Top-down Attention Supports Visual Loop Closing

Simone Frintrop Armin B. Cremers
Institute of Computer Science III,
Rheinische Friedrich-Wilhelms-Universität Bonn, Germany

***Abstract*— In this paper, we present a method to improve the loop closing behaviour for visual SLAM. Landmarks consist of a combination of attention regions and Harris-Laplace corners. The attention regions are detected by a visual attention system which combines image-based, bottom-up and target-related, top-down information. The ability to perform target-directed search is used to search for expected landmarks.**

We analyze the amount of correct and false matches for bottom-up and top-down matching depending on different matching thresholds. It shows that whereas bottom-up matching is useful for situations in which the scene changes only slightly like during tracking, top-down matching has advantages in loop closing situations by detecting a much higher amount of correctly matched landmarks.

***Index Terms*— Visual SLAM, loop closing, saliency, visual attention**

I. Introduction

An essential task of mobile robots which explore unknown environments is *SLAM (Simultaneous localization and mapping)*, the task of building a map and staying localized within it at the same time [3, 4, 18]. Special interest during the last years has been on *visual SLAM*, which uses cameras as main sensors [1, 10, 12, 17]. In contrast to laser-scanners, 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. Additionally, the rich visual information of camera images holds potential for better data association and more accurate 3D representations of the environment. Challenges in this field are the high amount of data which requires intelligent landmark selection strategies and the sensitivity of image data to illumination and viewpoint changes which requires robust tracking and matching methods. Additionally, when performing *bearing-only SLAM* with a single camera, depth estimation is difficult because it has to be estimated by triangulation from several frames.

One of the most challenging problems in SLAM is the *data association*, the task of associating current observations with map elements. In visual SLAM, this means to match currently detected visual landmarks to landmarks from a database. For consecutive frames, this problem is relatively easy, especially if additional odometry information is used, since usually images change only slightly between frames and since the odometry provides the system with rather accurate position estimates. The problem becomes much more difficult when the robot revisits a location after some time. This *loop closing* has to deal with illumination variations and viewpoint changes, and since the odometry estimation is much less accurate, large areas have to be considered for matching.

The choice of the feature detector is important to obtain useful landmarks which are on the one hand robust and easy to redetect and which have, on the other hand, high positional stability to obtain precise depth estimations when triangulating. Often, the landmarks are selected by a human expert or the kind of landmark is determined in advance, e.g., ceiling lights [17], artificial landmarks [2], Harris corners [12], SIFT features [13], or maximally stable extremal regions (MSERs) [15]. As pointed out by [19], there is a need for methods which enable a robot to choose landmarks autonomously. A good method should pick the landmarks which are most suitable for the current situation. An especially useful method to find landmarks autonomously depending on the current surrounding are visual attention systems [20, 11, 5]. They select regions that "pop out" in a scene due to strong contrasts and uniqueness. The advantage of these methods is that they determine globally which regions in the image are discriminative instead of locally detecting predefined properties. In previous work, we have shown that a combination of attention regions with Harris-Laplace corners is especially useful to obtain both, positional stability and good discrimination for loop closing [7, 6].

In this paper, we focus on an improvement of the loop closing module of our visual SLAM system. All approaches we are aware of match landmarks in a bottom-up manner, i.e., the same feature detection methods are applied to two frames and the detected features are compared afterwards [16, 15, 12, 8]. In contrast to this, we change the feature computations depending on the kind of landmarks we currently expect: we use the ability of the attention system to search in a top-down, target-directed manner for expected landmarks by explicitly supporting expected features. Information about which landmarks are expected is provided by the SLAM module, based on the estimate robot pose and the map.

We compare in real-world experiments the new top-down matching with the conventional bottom-up matching. It turns out that whereas the bottom-up matching shows advantages in easy matching situations like tracking, the top-down matching outperforms the bottom-up matching clearly in difficult matching situations with changing viewpoints. Therefore, the new method is more useful for loop-closing situations.

In the following, we first give an overview over the whole visual SLAM system (sec. II). Then, we describe the feature detection (sec. III), the feature matching (sec. IV), the feature tracking (sec. V), and the loop closing (sec. VI). Finally, we present several experiments on real-world data in sec. VII before we conclude (sec. VIII).

