Feature-based Initialization for Monocular Direct Visual Odometry

Mariia Gladkova

Advisor: Nikolaus Demmel
Supervisor: Prof. Dr. Daniel Cremers

Technical University of Munich

Contents

Abstract

Direct Sparse Odometry (DSO) [1] is a state-of-art approach that proposes direct and sparse solution to the monocular-based Visual Odometry (VO). In this work we have looked into its initialization part and investigated a possibility to incorporate methods from indirect VO solutions into its framework. In particular, we implemented an initialization module that utilizes features and computes relative camera transformation based on their correspondences analogous to ORB-SLAM work [2]. We further extended the solution by augmenting the extracted features and estimating the depth map based on the probabilistic approach formulated in [3]. A lot of attention was addressed to the optimization and map refinement by means of bundle adjustment and outlier filtering. Evaluations based on EuRoC [4], KITTI [5] and CARLA [6] image sequences have shown that the feature-based initialization outperforms the original approach proposed by DSO authors and increases overall robustness and accuracy.

1. Introduction

Among the core tasks of many autonomous systems like cars and unmanned aerial vehicles (UAVs) lie exploration of the environment (mapping) and positioning (localization) in the constructed map, jointly called Simultaneous Localization and Mapping (SLAM). Since the position of a vehicle depends on the map of the environment it navigates in, and, at the same time, the map is constructed based on the information about the agent's position, SLAM is considered a "chicken and egg" problem. The method for solving this problem, that we are investigating in this project, is monocular-based Visual Odometry (VO), real-time estimation of camera motion from a sequence of images obtained from a single camera. Although cameras are cheap, versatile and lightweight sensors, they impose a number of limitations in the environments with insufficient illumination, dynamic scenes or lack of visual cues (texture) [7]. Moreover, monocular-based VO suffers from scale uncertainty [5].

Direct Sparse Odometry (DSO) [1] proposes a robust and accurate VO method, which performs a novel sparse sampling of points with sufficient image gradient, thus reducing computational complexity of dense approaches, yet preserving the fine-grained 3D geometry [8]. Although DSO shows state-of-art performance with the image sequences from TUM Mono dataset [9], we have noticed that the initialization struggles for KITTI image sequences [5], where the depth range and camera motion are relatively large.

The goal of the project is to implement a robust and accurate initialization module for DSO system. In particular, we aim to implement a feature-based initializer proposed in ORB-SLAM system [10], populate its sparse map with the technique proposed in [3], optimize the estimations using robustified bundle adjustment and evaluate the results against the original method. Since initialization plays an important role in accuracy and convergence of the algorithm, it is anticipated that this module can increase the robustness and practical applicability of the algorithm and outperform the current solution, which uses uniform 3D structure initialization.

The rest of the report is organized as following: in section 2 ORB-SLAM and DSO initialization solutions are briefly discussed. Section 3 describes the approach implemented in the scope of the project, whereas in section 4 evaluation results of different metrics are presented, where the proposed module has been compared with the original solution and a closely-related work from the Master thesis by Xingwei Qu. Section 5 concludes the report with the remarks about future work.

2. Related Work

A lot of recent work in Computer Vision and Robotics communities has been devoted to design and implementation of robust and fast visual SLAM solutions. A resemblance between visual odometry methods and incremental Structure-from-Motion (SfM) has been observed, indicating the importance of good initialization for further map and pose optimizations [11]. In this section we describe two initialization solutions proposed by ORB-SLAM and DSO systems, which offer background to the work conducted in the scope of this project.

2.1. ORB-SLAM

ORB-SLAM [10] is a feature-based monocular SLAM system capable of real-time operation in versatile environments ranging from indoor to outdoor, hand-held to car-recorded image sequences. Moreover, it offers automatic map initialization, which attracts by its simplicity and effectiveness as demonstrated in [10]. Initialization process begins with the ORB feature extraction in the reference frame and feature matching in the target frame, which is followed by a parallel computation of two models, homography and fundamental matrix, to offer the relation between the correspondences. Model is selected based on its score and the proposed heuristic. Since we use this approach in our work, we describe the computation and selection in detail in section 3. Matrices are further decomposed into motion hypotheses and 3D positions of extracted feature points are computed via triangulation. Valid camera pose is selected based

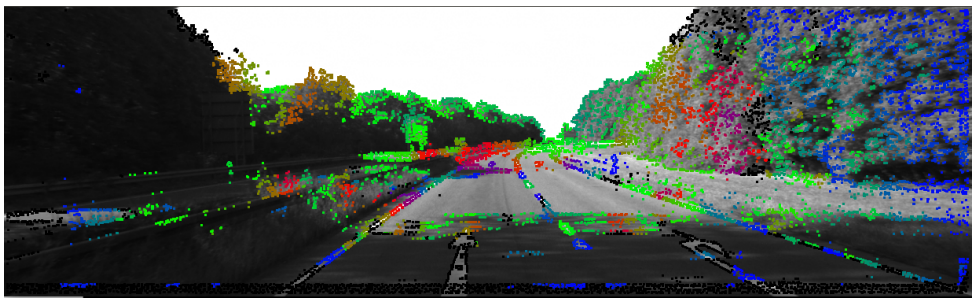

Figure 1: Example of inverse depth map initialization with KITTI sequence 01 (red patches indicate close to camera points, blue - far away ones).

on the number of points that appear in front of the camera, have sufficient parallax and small reprojection error values. More detailed description is available in section 4 of the paper [10].

2.2. Direct Sparse Odometry (DSO)

The basis of this work, Direct Sparse Odometry (DSO), offers a direct and sparse solution to monocular-based visual odometry. It relies on the photometric error optimization and enables generation of a more complete representation of the scene without additional abstraction to 2D keypoints or corner-based features [1]. The initialization starts with the region-based point selection, where an image is divided into sub-blocks and the adaptive region-based thresholds are computed. Points with sufficient gradient are selected for future optimizations, their initial inverse depths set to 1. Camera pose and inverse depths values are refined using Gauss-Newton optimization in a coarse-to-fine fashion [1]. Initialization is considered successful if after the last optimization the reprojection error of all candidate points is sufficiently low, otherwise the initialization is re-started.
Although the system has shown successful performance on a wide range of image sequences [1] and it has been used as a state-of-art approach in evaluation of many direct and indirect SLAM solutions, we have noticed that its initialization module cannot always guarantee fast-convergence and optimal solution of optimizations units. For instance, DSO requires to process many frames and undergo 2-3 attempts before successfully initializating the system in case of several outdoor sequences offered by KITTI dataset. As it can be seen in figure 1, the depth initialization can be very inconsistent and erroneous, which can further lead to non-convergence of the optimization pipeline and system failure. The importance of the initialization step has served as a motivation for the current work, which is described in section 3.

