MOD SLAM: Mixed Method for a More Robust SLAM without Loop Closing

Thomas Belos[1][a], Pascal Monasse[1][b] and Eva Dokladalova[2][c]

[1] *LIGM, Ecole des Ponts ParisTech, Université Gustave Eiffel, CNRS, F-77454 Marne-la-Vallée, France*

[2] *LIGM, Université Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée, France*

Keywords: SLAM, Simultaneous Localizaton and Mapping, Odometry, Drift, Robust, Online, ORB, DSO, Photometric, Features, Indirect, Direct, Mixed, Hybrid.

Abstract: In recent years, the state-of-the-art of monocular SLAM has seen remarkable advances in reducing errors and improving robustness. At the same time, this quality of results can be obtained in real-time on small CPUs. However, most algorithms have a high failure rate out-of-the-box. Systematic error such as drift remains still significant even for the best algorithms. This can be handled by a global measure as a loop closure, but it penalizes online data processing. We propose a mixed SLAM, based on ORB-SLAM2 and DSO: MOD SLAM. It is a fusion of photometric and feature-based methods, without being a simple copy of both. We propose a decision system to predict at each frame which optimization will produce the minimum drift so that only one will be selected to save computational time and resources. We propose a new implementation of the map that is equipped with the ability to actively work with DSO and ORB points at the same time. Our experimental results show that this method increases the overall robustness and reduces the drift without compromising the computational resources. Contrary to the best state-of-the-art algorithms, MOD SLAM can handle 100% of KITTI, TUM, and random phone videos, without any configuration change.

1 INTRODUCTION

Boosted by the automotive industry and robotics applications, visual monocular Simultaneous Localisation and Mapping (SLAM) algorithms have made remarkable progress in terms of robustness, accuracy and of computation cost reduction (Engel et al., 2017; Ferrera et al., 2021; Forster et al., 2017). Today, the state-of-the-art algorithms run on embedded CPUs in real-time and are being used extensively in many emerging applications such as navigation for drones (von Stumberg et al., 2017), self-driving cars (Singandhupe and La, 2019), 3D modeling of urban environments, and more (Lothe et al., 2009).

Despite this progress, the robustness and generated drift issues have still not been resolved (Chahine and Pradalier, 2018). Mainly due to the map, which will always contain a certain amount of gauges freedom (Strasdat et al., 2010), as the measured pixels 2D positions are discrete numbers. This drift propagates and worsens through the various optimization processes, especially during the gradient descent, and produces the loss of consistency of the results.

It is known that this drift significantly penalizes direct methods based on photometric optimization which are otherwise very efficient (Engel et al., 2017). On the other hand, feature-based methods, which are less affected by this phenomenon, can however have a strong drift in poorly textured environments (Mur-Artal and Tardós, 2017).

In general, to solve this issue a global correction such as loop closure is exploited (Gálvez-López and Tardós, 2012). However, the loop closure burdens the computing requirements of the online SLAM, and becomes impossible if a loop doesn't exist in the trajectory.

In this paper, we introduce a new mixed SLAM method (Figure 1) that we call Mixed ORB-SLAM2 and DSO (a.k.a. MOD SLAM). The principle of MOD SLAM consists in a mix of the currently most popular state-of-the-art methods: feature-based ORB-SLAM2 (Mur-Artal and Tardós, 2017) and direct method DSO (Gao et al., 2018). MOD SLAM relies on a new decision system allowing to predict in ad-

[a] https://orcid.org/0000-0002-2172-5962

[b] https://orcid.org/0000-0001-9167-7882

[c] https://orcid.org/0000-0003-1765-7394

Belos, T., Monasse, P. and Dokladalova, E.
MOD SLAM: Mixed Method for a More Robust SLAM without Loop Closing.
DOI: 10.5220/0010833600003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 5: VISAPP*, pages 691-701
ISBN: 978-989-758-555-5; ISSN: 2184-4321

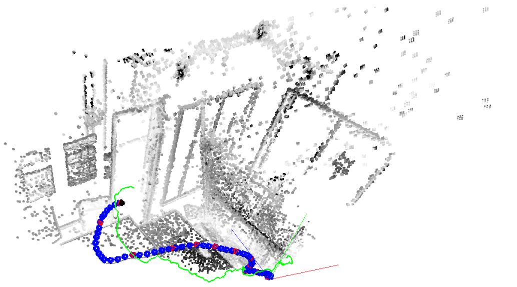


Figure 1: Screenshot of MOD SLAM software during an execution. The estimated trajectory is blue. The groundtruth is green.

vance which optimization will produce minimal drift. To ensure minimal required computing effort, MOD SLAM uses an original map that has the ability to actively work with ORB-SLAM2 and DSO information at the same time, without running both methods in parallel. We show on public datasets and on our smartphone datasets that this method improves the robustness and reduces the drift without increasing the computation resources.

The remaining part of this paper is organized as follows. Sections 1.1 and 1.2 introduce the related work and our contribution. Section 2 briefly describes ORB-SLAM2 and DSO procesing pipelines and principles. Section 3 presents the principles of the MOD SLAM method. The experimental results are collected and discussed in section 4. Finally, the last section is dedicated to the outline of conclusions and future work.

1.1 Related Work

Visual SLAM and odometry methods include both camera pose estimation and 3D scene reconstruction algorithms. We can distinguish four main classes.

