The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Scene Change Detection for Vision-based Topological Mapping and Localization

Navid Nourani-Vatani[1,2] and Cedric Pradalier[3]

[1]UQ QBI
QBI Building 79
St Lucia QLD, 4072 Australia

[2]CSIRO ICT Centre
1 Technology Crt
Pullenvale QLD, 4069 Australia

[3]ETH Zurich
CLA Building
Zurich, 8000, Switzerland

N.Nourani@uq.edu.au and Cedric.Pradalier@mavt.ethz.ch

Abstract— **A method for detecting changes in the environment using only vision sensors is presented. We demonstrate that optical flow can be used to detect these changes at key locations in outdoor scenarios in difficult and varying lighting conditions. These key locations are used as nodes in a topological mapping and localization framework. To close the loop we employ a bag-of-words methodology. We show that bag-of-words methods can be used in real-time on a standard computer to detect loop closures in sparse topological maps. Experimental results from field trials using our quad-rotor UAV demonstrate the capability of the proposed scene change detection method.**

I. Introduction

For a mobile robot to perform fully autonomous tasks it is generally essential that it can determine its location in the environment before it can navigate to carry out an assigned task. A truly autonomous robot should be capable of creating its own representation of the environment it is working within by creating maps and using these maps to localize itself. Mapping and localization are very complex tasks, which are further complicated when dealing with dynamic outdoor areas and using vision as the perception sensor. The main field of map-based localization is called Simultaneous Localization and Mapping (SLAM). Using SLAM we expect a system to be able to create a map of relevant features, estimate the instantaneous camera/robot motion, and correct for any motion estimation errors by closing the loop in the map. It is hence obvious that if only new areas are being explored the performance of a SLAM system will be no better than a standard visual odometry system [1]. This is because the main strength of a SLAM systems is its ability to correct for motion error using the built map. This also indicates that if the built map is corrupted, e.g. by mapping dynamic features, it can affect the localization adversely.

Loop closure is made difficult in metric SLAM due to errors in position estimation, which become progressively less accurate as the loop gets larger [2]. To overcome this, a new genre of visual SLAM, called Appearance-based Mapping and Localization, is being established. The concept here is to determine loop closure based on similarity in sensory data input - in the case of visual SLAM similarity in image frames - and ignoring the robot location.

The state-of-the-art in vision-based mapping and localization is currently to capture images at fixed intervals, use one or more types of image descriptors to describe each captured image, and create a database of these descriptors during the mapping phase. During localization, the database is searched for a match to close the loop or possibly to extend the map [3], [4]. The advantage of such *appearance-based* localization and mapping over *metric* SLAM is that the system can produce correct loop closures even when the robot is lost or has a very poor pose estimate. The disadvantage of this approach is that for every sensory input step the entire map has to be searched for matches. This gets less and less feasible in real-time as the map gets bigger. The system has to eliminate several false positives and false negatives, also in real-time. Furthermore, appearance maps cannot be used for control and planning.

Fig. 1. Our robotic platform is a quad-rotor UAV.

Topological localization and mapping is one level of abstraction above metric localization and mapping because it takes into account the connectivities, or relationships, between the various locations that are encountered in the process of building the map [5]. The abstraction comes from raising the attention from low-level local localization to the higher level connections between the locations. A topological map can be visualized as a map of nodes and edges, as shown in Fig. 2. The nodes of a topological map are key points or locations that are unique and recognizable when they are re-encountered. Each edge in a topological map describes how two nodes are connected.

In this paper, we propose a method for building topological maps using vision. The main contribution is automatic

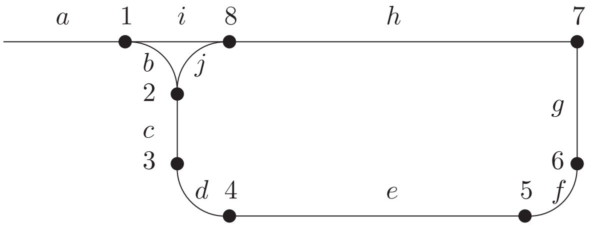


Fig. 2. An example of a topological map. The nodes (1-8) represent key locations that are connected by the edges ($a - i$)

detection of key locations in the environment using optical flow information; where these key locations serve as nodes in the topological map. To re-localize the vehicle, we use a visual appearance strategy.

