IEEE Transactions on Robotics (T-RO) paper, presented at ICRA 2023, London, UK. Cite as T-RO paper.

SOFT2: Stereo Visual Odometry for Road Vehicles based on a Point-to-Epipolar-Line Metric

Igor Cvišić, Ivan Marković and Ivan Petrović*

***Abstract*—Accurate localization constitutes a fundamental building block of any autonomous system. In this paper, we focus on stereo cameras and present a novel approach, dubbed SOFT2, that is currently the highest-ranking algorithm on the KITTI scoreboard. SOFT2 relies on the constraints imposed by the epipolar geometry and kinematics, i.e., it is developed for configurations that cannot exhibit pure rotation. We minimize point-to-epipolar-line distances, which makes the approach resilient to object depth uncertainty, and as the first step, we estimate motion up to scale using just a single camera. Then, we propose to jointly estimate the absolute scale and the extrinsic rotation of the second camera in order to alleviate the effects of varying stereo rig extrinsics. Finally, we smooth the motion estimates in a temporal window of frames by using the proposed epipolar line bundle adjustment procedure. We also introduce a multiple hypothesis feature matching approach for self-similar planar surfaces that accounts for appearance change due to perspective. We evaluate SOFT2 and compare it to ORB-SLAM2, OV2SLAM, and VINS-FUSION on the KITTI-360 dataset, KITTI train sequences, Málaga Urban dataset, Oxford Robotics Car dataset, and Multivehicle Stereo Event Camera dataset.**

***Index Terms*—Stereo visual odometry, road vehicle localization, point-to-epipolar-line metric, online calibration.**

I. Introduction

Localization constitutes a fundamental building block of any autonomous system. This is especially emphasized for autonomous vehicles that participate in urban traffic and need to maintain highly accurate estimates of their pose for navigation purposes. The localization can rely on proprioceptive sensors, like the wheel odometry and inertial measurement units, and exteroceptive sensors like cameras, laser range sensors and even radars. Indeed, localization of autonomous vehicles typically relies on the fusion of most of the aforementioned sensors, but, nevertheless, each modality should operate as accurately as possible to produce a reliably functioning autonomous system.

Visual localization is one of the key actors in modern localization systems for mobile robots and autonomous vehicles. Cameras offer a rich source of information at a fraction of the price of other competitive hardware. When it comes to estimating relative pose between two calibrated camera views, a reference approach boils down to the algebraic 5-point method [1] in combination with RANSAC [2]. An iterative approach that can estimate the relative pose between two views from an arbitrary number of points (but 5 minimum) was presented in [3], where it was also shown to be approximately twice as fast as the 5-point method. It is also an established results that minimizing reprojection errors in the image domain leads to better results since uncertainties are isometric in the image domain with respect to the 3D uncertainties of triangulated points from stereo setup [1]. Furthermore, it has also been previously shown that minimizing point-to-line distances produces greater robustness in odometry [4]. It is also worth noting that when a rigidness criteria is optimized on features with Gaussian noise from two relative views, the result suggests that it is optimal to minimize the point-to-epipolar-line metric [5].

All this makes cameras an attractive sensor modality and visual localization has been a subject of research for several decades now [1], [4], [6]. However, a unique recipe for successful visual odometry is difficult to define, thus multiple groups of approaches have been developed over the years. For example, one could tackle the problem by finding a rich set of features and track them over the frames to compute the odometry [7]–[11]. Another approach would be to disregard features and work directly on pixel intensities [12], [13], or even try to combine the best of both worlds [14], [15]. Furthermore, by assuming that the camera is placed on a specific vehicle and has limited maneuverability due to kinematic constraints, it is possible to further increase the performance of algorithms [16]. All of the aforementioned approaches have their benefits and drawbacks and for some further insights, comparison of visual odometry approaches for flying robots can be found in [17]. When cameras are combined in pairs, in the so-called stereo setup, they can also provide depth of the scene jointly with the standard image. Even though we are recently witnessing advent of deep monocular methods that can also estimate depth [18]–[20], stereo odometry for mobile robots and autonomous vehicles still offers many challenges and researchers are continuously striving to develop ever more accurate and reliable methods or leverage novel visual modalities [21]–[25]. This is an interesting notion despite the fact that to determine the relative pose between two views only 3 non-colinear 3D points are required and that lately a certifiable point cloud registration technique has been introduced [26].

