SLAM for Dynamic Environments

1st David Schmidig
ETH Zurich, D-MATH
Vision for Robotics Lab (V4RL)
Zurich, Switzerland
davschmi@student.ethz.ch

2nd Patrik Schmuck
ETH Zurich, D-MAVT
Vision for Robotics Lab (V4RL)
Zurich, Switzerland
pschmuck@ethz.ch

3rd Marco Karrer
ETH Zurich, D-MAVT
Vision for Robotics Lab (V4RL)
Zurich, Switzerland
karrerm@student.ethz.ch

4th Prof. Margarita Chli
ETH Zurich, D-MAVT
Vision for Robotics Lab (V4RL)
Zurich, Switzerland
chlim@ethz.ch

***Abstract*—This paper proposes a simple and computational cheap algorithm to extend and enhance state of the art SLAM systems. With this proposed algorithm such SLAM systems are capable of handling dynamic environments in a way that they can differentiate between reliable and unreliable landmarks and filter unreliable landmarks in their state estimation. Our method works especially well with texture-rich dynamic objects which stay inside the systems field of view (FOV) for a longer time and is capable of handling multiple objects as well as high occlusion of the FOV by dynamic objects. We cover the design, implementation and effect of the proposed algorithm in two state of the art SLAM systems. However the fundemental principle of the algorithm can be applied to all systems which use any sort of bundle adjustment for their state estimate optimization. Furthemore we compare and provide different datasets to be used with SLAM systems in dynamic environments.**

I. Introduction

A common studied problem in mobile robotics is the concurrent generation of the robots trajectory and a map of the environment. There are several state of the art approaches and implementations which solve the SLAM problem in a static environment in a robust manner. But only a small part of the literature considered also high-dynamic environemnts such as cities, boardwalks or multi-robot scenarios. In this project, we specifically target such high-dynamic environemnts and develop an algorithm which can be applied to any state of the art SLAM systems, which use a bundle adjustment optimization, without any major modification of the system itself. The approach does not use any additional data representation of the dynamic object and only uses a small computational window compared to the pose optimization of camera-poses and landmarks. The algorithm is capable of quickly classify landmarks on dynamic objects over consecutive frames and clusters them according to the detected shape of the object according to the landmarks. Further the algorithm uses movement prediction to detect newly introduced landmarks on dynamic obects. With this approach objects can also change their state from a dynamic to a static object, or vice-versa, and landmarks on the object are still captured to their corresponding type (static or dynamic).

This report is structured in three main chapters: Chapter II, where related literature is presented and compared to the use case defined in this project, Chapter III, where the design of the proposed algorithm is explained and Chapter IV, where the evaluation of two state of the art SLAM systems enhanced with our proposed algorithm and its performance on dynamic datasets is shown.

II. Related Work

State of the art SLAM systems such as [4], [5] have shown that their system work precise and robust in static environments. Usually such systems do not target high-dynamic environments and dynamic objects lead these systems to fail in terms of accuracy. There are approaches at mapping the static and dynamic environment and generate a static and a dynamic map such as [9] by D. Wolf et al. They use an additional laser range finder on the robot and show their results in a visual manner. While other publications like Wei Tan et al. [6] or Shimamura et al. [7] target augmented reality set-ups and treat landmarks on dynamic objects as outliers in their respective routine. Their approaches can handle such dynamic scenes according to the drift in augmented reality scenes but are not shown to work in larger environemnts and trajectories. A different proposition with object tracking to handle dynamic objects is presented by Chieh-Chih Wang et al. [10] while using additional prior map information and object detection. Co-SLAM by Danping Zou et al. [8] uses an approach based on the reprojection error between multiple different cameras to detect dynamic objects and even reconstruct their movement. This approach seems to be the most simple solution for handling dynamic objects in a SLAM problem without the need to map the dynamic objects with the only drawback that at least two seperate cameras are used.

III. SLAM in Dynamic Environments

This section, describes how the proposed and implemented algorithm works and explains each step in detail.

A. General Idea

In a best case scenario, the state estimation of the SLAM system should only rely on landmarks which correspond to a static environment, such that we can filter out unreliable landmarks (e.g. landmarks on dynamic objects) and only use reliable ones (e.g. landmarks in static environment) to be used in the state estimation. To adress some of the challenges mentioned in Chapter I and to be more flexible, we introduce a non-binary classification as a reliability-weight on each landmark.