II. System Overview

The visual SLAM architecture is displayed in Fig. 1. The main components are a *robot* which provides camera images

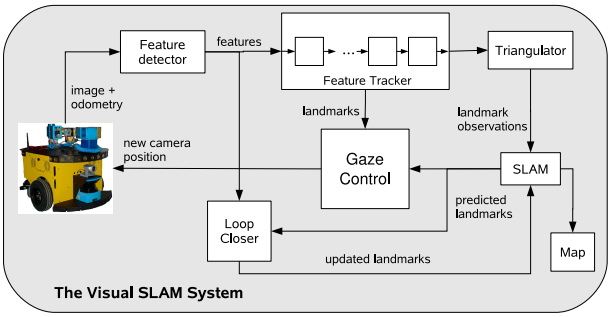


Fig. 1. The visual SLAM system

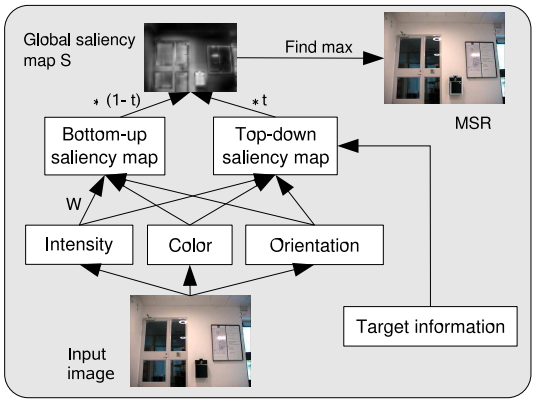


Fig. 2. ROI detection: The visual attention system VOCUS.

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 and Harris-Laplace 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 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 about the robot and the SLAM architecture can be found in [12].

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. 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* 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. Details on this module can be found in [8].

III. The Feature Detector

The feature selection is based on two different kinds of features: attentional ROIs and Harris-Laplace corners. In [7] 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. The corners on the other hand provide well localized points as required for precise depth estimation for structure from motion with a small baseline. Additionally, the combination improves the matching of landmarks (cf. sec. IV).

A. ROI Detection

The ROIs are detected with the attention system VOCUS (Visual Object detection with a CompUtational attention System) [5] (Fig. 2). It consists of a bottom-up part similar to [11], and a top-down part enabling goal-directed search; global saliency is determined from both cues.

1) Bottom-up computations: 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 $(0\,^\circ, 45\,^\circ, 90\,^\circ, 135\,^\circ)$ 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* $S_{bu} = \mathcal{W}(I) + \mathcal{W}(O) + \mathcal{W}(C)$. Details on the feature computations in [5].

To achieve real-time performance, the feature computations in VOCUS are efficiently performed on *integral images* [21]. 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 a 400×300 pixel image, 2.8GHz) that enables real-time performance (details in [9]).

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 (seeds) in S are found and second all neighboring pixels over a saliency threshold (here: 25% of the seed) are detected recursively with *region growing*. A ROI is defined as the smallest rectangle including the MSR. It is an approximation, to allow easier storing of features.

For each *MSR*, a bottom-up feature vector $\vec{v_{bu}}$ 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

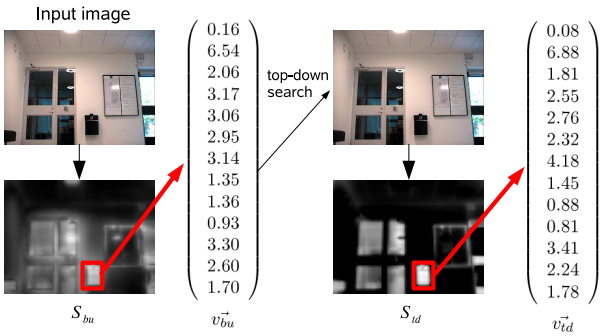


Fig. 3. Procedure to create a top-down vector $\vec{v_{td}}$: First, the bottom-up saliency Map S_{bu} is created from the input image. Then, for each MSR in S_{bu} the corresponding bottom-up vector $\vec{v_{bu}}$ is created. This vector is used to apply top-down search to the input image, yielding in a top-down saliency map S_{td}. The feature vector describing the corresponding MSR in S_{td} is the vector $\vec{v_{td}}$. The values in the vectors stand for the feature maps intensity on-off, intensity off-on, orientations $0°, 45°, 90°, 135°$, colors green, blue, red, yellow and for the conspicuity maps I, O, C.

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 the feature vector $\vec{v_{bu}}$ which corresponds to the wastebin. It tells us, e.g., that the region is dark on a bright background, since the highest value is the 2nd value of the vector, which represents the off-on intensity.