The rest of the paper is structured as follows. Next we look at the related research and compare our method to these. Section III describes how the scene change detection is performed using optical flow information in the image and how loop closures are detected using bag-of-words. Results from field trials using our quad-rotor UAV is presented in Section IV. In Section V we conclude and suggest future directions for the research.

II. Related Work

One example of key point detection was presented by Kortenkamp et al. [6], who introduced gateways as transition places between spaces. A gateway is different from a distinguishable feature or place in that all gateways are similar; an opening leading from one space to another. Using sonar sensors the authors detected doorways in hallways by monitoring the change in returned distance to wall. Hence, a gateway is a temporal signature of the opening, ranging from the detection of the opening until the detection of the doorway end.

Duff et al. [7] use a laser range finder in combination with a wheel odometer to navigate a Load-Haul Dump truck through underground mines. Incorporating a reactive wall following control and intersection detection using the laser range finder, they are capable of navigating through underground mazes. They take advantage of recognized landmarks to compensate for less precise odometry and localization. In underground mines, as well as when we drive on street roads, we can normally only drive forward or reverse. The only time we need to turn is when the opportunity arises in the form of an intersection - or a large open area. The authors therefore call this method of localization *opportunistic localization*.

Radhakrishnan et al. [8] proposed a vision-based transition detector for topological localization. They provided a topological map to their system and trained the system on transition points in the environment. Using a Naive Bayes Classifier in a two-tier approach their localizer was capable of coming up with transition thresholds automatically during training. These transition thresholds are used to determine when a transition from one space to another has occurred. The thresholds are local and directional. The resulting system produced almost 100% correct transition recognition. The limitation of their approach is that they have to provide a topological map and train the system on the transition points. Finally, their system has only been tested in structured indoor environments with controlled lighting and they point out that moving to large outdoor environments with uncontrolled lighting would be non-trivial.

Werner et al [9] propose a topological SLAM algorithm based on Bayesian Inference. In their approach they propose to use an omni-directional camera and extracted image histograms to perform the task in indoor self-similar environments. Their key point detection is not performed using vision. Their robot traverses the hallways using ultrasonic sensors and a Voronoi graph of the environment. The Voronoi nodes—a point equidistant to three (or more) obstacles—fall at T-junctions and intersections in the hallways. These are the key points in their topological map, and an image is taken when the robot reaches such a node.

III. Scene Change Detection

As described previously, we are interested in developing a sparse topological map of key interest points in the environment. The main requirement to do so is firstly to be able to detect key interest points and secondly to be able to describe these locations for future reference and relocalization. The key point detection using optical flow is described next. Loop closures detected using a bag-of-words methodology is described subsequently.

A. Key Point Detection

When driving or flying through structured or semi-structured environments, such as urban environments, significant scene changes occur when buildings and other structures appear or disappear next to the vehicle. In such environments, it is the detection of opening and closing of spaces that is considered significant. Considering openings and closings of space as significant is also often suggested in the literature, [6]-[10], whether in indoor or outdoor scenarios. As Duff et al. put it nicely, it is the *opportunity* that arrises at an opening, which allows for navigating to a different location, that is considered significant. Inversely, when the opening is no longer present it is the realisation that we are traversing a different path (or edge in the topological map). When using vision, these changes have to occur in the field of view of the camera and correspond to the changes observed in the induced optical flow signal.

We are using a similar approach as Kortenkamp et al. for detection of key locations. Instead of ultrasonic sensors, we use cameras to perceive the environment and can obtain information about it by observing the apparent motion induced in the camera and calculating the optical flow. This is because the magnitude of the induced optical flow in the image is dependent on two factors, namely the velocity at which the camera (or object, or both) is moving and the physical distance to the object generating the optical flow. In order to determine the distance to a static object, the component of the optical flow induced by the velocity of the vehicle must be subtracted. If the vehicle speed is known

TABLE I

OPTICAL FLOW FILTERING PARAMETERS AND SCENE CHANGE DETECTION PARAMETERS.

Hysteresis Accept	Hysteresis Reject	Smoothing Window	Min Change Threshold	Min Time Threshold
3	2	30	0.02	2.0

it can be used to normalize the magnitude of the optical flow vectors. If the velocity is unknown, but can be assumed constant, observed change in the optical flow will still reflect the change in distance to the observed object. Furthermore, on a dynamic platform such as an UAV with 6 degrees of freedom, it might be necessary to also remove any induced optical flow from roll, pitch, or yaw motion, if these are significant.