Following the approach of [1] and the fact, that the reprojection error, in theory, is larger over several consecutive frames for dynamic objects compared to static objects, we take the reprojection error as an indication for a landmark to be reliable or unreliable. These reliability-weights then can be used inside the bundle adjustment as modifier of the target cost function which is being optimized as seen in Figure 1.

B. Algorithm Design

Our algorithm consists of three phases: (1) Compute reliability weight, (2) Cluster unreliable landmarks, (3) Reduce weights according to clustering, that are executed each frame. A landmark reliability-weight and parameter intialization strategy is used for initialization of weights and algorithm parameters. The final step is to incoorporate the reliability-weight into the bundle adjustment.
The proposed algorithm is inserted between two bundle adjustement steps of the state estimation loop and modifies the cost function inside the bundle adjustment according to the computed reliability weight.

Phase 1 (compute reliability-weight)
The reprojection error $e_{i,t}$ of landmark i at frame t as a result from the bundle adjustment is used to compute a first weight $W_{t,i}$ of the landmark i as follows:

$$W_{t,i}(W_{t-1,i}, \hat{e}_{t,i}) = \frac{1}{2}W_{t-1,i} + \frac{1}{2}F(\hat{e}_{t,i}) \tag{1}$$

where,

$$\hat{e}_{t,i} = \frac{e_{t,i}}{e_{max}} \tag{2}$$

and,

$$F(\hat{e}_{t,i}) = \begin{cases} 0 & \text{if } \hat{e}_{t,i} \geq 1, \\ 1 & \text{if } \hat{e}_{t,i} \leq 0 \\ 1 - \frac{a^{\hat{e}_{t,i}}-1}{a-1} & \text{otherwise} \end{cases} \tag{3}$$

The reprojection error $e_{i,t}$ is computed after the 3D-poses of the landmarks and the camera are optimized according to the bundle adjustment as follows:

$$e_{i,t} = \sqrt{x_{i,t}^2 - (X_{i,t}P)^2}, \tag{4}$$

where X is the 3D-landmark-pose and $P = K\begin{bmatrix}R\mathbf{t}\end{bmatrix}$ the projection matrix of the camera with the cameras intrinsics and extrinsics parameters.

Phase 2 (cluster unreliable landmarks)
First, all landmarks with a reliability weight below c_{max} are considered as a set of cluster candidates. DBSCAN [2] is applied on the cluster candidates and results in a list of clusters with their respective landmarks, a convex hull and an average optical flow computed from the optical flow of the landmarks inside the specific cluster. Using the average optical flow, we can project all clusters from the last n_{wc} frames onto the current frame and use them in phase 3. n_{wc} is in this case the window-size of the cluster-movement prediction. This step provides the algorithm with a simple cluster-movement prediction onto the next frame which are then used in phase 3.

Phase 3 (reduce weights according to clustering)
Let N_i be the number of projected clusters (to the current frame) in which landmarik i lies. We now reduce the weight $W_{t,i}$ with a factor of r as

$$\hat{W}_{t,i} = W_{t,i} * r(1 - \frac{N_i}{n_{wc}}), \tag{5}$$

where r is a defined reduction parameter.

Landmarks Initalization
Since landmarks introduced in timestep t do not have any weights to be used in the bundle adjustemnt in timestep $t+1$, we set the reliability-weight of newly detected landmark to 0.5. This favors a neutral decision of the algorithm on this landmarks reliability. The problem of lost and new introduced landmarks are then captured with clustering and movement prediction of the clusters as mentioned in phase 3.

Parameter Initalization
Parameter e_{max} from Equation 2 is set in an initial scene, where no dynamic objects in the robots field of view are assumed, as the median of the set of maximum reprojection errors over 80 frames. c_{max} is set to 0.3 and n_{wc} to 8 in our experiments because they have given the best results. r from Equation 5 is set to $r = 0.59 \leq \frac{c_{max}}{0.5}$ such that a newly introduced landmark with weight 0.5 which appears to be in all clusters from previous frames has a weight of $W_{t,i} \leq c_{max}$ and is immediately considered as a cluster candidate.

Bundle Adjustment
The final weights $\hat{W}_{t,i}$ are then used inside the cost function of the bundle adjustment as a modifier of the residual of the landmarks, which is in this case simply the reprojection error itself. Note, that we do not multiply the reliability-weight with the reprojection error but with the residual inside the cost function, such that we evaluate the undistorted reprojection-error to be used in phase 1.