Indirect SLAM. (also known as feature-based SLAM) proceeds by extracting salient image feature sets and matching them by using feature specific descriptors. The optimization of the camera poses and the 3D points are done by minimizing the reprojection error using the 2D-3D matches. A typical example is Parallel Tracking and Mapping for Small AR Workspaces (Klein and Murray, 2007) by Klein et al. that has inspired numerous indirect SLAM methods. This is the case of ORB-SLAM2 (Mur-Artal and Tardós, 2017) (improved ORB-SLAM (Mur-Artal et al., 2015)) by Mur-Artal et al. which is currently one of reference state of the art indirect method equipped with a loop closure system and has relocalization capabilities. The drawback is relatively high computing time and the need for robust estimation techniques. If ORB-SLAM2 is known to perform well on KITTI odometry car dataset (Geiger et al., 2012), it can fail to handle the videos of TUM Dataset (Engel et al., 2016) with low texture information.

Direct SLAM. (also known as photometric optimization-based SLAM) methods estimate motion and structure by optimizing directly error measure at pixel intensities level. Dense direct method like DTAM (Newcombe et al., 2011) estimates dense depth map and optimizes error by using all image pixels. These kinds of methods generally need a big amount of resources and a GPU. Whereas sparse direct methods as (Gao et al., 2018) by Engel et al. performs an optimization based on a subset of interest points in the images. Contrary to ORB interest points, DSO interest points do not need repeatability, but they are required to be photometrically trackable. This method track a huge number of points very precisely, with very low CPU load, resulting in a good quality point cloud. Despite these advantages, direct methods suffer from drift problem in dynamic and fast motion scenes. It needs perfect internal and photometric calibration and good image quality.

Semi-Direct SLAM. is an in-between direct and indirect SLAM. We can cite SVO (Forster et al., 2014; Forster et al., 2017) by Forster et al. as the state of the art semi-direct method example. It photometrically tracks and matches the features, but still minimizes the re-projection error. Its alternative CNN-SVO by Yan Loo et al. (Loo et al., 2019) aims to improve monocular SLAM without closing loop, by sustaining SVO with deep learning estimation. The authors succeeded to reduce significantly the drift, but at the price of GPU support. Notice that CNN-SVO is one of the few articles which evaluate on monocular ORB-SLAM2 without closing loop.

Mixed Methods. are based on tight coupling of direct and feature-based methods. Hun Lee et al. (Lee and Civera, 2018) proposed Loosely-Coupled Semi-Direct Monocular SLAM which runs ORB-SLAM2 and DSO in parallel. The authors execute a standard DSO, and forward the marginalized frame to a new ORB SLAM with loop closure. The principal benefit of their system is to have a photometric SLAM with loop closure. However, without loop closure, they see no improvement over regular ORB-SLAM2, probably because ORB-SLAM2 is always taking the lead (Section 4.4).

The aim of this paper is to propose a new mixed ORB-SLAM2 and DSO method, similar in general principle to (Lee and Civera, 2018). The main

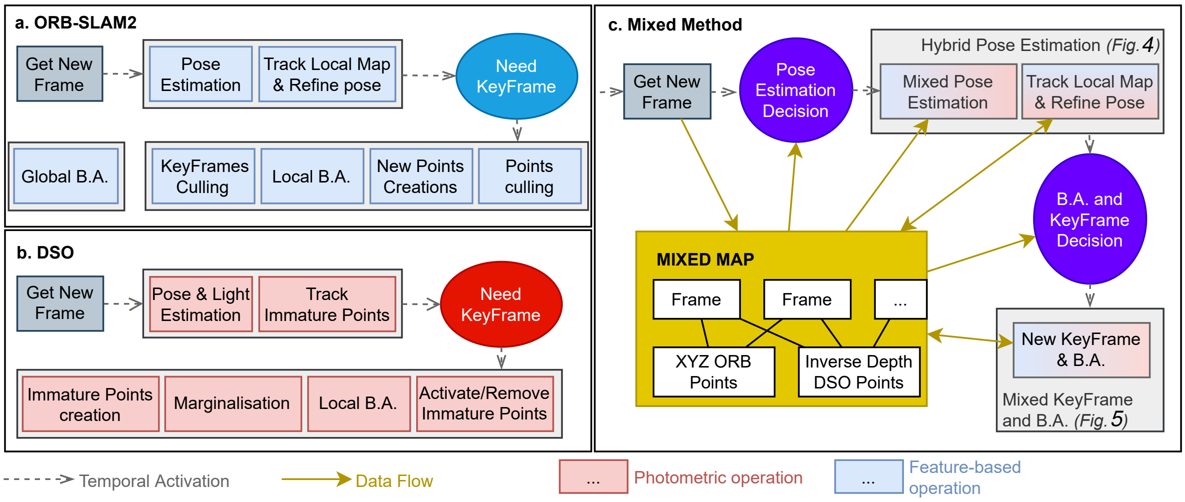


Figure 2: a. ORB-SLAM2 pipeline without loop closure; b. DSO pipeline; c. MOD SLAM pipeline.

means is to reimplement DSO and ORB-SLAM2 more closely (Figure 2). In this sense, we have a unique shared map and never run both ORB-SLAM2 and DSO in parallel. Instead, we define an estimation method allowing to predict the drift measure and to minimize it frame by frame.

1.2 Contributions

The principal aims of this paper is to increase the robustness, and to produce the minimal drift of ORB-SLAM2 and DSO, by introducing a new original mixed SLAM. The main contributions of our method are the following:

- We define a novel SLAM architecture which is ORB-SLAM2-based (Mur-Artal and Tardós, 2017) and DSO-based, with a mixed map providing a fluid and consistent sharing of the data (including frame camera pose and 3d points coordinates) between ORB-SLAM2 and DSO.
- We formulate and evaluate a decision method, allowing to predict in advance which of ORB-SLAM2 or DSO will be more robust, at each step of the SLAM.
- We show in this paper that, by mixing ORB-SLAM2 and DSO, MOD SLAM can handle 100 percent of the videos, reduce the drift (without using the loop closure) on KITTI dataset (Geiger et al., 2012) and on our own phone dataset.