Subsequently, by monitoring the change in the induced optical flow in the image, change in the environment can be detected. Key locations in the environment are defined as places with significant change in the optical flow signal. Key locations are nodes in the topological map.

In our approach, we use a Canny edge detector in combination with a pyramidical Lucas-Kanade algorithm to estimate the optical flow. The optical flow signal is noisy, and in order to extract the necessary information about the scene change from the induced optical flow it must be filtered. For this purpose, the flow vectors are first transformed to polar co-ordinates:

$$\theta = atan(\Delta y/\Delta x) \tag{1}$$

$$r = x_1 \cdot cos(\theta) + y_1 \cdot sin(\theta) \tag{2}$$

where x_1 and y_1 are the vector start position, and the length of the vector is $\sqrt{\Delta x^2 + \Delta y^2}$.

To achieve a smooth signal a two step approach is used. Firstly, a hysteresis voting scheme is applied in the polar domain. This ensures that only flow vectors present over several consecutive frames are accepted. A flow vector is discarded again, once it hasn't been observed in a few frames and its vote falls under the minimum hysteresis threshold.

The scene flow response is calculated by deriving the mean vector lengths in the polar coordinate frame:

$$\mu(r) = \frac{1}{n}\Sigma_{i=1}^{n} r \tag{3}$$

where n is the number of vectors.

Next, a fixed width window smoother is applied in the time domain to remove high frequency information; corresponding to small changes. Changes in the environment are detected by finding peaks and valleys in the filtered response signal. These changes correspond to key points in the environment and nodes in the topological map.

It might be desirable to ignore nodes corresponding to small spikes by accepting only changes larger than a threshold, or to have a minimum distance between two nodes. The strength of the system is that it will repeatedly trigger on similar changes, as shown below. For the data presented in this paper we use the parameters in Table I.

The process of filtering and key point detection for the L-shaped path in Fig. 4 is shown in Fig. 3. The figure shows the raw optical flow signal (3(a)), the smoothed signal (3(b)), and the detection of key points (3(c)). We have no visual odometer but assuming constant distance to the buildings we can use the induced horizontal optical flow signal (red dashed line in Fig. 3(b)) as a *scaleless* estimate of the vehicle forward velocity. For visualization purposes only, this velocity is used in conjunction with the yaw angle from an onboard Inertial Measurement Unit (IMU) to build a map. Fig. 3(d) shows an approximate scaleless 2-dimensional (2D) path of the traversal, the flow magnitude, and the node locations.

B. Loop Closure

When the creation of a key location is triggered, as explained above, it is time to check if this is a known location we have returned to and then add the image to a database. To speed up the image matching process we use image descriptors to describe the image and then use a bag-of-words approach to find matches and to store the image.

As image descriptor, we use SIFT features [11] extracted from the images that trigger key point detection.

These SIFT descriptors are fed to a bag-of-words algorithm [12] to build up a database. When querying for a match, we have through empirical verification found out that, the correct match, if any, is in 95% of the times in the top $n = 3$ matches. To ensure that there indeed is a correct match we perform SIFT matching between the query image and the database image. A loop closure has been detected if at least 50% of the SIFT features match. This process is faster but not as robust as performing geometric verification

