Published at the Workshop on State Estimation and Terrain Perception for All Terrain Mobile Robots held in conjunction with the International Conference on Intelligent Robots and Systems (IROS), 2016.

Map Enhancement with Track-Loss Data in Visual SLAM

Stefan Schubert, Sven Lange, Peer Neubert and Peter Protzel

***Abstract*— Mobile robots often do not know their environment in advance and have to explore their surroundings after mission start. So, a robot has to localize itself and build a map in parallel, which represents the well-known SLAM problem. In case of a three-dimensional environment, a visual SLAM system is a reasonable choice and different SLAM systems exist. A promising approach of existing algorithms is ORB-SLAM which provides a computationally efficient (CPU-only), robust and open-source SLAM system.**

However, we noticed a drawback which causes a loss of map information after a *track loss* – e.g. in case of a fast rotational movement, ORB-SLAM fails to localize itself and starts a relocalization loop as long as it cannot recognize a previously visited place. Meanwhile, the incoming images are solely used for relocalization in the map and not used for enhancing the map. We coin these images *track-loss data* and hypothesize that this data can be beneficially used to enhance the map after a successful relocalization. In this paper, we discuss ways to exploit the track-loss data for map enhancement in ORB-SLAM, and show the result of an extension in comparison to the original algorithm with a proof-of-concept outdoor experiment.

I. INTRODUCTION

Perceiving the environment and estimating its own location are fundamental skills of a mobile robot. In case of an unknown map, the robot has to localize itself and build a map in parallel in order to maintain a global localization – the well-known SLAM problem (Simultaneous Localization And Mapping) occurs. There are different approaches for SLAM with different existing solutions for different scenarios like the 2nd *SpaceBot Cup*, a German national robotics contest, we participated in 2015 with two robots [1]; the task was to maneuver in a three-dimensional moon-like environment, and look for and carry objects. For such three-dimensional tasks, visual SLAM seems to be a reasonable choice since stereo-, RGBD- and mono-cameras are cost-efficient and provide high frame rate and resolution as well as color or intensity information.

A reasonable choice for visual SLAM is ORB-SLAM by Mur-Artal et al. [2] which is computationally efficient, robust and open-source[1]. In its current version, stereo- and RGBD-cameras can be used additionally which enable ORB-SLAM to determine the scale-factor in order to return a map with real-world dimensions. However, despite the good performance of ORB-SLAM we noticed a remaining drawback: Imagine, our robot is exploring a path during the SpaceBot Cup, accordingly, it estimates its location and builds a map in parallel. This is done with quite high precision by ORB-SLAM, so, the actual and the estimated path defer only with small deviation. However, in case of a fast rotational movement of the robot, ORB-SLAM looses its pose estimation and a *track loss* appears: The SLAM system fails to localize itself in the map and starts the relocalization mode. During relocalization, ORB-SLAM uses acquired images solely for relocalization; as it fails, the image is thrown away without further processing. As soon as a known place is recognized, the SLAM system continues its normal functionality without any exploitation of the acquired images during the track loss. However, by using this *track-loss data*, the map can be enhanced tremendously. Section III gives a real-world example for this problem (Fig. 1), provides two possible extensions of ORB-SLAM in order to exploit the track-loss data, and shows the potential of the discussed solutions.

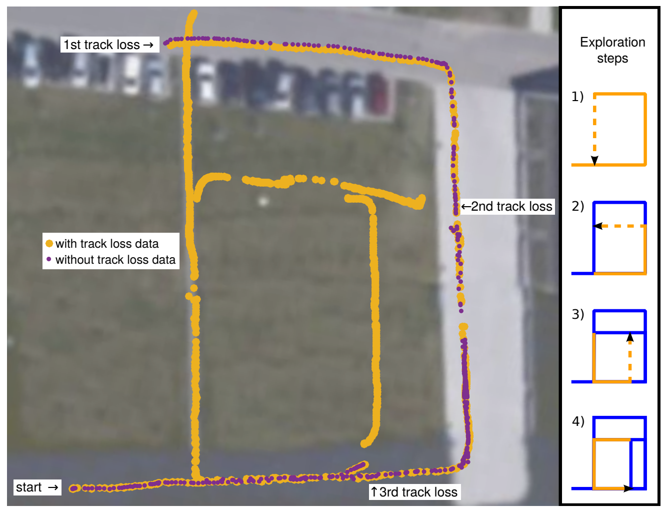

Fig. 1. Comparison of the original ORB-SLAM (purple) with a modified version (yellow) for a proof-of-concept experiment – each point represents the position of a keyframe on the track. Each track loss is caused by a fast rotational movement. The used extension of ORB-SLAM, *Reverse Replay*, is presented in section III. The right column gives an intuition about the traversed path; dashed lines mark path segments during track loss.

II. ORB-SLAM

ORB-SLAM [2] is a vision-only SLAM approach with a free open-source implementation[1]. It is designed to run efficiently on a CPU in real-time with long-term and large-scale ability. In its first version, it was designed to work with mono-cameras, however, the current version of ORB-SLAM provides additional support for stereo- and RGBD-cameras, which enables instantaneous scale estimation – this is, resulting dimensions of the point cloud are identical to the actual real-world dimensions.