2 ORB-SLAM2 AND DSO PIPELINES

The left part of figure 2 shows the ORB-SLAM2 and DSO pipelines. This section intends to give some definitions, and to describe briefly the pipelines of ORB-SLAM2 and DSO, necessary to understand the following description of MOD SLAM.

2.1 Definitions

Immature Point. Initial and temporary status of a 3D point, before deciding whether it is an inlier or outlier. At keyframe creation of DSO, newly detected interest points are mapped as 3D points (with unknown depth) and classified as immature. At the next keyframe creation, immature points become inlier or outlier. Outliers can be discarded as there is no loop closing, hence no chance to become inliers later.

Local Map, Active Points and Frames. The local map is a subgraph of the map containing all the active points and frames. The active frames contain the current frame and some previous frames, based on covisibility. The active points are all the 3D points visible in at least one active frame. The local map is updated during the local map tracking (Section 3.2.4). Only active points are tracked on the new frames and only the local map is bundle adjusted during keyframe creation.

2.2 Indirect Pipeline: ORB-SLAM2

At each new frame, a pose estimation is performed: tentative matches between 2D ORB interest points and 3D points are found. The descriptor of the 3D point is the coordinate-wise median of the descriptors of ORB points having given birth to it. A best to second-best descriptor distance ratio is compared to a threshold to ignore ambiguous 2D/3D point correspondences. The sum of squared residuals of point projections is then minimized, as a function of the frame pose (rotation and translation), where at each iteration a large residual removes the correspondence from the next energy computation, which can be reinserted at a future iteration if its residual becomes small again.

From this initial pose, numerous new 2D/3D correspondences can be found by descriptor comparison because the search is restricted to the neighborhood of the 3D point projection. The same minimization with all new correspondences allows refining the pose.

When a new keyframe is needed, duplicate 3D points are merged (culling), new 3D points may be created by triangulation with matching ORB points in previous keyframe, and a local bundle adjustment is performed, with variables the new keyframe pose and the positions of 3D points. Finally, the new keyframe is compared to the previous two keyframes to check whether the middle one is redundant, in which case it loses its status of keyframe.

2.3 Direct Pipeline: DSO

A photometric energy minimization is performed wrt pose and light change parameters. Immature points, 3D points with still unknown depth, generated by the DSO interest points in the preceding keyframe, are then tracked in the new frame. The success of this tracking is then used to decide whether to activate the 3D point as mature or to remove it. A local bundle adjustment is then performed and the marginalisation can fix some previous keyframes: they are removed from the local map. Finally, the new keyframe generates from its 2D interest points immature 3D points.

3 MIXED METHOD

Our mixed method is a fusion of ORB-SLAM2 (Mur-Artal and Tardós, 2017) and DSO (Engel et al., 2017). The pipeline shown in figure 2 borrows modules from both, though both run paths are exclusive: Two functions predict in advance if it is best to use photometric or feature based method to minimize drift. The first chooses between tracking with ORB and refining with DSO, or the contrary, and the second one whether to use an ORB or a DSO bundle adjustment.

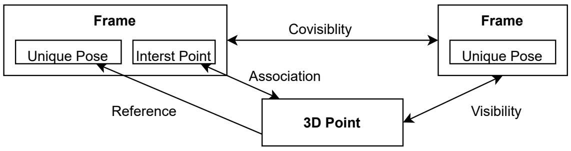


Figure 3: High level representation of the edges in the map graph.

3.1 Map

3.1.1 Mixed Map Structure

Our map is a non-oriented graph, whereas DSO and ORB maps are oriented graphs, with edges going from the frames to points. The nodes are:

- **The frames** including: their camera pose, image, DSO interest points, ORB interest points and their respective descriptors, their internal and photometric calibrations...
- **The 3D ORB points** with their position, median binary descriptor, uncertainty.
- **The 3D DSO points** with their position represented with inverse depth parametrization (Civera et al., 2008), Hessian.

For each frame, the pose estimation is unique. All poses and 3D points, whether issued from ORB or DSO, use the same coordinate system.

The edges of the graph represented in figure 3 are:

- **Between frames:** The covisibility edge between frames when they see the same points.
- **Between Frame and 3D ORB Point:**
  - Used for the correspondence of the ORB-SLAM 3D Points with the ORB interest points
  - Used when the point is assumed visible inside the frame according to the camera pose, but is not matched with an interest point. This allows knowing the number frames in which the point is visible but not matched.
- **Between Frame and 3D DSO Point:**
  - When the DSO point is extracted from the frame, and mapped, hence each DSO point has only one such edge.
  - Used when the point is assumed visible inside the frame, to track the point and to bundle adjust photometrically.
  - Used when the point position is relative to its reference camera, so that the point needs to move with the camera pose.

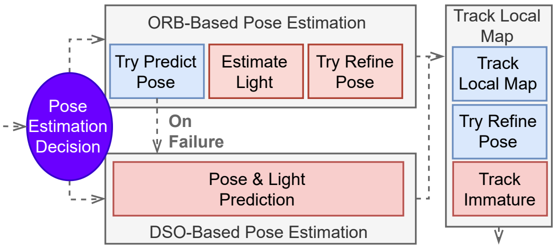


Figure 4: Mixed pose estimation.

3.1.2 Information Transmission

The map is mixed, as it can store, use and transmit information independently of their representation. It ensures the consistent storage and transmission of the information.

DSO and ORB-SLAM2 share and reuse their data through this map. They have common frame structure and information (including the camera pose). Optionally, DSO points can be used in ORB part, and ORB points can be used in DSO part.

3.2 Mixed Pose Estimation

The mixed pose estimator represented in the figure 4 works in 3 steps: deciding the pose estimation method, estimating the pose, tracking and refining the local map.