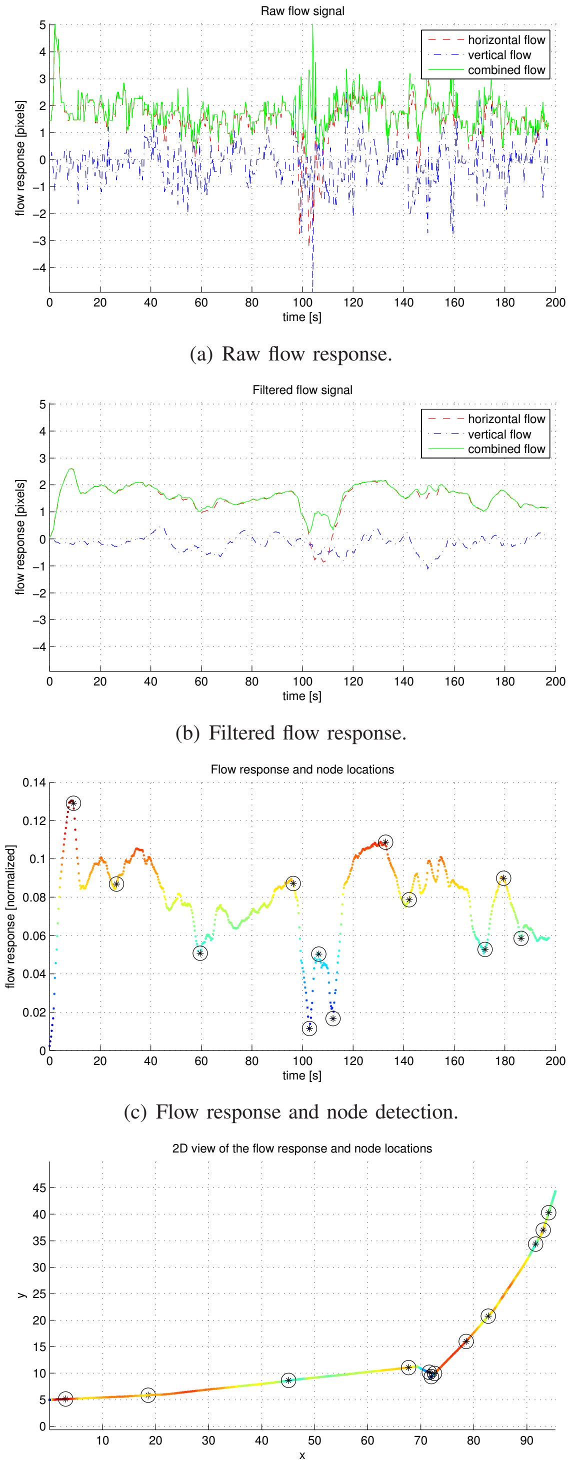
$$\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0 \tag{4}$$

where, $\mathbf{x}_1$ and $\mathbf{x}_2$ are the locations of the SIFT descriptors in the two images returned by the matching function and $\mathbf{F}$ is the principal matrix of the camera. Only if the two images are from the same locations, and the SIFT features represent the same points in the scene, will a solution exist for this equation.

Once a match has been detected, a choice must be made whether the newly matched node should be added to the map or not. Adding the node to the database will signify that there will exist two nodes representing the same location and hence more than one correct match in database. Not adding the new node will keep the map more concise and smaller, but small changes due to lighting and dynamic changes can be lost too. In our implementation we have chosen the former method and add all detected nodes to the database. This is justified because the created topological map is very sparse. During loop closure, however, we stop looking for more matches once the first match has been found.

IV. RESULTS

Our robotic platform is a quad-rotor Unmanned Aerial Vehicle (UAV) from Ascending Technologies GmbH (Fig. 1). The UAV has onboard an IMU, GPS, camera, and an Atom



(a) Raw flow response.

(b) Filtered flow response.

(c) Flow response and node detection.

(d) 2D view of the flow response and the node locations.

Fig. 3. Scene change detection from the optical flow response signal.

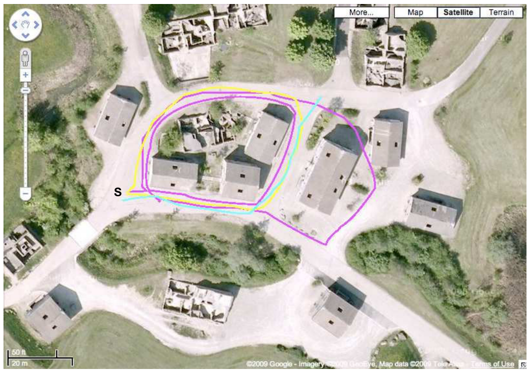

More...
Map
Satellite
Terrain
S
50 ft
20 m
©2009 Google - Imagery ©2009 DigitalGlobe, Map data ©2009 Tele Atlas - Terms of Use

Fig. 4. Overview of the field test area. The approximate paths of the three experiments are shown and the start position is marked with an 'S'. Experiment 1 (magenta) was a double loop of $\sim 375\,\mathrm{m}$, experiment 2 (yellow) was a single loop of $\sim 155\,\mathrm{m}$, and experiment 3 (cyan) was an L-shaped path of $\sim 85\,\mathrm{m}$.