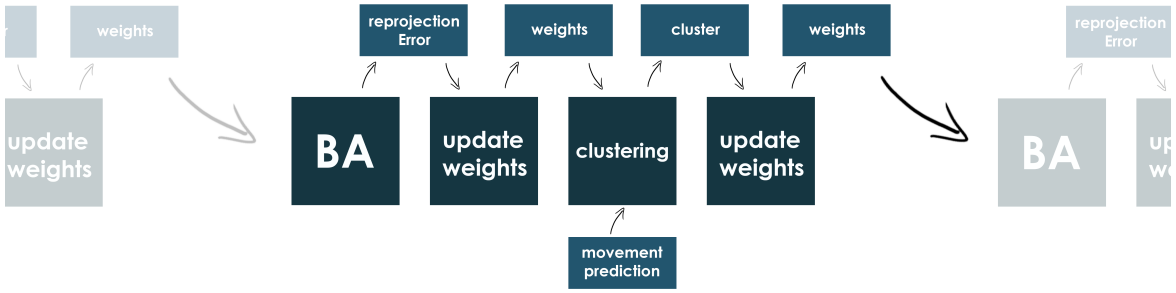


Fig. 1: Graphical overview of the proposed algorithm inserted after the bundle adjustment (BA).

IV. Experiments and Results

In this section the created and used dataset are presented and then performance of two different state of the art SLAM systems are compared to the specific system enhanced with our proposed algoritm is shown.

A. Datasets

Because there are no datasets available for dynamic environments with ground truth, we created our own datasets. A mixed-reality dataset is created using the EUROC [3] dataset and adding artifical dynamic objects (see Figure 2a) onto the image plane using paintings with different movements (modes). This dataset has a ground truth trajectory and thus can be used to evaluate the accuracy of SLAM systems precisely. Furthermore another dataset consisting of one or two texture-rich dynamic objects moving infront of a camera (see Figure 2b) inside a room are captured. Within this dataset, different movements, such as following an object, hovering or sideways movement are present. Ground truth has been captured with the Vicon system. For the final and most real-world dataset a boardwalk-scene with passing trams is recorded (see Figure 2c). This dataset does not have ground truth but one can still evaluate a final drift, since start and end points of the trajectory align.

B. Error measurement

For performance evaluation of state of the art SLAM systems compared to our proposed enhancement of the respective system in dynamic environemnts, two different error measurements are used. For datasets where a ground truth trajectory is available (i.e. mixed reality and vicon) the estimate and ground truth trajectory are aligned and the root-mean-squared error (RMSE) at different trajectory length is averaged and visualised as seen in Figure 3a. If no ground truth trajectory is available, as in the Bahnhofstrasse dataset, the trajectories are chosen such that start and endpoint overlap. With this we can evaluate a drift between the start- (or end-) point and the estimate end-point as euclidean distance (also see Figure **??**).

C. VINS-Mono

VINS-Mono [5] is a feature tracking based SLAM system which means that landmarks between consecutive frames are matched according to the feature tracker on the landmarks. VINS also does not do any additional geometrical consistency checks while using landmarks in the state estimation, which is prone to fail in scenes where a large amount of landmarks on dynamic objects are initializable.

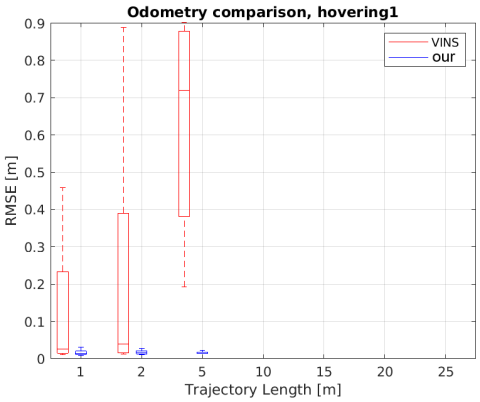
1) Vicon: In Figure 4 a selection of scenes from the vicon dataset is shown. First of all we notice that the proposed enhancement compared to VINS' system performes better on all those scenes. One can also distinguish between scenes where VINS' error and trajectory diverges with an RMSE above $4m$ as seen in the hovering scene with two objects (Figure 3b) and scenes where dynamic objects have a significant influence on VINS' accuracy as in a hovering scene with just one dynamic object (Figure 3a). In both cases our proposed algorithm detects and classifies landmarks on dynamic objects successfully and keeps the system from diverging since the RMSE in both scenes is below $0.2m$, $0.05m$ respectively. Since these are hovering scenes the RMSE should be much lower compared to datasets where the robot has a larger trajector. Looking at two different non-hovering scenes in Figure 3c and Figure 3d we observe that VINS handles these scenes rather well and our method performes as well as VINS