3.2.1 Pose Estimation Decision

The role is to decide in advance which of the two pose methods will produce the less drift:

- An ORB pose prediction refined with DSO, then refined with ORB in the local map tracking.
- A DSO pose prediction refined with ORB in the local map tracking.

MOD SLAM use the covariance matrices of the last estimated poses to make the decision, as they are an indicator of the uncertainty of DSO and ORB-SLAM2 methods.

$$cov_{i|orb} = (J_i^\top J_i)^{-1} \qquad cov_{i|dso} = H_i^{-1} \tag{1}$$

where J_i is the last i^{th} ORB-SLAM2 pose estimator Jacobian, and H_i is the last i^{th} DSO pose estimator Hessian.

J_i and H_i are computed for the same pose, with points in the same coordinates system. Hence, $cov_{i|orb}$ and $cov_{i|dso}$ use the same quantity, and are comparable.

As ORB-SLAM2 and DSO have different representations of rotation, MOD SLAM uses the vector containing the variance of the translation of the pose $[x,y,z]$. The variance of x (noted $cov_{orb|x,x}$ and $cov_{dso|x,x}$) appears in the diagonal of the covariance matrices cov_{orb} and cov_{dso}:

$$\begin{aligned} u_{i|orb} &= [cov_{i|orb|x,x}; cov_{i|orb|y,y}; cov_{i|orb|z,z}] \\ u_{i|dso} &= [cov_{i|dso|x,x}; cov_{i|dso|y,y}; cov_{i|dso|z,z}] \end{aligned} \tag{2}$$

where u_{orb} and u_{dso} are the vectors containing the variance of the translation part of the pose. $||u_{orb}||$ and $||u_{dso}||$ are uncertainty estimates for ORB and DSO.

We observe that when $||u_{orb}||$ or $||u_{dso}||$ becomes high, it is likely to remain high in the next frames. For robustness, the covariance over N preceding keyframes is considered :

$$v_{orb} = \frac{1}{N}\sum_{i=1}^{N} u_{i|orb} \qquad v_{dso} = \frac{1}{N}\sum_{i=1}^{N} u_{i|dso} \tag{3}$$

The algorithm take a decision by comparing $||v_{orb}||$ and $||v_{dso}||$. We introduce a parameter w_{pe} as a weight which will influence the decision. A weight below one will favor DSO tracking, and above one will favor ORB tracking. (see the table 5 for our chosen value of the parameters).

$$decision = \begin{cases} orb, & \text{if } ||u_{orb}|| < w_{pe} \times ||u_{dso}|| \\ dso, & \text{otherwise} \end{cases} \tag{4}$$

3.2.2 DSO-based Pose Estimation

The DSO-Based Pose Estimator works unmodified. A batch of motion guesses is computed by perturbing in different ways the motion of the previous frame. For each, a photometric pose estimation is done. The estimator will keep the pose with the lowest photometric RMSE.

This pose estimation is followed by an optional ORB refinement, inside the local map tracking (Section 3.2.4). Therefore, this tracking method is still mixed.

On most of the images, this yields fast a precise pose estimation. However, the estimation is not always the best, especially on video like the Kitti dataset where the photometric error of the tracked points is often high.

3.2.3 ORB-based Pose Estimation

The ORB-Based Pose Estimation is a combination of the ORB-SLAM2 pose estimation, and DSO pose optimization:

1. First, the SLAM will try to track with a constant velocity motion model using only ORB points,
2. If the number of tracked points is too low, try to track with a standard PnP model.

3. If the number of tracked points is still below a threshold, ORB tracking is considered a failure, and it falls back to a DSO-Based Pose Prediction.
4. Then MOD-SLAM tries to refine with DSO, by minimizing the photometric error using only the DSO points. The refined camera is accepted only if the ratio of inliers points r_{dso} (with a photometric error below a threshold, Table 5) is low enough. In all cases, DSO refinement will compute and store the light change.

As observed in section 3.2.2, DSO Pose estimator does not always return the best result. Still, when DSO finds a good initial motion, it generally results in a more precise camera position than ORB. The idea behind our ORB-Based Pose Prediction is to feed the DSO pose optimizer directly a good motion found by ORB, to find the most precise camera pose.

When the DSO refinement is not accepted, the assumption is that DSO was too weak, and that it would have added a lot of drift. When ORB failed, the SLAM switches to a DSO-based pose estimation, because of the ability of the DSO coarse pose estimator to almost always find a pose, with a decent quality despite the drift.

3.2.4 Local Map Tracking

After each pose estimation, MOD SLAM tracks the ORB local map like a standard ORB-SLAM2. The active ORB 3D points are re-projected into the current frame, to make 2D-3D matches. New active ORB 3D points are computed, according to the previous neighbor frames. Then, the algorithm can choose between just computing the covariance of the ORB-SLAM2 pose estimator, or refining by indirectly optimizing the pose of the camera. When DSO was not used to refine or track, ORB based refinement is launched, as the result is likely to improve. Otherwise, MOD SLAM uses the inlier ratio of photometric points r_{orb} (Table 5) as an indicator of the quality of the DSO pose estimator. If it is higher than a threshold, MOD SLAM refines with ORB, else computes only the covariance matrix of ORB.

3.3 Mixed Keyframe Creation and Bundle Adjustment

The mixed keyframe creation and bundle adjustment choice pipeline is represented in the figure 5.

3.3.1 Bundle Adjustment Decision

Bundle Adjustment decision has a huge impact on the drift.

