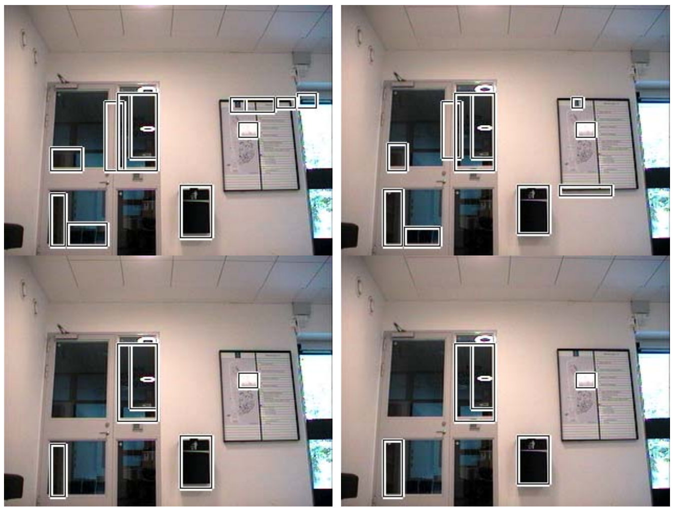


Fig. 7. Top: all found ROIs in two consecutive frames. Bottom: The ROIs which were matched between the frames only depending on their feature vector. All matches are correct.

*1) Tracking:* First, we investigate the matching of ROIs between consecutive frames. To investigate the matching by similarity independently from proximity, we first only consider matching of the ROIs based on the size of the ROIs and the similarity of the feature vector and second combine it with the position prediction.

Fig. 7 shows the matching for an example case with $\delta = 1.7$. The top row shows all ROIs that were found in two consecutive frames. In the first frame, 12 ROIs were found, in the second 10. Now, the matching depending on size and similarity of the feature vector is performed. Remember, that we do not consider any position information here. In the bottom row, the 5 ROIs which survive the matching are displayed. All of them are correct.

We investigated how the number of landmarks, the length of the landmarks (i.e. the number of associated ROIs), and the quality of the matching (nb. false matches) depends on the choise of the matching threshold $\delta$ (Tab. I, top). It shows that the number of landmarks increases when $\delta$ increases as well as the length of the landmarks, but on the other hand there are also significantly more false matches.

Finally, we performed the same experiments with additional position prediction (Tab. I, bottom). It shows that the false matches are avoided and the matching is robust even for higher matching without prediction:

| Threshold | # landm. | # ROIs | # ROIs/Landmark (av) | # false m. |
|---|---|---|---|---|
| 1.7 | 62 | 571 | 9.2 | 1 |
| 2.0 | 73 | 730 | 10.0 | 7 |
| 2.5 | 88 | 957 | 10.9 | 15 |
| 3.0 | 102 | 1109 | 10.9 | 42 |
| 5.0 | 130 | 1568 | 12.0 | 147 |

matching with prediction:

| Threshold | # landm. | # ROIs | # ROIs/Landmark (av) | # false m. |
|---|---|---|---|---|
| 1.7 | 61 | 566 | 9.3 | 0 |
| 2.0 | 73 | 724 | 9.9 | 0 |
| 2.5 | 88 | 955 | 10.9 | 0 |
| 3.0 | 98 | 1090 | 11.1 | 0 |
| 5.0 | 117 | 1415 | 12.1 | 0 |

TABLE I

MATCHING OF ROIs FOR DIFFERENT THRESHOLDS $\delta$

values of $\delta$. This means that for a high threshold combined with position prediction, a higher amount of landmarks is achieved while preserving the tracking quality. Fig. 8 shows an example of tracking a ROI on a telephone over 20 frames. It shows that although the viewpoint changes strongly, the tracking is stable and robust. We also tried to perform the matching based only on the position prediction without considering the ROI similarity at all, but it turned out that nested ROIs can not be distinguished any more. This means, the results are still good, but the shape of the ROIs is less stable. Therefore, we recommend using both criteria for matching.

*2) Loop closing:* In the next experiment, we investigate the matching quality in loop closing situations. In our experimental setup, the robot drives 2.5 times in a circle. This means, most of the visited regions appear again after a while, a perfect setting to investigate the ability to close loops.

We proceed as follows: for each ROI that is detected in a new frame, we match to all ROIs from all landmarks that occurred so far. We used a tight threshold for the matching ($\delta = 1.7$) and assumed that no position information is available for the ROIs. This case corresponds to the "kidnapped robot problem", the most difficult version of loop closing. If additional position information from odometry is added, the matching gets more precise, false matches are avoided and the threshold might be chosen higher.

In these experiments, the system attempted to perform $974\,256$ matches between ROIs (all current ROIs were matched to all ROIs of all landmarks that were detected so far). Out of these possible matches, $646$ were matched, $575$ ($89\%$) of these matches were correct. Fig. 9 shows four examples of correct matches, two examples of false matches are depicted in Fig. 10.