1.6 GHz processor running Ubuntu 9.10. The Point Grey Firefly camera is mounted on top of the UAV and produces grey scale images in 752×480 pixels at 30 fps. For the key point detection, however, only every fifth frame at a resolution of 376×240 is used while the IMU and GPS units are not used at all. Images are captured and logged onboard the UAV but post-processed in real-time on a Macbook Santa Maria 2GHz Dual2 2GB notebook running Ubuntu 9.04.

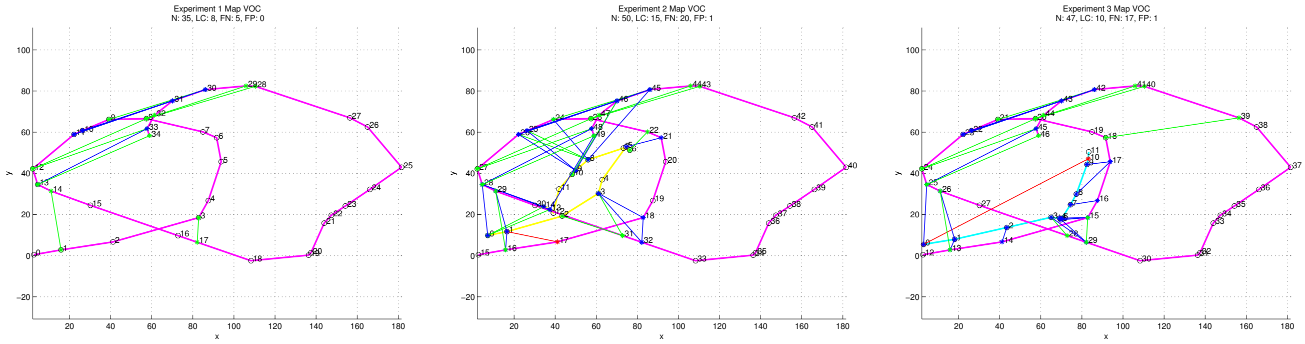
The field trials took place at Birmensdorf Barracks village, Switzerland. This is an artificial village with one and two level buildings and also some demolished buildings. Three experiments were carried out, see Fig. 4. In the first experiment we flew a double loop and captured 8720 frames (magenta path). The second experiment contains the inner loop and 6013 frames (yellow path) while in the third experiment we flew an L-shape through the first passage, capturing 5622 frames (cyan path). All flights were human remote controlled, and since no other sensor than the camera is used we assume the flight speed to be constant.

As shown by Hrabar and Sukhatme [13], the camera should be facing 90° to the direction of motion to measure the maximum optical flow to increase the signal-to-noise ratio. In the same study the authors suggest, though, the optimum angle of the camera to be 45° for control purposes. Therefore, in experiments 1 and 3 we used a 90° mount angle, while in experiment 2 we used a 45° mount angle for the camera.

Experiment 1 is our baseline experiment. By flying a double loop, key points should be detected at similar locations where there is overlap. In experiments 2 and 3, key points from these flights are matched to key points detected during experiment 1. There are several hours between the tests flights and the lighting condition changed significantly.

The results for these three experiments are visualized in Figs. 5(a)-5(c). Notice that these topological maps are for visualization only. They are scaleless - in fact they have been scaled differently for better visualization. The performance of the key point detection and the loop closure procedures are measured separately and can be seen in Tables II-III. In these results, the actual number and location of loop closures (LCs), false positives (FPs), and false negatives (FNs) are manually tagged. Please also refer to the attached video[1] for a demonstration of how the key point detection and loop

[1] http://nourani.dk/?pId=ROBOTICS&subId=TUBES



(a) Experiment 1

(b) Experiment 2

(c) Experiment 3

Fig. 5. Detected loop closures are shown in green, false negatives in blue, and false positive loop closures in red. Notice that x and y-axis are not labelled as the maps are scaleless and the paths have been scaled differently for visual clarity.

TABLE II

NUMBER OF IMAGE FRAMES, KEY POINTS DETECTED, FALSE POSITIVES (FP) AND FALSE NEGATIVES (FN).

Experiment	Num Frames	Num Nodes	Num FP	Num FN
1	8720	35	3	1
2	14733 (8720 + 6013)	50	0	2
3	14342 (8720 + 5622)	47	3	1

TABLE III

NUMBER OF NODES IN THE TOPOLOGICAL MAP, POSSIBLE LOOP CLOSURES (LC), ACTUAL LCS AND FALSE LOOP CLOSURES (FP).