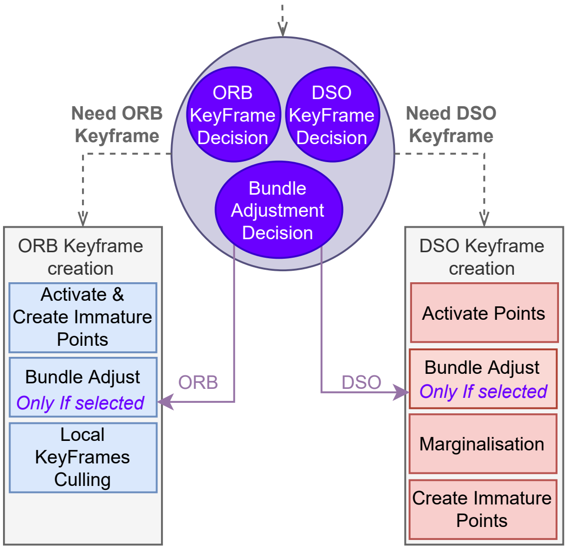


Figure 5: Mixed Keyframe creation and Bundle Adjustment.

1. If the number of points tracked by ORB is too low, only DSO can give a good result, even if it can be also weak.
2. If the ratio of photometric outliers points higher than a fixed threshold r_{ba}, ORB is favored.
3. The bundle adjustment method is chosen by comparing the number of inliers tracked by DSO and by ORB. We introduce a weight w_{ba} which will influence this decision (see the table 5 for the parameters).

3.3.2 Keyframe Creation

Two different pipelines are used to create ORB-SLAM2 and DSO key frame. If a DSO bundle adjustment is requested, MOD SLAM try first to create a DSO key frame, then a ORB-SLAM2 key frame. And if an ORB-SLAM2 bundle adjustment is requested, MOD SLAM tries first to create an ORB-SLAM2 key frame, then a DSO key frame.

ORB-SLAM2 and DSO key frames are created based on their own criteria: for ORB-SLAM2, it is based on the number of tracked points, whereas for DSO it is based on the optical flow.

Let t_n and a_n be the number of tracked ORB points and active ORB 3D points at time n. The original ORB condition for keyframe creation is $a_n < t_n \times 0.9$ wheras MOD SLAM condition is $t_n < exp(log(a_n) \times 0.975)$. This gives almost the same result for a low number of points, but MOD SLAM condition produces fewer keyframes for a high number of points. High movement of the pose or high residual will produce a DSO keyframe.

DSO key frames creation remains unchanged. ORB-SLAM2 key frame creation keeps the same method, but use the same principle of immature point as DSO, with a stricter new point filtering, based on the covariance of the coordinates of the point.

During the keyframe creation, only the B.A. of the chosen type will be run. Therefore, it is possible to choose a B.A. of a type, but to not choose to create a keyframe of this type. In this case, no B.A. will be executed. DSO and ORB have their own local map. It means that DSO will bundle adjust its own local map and ORB its own local map.

ORB bundle adjustment consists of a Levenberg-Marquardt optimization of the reprojection error of the active map. DSO bundle adjustment consists of a Gauss-Newton optimization of the photometric error on a window of 6 frames with their points.

4 RESULTS

Our SLAM library and MOD SLAM source code used to obtain the results below are ready to be released upon publication of the article.

4.1 Parameters

The table 5 represent the parameters that we choose for MOD SLAM, compared to those of ORB-SLAM2 and DSO. Note that these parameters have not been strongly tuned. They have been chosen as a compromise with speed and accuracy. The original ORB-SLAM2 implementation uses 2000 points for the KITTI dataset, and 1000 points for the TUM dataset, whereas we use 1250 for both. The original DSO has two modes: the normal mode with a desired density of 2000 points (used for the evaluation), and a fast mode with a desired density of 800 points. We choose to use the same parameters as the fast mode of DSO.

4.2 Results and Performance

KITTI. KITTI Odometry dataset (Geiger et al., 2012) is made of multiple stereo car sequences, from 1 to 8 minutes, taken with a resolution of 1241x376 at 10 fps. Only the grayscale left images of the stereo pair have been used for the evaluation. The table 1 shows the result on the KITTI dataset. In most cases, we get the best of ORB-SLAM2 and DSO. We observed for some videos a significant reduction of the drift. Some trajectories have been plotted in the figure 6. Even if our SLAM is a mix of ORB and DSO, each computation is never done twice in an indirect or a direct manner. On an i9-10900 with 20 threads, the SLAM needs 5% of CPU usage to run in real-time on KITTI dataset, and only 2 GB of RAM with the biggest video of KITTI comprising 4541 frames.

Table 1: Absolute Trajectory Error on KITTI datasets (Geiger et al., 2012). The last line is the number of times a method has given the best result. ORB-SLAM2 has been evaluated without loop closure, so we obtained different results from the original paper, but7 almost the same result as CNN-SVO (Loo et al., 2019).

	Ours	ORB.	DSO
00	105m	**67m**	114m
01	**12m**	x	x
02	**43m**	**43m**	120m
03	1.8m	**1.0m**	2.1m
04	1.1m	**0.9m**	1.5m
05	**40m**	43m	52m
06	**44m**	49m	59m
07	**16m**	17m	17m
08	**54m**	58m	111m
09	**39m**	60m	63m
10	11m	**9m**	16m
Best	7	6	0

TUM. TUM Monocular Visual Odometry Dataset (Engel et al., 2016) is made of multiple grayscale monocular interior sequences. The images have been captured with a resolution of 1280x1024 reduced to 640x480 at runtime, at 30 fps. The dataset contains precise internal and photometric calibration. ORB-SLAM2 failed in almost half of the videos, whereas DSO and MOD SLAM ran successfully on all the videos. The results between DSO and MOD SLAM are tight. MOD SLAM could not improve more the result of DSO as it already gave almost perfect results. On an i9-10900 (10 cores, 20 threads, 2.8 to 5.2 GHz), the SLAM needs 20% of CPU usage to run in real-time on TUM dataset.