2) Top-down computations: 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 $\vec{v}$. In *search mode*, VOCUS multiplies the feature and conspicuity maps with the corresponding weights of $\vec{v}$. 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 [5]). Here, we use $t = 0$ for bottom-up and $t = 1$ for top-down computations.

Fig. 3 shows a bottom-up and a top-down saliency map: the bottom-up saliency map highlights all regions which might be of interest, regardless of a certain target. The top-down map highlights especially the target region (the black wastebin) and suppresses regions which do not look similar.

If the similarity of two ROIs shall be compared (see sec. IV), we cannot compare a top-down ROI with a bottom-up ROI because the feature values result from different computations. Instead, we additionally compute a top-down vector $\vec{v_{td}}$ for each bottom-up ROI. This is done by using the bottom-up vector $\vec{v_{bu}}$ as target information and search for this region within the same image. This results in a top-down saliency map in which the top-down MSR within the target region, defined by the bottom-up ROI, is determined. Fig. 3 shows the procedure to create such a top-down vector $\vec{v_{td}}$.

B. Harris-Laplace detector:

To detect features with high position stability inside the ROIs, we used the Harris-Laplace feature detector [14] – an extension of the Harris corner detector to Laplacian pyramids which enables scale invariance. For convenience, we talk briefly about Harris corners in the following. The method finds a few (av. 1.6) points per ROI. To allow matching of points, a SIFT descriptor is computed for each detected corner [13].

IV. Feature Matching

Feature matching is performed between consecutive frames (in the feature tracker) and with features from the database (in the loop closer). The general matching procedure is the same in both modules. It is based on two criteria: proximity and similarity. First, the features in the new frame have to be close enough to the predicted position. Second, 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 $\vec{v}$ and $\vec{w}$ is determined by eq. 1; the smaller the distance $d(\vec{v}, \vec{w})$, the higher the similarity of the ROIs. If $d(\vec{v}, \vec{w})$ is below a certain threshold δ, the ROIs match (see sec. VII for the choice of δ). The computation is similar to the Euclidean distance of the vectors, but it treats the feature map values (v_1,..,v_{10}) differently than the conspicuity map values (v_{11},..., 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 v_{13} means the values of the color feature maps (v_7,...,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 that this matching procedure outperforms the simple Euclidean distance of the feature vectors.

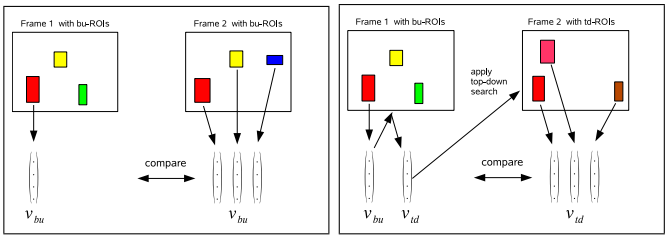
We distinguish two matching approaches: *bottom-up* and *top-down matching*. They differ in the kind of vectors which are used to determine the similarity. We describe both in the following.

A. Bottom-up matching

For bottom-up matching, each ROI from a frame f_1 is compared to each ROI from a frame f_2. If the matching distance $d(\vec{v}, \vec{w})$ for the vectors $\vec{v}$ and $\vec{w}$ of two ROIs is below the matching threshold δ, the ROIs are considered as a match. If several ROIs from f_2 match to the same ROI from f_1, the best match with the smallest distance is chosen. The bottom-up matching procedure is illustrated in Fig. 4, left.

The bottom-up matching works especially well, if the two frames differ only slightly. This is the case for tracking. For loop-closing, the bottom-up matching works well if the viewpoint of the landmark differs only slightly to the viewpoint it had when seeing the landmark for the first time. For more different viewpoints, the top-down matching is preferable.

$$d(\vec{v},\vec{w}) = \sqrt{\frac{v_{11}w_{11}\sum_{i=1,2}(v_i-w_i)^2 + v_{12}w_{12}\sum_{i=3,..,6}(v_i-w_i)^2 + v_{13}w_{13}\sum_{i=7,..,10}(v_i-w_i)^2}{v_{11}w_{11}+v_{12}w_{12}+v_{13}w_{13}}} \quad (1)$$



(a) bottom-up matching (b) top-down matching

Fig. 4. Left: bottom-up matching. To find a match for a ROI from frame 1, it is compared to each ROI from frame 2. Right: top-down matching. To find a match for a ROI from frame 1, its top-down feature vector v_{td} is used as target information to search for this ROI in frame 2. The resulting ROIs all look similar to the ROI from frame 1.