3. Methodology

The project idea lies in the implementation and integration of a new module that is responsible for the initialization of 3D structure and 2 keyframes, where the first one is always an identity transformation. Since the design of a new unit is based on ORB-SLAM initialization, in the scope of this work we call it ORB Initializer, whereas the original module is named Coarse Initializer after its class name in (DSO code implementation).

3.1. Set-up routine

As soon as the system is launched, the ORB Initializer receives the reference frame from the caller. Since we follow feature-based approach, feature points are extracted from the reference frame and tracked in the following N frames using Lucas-Kanade optical flow approach ($N = 4$ has been experimentally chosen). In this work we consider features based on the most prominent corners in the image [12]. The reference frame is reset if the number of tracked features drops

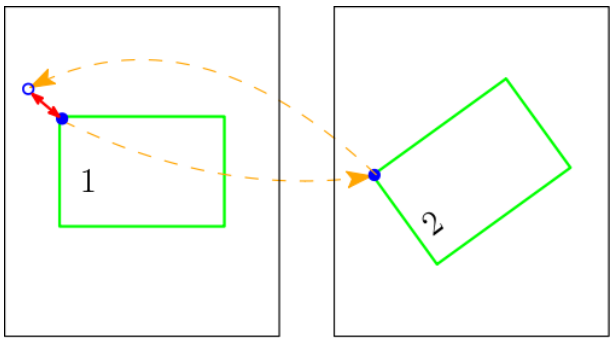

Figure 2: Example of a feature point tracking $1 \rightarrow 2 \rightarrow 1$

below a predefined threshold. For feature handling (extraction and tracking) we utilize functions from OpenCV library[13], in particular, *goodFeaturesToTrack* and *calcOpticalFlowPyrLK* methods.
The points are tracked in a frame-to-frame fashion, starting from the extraction in the reference image. To improve robustness of the feature matching we look only for the points that can be successfully tracked in both directions: from frame 1 to frame 2 and, back, from 2 to 1, and consider only those features which pixels' positions in the image 1 land in the close vicinity to the original value after completion of a tracking-loop (i.e. $1 \rightarrow 2 \rightarrow 1$). Figure 2 demonstrates the aforementioned concept.

3.2. Computation of a relative pose and geometric bundle adjustment

After tracking N - 1 frames we attempt to establish a relation between the features from the reference image and their matches in the Nth frame. Here we directly follow the approach proposed in ORB-SLAM by computing the homography (H sub-index) and fundamental matrix (F sub-index) models and selecting the winner based on the heuristic (eq. 1).

$$\text{Model} = \begin{cases} \text{Homography} & \frac{S_H}{S_H+S_F} > 0.4 \\ \text{Fundamental Matrix} & else \end{cases} \tag{1}$$

S_H and S_F are model scores, which are computed according to eq. 2, where d^2_{cr} and d^2_{rc} are symmetric transfer errors (eq. 4)[14], N - number of points, $T_H = 5.99$, $T_F = 3.84$ and $\Gamma = T_H$.

$$S_M = \sum_{i=0}^{N} \left(\rho_M(d^2_{cr}(\mathbf{x}^i_c, \mathbf{x}^i_r)) + \rho_M(d^2_{rc}(\mathbf{x}^i_c, \mathbf{x}^i_r)) \right) \tag{2}$$

$$\rho_M(d^2) = \begin{cases} \Gamma - d^2 & \text{if } d^2 < T_M \\ 0 & else \end{cases} \tag{3}$$

$$d_{cr}(\mathbf{x}_c, \mathbf{x}_r) \coloneqq \begin{cases} \mathbf{x}_c - \pi \cdot H \cdot \begin{pmatrix} \mathbf{x}_r \\ 1 \end{pmatrix} & \text{if } M = H \\ \frac{1}{||\text{epline}||^2} \cdot \left(\begin{pmatrix} \mathbf{x}_c \\ 1 \end{pmatrix}^T \cdot F \cdot \begin{pmatrix} \mathbf{x}_r \\ 1 \end{pmatrix} \right) & \text{if } M = F \end{cases} \tag{4}$$

$$d_{rc}(\mathbf{x}_r, \mathbf{x}_c) \coloneqq \begin{cases} \mathbf{x}_r - \pi \cdot H^{-1} \cdot \begin{pmatrix} \mathbf{x}_c \\ 1 \end{pmatrix} & \text{if } M = H \\ \frac{1}{||\text{epline}||^2} \cdot \left(\begin{pmatrix} \mathbf{x}_r \\ 1 \end{pmatrix}^T \cdot F^T \cdot \begin{pmatrix} \mathbf{x}_c \\ 1 \end{pmatrix} \right) & \text{if } M = F \end{cases} \tag{5}$$

It is expected that the model based on homography matrix is able to explain the relations of points that lie on (nearly) a plane or have low parallax, whereas a model based on fundamental

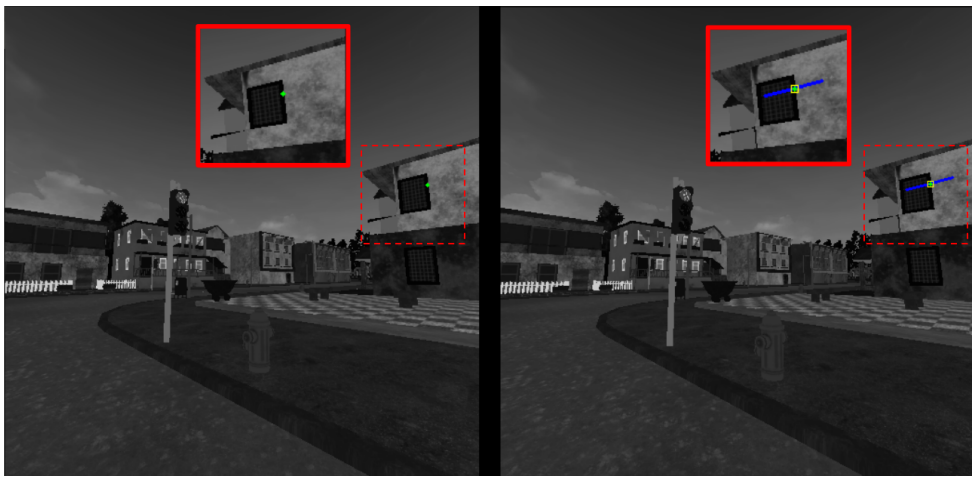

Figure 3: Epipolar line search example with image sequence from carla dataset (left: reference frame, right: current frame). Epipolar line is depicted in blue, matched point is emphasized with yellow border.

matrix should take care of other spatial distributions [10].
After selecting the model, the motion hypotheses are retrieved. We utilize *decomposeHomographyMat* function from OpenCV library to obtain the transformation proposals. In case of a fundamental matrix model we first recover the essential matrix with $E = K^T F K$ and decompose it with *decomposeEssentialMat* from OpenCV library. To select the most-likely camera relative pose we adopt the approach proposed in the original ORB-SLAM paper and look for the hypothesis that is supported by the most number of points that lie in front of the camera and have sufficient parallax [10]. Geometric bundle adjustment is further performed to refine the 3D structure and camera poses for every frame that has been used to track the features. This way we aim to establish the baseline for further steps of structure population and motion optimization described in the next subsections 3.3 and 3.4. Bundle adjustment is followed by outlier removal, where points with large reprojection error (aka "severe" outliers) are removed. We iterate with the optimization and outlier removal scheme until all severe outliers are removed. After all such points are removed we check for "normal" outliers, i.e. points that have a reprojection error less than 3 pixels, and eliminate them too. This way we make sure that the poses are well-refined and points are reliable for further steps based on the photometric information.