Smartphone Videos. Using a Google Pixel 3a camera, we have filmed perfect loops inside a room, at a resolution of 1920x1080 downsampled to 640x360. The camera was calibrated with the OpenCV library (Bradski, 2000) and a chessboard. We have run the three SLAM methods on each video to evaluate the drift generated by each method. The results are plotted in the figure 7. ORB-SLAM2 got lost on all the videos. Observe how the loop closes: MOD SLAM and DSO get almost the same result, excepting for some videos where we can observe a small improvement for MOD SLAM.

Robustness. Table 2 shows that MOD SLAM is the only method that can run on all the videos, without changing any parameters, while still keeping the accuracy of ORB on KITTI, and the density of the generated point cloud of DSO on TUM dataset (Figure 1).

Table 2: Number of success on TUM Monocular Dataset (Engel et al., 2016), KITTI dataset (Geiger et al., 2012), and our own smartphone videos.

		Ours	ORB.	DSO
TUM	50	50 (100%)	28 (56%)	50 (100%)
KITTI	11	11 (100%)	10 (91%)	10 (91%)
P3	5	5 (100%)	0 (0%)	5 (100%)
Total	66	**66 (100%)**	38 (58%)	65 (98%)

4.3 Ablation Study

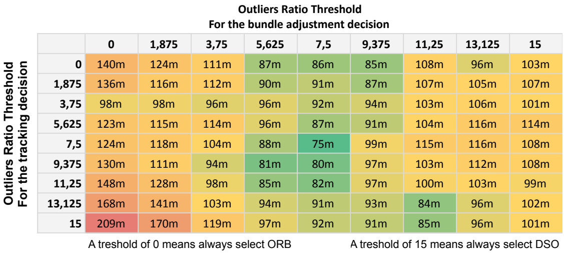
Table 3: Decision effect in our MOD SLAM methods. Each value is an average of the absolute trajectory error in the KITTI dataset with a Forced ORB/Forced DSO on Mixed Tracking/Mapping decision. An empty case means that the ATE is averaged with all the possibilities (ORB / DSO / Mixed).

Tracking	Mapping	Averaged ATE on KITTI
ORB		62m
	ORB	65m
DSO		46m
	DSO	52m
Mixed		39m
	Mixed	**36m**

MOD-SLAM uses multiple decision systems to predict which of ORB-SLAM2 and DSO will perform best. To make these decisions in advance, it is necessary to evaluate the robustness of ORB-SLAM2 and DSO. SLAM internal measurement, such as the covariance matrices, or the number of inliers, can be tested to find the most robust method. The table 3 evaluates the impact of the decision of MOD SLAM on the final result. We have tested to favor ORB-SLAM2 then DSO during the tracking decision, and the mapping decision. We averaged the absolute trajectory error on the KITTI dataset in the table 3. We observe clearly that both tracking and bundle adjustment conditions have a big impact on the final results.

Table 4 is an example of parameter study. It shows the impact of an outlier ratio threshold during the tracking (vertical axis) and the bundle adjustment (horizontal axis). A threshold of 0 means that the SLAM always selects ORB, and a threshold of 15 means that the SLAM always selects DSO. The SLAM seems to perform better with an in-between threshold of 7.5, suggesting that the SLAM gives better result in a mixed manner.

Table 4: ATE w.r.t. DSO outliers ratio threshold for tracking and BA decision on KITTI 08 (meters).



Outliers Ratio Threshold
For the bundle adjustment decision

Outliers Ratio Threshold
For the tracking decision

	0	1,875	3,75	5,625	7,5	9,375	11,25	13,125	15
0	140m	124m	111m	87m	86m	85m	108m	96m	103m
1,875	136m	116m	112m	90m	91m	87m	107m	105m	107m
3,75	98m	98m	96m	96m	92m	94m	103m	106m	101m
5,625	123m	115m	114m	96m	87m	91m	104m	116m	114m
7,5	124m	118m	104m	88m	75m	99m	115m	116m	108m
9,375	130m	111m	94m	81m	80m	97m	103m	112m	108m
11,25	148m	128m	98m	85m	82m	97m	100m	103m	99m
13,125	168m	141m	103m	94m	91m	93m	84m	96m	102m
15	209m	170m	119m	97m	92m	91m	85m	96m	101m

A treshold of 0 means always select ORB

A treshold of 15 means always select DSO

4.4 Discussion

ORB-SLAM2 has a lot of trouble handling the TUM dataset and our videos. Weird camera movement may be responsible, for instance when the camera goes under a desktop table on the TUM dataset. Also some images do not contain enough reliable ORB points, as the filmed surfaces are too flat. On the contrary, DSO gives the worst result on KITTI dataset, probably because the photometric optimization has trouble handling the big motion and the noisy images.

MOD SLAM did not fail on any video, contrary to ORB-SLAM2 and DSO, despite using the same parameter set for all the datasets (which have not been strongly tuned). Hence, it proves the robustness of MOD SLAM.

The method (Lee and Civera, 2018) mentioned in the introduction runs a full version of DSO and ORB-SLAM2, side by side, with two separate maps. It allows having the loop closure and BA system of ORB-SLAM2 with DSO, in the cost of multiplying by 2 the computational CPU and RAM resources. But it shows no robustness and no drift improvement on videos without a loop as one method is always privileged. On the other hand, MOD SLAM uses one unique map, and has to choose the tracking method at each frame (Figure 2). The computational resources do not increase: a refinement by the complementary method (ORB-SLAM2 or DSO) is done, which is fast since the pose is already close to the solution. Contrary to (Lee and Civera, 2018), MOD SLAM has proven to be more robust and to yield lower drift.

5 CONCLUSION

Our flexible map has permitted the creation of a mixed ORB-SLAM2/DSO architecture. We have shown with the result on the KITTI dataset (Table 1) and on our own dataset the potential of MOD SLAM compared to the original ORB-SLAM2 and DSO: on most videos, MOD SLAM succeeds in giving the best of the two methods. Observe that on KITTI 01, MOD

Table 5: Parameters used for the evaluation. N.A. = Non-applicable.