B. Top-down matching

For top-down matching, we determine for each ROI a top-down feature vector $\vec{v_{td}}$, as described in sec. III-A.2. These vectors are later used for comparison.

To find a match for ROI r_1 from frame f_1 in frame f_2, the vector $\vec{v_{td}}$ which describes r_1 is used to apply top-down search to f_2. From the resulting top-down saliency map, the most salient ROIs are extracted and their top-down feature vectors are compared to $\vec{v_{td}}$. As for the bottom-up matching, the ROIs are considered as a match if the matching distance d is below the matching threshold δ, and if several ROIs from f_2 match to r_1, the best match with the smallest d is chosen. The top-down matching procedure is illustrated in Fig. 4, right. The colors and the shape of the ROIs illustrate their similarity. Top-down matching compares a ROI only to similar regions, whereas the bottom-up matching compares it with all salient regions.

Top-down matching pays off especially if the appearance of two frames differs strongly. Since this is usually the case in loop closing situations, we apply the top-down matching for loop-closing. To search for an expected ROI does not mean that all computations of VOCUS have to be repeated for each expected ROI. The most time consuming computations, the computations of the feature maps, do not have to be done again. They are the same for the bottom-up computations and for each expected ROI. Therefore, these computations are still possible in real-time.

V. The Feature Tracker

In the feature tracker, the frames are stored in a buffer with length n (here: $n = 30$) and features are tracked over several frames. 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*.

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. We apply bottom-up matching here. 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. The same procedure is used to create the Harris-landmarks.

At the end of the buffer, the landmarks are transferred to the triangulator, which first checks whether the landmarks are long enough (≥ 5). Then, the Harris corners inside of ROIs are determined, and it is checked whether the corresponding Harris-landmarks are long enough and stable enough. Finally, the Harris-landmarks which survive the process are reported to the SLAM module.

VI. The Loop Closer

The loop closer obtains landmark predictions from the SLAM module and checks if these landmarks are visible in the current frame. In bottom-up matching mode, it compares each ROI from the expected landmarks to each ROI of the current frame. In top-down mode, it takes each ROI from each expected landmark, uses it as target information, and searches for it with top-down attention within the current frame. Then, the resulting top-down ROIs are compared to the ROIs from the expected landmarks with top-down matching. If there are several matches in the current frame, the best match is taken.

If there is a ROI-match, all of the Harris corners within the matching ROIs are compared based on their SIFT descriptor. If there is also a match, the corresponding landmark is reported to the SLAM module, to update the map. The combination of ROI and Harris matching enables a reliable matching with almost no false positives.

VII. Experiments and Results

In this section, we illustrate the differences between bottom-up and top-down matching and the advantages of the top-down matching for loop closing. We investigated the system behaviour twice for the same data obtained from the trajectory displayed in Fig. 5. The robot drove through a room, left the room, drove through the corridor, and entered the room again through a different door. After entering the room, it faced the same region as in the beginning. At this point, it should be

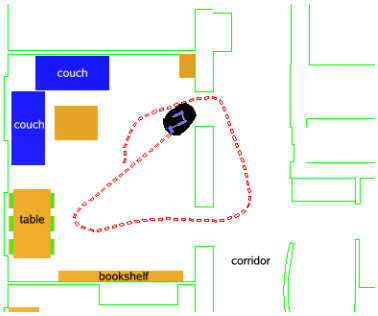


Fig. 5. The robot environment and the driven trajectory.

able to detect that it closed a loop. Although the loop is very small compared to some other SLAM-scenarios, it is sufficient here to show that top-down matching outperforms bottom-up matching in loop-closing situations. The effect of larger loops would be a higher uncertainty of robot and landmark positions, resulting in larger search areas in the images, in the worst case the whole image. In these cases, the advantage of top-down matching is expected to be even more important.

The visual SLAM system runs online in real-time, but for our experiments we needed offline data to enable experiments on the same data for both matching methods. Therefore, we stored the image sequence, consisting of 283 images, as well as the odometry information. We ran the system twice on this sequence, once the loop closing was implemented with the bottom-up matching and once with the top-down matching. Note, that in offline mode the gaze control module cannot be used. But since gaze control and top-down matching are two largely independent mechanisms (gaze control controls the camera actively whereas top-down matching focuses the processing actively to regions of interest within the current image), this does not affect the current experiments.