3.3. Semi-dense reconstruction using epipolar line search

To improve the initialization quality and bridge the gap between our indirect initialization module and the direct system, we extract additional points in the same way the PointSelector in DSO works [1]. Initially, the region-based thresholds are computed by splitting the reference image into 32×32 blocks as $\bar{t}_i + T$, where $\bar{t}_i$ is the median absolute gradient in i-th block and $T = 7$. The selection is performed in 6×6 image patches by choosing the pixel points with maximum absolute gradient value in the patch that exceeds the corresponding region threshold. After obtaining additional points we seek to reconstruct their 3D position following the approach proposed in [3]. We adopt the probabilistic inverse depth map representation and the way of refining the values by propagating them in a frame-to-frame fashion. Depth map of the feature points serves as a prior distribution.
Based on the estimated camera transformations we can compute the equations for an epipolar line and search for point correspondences using Sum of Square Differences (SSD) cost measure over a window of pixels. The point with the lowest SSD cost is considered as a match (see figure 3). Having the best matching position (i.e. disparity λ) we can compute the inverse depth d by considering two types of errors: a geometric error $\sigma_{\lambda(\xi,\pi)}$ caused by noise on relative orientation ξ and projection function π together with a photometric disparity error $\sigma_{\lambda(I)}$. The computation of the geometric error follows eq. 6, where l is an epipolar line direction, g is image

gradient (normalized), σ_l - standard deviation of the Gaussian-distributed noise ϵ_l imposed on the epipolar line l, which is approximated by the norm of the absolute image gradient at the matching point position.

$$\sigma^2_{\lambda(\xi,\pi)} = \frac{\sigma_l^2}{\langle g, l\rangle^2} \tag{6}$$

Photometric error can be computed according to eq. 7, where g_p is an image gradient along the epipolar line, σ_i^2 - variance of the image intensity noise, which is taken as a constant parameter and equals to 4· camera_pixel_noise with camera_pixel_noise = 10.

$$\sigma^2_{\lambda(I)} = \frac{2\sigma_i^2}{g_p^2} \tag{7}$$

According to [3] the observational variance of the inverse depth becomes

$$\sigma_o^2 = \alpha^2 \left(\sigma^2_{\lambda(\xi,\pi)} + \sigma^2_{\lambda(I)}\right) \tag{8}$$

where α represents a pixel to inverse depth ratio and can be computed as

$$\alpha = \frac{\delta_d}{\delta_\lambda} \tag{9}$$

with δ_d being the length of a searched inverse depth interval and δ_λ - the corresponding length of the searched epipolar line. Please refer to the original paper [3] for the derivations of the aforementioned equations.
Since the inverse depth has a probabilistic nature, we assume that every pixel can be represented by a Gaussian distribution, which is updated from frame to frame via multiplication of a prior and a noisy observation distributions:

$$\begin{aligned} P(D_p|D_o) &\propto P(D_o|D_p)P(D_p) = \mathcal{N}(d_o, \sigma_o)\mathcal{N}(d_p, \sigma_p) \\ &\propto \mathcal{N}\left(\frac{\sigma_p^2 d_o + \sigma_o^2 d_p}{\sigma_p^2 + \sigma_o^2}, \frac{\sigma_p^2\sigma_o^2}{\sigma_p^2 + \sigma_o^2}\right). \end{aligned}$$

Observational inverse depth d_o is computed via triangulation and σ_o^2 is computed as in eq. 8.

3.4. Photometric Bundle Adjustment and Outlier Removal

As an additional step to refine the poses and the 3D structure we have included photometric bundle adjustment (PBA) step. The main idea behind PBA lies in the photo-consistency assumption, i.e. the pixel value should remain similar when we reproject it from one image frame to another. This gives rise to a residual r_i for 3D point x_i, which is defined as the difference of the pixel brightness between image 1 and image 2 that is warped with relative transformation ξ_{12}.

$$r_i(\xi_{12}, \mathbf{x}_i) = I_2(\pi(\xi_{12}\mathbf{x}_i)) - I_1(\pi(\mathbf{x}_i)) \tag{10}$$

Similarly to [1], the optimization problem is defined as a weighted Sum of Squared Differences (SSD) over all points $\mathbf{x}_i \in \mathcal{P}$ hosted in the reference keyframe and observed in frame $j \in obs(\mathbf{x}_i)$. Moreover, each point is represented by a small patch of pixels $\Delta \in \mathcal{N}(\mathbf{p}_i)$, where point's projection serves as a central pixel $\mathbf{p}_i$ (figure 4). The latter is extracted from the reference frame I and projected to the target frame I_j (eq. 11).

$$\min_{\{\xi_j\}_{j=1\ldots|\mathcal{C}|}, \{\mathbf{x}_i\}_{i=1\ldots|\mathcal{P}|}} \sum_{i=1}^{|\mathcal{P}|} \sum_{j\in obs(\mathbf{x}_i)} \sum_{\Delta\in\mathcal{N}(\mathbf{p}_i)} ||I_j(\pi(\xi_j(\mathbf{x}_i + \pi^{-1}\Delta))) - \psi I(\pi\mathbf{x}_i + \Delta)||_\gamma \tag{11}$$

Optimization is performed over camera poses ξ_j and inverse depth values of each point $\mathbf{x}$. To account for the outliers, Huber norm $||\cdot||_\gamma$ is used. Moreover, since the mean intensity values of

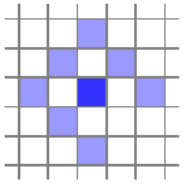

Figure 4: Neighborhood pattern used to compute the objective cost in PBA ([1]).

the patches in two images can differ, we normalize the values with an additional factor ψ, which corresponds to eq. 12. In case a part of the patch lies outside the image frame, the values on the border are repeated.

$$\psi = \frac{\sum_{\Delta \in \mathcal{N}(\mathbf{p})} I_j(\pi \mathbf{T}_j(\mathbf{x} + \pi^{-1}\Delta))}{\sum_{\Delta \in \mathcal{N}(\mathbf{x})} I(\pi \mathbf{x} + \Delta)} \tag{12}$$