(a) Mixed Reality

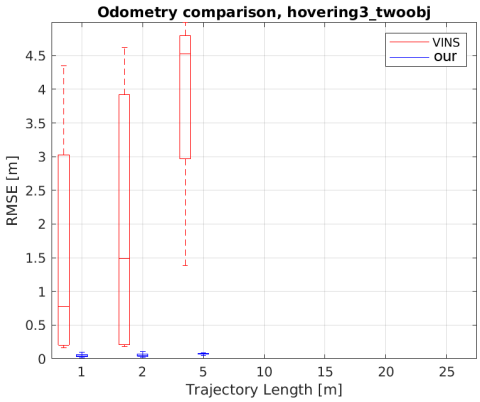
(b) Vicon

(c) Bahnhofstrasse

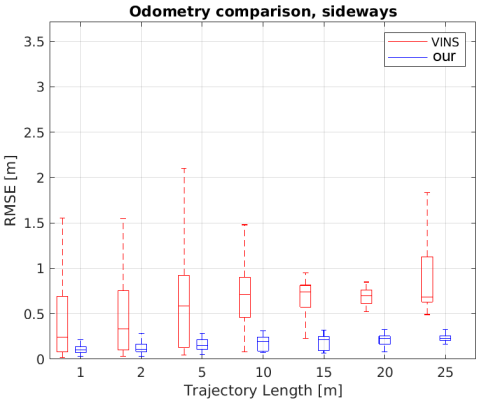
Fig. 2: Example scenes from all three datasets.



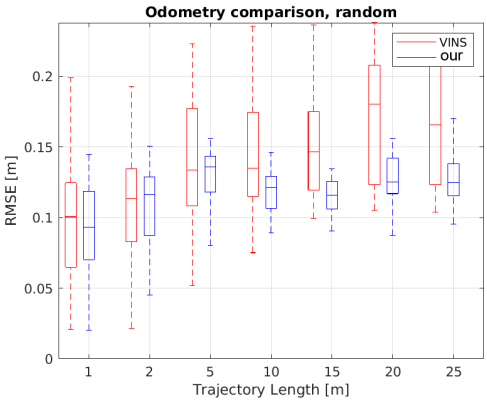
(a) Hovering with a single dynamic object.



(b) Hovering with two dynamic objects



(c) The dynamic object only moves lateral respect to the camera.

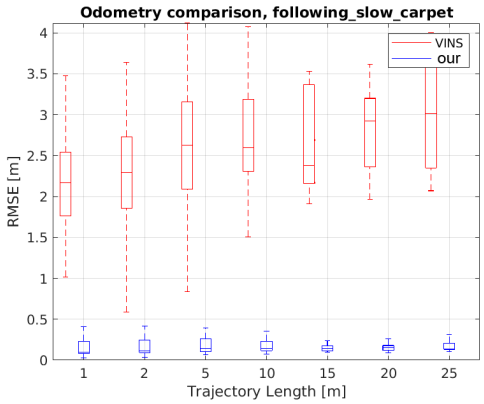


(d) Random movement of the dynamic object inside the cameras FOV.

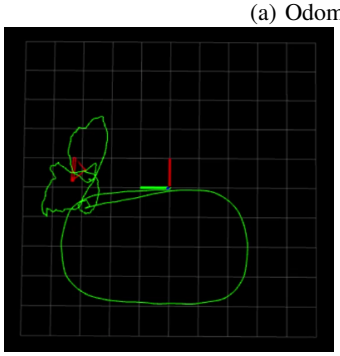
Fig. 3: Selection of VINS comparison on the vicon dataset.

in this case. This concludes, that the enhancement does not influence the system in cases where dynamic objects do not have any influence on the state estimation at all. In Figure 4 another scene where VINS diverges is presented along with the generated trajectory of VINS and VINS + our method. Already from a qualitative standpoint, one can tell that VINS' result is not usable in this case while our methods result is. One can inspect the classification and clustering of the algorithm visually in Figure 5. The weights of each landmark are color coded. A landmark with a weight of 1 has a bright pink color and a landmark with a weight tending to zero is black. Inbetween, a linear color-transition is used. As one can see in Figure 5b, not all landmarks on the left dynamic object are clustered. However they have a lower weight according to their color-coding and thus have a smaller influence in the state estimation.