Method	Ours	DSO	ORB
Point Density			
Number of ORB Corner	1250	N.A.	2000
Desired DSO Point Density	800	2000	N.A.
Number of DSO Corner	600	1500	N.A.
DSO Bundle Adjustment			
Iteration	4	6	N.A.
Optimization Window Size	6	7	N.A.
ORB Bundle Adjustment			
Iteration	5	N.A.	0 to 5
Refinement Iteration	0	N.A.	0 to 10
Pose Estimation Decision (Section 3.2.1)			
Covariance weight w_{pe}	0.75	N.A.	N.A.
Average Window Size N	7	N.A.	N.A.
Pose Estimation and refinement (Section 3.2.3 and Section 3.2.4)			
Inliers Ratio r_{dso}	0.85	N.A.	N.A.
Inliers Ratio r_{orb}	0.85	N.A.	N.A.
Bundle Adjustment Decision (Section 3.3.1)			
Outliers Ratio r_{ba}	0.15	N.A.	N.A.
Number of point weight w_{ba}	0.0125	N.A.	N.A.

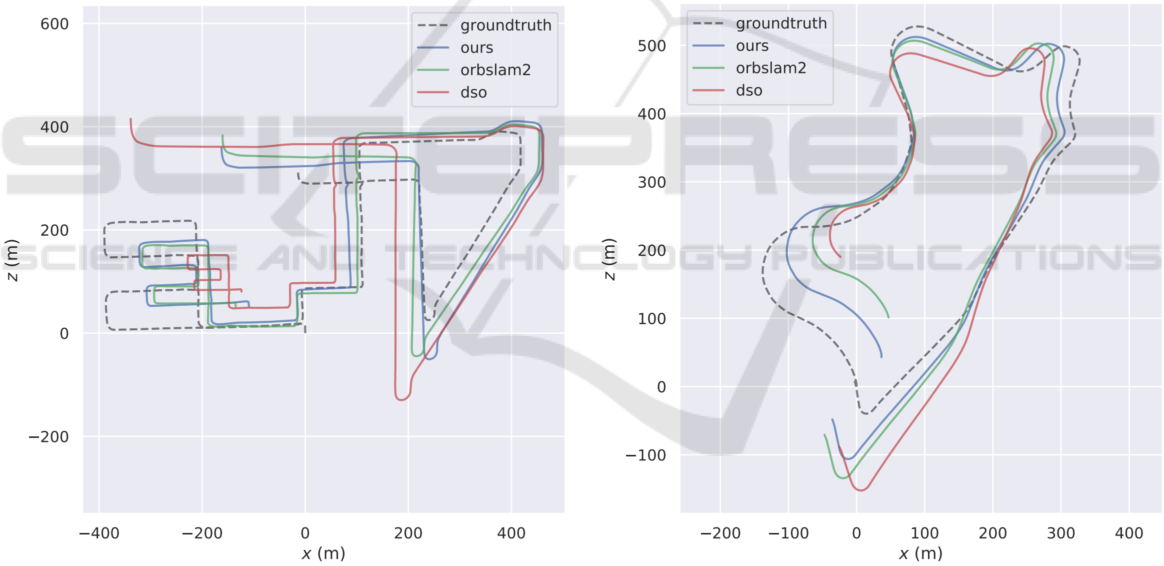


Figure 6: Kitti (Geiger et al., 2012) Trajectories 08 and 09.

SLAM is the only method able to process the full video, and that on KITTI 09 the error is drastically lower.

A finer decision function would have the potential to still lower the drift. Therefore, our future works include to work on the decisions functions.

ACKNOWLEDGMENTS

We thank Pierre-Alain Langlois for assistance with using OpenMP, Ceres, and other C++ libraries; Abderahmane Bedouhene for comments that greatly improved the paper; and Thomas Daumain for double checking the results. This work has been supported by the project UrbanVision. Initiated in 2018, UrbanVision is a project of the I-SITE FUTURE, a French initiative to answer the challenges of sustainable city.

REFERENCES

Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Chahine, G. and Pradalier, C. (2018). Survey of monocular SLAM algorithms in natural environments. In *2018 15th Conference on Computer and Robot Vision (CRV)*, pages 345–352.

Civera, J., Davison, A. J., and Montiel, J. M. (2008). Inverse depth parametrization for monocular SLAM. *IEEE transactions on robotics*, 24(5):932–945.

Engel, J., Koltun, V., and Cremers, D. (2017). Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625.

Engel, J., Usenko, V., and Cremers, D. (2016). A photometrically calibrated benchmark for monocular visual odometry. In *arXiv:1607.02555*.

Ferrera, M., Eudes, A., Moras, J., Sanfourche, M., and Besnerais, G. L. (2021). OV^2SLAM : A fully online and versatile visual slam for real-time applications.

Forster, C., Pizzoli, M., and Scaramuzza, D. (2014). SVO: Fast semi-direct monocular visual odometry. In *2014 IEEE international conference on robotics and automation (ICRA)*, pages 15–22. IEEE.

Forster, C., Zhang, Z., Gassner, M., Werlberger, M., and Scaramuzza, D. (2017). SVO: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics*, 33(2):249–265.

Gálvez-López, D. and Tardós, J. D. (2012). Bags of binary words for fast place recognition in image sequences. *IEEE Transactions on Robotics*, 28(5):1188–1197.

Gao, X., Wang, R., Demmel, N., and Cremers, D. (2018). LDSO: Direct sparse odometry with loop closure. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2198–2204. IEEE.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*.

Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *2007 6th IEEE and ACM international symposium on mixed and augmented reality*, pages 225–234. IEEE.