If we look into the optimization (eq. 11) from the probabilistic perspective and represent the residuals (eq. 10) as a distribution, then, according to [15, 16], adopting the assumption of the Gaussian nature of such distribution, where very low and high residuals are very unlikely, will not portrait the real data with many outliers very well. Nevertheless, it has been shown in [15] that fitted t-distribution with its heavy tails matches better the residual distribution (figure 5). In comparison with two parameters, mean μ and variance σ^2, of a Gaussian distribution,

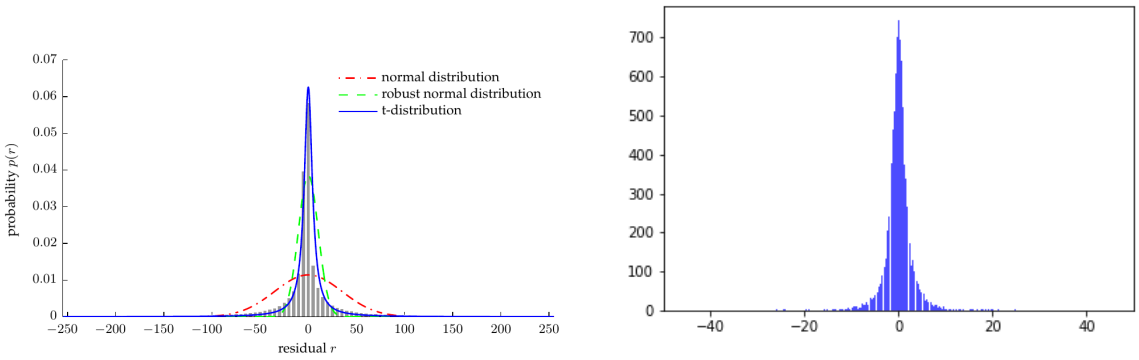

Figure 5: Left: probability density functions fitted to accumulated residual histogram (fr2/desk sequence) [16], right: residual histogram example (MH_03_medium sequence).

t-distribution is parameterized by degrees of freedom ν which value we assign to 5 as in [15]. In our experiments we first filter the residuals assuming the underlying normal distribution with zero mean and standard deviation as defined in eq. 13 using the Median Absolute Deviation (MAD) estimator, where $c = 1.4826$ [16].

$$\sigma_{MAD}(\mathbf{x}) = c \cdot \text{median}(|\mathbf{x} - \text{median}(\mathbf{x})|) \tag{13}$$

The pixels with standard deviation $|\sigma| < 3 \cdot \sigma_{MAD}$ are used to fit the distribution. The "fitting" is performed by iterative approximation of the scale parameter (until its convergence) based on eq. 14 from [16]. Each point that is observed in an image contributes 8 residuals (patch pattern as in figure 4).

$$\sigma^2_{k+1} = \frac{1}{n} \sum_{i}^{n} \frac{\nu + 1}{\nu + r_i^2/\sigma_k^2} r_i^2 \tag{14}$$

It should be noted that we fit t-distribution in every image of our initialization sequence (i.e. from the reference to the current frame). Alternatively, one can collect the residuals from all frames and approximate only one distribution. As it is mentioned in section 4, we have not observed significant differences in the quality of estimations between fitting one or one-per-frame

distributions.
As the last step, we would like to use the scale parameters and filter out the residuals that lie outside the 90% percentile of the kth image fitted distribution, i.e. $|r_i| < 2.571 \cdot \sigma_k$[1]. The observation of a point in kth frame is removed if more than half of the contributed residuals are discarded during the fitting. A 3D point is removed if it is observed in less than two frames.
In our pipeline, the t-distribution fitting is performed after the first photometric bundle adjustment. The filtered 3D pointcloud together with the camera poses are then passed to PBA for a final refinement.

3.5. Method overview

To summarize, a high-level representation of the underlying process is presented in algorithm 1. Please note the comments regarding ORBInitializer v1 - v3 for the discussion in section 4.

Algorithm 1: trackFrame routine in ORBInitializer

```
keypoints_new ← ∅;
# set-up and subroutine explanations in 3.1
num_successful ← trackFeatures(keypoints_prev_frame, keypoints_new);
if num_successful < MIN_NUM_MATCHES then
    map.clear();
    resetRefFrame();
    return FAILURE;
map.updateCameraInformation(keypoints_new, frame);
if |frame.frame_id - ref_frame_id| ≥ MIN_NUM_FRAMES then
    # as described in 3.2
    success ← map.computeRelativeTransform(frame, ref_frame);
    if success then
        do
            map.performGeometricBA();
            severeOutliers ← map.removeSevereOutliers();
        while severeOutliers > 0;
        # as described in 3.3
        map.populateStructure3D();
        # until here → ORBInitializer v1
        do
            map.performGeometricBA();
            severeOutliers ← map.removeSevereOutliers();
        while severeOutliers > 0;
        # as descibed in 3.4
        if map.performPhotometricBA() then
            # until here → ORBInitializer v2
            map.removeOutliers();
            map.performPhotometricBA();
            # until here → ORBInitializer v3
        return SUCCESS;
return FAILURE;
```

4. Evaluation

The evaluation has been done using image sequences and odometry ground truth from 2 datasets: EuRoC [4] and KITTI [17]. Since there is no ground truth for the reconstructed 3d maps, we

[1]The factor of 2.571 is taken from the table of common t-distribution values (link to the table)

also considered a sequence generated from CARLA simulator [6]. For every image sequence we performed 5 forward and 5 backward runs, where we recorded an initial pose that is provided by an initializer (i.e. not considering its future optimization by DSO BA), number of frames taken for the initialization and, if applicable, 3d pointcloud. In addition, all keyframe poses are recorded in order to observe the impact of the initialization on the whole system. To test the robustness of the initialization pipeline each forward and backward pass has a different starting frame.
To illustrate the development of our ORBInitializer and evaluate the impact of each additional component as described in section 3, we additionally present the results from ORBInitializer v1 (plotted in yellow line), v2 (blue line) and v3 (green line) compared with results from the original CoarseInitializer (red line). Firstly, we performed only geometric bundle adjustment of tracked features and sampling of the additional points (ORBInitializer v1). Secondly, we extended ORBInitializer v1 with the photometric bundle adjustment of all points (ORBInitializer v2). Finally, we added fitting of the t-distribution into residuals to the second version with the subsequent photometric bundle adjustment (ORBInitializer v3).
For evaluations with image sequences from KITTI we have asked Xingwei Qu to provide the results of the version of ORBInitializer from his Master thesis (plotted in magenta). In his work Xingwei extracts ORB features in the reference frame and matches them in the target frame, or, if there are not enough matching pairs, performs optical flow to find the correspondences. Afterwards, similar to our approach, he finds a relative transformation between reference and target frames by, firstly, computing the homography and fundamental matrices in parallel and then decomposing the one which model has the most number of inliers. Based on the transformation he populates the extracted points and refines their depth by epipolar line search. In case the system fails on any of the steps, the initialization is restarted from the beginning and a new pair of frames is probed.