The experiments show that the attentional ROIs are useful landmarks which are both successfully tracked over consecutive frames and suitable to be redetected after visiting an area again from a different viewpoint.

## VI. CONCLUSION

Image based localization is an emerging methodology that has a significant potential for new types of robot applications.

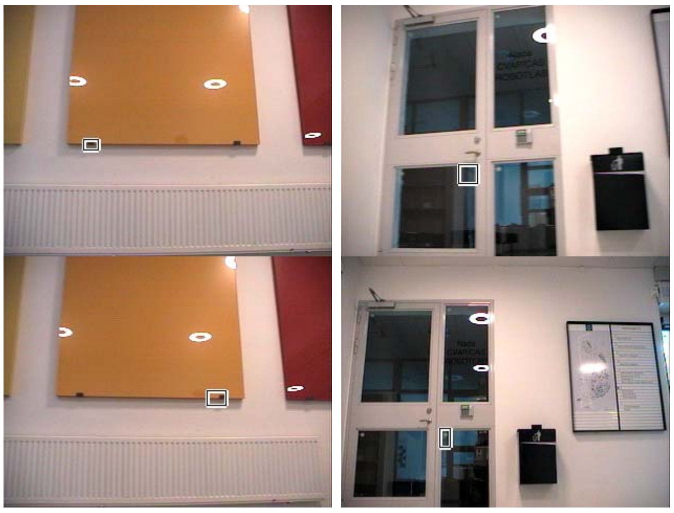


Fig. 10. False matches in loop closing situations: the first row shows a ROI in a current frame, the second row shows a previously stored ROI the current ROI was matched with. The matched regions are associated wrongly since they are highly ambiguous.

To make image based localization tractable, there is a need to detect major salient structures in the scene. In this paper, we have presented a method for landmark selection based on a biologically motivated attention system which detects salient regions of interest. Scene prediction and similarity of features are used to achieve stable matching results. The matching shows to be successful not only in short-term tracking but also in loop closing. The detection and matching method has been tested in a SLAM scenario to demonstrate the basic performance for in-door navigation.

The present work has been performed off-line and future work is consequently going to consider an on-line evaluation. In addition, there is a need to study the performance of the method for loop closing and estimation in the presence of significantly higher scene complexity.

## ACKNOWLEDGMENT

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. Additional thanks to Maria Klodt for speeding up VOCUS to enable real-time performance.

## REFERENCES

[1] J. Leonard and H. Durrant-Whyte, "Mobile robot localization by tracking geometric beacon," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 376–382, June 1991.

[2] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, "Monte carlo localization - efficient position estimation for mobile robots," in *Proc. of the National Conference on Artificial Intelligence (AAAI-99)*, Orlando, Florida, USA, July 18-22, 1999.

[3] K. Arras, N. Tomatis, and R. Siegwart, "Multisensor on-the-fly localization using laser and vision," in *Proc. of IROS'00*, vol. 1, 2000, pp. 462–476.

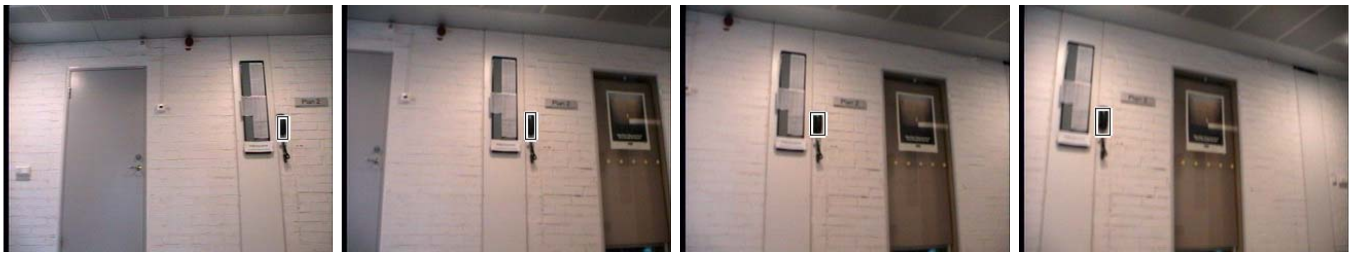


Fig. 8. Tracking of a landmark after 1, 10, 15, 20 frames with $\delta = 5.0$ and additional position prediction.