Experiment	Num Nodes	Possible LC	Num LC	Num FP
1	35	13	8	0
2	50	35	15	1
3	47	27	10	1

closure procedures performed during experiment 1.

It is rather difficult to measure and justify what is the correct number of key points and where they should be. Similarly, it is difficult to determine what is a false negative—key point detection missed—and what is a false positive—unnecessary key point triggering. Manual tagging of the video sequence from experiment 1 resulted in 31 key points—compared to the automatically detected 35 nodes—and did not result in better loop closure results. The most important features of the key point detection must be its ability to repeatedly trigger at same key locations while keeping the map sparse with very few FPs. From Table II we can see that the number of missed key points, FNs, is between 1-2 for each of the experiments, while the number of FPs is between 0-3. Comparing to the number of image frames, $8720 - 14733$, these are extremely low numbers showing the discriminatory behaviour, robustness, and repeatability of the method. Fig. 6 shows an example of FP key point detection. During this 90° turn around the building the operator is changing flight height. These changes in vertical velocity result in spikes in the flow signal which are detected as change in the environment. We have no threshold for minimum distance between two nodes and are currently ignoring FP key point detection due to height changes as they are due to manual flight operation and will not be experienced during automated flights or if vehicle velocity information was available.

TABLE IV

AVERAGE PROCESSING TIMES IN *milliseconds* FOR THE LOOP CLOSURE PROCEDURE.

	SIFT Descriptor	DB Query	DB Update	Total
Time [ms]	1654	350	15	2034
Percent [%]	81.92	17.34	0.74	100.0

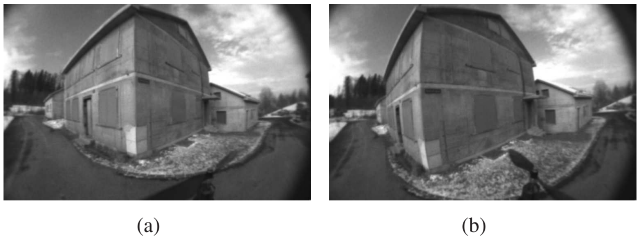
The loop closure has three parts to it: SIFT descriptor generation, DB query, and DB update[2]. Table IV shows the average calculation times for each of these processes. Using a more efficient implementation on a GPU can bring down the SIFT descriptor processing time to the neighbourhoods of $200\,\mathrm{ms}$. This, however, is not feasible on an embedded system like the one used on the UAV. From this table it can be observed that the vast majority of the processing time is spent generating SIFT descriptors. The overall loop closure procedure is around $2\,\mathrm{s}$, however, it is only called after a key point detection event. The mean and standard deviation time between key point detections in the three experiments are $\mu = 11.03\,\mathrm{s}$, $\sigma = 7.05\,\mathrm{s}$ and the mean optical flow calculation time including filtering and voting is $20\,\mathrm{ms}$, which indicates that the loop closure can be performed in real-time, even on an embedded system.

An example of a FP loop closure is shown in Fig. 7. There is approximately $10\,\mathrm{m}$ between the location where these nodes were generated in experiments 1 and 3, respectively. There is enough overlap between the two images and the SIFT descriptors such that the algorithm deems this a loop closure.

V. CONCLUSION

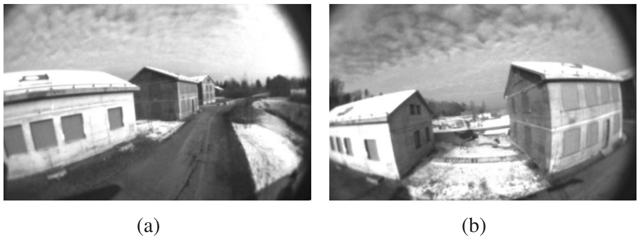
We have demonstrated the robustness of optical flow as a measure to detect significant scene changes in outdoor

[2] Friedrich Fraundorfer's bag-of-words algorithm expects the SIFT features to be extracted using Andreas Vevaldi's SIFT extraction code. It was therefore not possible to try other SIFT implementations or other feature descriptors, such as SURF.



(a) (b)

Fig. 6. Example of a false positive key point detection during experiment 3 (Nodes 5 and 6). The second key point (b) was detected due to a sudden change in height.