4.1. Metrics

Firstly, we looked into relative pose error (RPE) for the keyframe pose obtained from an initializer. For this purpose, the relative pose error RPE can be defined as in eq. 15

$$E_{i,j} = (Q_i^{-1} Q_j)^{-1} (P_i^{-1} P_j) \tag{15}$$

where Q_i and Q_j are ground truth global poses, P_i and P_j - estimated global poses [18]. Moreover, for EuRoC dataset the poses are linearly interpolated, since not all the image timestamps have a corresponding ground truth pose.
From the relative error (eq. 15) we extracted the translational part (eq. 16) and considered all pose pairs with $\Delta = 1$. According to [18] it is sufficient to evaluate only translational part of the transformation, since the rotational error is correlated.

$$E_e^t = ||transl(E_e)||_2^2 \tag{16}$$

ATE metric is defined in eq.18, where n being the number of keyframes, F_i (eq. 17) is the absolute trajectory error after their Sim(3) alignment.

$$F_i = Q_i^{-1} S P_i \tag{17}$$

$$ATE_{1...n} = \left(\frac{1}{n} \sum_{i=1}^{n} ||transl(F_i)||_2^2 \right)^{\frac{1}{2}} \tag{18}$$

As for the evaluation of the RPE, we evaluated the translational error as proposed in [18] and defined in eq. 19.

$$RPE_{1...n} = \left(\frac{1}{n-1} \sum_{i=1}^{n-1} ||transl(E_{i,i+1})||_2^2 \right)^{\frac{1}{2}} \tag{19}$$

For all results we normalized the error by the distance traveled, therefore all values are represented per meter.

4.2. CARLA

For evaluation an image sequence created using CARLA simulator [6] has been used. The visualization of the trajectory and pointcloud produced by DSO can be seen in figure 6. On

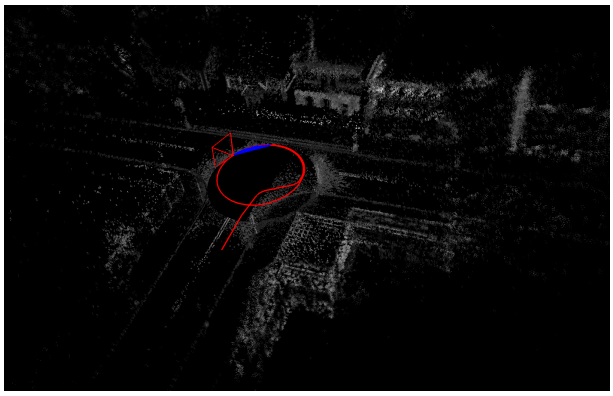

Figure 6: Visualization of CARLA sequence used for evaluation.

average, ORB Initializer required 4.25 frames for initialization whereas Coarse Initializer used 7.87 frames. Regarding the provided initial pose and map, ORB Initializer has shown a superior performance in the first case and nearly similar results in the second one (figure 7). The drop in depth map accuracy can be caused by fitting t-distribution into the point residuals and removal of the good ones, which can have a negative effect on the bundle adjustment and optimization convergence (figure 8). The improvement in the depth map does not seem to impact the quality of the initial pose and ORB Initializer v2 achieves the best compromise for these metrics (figure 8).

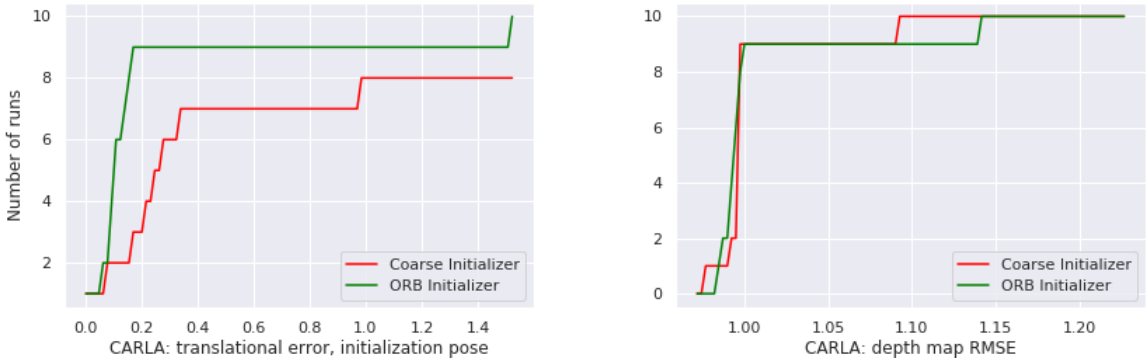

Figure 7: Left: cumulative plot for initial keyframe pose, right: initial 3d map.

Based on the ATE and RPE metrics our initializer shows an improvement in the overall robustness and accuracy (figure 9). Comparing different versions of ORB Initializer (figure 10) we can spot a bad run that affects the results of all of them (due to their sequential dependency). In that case the optimization for GBA converges to a non-optimal solution and many outliers are generated, which is nevertheless slightly improved in the next steps of PBA and outlier removal.

4.3. KITTI

First, we present the results collected for all KITTI image sequences that have ground truth data (i.e. 00 - 10). From our observations KITTI has been a challenging dataset with its fast movement, dynamic objects and, sometimes, lack of visual cues (e.g. sequence 01 due to its highway scene dominance). On average, it took around 4.78 frames for ORB Initializer to perform map initialization, whereas Coarse Initializer required 7.3 frames. Concerning the initialization

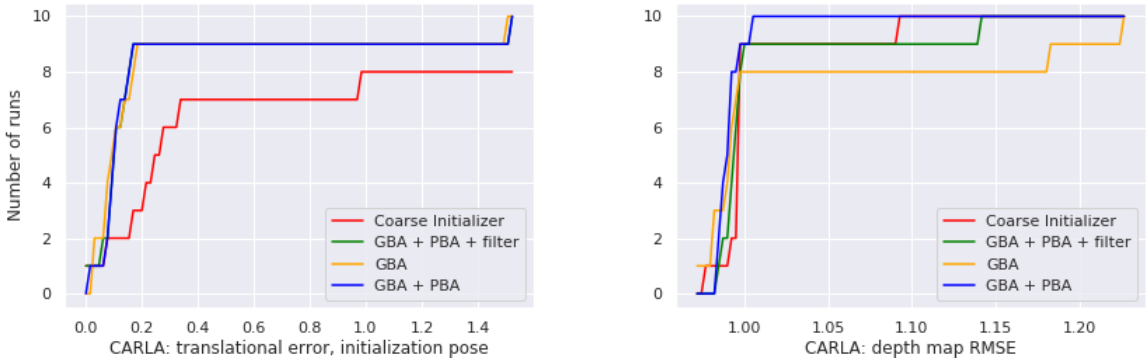

Figure 8: Left: cumulative plot for initial keyframe pose $[m^{-1}]$, right: initial 3d map.