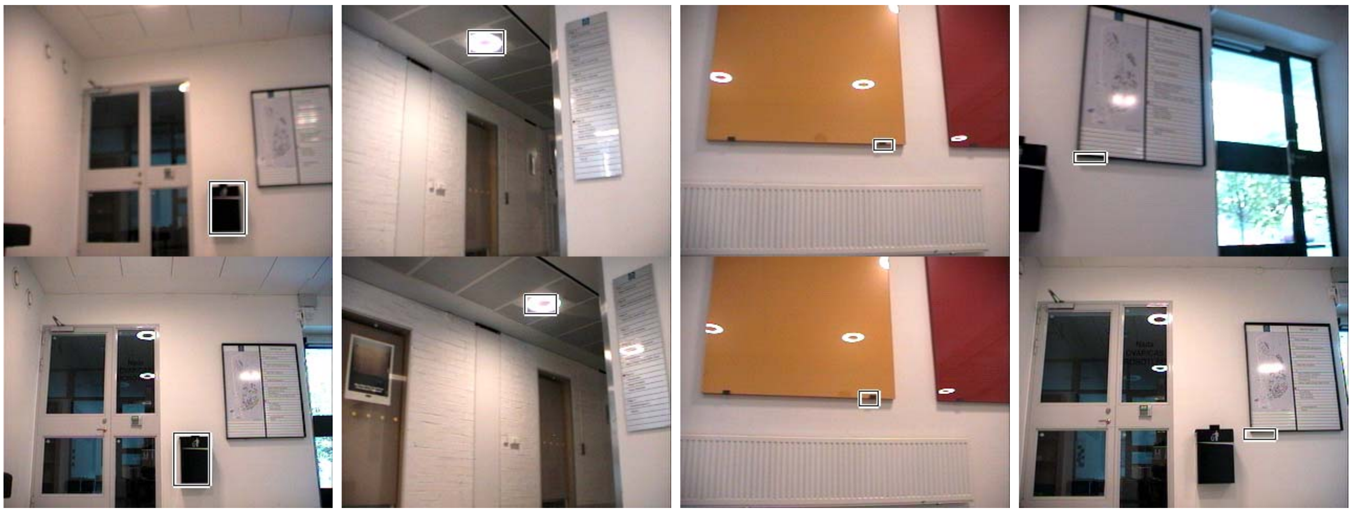


Fig. 9. Correct matches in loop closing situations: the first row shows a ROI in a current frame, the second row shows a previously stored ROI the current ROI was matched with. This situation usually occurred after the robot drove a circle and revisited the same area again.

[4] B. J. Kuipers, "Representing knowledge of large-scale space," MIT Artificial Intelligence Laboratory, Tech. Rep. TR-418 (revised version of Doctoral thesis May 1977, MIT Mathematical Department), July 1977.

[5] D. Kortenkamp and T. Weymouth, "Topological mapping for mobile robots using a combination of sonar and vision sensing," in *Proc. of the National Conference on Artificial Intelligence (AAAI-94)*, 1994.

[6] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," in *Proc. of ICRA'00*, vol. 2, Apr. 2000, pp. 1023–1029.

[7] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *Proc. of the ICCV*, oct 2003.

[8] 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.

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

[10] 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.

[11] 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 Robotics Research*, vol. 19, no. 11, 2000.

[12] S. Thrun, "Finding landmarks for mobile robot navigation," in *Proc. of the 1998 IEEE International Conf. on Robotics and Automation (ICRA '98)*, Leuven, Belgium, May 1998.

[13] 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.

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

[15] S. Frintrop, G. Backer, and E. Rome, "Goal-directed search with a top-down modulated computational attention system," in *Proc. of DAGM 2005*, Sept. 2005.

[16] V. Navalpakkam, J. Rebesco, and L. Itti, "Modeling the influence of task on attention," *Vision Research*, vol. 45, no. 2, pp. 205–231, 2005.

[17] D. Walther, U. Rutishauser, C. Koch, and P. Perona, "Selective visual attention enables learning and recognition of multiple objects in cluttered scenes," *Computer Vision and Image Understanding*, to appear 2005.

[18] N. Ouerhani, N. Archip, H. Hügli, and P. J. Erard, "A color image segmentation method based on seeded region growing and visual attention," *Int'l J. of Image Processing and Communication*, vol. 8, no. 1, pp. 3–11, 2002.

[19] S. Vijayakumar, J. Conradt, T. Shibata, and S. Schaal, "Overt visual attention for a humanoid robot," in *Proc. of IROS 2001*, Hawaii, 2001, pp. 2332–2337.

[20] 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.

[21] N. Ouerhani, A. Bur, and H. Hügli, "Visual attention-based robot self-localization," in *Proc. of ECMR*, 2005, pp. 8–13.

[22] S. Frintrop, "VOCUS: A visual attention system for object detection and goal-directed search," Ph.D. dissertation, Bonn, Germany, Juli 2005, published 2006 in Lecture Notes in Artificial Intelligence (LNAI), Vol. 3899, Springer Verlag.

[23] P. Viola and M. J. Jones, "Robust real-time face detection," *Int'l J. of Computer Vision (IJCV)*, vol. 57, no. 2, pp. 137–154, May 2004.

[24] S. Frintrop, P. Jensfelt, and H. Christensen, "Pay attention when selecting features," in *Proc. of the 18th Int'l Conf. on Pattern Recognition (ICPR 2006) (accepted)*, Hong Kong, Aug 2006.

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