The complete system is divided into three parts, which run on three different threads on a CPU: a pose tracking which

The authors are with the Department of Electrical Engineering and Information Technology, TU Chemnitz, 09126 Chemnitz, Germany. firstname.lastname@etit.tu-chemnitz.de

[1]ORB-SLAM2: https://github.com/raulmur/ORB_SLAM2

estimates the local motion; a local mapping which performs a local bundle adjustment, and creates or culls keyframes; and a loop closing module which searches for loop closures for every new keyframe and maintains a global consistency of the map by graph optimization.

ORB-SLAM uses solely ORB-features (Oriented FAST and Rotated BRIEF) [3] for all components – this makes the system more efficient, simple, and reliable. Relocalization as well as loop-closure detection are performed with the ORB-features in combination with a bag-of-words approach [4].

III. ENHANCED MAPPING WITH TRACK-LOSS DATA

ORB-SLAM provides a good pose estimation, which is indicated by a small deviation between the actual and the estimated pose in the real world. Additionally, in case of a loop closure the algorithm improves its estimations by a graph optimization. However, as soon as the algorithm becomes unable to determine its current position (track loss), it only tries to relocalize itself in previously recorded keyframes with current camera frames – if a relocalization attempt fails, the current frame is thrown away without any further saving or processing, which causes a full loss of map information during this period of track loss.

A performance improvement can be achieved in different ways with different properties and effort. Below, we present two approaches, *Reverse Replay* and *Parallel Tracking*, as an extension of the SLAM system in order to achieve a better mapping performance by exploiting data after a track loss.

Reverse Replay: The yellow points in Fig. 1 are proof-of-concept results of the following modification of ORB-SLAM: After a track loss, a frame sequence is recorded until the relocalization module succeeds. Subsequently after relocalization, the recorded frames are played back reversely into the system. The reverse replay provides the SLAM system a second chance to attach the track from the other side to the main track. However, as saving raw images can cause a memory overflow pretty fast, we preprocess the incoming color and depth images by extracting keypoints with corresponding ORB-features, and store these features instead of the whole images. A drawback is the missing real-time performance, as the replay replaces the new incoming image frames, however, since the images were already pre-processed the replay runs faster than real-time. Moreover, this strategy can simply be integrated into the existing ORB-SLAM implementation.

Parallel Tracking: The *Reverse Replay* method is easy to implement but not capable for real-time. A better approach is to run an additional process in parallel: In case of a track loss, this process starts a completely new path tracking until the relocalization of the main track signals a successful relocalization attempt. Then, the full new path, containing keyframes and relative positions, can be simply attached to the main track. This approach is promoted by the fact, that the CPU workload is reduced during the track loss period, as the local mapping and the loop closing threads sleep as long as no new keyframe is created. Accordingly, the additional CPU workload of two parallel ORB-SLAM systems with one shared track is relatively small. An alternative to a direct attachment of keyframes is to build a second pose graph and combine it with the main pose graph. This bypasses the need for a full integration of the second tracking result into the main track after a relocalization, and allows an external management of the single tracks. Such an approach could be advantageous, as it enables the system to reverse an attachment with the potential of higher robustness.

Fig. 1 depicts a real-world path tracking experiment: An outdoor environment of about $25\,\mathrm{m} \times 38\,\mathrm{m}$ size with buildings, parking cars, and a meadow with footpath is explored with an Intel Realsense RGBD-camera acquiring 640×480 images at $60\,\mathrm{Hz}$. A rapid turn is performed on three different occasions which causes a track loss. Each point in Fig. 1 represents an estimated keyframe position for both an original and a modified ORB-SLAM version. As can be seen, the original ORB-SLAM (purple points) looses its track after the first fast rotational movement, and starts to run a relocalization. As soon as it succeeds, the pose tracking is continued at the recognized place without any further enhancements of the map. In contrast, the yellow points show the pose tracking result for the proposed *Reverse Replay* with ORB-SLAM. The tracked path almost coincides with the actual path except for some small gaps, which are caused by another track loss during the replay phase. This shows the potential of ORB-SLAM to enhance the map further despite a track loss, which additionally enables the system to localize the camera on the previously lost path during a second traverse. In conclusion, the original ORB-SLAM creates only a map for the first path segment from the starting point to the first track loss, whereas the proposed extension maps almost the whole actual path.

IV. CONCLUSION AND FUTURE WORK

In this paper, we proposed an extension of ORB-SLAM, which we believe could become a standard algorithm for visual SLAM in mobile robotics as it provides multiple advantages. Further, we could demonstrate the high gain of map information for our proposed track-loss data exploitation.

In our future work, we intend to implement the more sophisticated *Parallel Tracking* approach of our proposed extensions in order to loose less or no information after a relocalization. The results would be published and provided as a free open-source patch for ORB-SLAM.

REFERENCES

[1] S. Lange, D. Wunschel, S. Schubert, T. Pfeifer, P. Weissig, A. Uhlig, M. Truschzinski, and P. Protzel, "Two Autonomous Robots for the DLR SpaceBot Cup – Lessons Learned from 60 Minutes on the Moon," in *Int. Symp. on Robotics*, Munich, Germany, 2016.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Trans. on Robotics*, 2015.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Int. Conf. on Computer Vision*, Barcelona, Spain, 2011.

[4] D. Gálvez-López and J. D. Tardós, "Bags of Binary Words for Fast Place Recognition in Image Sequences," *IEEE Trans. on Robotics*, 2012.