Each ROI of each expected landmark was considered for matching. Fig. 6 shows the matching results for different thresholds δ. It shows, that the increase of false matches (red, dashed line) for increasing thresholds is about the same for bottom-up and top-down matching, whereas the increase of correct matches (blue, solid line) is steeper for the top-down matching. That means, more correct matches are obtained in top-down mode.

To illustrate the correspondence between false and correct matches in more detail, Fig. 7 displays the correct matches depending on the number of false matches. This figure is similar to a ROC (receiver operating characteristic) curve, but note that here the axes denote numbers of matches instead of ratios. This is sufficient here, because in contrast to recognition tasks, where the ratio of correct matches is important, we are not interested in detecting all possible matches; some matches are sufficient to close the loop. However, a higher detection rate is still preferable, because it speeds up the loop closing process and makes it more stable.

We expected the top-down matching to outperform the bottom-up matching. Interestingly, this was not always the case. For low thresholds which accept only very few or no false detections at all, the bottom-up matching showed to be better and provided more correct matches. The turning point is between 8 and 15 false matches, where both bottom-up and top-down matching perform equally. For higher thresholds which accept more false matches, the top-down matching outperformed the bottom-up matching, resulting in a considerably higher number of correct matches: for 50 false matches, the bottom-up matching detected 183 correct matches whereas the top-down matching achieved 261 correct matches, which is an increase of 42%.

Note that this number of false ROI matches is not the number of false landmark matches which is reported to SLAM. First, several of the matched ROIs belong to the same landmark, since each ROI from an expected landmark is matched to current ROIs. For example, the 50 false matches of the top-down matching belonged to only 5 different ROI-landmarks with 10 matches on average and the 261 correct matches belonged to 9 ROI-landmarks with 29 matches on average. Since there are usually considerably more matches from correctly matching landmarks then from not-matching landmarks, the number of matching ROIs per landmark is an additional hint whether a landmark is redetected. We plan to consider this for future work. Second, since we additionally use the sift matching of the Harris corners, we are able to get rid of almost all of the remaining false ROI-landmark matches. In this example, only one Harris-landmark was classified wrongly with the top-down matching. Interestingly, this false match does not result from a false ROI match but from a wrong association of a Harris corner in the top-right corner of a ROI to one in the bottom-right corner.

To investigate the difference between the cases in which the bottom-up matching performed better and the ones in which top-down matching performed better, we had a closer look at the matches. It turned out that "easy" matches are better redetected with bottom-up matching. Easy matches are those in which ROIs are seen under almost the same conditions (i.e. from almost the same viewpoint and under almost the same lightning conditions) as when they were detected the first time. One example of such an easy match is displayed in Fig. 8, left. More difficult matches are better redetected with top-down matching. In some of these examples, the ROI is seen from a quite different viewpoint as the example in Fig. 8, right. These examples are of course more interesting, since usually the robot does not face a landmark from exactly the same position as before, so a viewpoint tolerance is necessary.

Since the matches in tracking situations are usually easy matches, we suggest to use the bottom-up matching in the feature tracker and top-down matching in the loop closer.

VIII. Conclusions

In this paper, we have presented a method to improve the loop closing behaviour for visual SLAM. The visual attention system, which detects regions of interest in a frame, is tuned in a top-down manner to search for expected landmarks. Whereas in easy matching situations the bottom-up matching is preferable, the top-down matching outperforms the bottom-up approach clearly in difficult matching situations: especially when the viewpoint changes, the top-down matching enables a

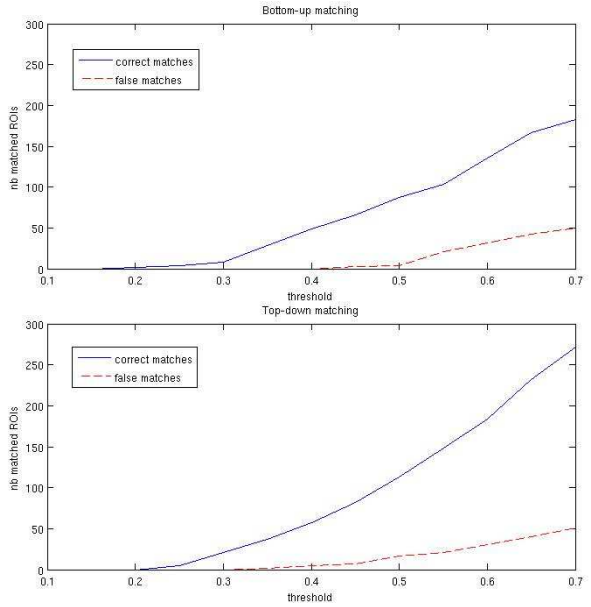


Fig. 6. Correct and false ROI matches for bottom-up (top) and top-down matching (bottom) depending on the matching threshold δ.