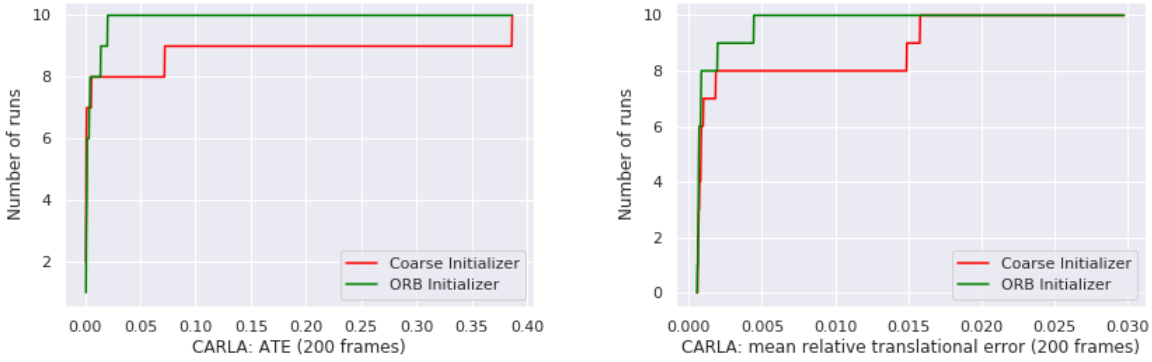

Figure 9: CARLA: cumulative plot for trajectory error (over 200 frames) $[m^{-1}]$.

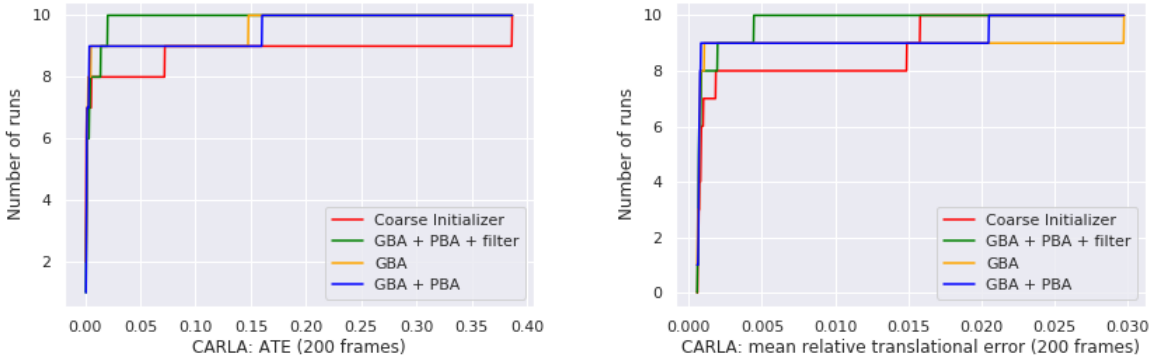

Figure 10: CARLA: cumulative plot for trajectory error, ORB Initializer versions comparison (over 200 frames) $[m^{-1}]$.

pose ORB Initializer performs significantly better than Coarse Initializer, which persists for all its versions 11. This way we can confirm our original project's incentive that feature-based approach even in its simplest version (ORB Initializer v1) can improve the performance by a significant margin.

We also improved the performance according to ATE and RPE metrics 12. If we look into different versions of our initializer (figure 13), we can observe that the PBA ones dominate against GBA-only, latter being slightly better for the initial pose estimation. This can indicate that despite a loss in the pose accuracy map refinement with PBA and outlier removal have a positive impact on the robustness and overall accuracy.

Additionally, we performed a comparison with Xingwei's work over ATE metric and chose the sequences 00 - 02 (see figure 14) for evaluation.

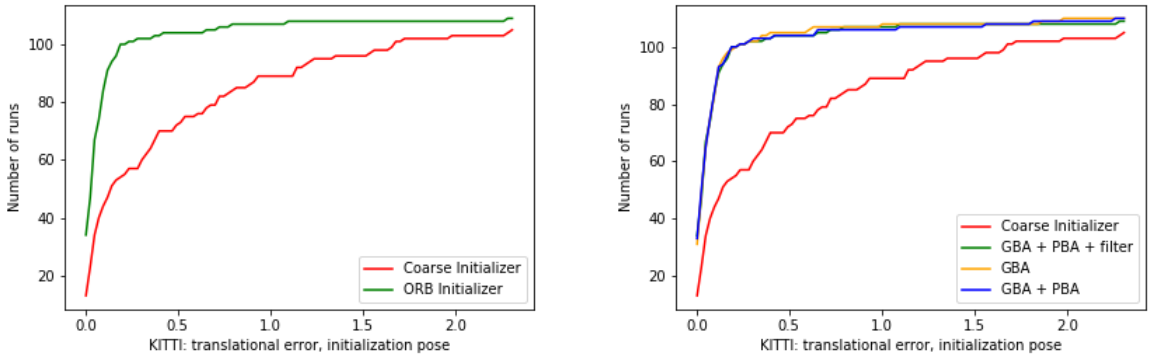

Figure 11: KITTI: cumulative plot for initial keyframe pose $[m^{-1}]$.

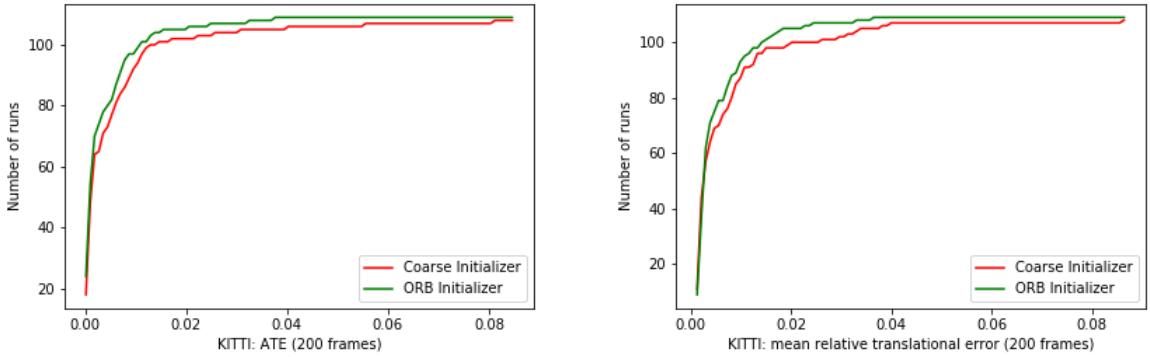

Figure 12: KITTI: cumulative plot for trajectory errors (over 200 frames) $[m^{-1}]$.