An important factor in driving the process of obtaining accurate and robust stereo odometry was and still is the KITTI dataset [27], which has been acting as a public benchmark for road vehicles since 2012. Multiple stereo vision methods currently achieve less than 1% translation error [9], [22], [28]–[33], demonstrating the ability of cameras to produce

This work has been supported by the European Regional Development Fund under the grant KK.01.1.1.01.0009 (DATACROSS).

*Authors are with the University of Zagreb Faculty of Electrical Engineering and Computing, Laboratory for Autonomous Systems and Mobile Robotics, Croatia. {igor.cvisic, ivan.markovic, ivan.petrovic}@fer.hr.

highly accurate road vehicle trajectories. Besides the KITTI dataset, several other public datasets have been recorded by road vehicles with a suite of sensors operating in an urban environment, such as the KITTI-360 dataset [34], the Málaga Urban dataset [35], the Oxford Robotics Car dataset [36], and the Multivehicle Stereo Event Camera Dataset (MVSEC) [37] – all of which have been used in the present paper for experimental evaluation. Although other high quality automotive datasets are also publicly available [38]–[42], they are more oriented towards object detection, classification, semantic segmentation, and are either missing stereo cameras or ground truth trajectory for odometry evaluation.

In this paper we propose SOFT2, a highly accurate automotive stereo visual odometry designed for operation in real-time – the successor to our SOFT odometry [9], [22]. SOFT2 currently scores 0.53% in translation error and 0.0009 deg/m in rotation error rendering it currently the highest ranking algorithm on the KITTI scoreboard[1], over all sensor configurations. We introduce the following novelties: (i) relative pose estimation (up to scale) is based on minimizing the point-to-epipolar line distance from frame to frame in a single camera and also in a window of frames in a type of *epipolar line bundle adjustement*, (ii) we estimate the absolute scale jointly with the relative orientation of the second camera, thus performing online calibration at each frame, and (iii) we develop a multiple hypothesis feature matching method based on perspective correction for self-similar planar textures – this mostly affects features on the road and thus enhances translation estimation. To summarize, by focusing on the point-to-epipolar line distance minimization we obtain increased robustness to feature position uncertainty, lens radial distortion, and motion blur, while by maximally utilizing data from just a single camera, we account for possible variations in the stereo pair extrinsic parameters and rig inflextion during operation. SOFT2 omits 3D projections to avoid depth uncertainty and consequently does not offer mapping and loop closing – it is strictly an odometry algorithm. Finally, we also compare SOFT2 to ORB-SLAM2 [10], OV2SLAM [33], and VINS-FUSION [11], [43], on the KITTI-360 dataset, KITTI train sequences, Málaga Urban dataset, Oxford Robotics Car dataset, and Multivehicle Stereo Event Camera Dataset.

The paper is organized as follows. In Section II we present fundamentals of stereo vision and odometry to set the grounds for the paper. Section III then presents an overview of the proposed system to acquaint the reader with the whole picture of the SOFT2 pipeline. Then, Sections IV and V describe in details each of the modules and core paper contributions, i.e., our 2D-2D method that minimizes the point-to-epipolar line metric, scale and online extrinsic rotation calibration, and multiple hypothesis matching for self-similar planar surfaces, like the road, via perspective correction of feature appearance. Section VI presents experimental results and comparison to three competitive approaches on five different automotive datasets, while Section VII concludes the paper.

[1]http://www.cvlibs.net/datasets/kitti/eval_odometry.php

II. Stereo Visual Odometry Background

In this section we give a brief background on stereo vision and stereo odometry that we find necessary to setup the grounds for the proposed method. We assume that the reader is familiar with notions from the fields of computer vision and geometry of pairwise views, while for details on the subject we direct the reader to [44].