(a) (b)

Fig. 7. The similarity between the buildings triggers some false positive loop closures. Here an example of false positive loop closure between nodes 1 and 17 in experiment 2.

environments in varying and difficult lighting conditions. We have also shown that optical flow can be used to detect large changes robustly even when used onboard a highly dynamic platform, such as a UAV.

The scene change detections were used to define key locations as nodes in a topological framework. To detect loop closure we applied a bag-of-words approach.

The system was used to perform topological mapping and localization on data sets gathered from field trials using our quad-rotor UAV. A total of four key locations where missed during the three flights where 132 nodes were detected, while only generating six false positive nodes in the maps. These results show that the scene change detection is capable of detecting key points repeatedly at the same locations in the environment over several fly-throughs with varying flight and lighting conditions.

We also experimented mounting the camera at both 45° and 90° to the direction of motion. The signal magnitude and hence also the signal-to-noise ratio is lower at 45° (experiment 2) . Still, as the results show, Fig. 5(b), the scene change detection is not compromised by the change of camera angle. In terms of control for navigation, it is hence desirable to use 45° camera angles for scene change detection also.

During loop closure, to avoid false positive matches, the matching threshold of the SIFT matching is set at a high 50%. This results in a number of key locations not being matched. One possibility for improving the loop closure is to use geometric verification as mentioned earlier.

The UAV is currently equipped with a single camera pointing to the left. Addition of a second camera pointing in the opposite direction, will allow to perform cross-check between the matches found for each camera. The loop closure performance can be further improved by taking advantage of the topology; rejecting matches that are too far away in terms of topology, e.g. nodes 10 and 0 in experiment 3.

Although the key point detection and loop closure times allow for real-time processing the results are from offline processing. Future work also includes implementation of the vocabulary search for onboard loop closure. This will allow us to feedback the mapping and localization results for navigational purposes.

ACKNOWLEDGMENTS

Navid Nourani-Vatani is funded jointly by the Australian government through the Endeavour IPRS scheme, the University of Queensland, the Queensland Brain Institute, the University of Queensland Graduate School Research Travel Grant, and the CSIRO ICT Centre. Cedric Pradalier is partially funded by the EU project Robots@home (IST-6-045350). The authors would like to thank their colleagues for their help and support. In particular Stephan Weiss and Markus Achtelik from ASL ETH for gathering the data set and Friedrich Fraundorfer for helping with the bag-of-words algorithm.

REFERENCES

[1] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *Journal of Field Robotics*, vol. 23, no. 1, pp. 3–20, 2005.

[2] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[3] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[4] A. Angeli, D. Filliat, S. Doncieux, and J. Meyer, "A Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words," *Conditionally accpeted for publication in IEEE Transactions On Robotics, Special Issue on Visual SLAM*, 2008.

[5] B. Kuipers, "Modeling spatial knowledge," *Cognitive Science*, vol. 2, no. 2, pp. 129–153, 1978.

[6] D. Kortenkamp, L. Baker, and T. Weymouth, "Using gateways to build a route map," *Intelligent Robots and Systems, 1992., Proceedings of the 1992 IEEE/RSJ International Conference on*, vol. 3, pp. 2209–2214, Jul 1992.

[7] E. Duff, J. Roberts, and P. Corke, "Automation of an underground mining vehicle using reactive navigation and opportunistic localization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 4, 2003, pp. 3775–3780.

[8] D. Radhakrishnan and I. Nourbakhsh, "Topological robot localization by training a vision-based transition detector," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, 1999, pp. 468–473.

[9] F. Werner, F. Maire, and J. Sitte, "Topological slam using fast vision techniques," in *Proceedings of the FIRA RoboWorld Congress 2009 on Advances in Robotics*. Springer, 2009, p. 196.

[10] A. Ranganathan and F. Dellaert, "Bayesian surprise and landmark detection," in *Proceedings of the 2009 IEEE international conference on Robotics and Automation*. Institute of Electrical and Electronics Engineers Inc., The, 2009, pp. 1240–1246.

[11] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[12] F. Fraundorfer, C. Wu, J. Frahm, and M. Pollefeys, "Visual word based location recognition in 3d models using distance augmented weighting," in *Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.

[13] S. Hrabar and G. Sukhatme, "Optimum camera angle for optic flow-based centering response," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3922–3927.