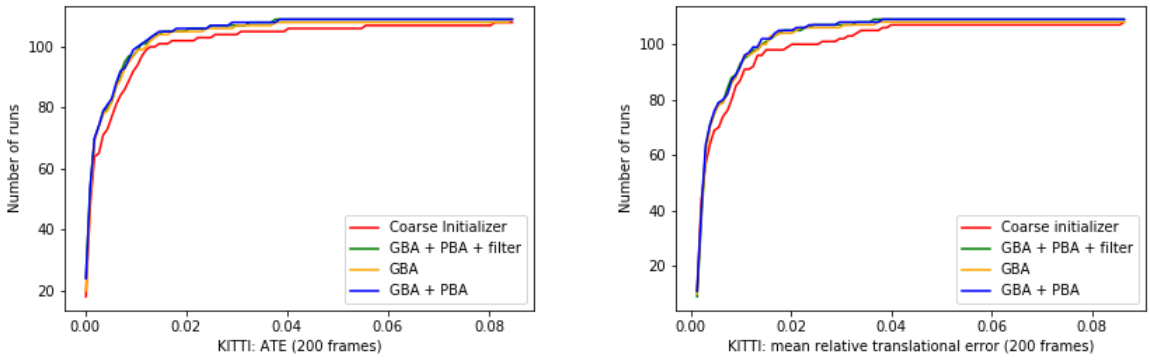

Figure 13: KITTI: cumulative plot for trajectory errors, ORB Initializer versions comparison (over 200 frames) $[m^{-1}]$.

Figure 14: KITTI dataset, frame 000023 (left: sequence 00, right: 02).

Regarding the number of frames used, Xingwei's solution has shown on average initialization of 12 frames. Our initializer was able to initialize the system within 5 frames, whereas original

Coarse Initializer took 7 frames on average. We also prove a better overall accuracy with respect to both of the versions (figure 15).

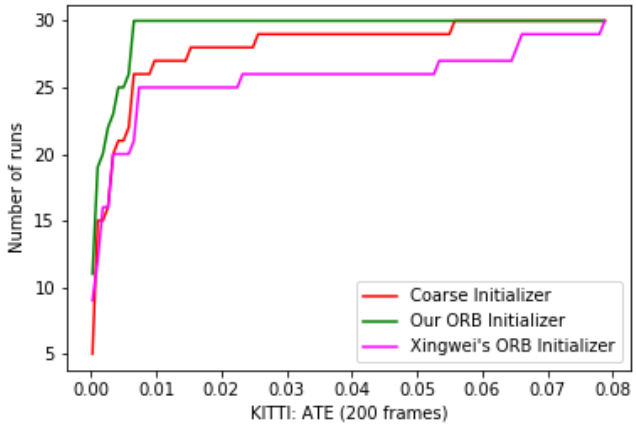

Figure 15: KITTI: cumulative plots for ATE $[m^{-1}]$.

4.4. EuRoC

For evaluations with EuRoC dataset we considered all image sequences: MH_O1_easy, MH_O2_easy, MH_O3_medium, MH_O4_difficult, MH_O5_difficult, V1_O1_easy, V1_O2_medium, V1_O3_difficult, V2_O1_easy, V2_O2_medium, V2_O3_difficult.
EuRoC dataset has posed another challenge for the initialization module, with jerky moves and long pauses. Although we tried to test the robustness by starting at different points of an image sequence, we avoided the no-motion moments. Overall, it has taken on average 5 frames for ORB Initializer and 7 for Coarse Initializer. From figure 16 we can see that the performance of

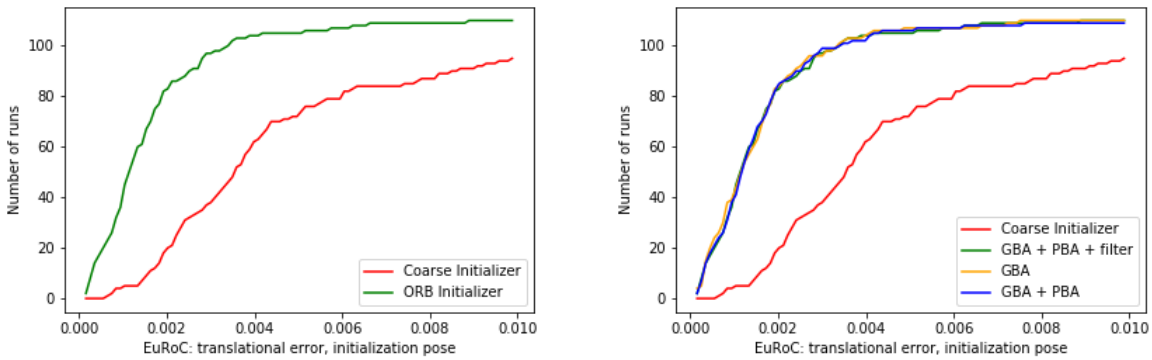

Figure 16: EuRoC: cumulative plots for initial keyframe pose $[m^{-1}]$.

ORB Initializer outperforms Coarse Initializer (figures 19 and 20). From the comparison of ORB Initializer versions (figure 20) one cannot see a clear improvement after inclusion of PBA and outlier removal steps. When looking into all frames (not only keyframes) we have observed that the gap between the original and our initializer has narrowed down, but now we can observe the positive impact of additional optimizations, which restates the importance of these refinements.

5. Conclusion

In the scope of this work we looked into an alternative way of initializing DSO system using an indirect method. Our initialization approach was inspired by ORB-SLAM work [2] and it

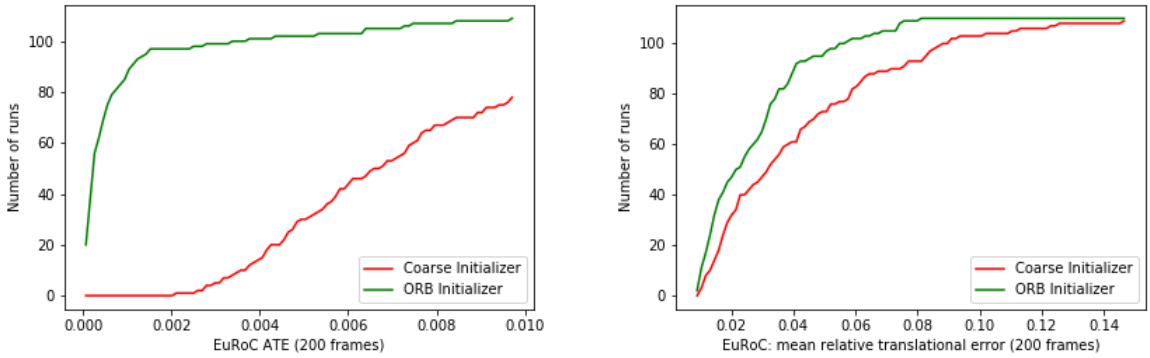

Figure 17: EuRoC: cumulative plot for trajectory errors (only keyframes) $[m^{-1}]$.