Stereo visual odometry relies on a pair of cameras to estimate the ego-motion, i.e., the accumulated relative displacement of the cameras. Mathematically, the displacement between the two views is defined by the translation vector $t \in \mathbb{R}^3$ and the rotation matrix $R \in \mathsf{SO}(3)$ that belongs to a special class of matrices called the special orthogonal group – matrices whose columns/rows form an orthonormal basis and have determinant of one. Together, the pair (R, t) forms the so-called special Euclidean group

$$\mathsf{SE}(3) = \left\{ \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \mid t \in \mathbb{R}^3, R \in \mathsf{SO}(3) \right\}, \tag{1}$$

and the final stereo pair trajectory is obtained by successive concatenation of the estimated relative displacements.

The stereo pair is set apart at a known distance, known as the baseline, while the orientation of the two cameras is targeted to be equal in order to make the camera image planes as parallel as possible. Unlike the monocular counterpart, the stereo setup, thanks to the known baseline, allows to resolve the scale and obtain a metric trajectory. Given that, besides the intrinsic calibration parameters, the extrinsic parameters of the stereo setup, namely the relative orientation and translation of one camera with respect to the other, need to be determined. This is usually performed in a standard fashion using a known calibration target, and toolboxes such as Kalibr [45] offer to do both simultaneously. However, although intrinsic and extrinsic calibration parameters appear as fixed design parameters, due to various influences and motion they can change during the operation, thus inevitably introducing additional errors to the odometry system.

Assuming that the stereo setup is calibrated, the standard approach in feature based stereo odometry proceeds by detecting and matching feature pairs in the stereo image. This approach is facilitated by the fact that the geometry of pairwise view sets the so called *epipolar constraint* for the feature pair. Let $x', x \in \mathbb{R}^3$ be the homogeneous coordinates of the same point P projected on the two camera image planes. Then, the two image points satisfy the epipolar constraint

$$x'^{\mathsf{T}} [t]_\times R\, x = 0, \tag{2}$$

where $[t]_\times \in \mathbb{R}^{3\times3}$ is the skew-symmetric matrix of the relative position vector and R is the relative orientation between the two views. Furthermore, t and R form together the epipolar matrix $E = [t]_\times R$, and the constraint is usually written as

$$x'^{\mathsf{T}} E x = 0. \tag{3}$$

Since for the calibrated stereo pair we know the relative pose of the cameras and their intrinsic parameters, we can use the

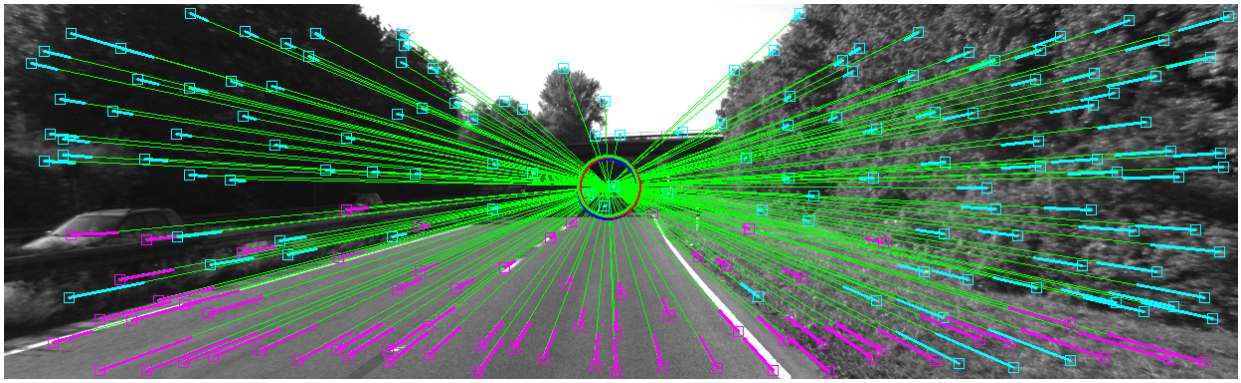


Fig. 1: Illustration of epipolar lines for the KITTI 01 sequence. The camera principal point (red) approximately coincides with the epipoles of the current (green) and previous (blue) frame – for better visualization, points are at the centers of the enlarged circles. Squares illustrate detected features in the current frame, while the tails connect them with matched features from the previous frame. Green lines represent epipolar lines associated to previous frame features. Color differentiates ground features from the other features. Our approaches minimizes the perpendicular distance of all features to the pertaining epipolar lines.

epipolar constraint to more efficiently search for matches in the stereo pair of images. Namely, in ideal conditions, for the point x in the first camera, the corresponding image point x' should lie on the epipolar line in the second camera.