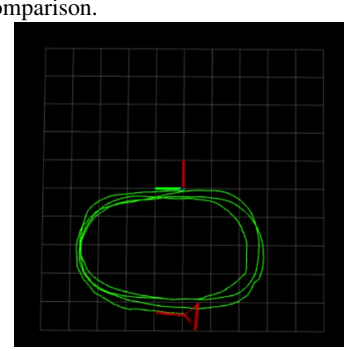
2) Bahnhofstrasse: The evaluation for the Bahnhofstrasse dataset is presented in Figure 6a and shows that also in this case, our method combined with VINS performs up to $2.5x$ more accurate than VINS. Also VINS' estimate does not diverge in this case, since the amount of tracked landmarks on dynamic objects and hence their influence on the state



(a) Odometry comparison.

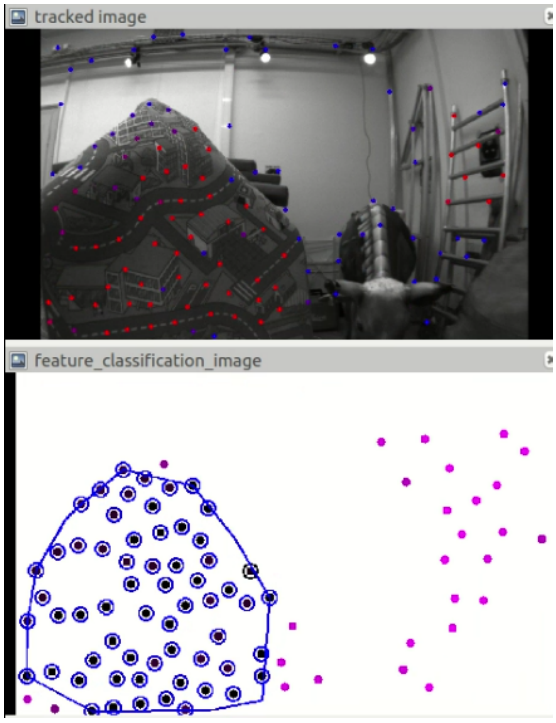


(b) Trajectory without our method.

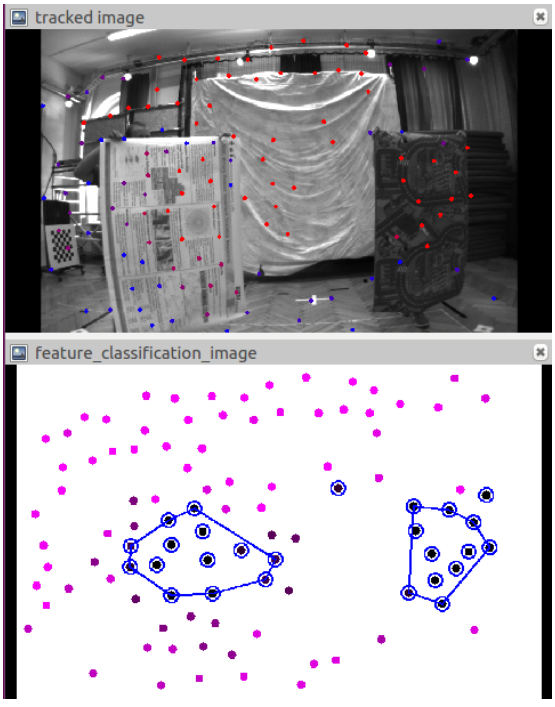


(c) Trajectory with our method.

Fig. 4: Following a dynamic object scenario. Quantitative and qualitative comparison.



(a) Single obejct



(b) Two objects

Fig. 5: Classifying and clustering visualized.

estimation is smaller than in the vicon dataset.

D. Okvis

Okvis [4] is a feature descriptor matching based SLAM systems that computes feature correspondence between frames and landmarks based on a descriptor vector. This system also does some consistency checks between the landmarks and the robots trajectory according to a geometry constraint. Thus Okvis already has additional checks to limit the influence of unreliable landmarks to prevent descriptor mis-matches from distorting state estimation. Thus this also largely prevents landmarks on dynamic objects to be used in more than just a few (up to 10) frames.