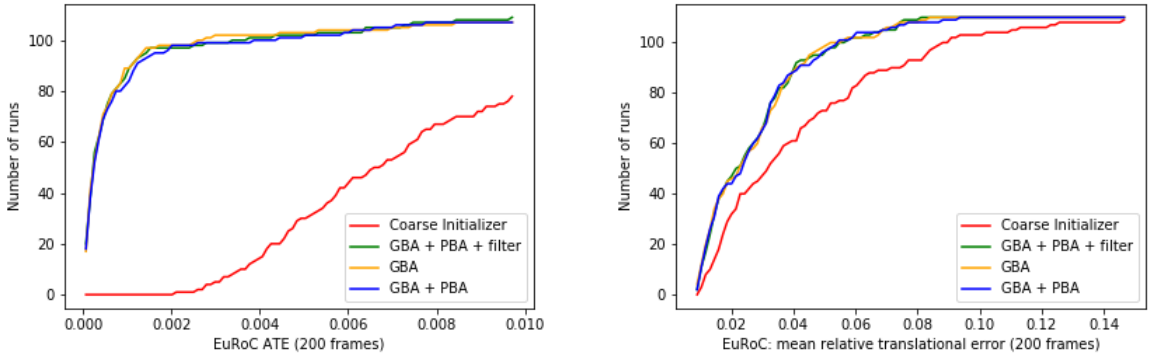

Figure 18: EuRoC: cumulative plot for trajectory errors, ORB Initializer versions comparison (only keyframes) $[m^{-1}]$.

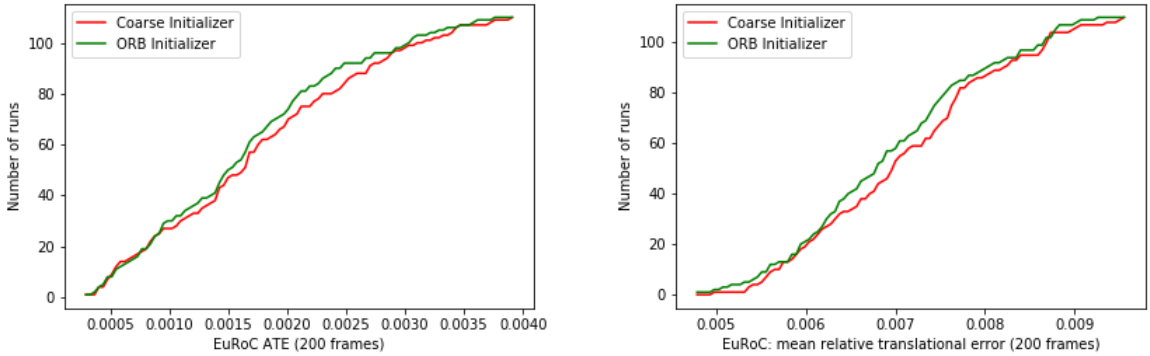

Figure 19: EuRoC: cumulative plot for trajectory errors (over 200 frames) $[m^{-1}]$.

was further changed and extended with the photometric-based methods like epipolar line search and photometric bundle adjustment. We also looked into a probabilistic approach of filtering outliers by fitting t-distribution into the photometric residuals. From our extensive evaluations on EuRoC and KITTI datasets we have confirmed that indirect initialization for direct system has a positive effect on the overall performance of the DSO system.

As a perspective future work one can definitely try to speed-up the initialization process without influencing its accuracy. Although it takes, on average, 4 - 5 frames for initialization, all the optimizations are very computationally expensive and require several seconds to process the data, which is not desirable in the system with real-time requirements. Moreover, one can look into the ways of estimating the affine light parameters within our framework or integrate learned features like SuperPoint [19] that have shown to be reliable and robust to changes in

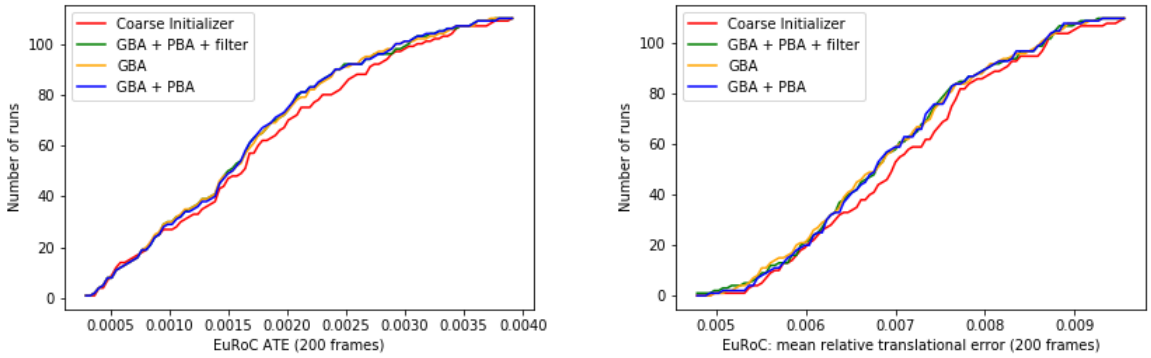

Figure 20: EuRoC: cumulative plot for trajectory errors, ORB Initializer versions comparison (over 200 frames) $[m^{-1}]$.

light conditions and scale.

References

[1] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[2] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[3] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE international conference on computer vision*, pages 1449–1456, 2013.

[4] Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[5] Bernd Manfred Kitt, Joern Rehder, Andrew D Chambers, Miriam Schonbein, Henning Lategahn, and Sanjiv Singh. Monocular visual odometry using a planar road model to solve scale ambiguity. 2011.

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.

[7] Davide Scaramuzza and Friedrich Fraundorfer. Tutorial: visual odometry. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[8] Nan Yang, Rui Wang, and Daniel Cremers. Feature-based or direct: An evaluation of monocular visual odometry. *arXiv preprint arXiv:1705.04300*, 2017.

[9] J. Engel, V. Usenko, and D. Cremers. A photometrically calibrated benchmark for monocular visual odometry. In *arXiv:1607.02555*, July 2016.

[10] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[11] Chengzhou Tang, Oliver Wang, and Ping Tan. Gslam: Initialization-robust monocular visual slam via global structure-from-motion. In *2017 International Conference on 3D Vision (3DV)*, pages 155–164. IEEE, 2017.

[12] Jianbo Shi et al. Good features to track. In *1994 Proceedings of IEEE conference on computer vision and pattern recognition*, pages 593–600. IEEE, 1994.

[13] Gary Bradski and Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.", 2008.

[14] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[15] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for rgb-d cameras. In *2013 IEEE international conference on robotics and automation*, pages 3748–3754. IEEE, 2013.

[16] C. Kerl. Odometry from rgb-d cameras for autonomous quadrocopters. Master's thesis, Technical University Munich, Germany, Nov. 2012.

[17] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[18] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580. IEEE, 2012.

[19] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 224–236, 2018.