Lee, S. H. and Civera, J. (2018). Loosely-coupled semi-direct monocular ”slam”. *IEEE Robotics and Automation Letters*, 4(2):399–406.

Loo, S. Y., Amiri, A. J., Mashohor, S., Tang, S. H., and Zhang, H. (2019). CNN-SVO: Improving the mapping in semi-direct visual odometry using single-image depth prediction. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5218–5223. IEEE.

Lothe, P., Bourgeois, S., Dekeyser, F., Royer, E., and Dhome, M. (2009). Towards geographical referencing of monocular SLAM reconstruction using 3d city models: Application to real-time accurate vision-based localization. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2882–2889.

Mur-Artal, R., Montiel, J. M. M., and Tardos, J. D. (2015). ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics*, 31(5):1147–1163.

Mur-Artal, R. and Tardós, J. D. (2017). ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262.

Newcombe, R. A., Lovegrove, S. J., and Davison, A. J. (2011). DTAM: Dense tracking and mapping in real-time. In *2011 international conference on computer vision*, pages 2320–2327. IEEE.

Singandhupe, A. and La, H. M. (2019). A review of SLAM techniques and security in autonomous driving. In *2019 Third IEEE International Conference on Robotic Computing (IRC)*, pages 602–607.

Strasdat, H., Montiel, J., and Davison, A. J. (2010). Scale drift-aware large scale monocular SLAM. *Robotics: Science and Systems VI*, 2(3):7.

von Stumberg, L., Usenko, V., Engel, J., Stückler, J., and Cremers, D. (2017). From monocular SLAM to autonomous drone exploration. In *2017 European Conference on Mobile Robots (ECMR)*, pages 1–8.

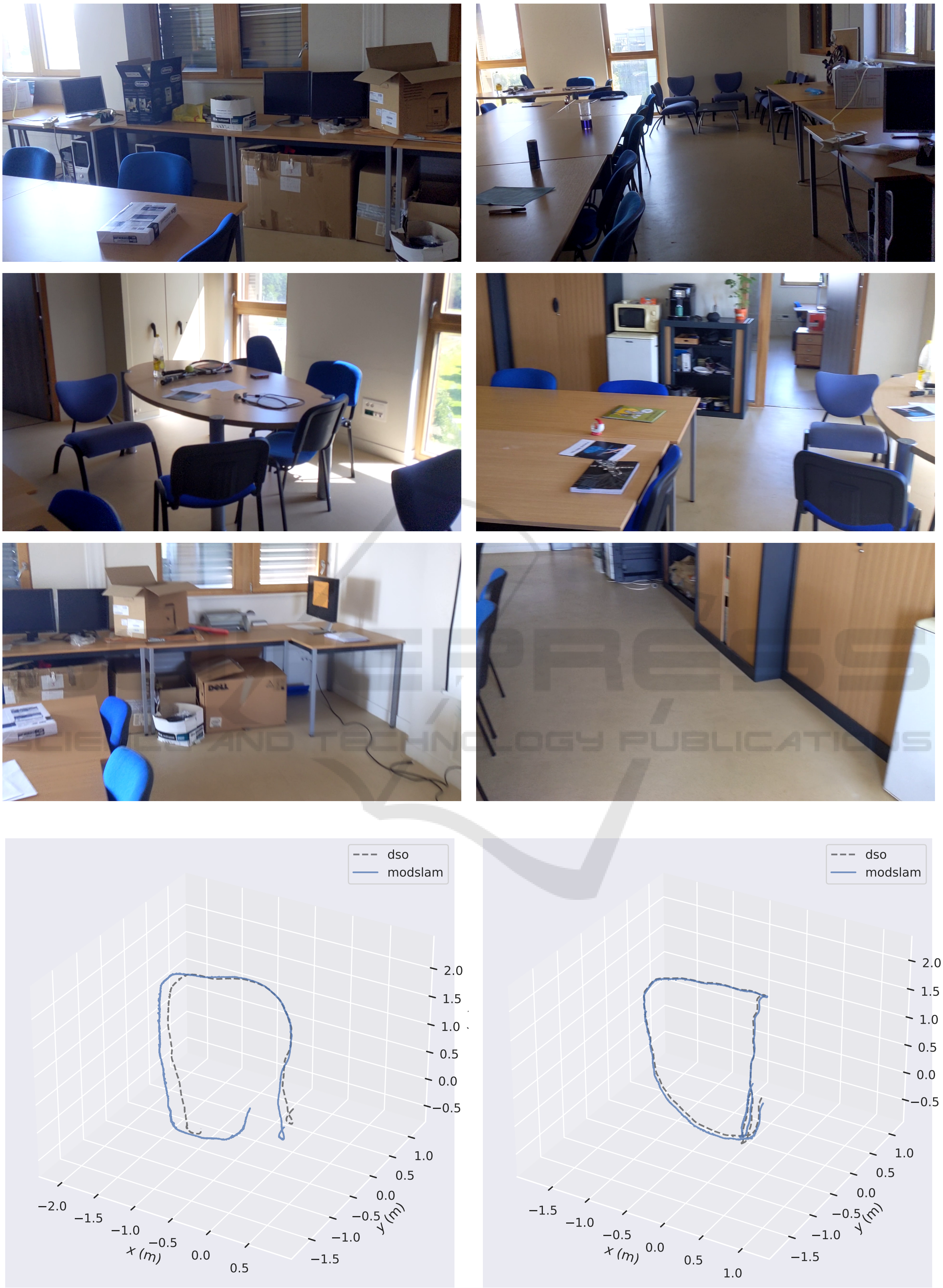


Figure 7: Trajectories of videos taken with a Google Pixel 3a. The videos end at the same place they started. The closer the two trajectory ends, the lower the drift. ORB-SLAM2 got lost on all the videos.