1) Mixed Reality: As already mentioned, the additional checks on landmarks which Okvis already performs handle scenes like the one from the mixed reality dataset well without drifting significantly or even diverging as seen in Figure 7. Thus the proposed enhancement with our method does not hold any improvements over Okvis' algorithm, however does also not produce worse results. Again, on scenes which are can be handled by Okvis itself, our algorithms has no negative influence on the accuracy. The insignificant difference between both methods in Figure 7 originates from the fact that Okvis has a random nature in feature detecting, matching and outlier detection (e.g. RANSAC) and can be neglected. We note, that scenes from Room 2 in Figure 7c and Figure 7d okvis have a much higher error compared to the scenes from room 1. This is due to the dynamic object occluding the image such that there are not enough features left to be used in the state estimation. In this case our proposed method does not give any advantages but also performs not worse,

2) Vicon: in Figure 8, the same observations as in the mixed reality dataset can be made on the vicon dataset. Okvis handles all of these scenes without diverging and a reasonable error, while our method does not give any advantages nor penalties according to the accuracy.

3) Bahnhofstrasse: Unfortunately Okvis and Okvis with our method fails on the Bahnhofstrasse dataset and hence cannot be evaluated properly.

E. Benchmark Case Study

As seen in Figure 6b, classification and clustering uses less than 1% of the time, optimization and other parts of the state estimation uses. Hence our proposed method does not have any impact in the systems real-time capability.

F. Conclusion and Future Work

We propose a simple but effective method to detect, classify and cluster landmarks which influence the state estimation in a negative manner according to the trajectories accuracy. It is able to filter such unreliable landmark without losing the systems initial accuracy on scenes where no dynamic object influence its state estimation. Further our method is computationally cheap and can be implemented in every system that uses a bundle adjustment as the optimization step in the state estimation. In scenes where VINS diverges because of too many unreliable landmarks, our method generates an accurate and usable trajectory. This method also works under high occlusion ($> 60\%$) of the image by dynamic objects (seen in the *following-scene* in Figure 4) and can also handle detection and clustering of multiple objects. This method works especially well, if features of unreliable landmarks are tracked over a long period of time, such as in a scene, where the robot has the same dynamic object in its FOV for a longer time from the same angle. The classification depends largely on e_{max} where some tuning can be done. Since we assume no dynamic objects in the initialization phase, one can try to adapt e_{max}'s initialization with dynamic objects. Another approach is to create a strategy to adaptively adjust e_{max} according to the robots movement velocity (e.g. via IMU), since we also assume that our robot has the same movement throughout the whole scene. In Okvis' enhancement, one could apply a more robust movement detection, such that newly introduced landmarks on dynamic objects are captured independent from the time they have been clustered the last time (e.g. no cluster-window).

REFERENCES

[1] D. Zou and P. Tan, "CoSLAM: Collaborative Visual SLAM in Dynamic Environments," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 2, pp. 354-366, Feb. 2013.

[2] Martin Ester, Hans-Peter Kriegel, Jrg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96), Evangelos Simoudis, Jiawei Han, and Usama Fayyad (Eds.). AAAI Press 226-231.

[3] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. Achtelik and R. Siegwart, The EuRoC micro aerial vehicle datasets, International Journal of Robotic Research

[4] Stefan Leutenegger, Paul Timothy Furgale, Vincent Rabaud, Margarita Chli, Kurt Konolige, Roland Siegwart. Keyframe-Based Visual-Inertial SLAM using Nonlinear Optimization. In Proceedings of Robotics: Science and Systems, 2013

[5] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," in IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, Aug. 2018.

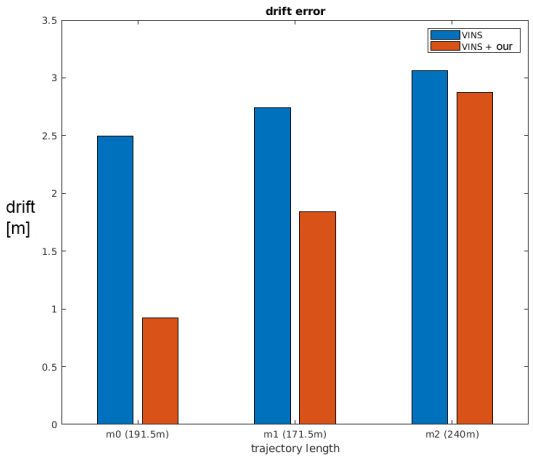
[6] Wei Tan, Haomin Liu, Z. Dong, G. Zhang and H. Bao, "Robust monocular SLAM in dynamic environments," 2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Adelaide, SA, 2013, pp. 209-218.