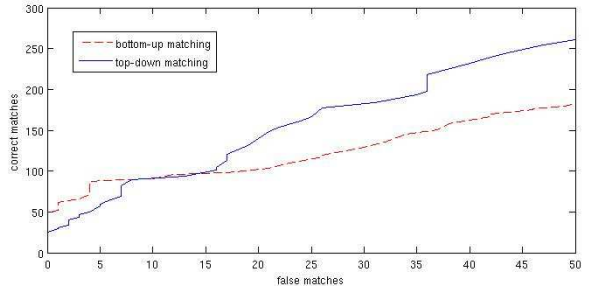


Fig. 7. Correct matches for bottom-up and top-down matching depending on the error rate: For a low number of false detections, bottom-up matching results in more correct matches. If more false matches are acceptable, top-down matching provides more correct matches.

more stable redetection with a considerably higher amount of correct matches. Remaining false detections are removed with an additional SIFT matching of Harris corners. This makes the method useful for loop closing situations. In future work, we plan to make the matching even more robust by considering the matching stability of features over time and the constellation of landmarks to each other within frames. Another topic of research will be to investigate the limits of the method, i.e., to check how strongly the viewpoint may differ to still enable redetection.

Fig. 8. The rectangles denote correctly matched ROIs. Top: current frame. Bottom: frame with expected ROI from database. Left: "easy" matching situation. Right: "difficult" matching situation from different viewpoint.

REFERENCES

[1] Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proc. of the ICCV*, oct 2003.

[2] Andrew J. Davison and David W. Murray. Simultaneous localisation and map-building using active vision. *IEEE Trans. PAMI*, 2002.

[3] 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.*, 17(3):229–241, 2001.

[4] U. Frese, P. Larsson, and T. Duckett. A multigrid algorithm for simultaneous localization and mapping. *IEEE Trans. Robot.*, 21(2):1–12, 2005.

[5] Simone Frintrop. *VOCUS: A Visual Attention System for Object Detection and Goal-directed Search*. PhD thesis, 2005. Published 2006 in LNAI, Vol. 3899, Springer.

[6] Simone Frintrop, Patric Jensfelt, and Henrik Christensen. Attentional Landmark Selection for Visual SLAM. In *Proc. Int'l Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[7] Simone Frintrop, Patric Jensfelt, and Henrik Christensen. Pay attention when selecting features. In *Proc. Int'l Conf. on Pattern Recognition (ICPR 2006)*, 2006.

[8] Simone Frintrop, Patric Jensfelt, and Henrik Christensen. Attentional robot localization and mapping. In *ICVS Workshop on Computational Attention & Applications (WCAA)*, 2007.

[9] Simone Frintrop, Maria Klodt, and Erich Rome. A real-time visual attention system using integral images. In *Proc. of Int'l Conf. on Computer Vision Systems (ICVS)*, 2007.

[10] 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.

[11] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11), 1998.

[12] Patric Jensfelt, Danica Kragic, John Folkesson, and Mårten Björkman. A framework for vision based bearing only 3D SLAM. In *Proc. of ICRA'06*, Orlando, FL, May 2006.

[13] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of ICCV*, pages 1150–57, 1999.

[14] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. of ICCV*, pages 525–531, 2001.

[15] Paul Newman and Kin Ho. SLAM-loop closing with visually salient features. In *Proc. Int'l Conf. on Robotics and Automation, (ICRA 2005)*, 2005.

[16] Nabil Ouerhani, Alexandre Bur, and Heinz Hügli. Visual attention-based robot self-localization. In *Proc. of European Conf. on Mobile Robotics (ECMR 2005)*, 2005.

[17] 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*, 19(11), 2000.

[18] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Y. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Int. J. Robot. Res.*, 23(7-8):693–716, 2004.

[19] Sebastian Thrun. Finding landmarks for mobile robot navigation. In *Proc. of ICRA*, 1998.

[20] John K. Tsotsos, Sean M. Culhane, Winky Yan Kei Wai, Yuzhong Lai, Neal Davis, and Fernando Nuflo. Modeling visual attention via selective tuning. *AI*, 78(1-2), 1995.

[21] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int'l Journal of Computer Vision (IJCV)*, 57(2):137–154, May 2004.