[7] J. Shimamura, M. Morimoto, and H. Koike. Robust vSLAM for dynamic scenes. In MVA, pages 344347, 2011.

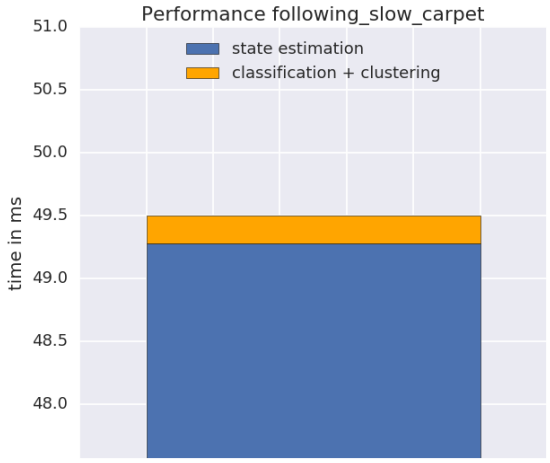
[8] Danping Zou and Ping Tan, "CoSLAM: Collaborative Visual SLAM in Dynamic Environments," in IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 35, no. 2, pp. 354-366, 2013.

[9] D. Wolf and G. S. Sukhatme, "Online simultaneous localization and mapping in dynamic environments," IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004, New Orleans, LA, USA, 2004, pp. 1301-1307 Vol.2.

[10] Chieh-Chih Wang, C. Thorpe and S. Thrun, "Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas," 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422), Taipei, Taiwan, 2003, pp. 842-849 vol.1.

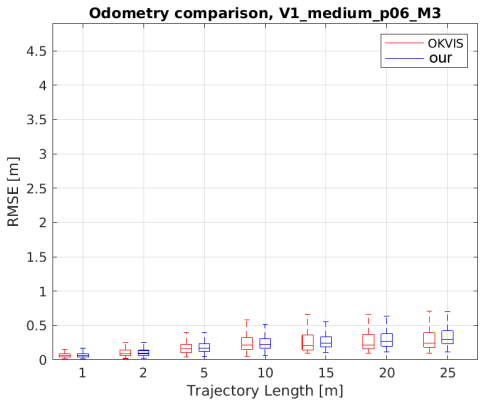


(a) Drift error on the Bahnhofstrasse dataset.

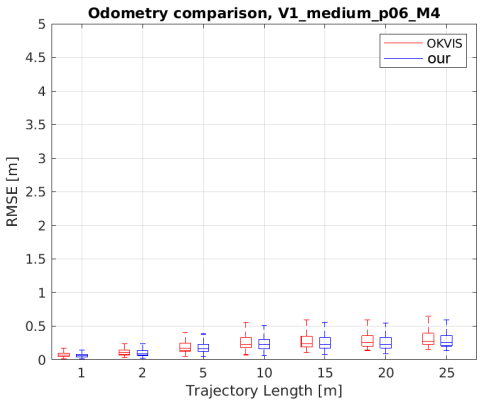


(b) Time consumption of classification, clustering compared to state estimation on a vicon scene.

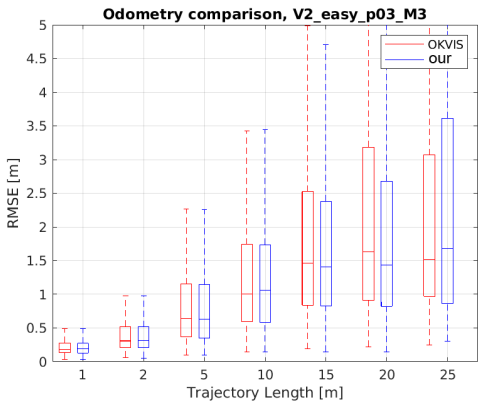
Fig. 6



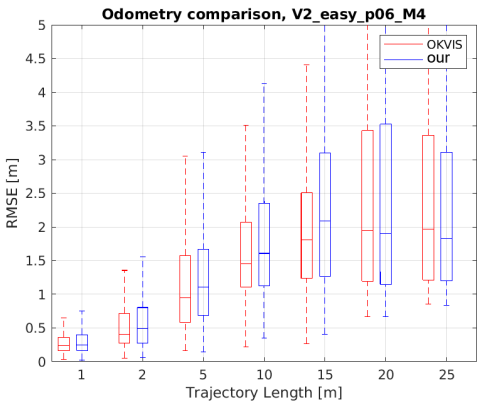
(a) Mixed reality mode 3, Room 1



(b) Mixed reality mode 4, Room 1

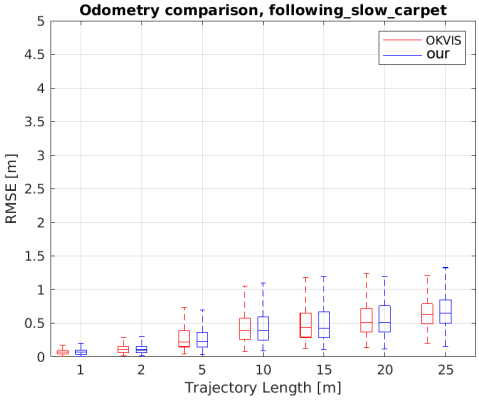


(c) Mixed reality mode 3, Room 2

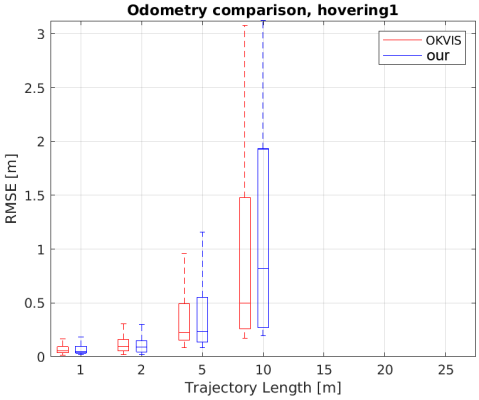


(d) Mixed reality mode 4, Room 2

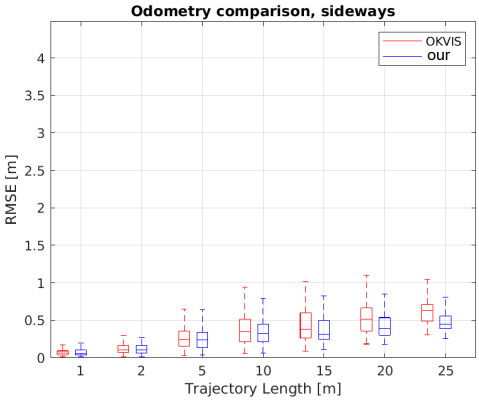
Fig. 7: Odometry comparison for Okvis on scenes of the mixed reality dataset.



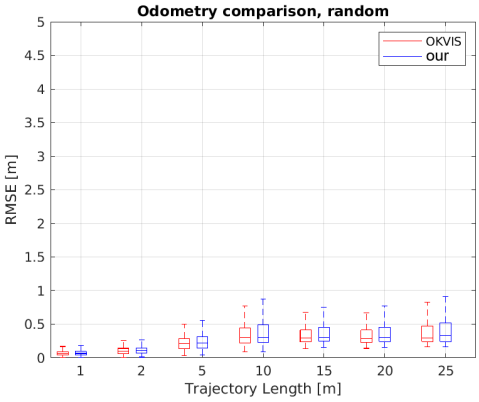
(a) Following a dynamic object.



(b) Hovering with a dynamic object.



(c) The dynamic object only moves lateral respect to the camera.



(d) Random movement of the dynamic object inside the cameras FOV.

Fig. 8: Odometry comparison for Okvis on scenes of the vicon reality dataset.

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Declaration of originality

The signed declaration of originality is a component of every semester paper, Bachelor's thesis, Master's thesis and any other degree paper undertaken during the course of studies, including the respective electronic versions.

Lecturers may also require a declaration of originality for other written papers compiled for their courses.

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervisor.

Title of work (in block letters):

SLAM FOR DYNAMIC ENVIRONMENTS

Authored by (in block letters):

For papers written by groups the names of all authors are required.

Name(s):	**First name(s):**
SCHMIDIG	DAVID

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism.

Place, date	**Signature(s)**
EMMENBRÜCKE, 13.10.2018	

For papers written by groups the names of all authors are required. Their signatures collectively guarantee the entire