The geometry of pairwise views can also be applied to images taken by the same camera, but at displaced locations. However, in this case, we aim to determine the unknown relative pose defined by $(R,t) \in \mathsf{SE}(3)$ between the views. For the monocular example, we can obtain the definite relative orientation R, but the relative position t only up to scale. In this case, the epipolar matrix is first estimated from point correspondences, typically with the 5-point method [1] followed by the singular-value decomposition (SVD) to obtain (R,t). In this paper we assume that the camera is placed on a platform that cannot exhibit pure rotation, like the Ackermann drive or a rail vehicle.

The stereo setup, on the other hand, offers the possibility to estimate the definite t with proper scale. One approach would be to have image correspondences triangulated in both stereo views to obtain two 3D point clouds, and apply an algorithm from the iterative closest point (ICP) family to estimate the relative pose. However, it was shown in [1], [4] that motion estimation obtained by minimizing the points reprojection error in the image space yields more accurate results than minimizing the Euclidean error in 3D space. The reason being that point uncertainties in the Euclidean space can be highly anisotropic, while in the image space their projections have a more balanced uncertainty. In such approaches, the 3D point cloud is triangulated from the previous stereo view and the points are projected on the image plane in the current view. For a 3D point $X = \begin{bmatrix}\tilde{X} & \tilde{Y} & \tilde{Z} & 1\end{bmatrix}^\mathsf{T}$ in homogeneous coordinates this evaluates to the following expression

$$\begin{bmatrix}\tilde{x}\\ \tilde{y}\\ 1\end{bmatrix} = \pi(X,R,t) = \begin{bmatrix}f_x & 0 & c_x & 0\\ 0 & f_y & c_y & 0\\ 0 & 0 & 1 & 0\end{bmatrix}\begin{bmatrix}R & t\\ 0 & 1\end{bmatrix}\begin{bmatrix}\tilde{X}\\ \tilde{Y}\\ \tilde{Z}\\ 1\end{bmatrix}, \quad (4)$$

where $x = \begin{bmatrix}\tilde{x} & \tilde{y} & 1\end{bmatrix}^\mathsf{T}$ are homogeneous image coordinates, $\pi(\cdot)$ is the projection function, f_x and f_y are the focal lenghts, $\begin{bmatrix}c_x & c_y\end{bmatrix}^\mathsf{T}$ is the image principal point, and (R,t) is the relative pose between the two stereo views. To estimate (R,t) one would then typically minimize the following objective function for both images of the stereo setup

$$\underset{R,t}{\operatorname{argmin}} \sum_i ||x_i - \pi(X_i,R,t)||^2, \quad (5)$$

where x_i is the image point in the current stereo view, while $\pi(X_i,R,t)$ is the projected image point from the 3D point cloud constructed from the previous stereo view. Unlike a pure ICP approach, which is a 3D-3D method, minimizing (5) is a 3D-2D method, since it requires estimated feature depth before reprojection. On the other hand, the proposed approach is a 2D-2D method, since it works directly on image coordinates and does not require feature depth. Note that the projection function $\pi(\cdot)$ also entails a nonlinear part in charge of correcting lens distortions, mostly the radial distortion [45].

III. SOFT2: General System Overview

The SOFT2 pipeline is depicted in Fig. 2. Upon receiving a pair of stereo images from the left and right camera (L and R), we detect and match features, producing left and right camera features sets, $\mathcal{F}_L$ and $\mathcal{F}_R$, respectively. The correction of feature patch appearance due to perspective is an optional step and described at the end of this section; at the moment we leave it out of discussion. By relying solely on $\mathcal{F}_L$, we iteratively estimate the epipolar matrix E using the whole set of inliers [3], from which we can determine the relative rotation R and translation direction $\hat{t}$. Then, by including $\mathcal{F}_R$, we estimate jointly the scale s and relative orientation of the right camera with respect to the left one. This step produces absolute relative motion defined by the rotation matrix R and translation vector $s\,\hat{t}$. To enhance our final estimate, we include information from the temporal window of previous left camera frames in a bundle adjustment procedure, which yields

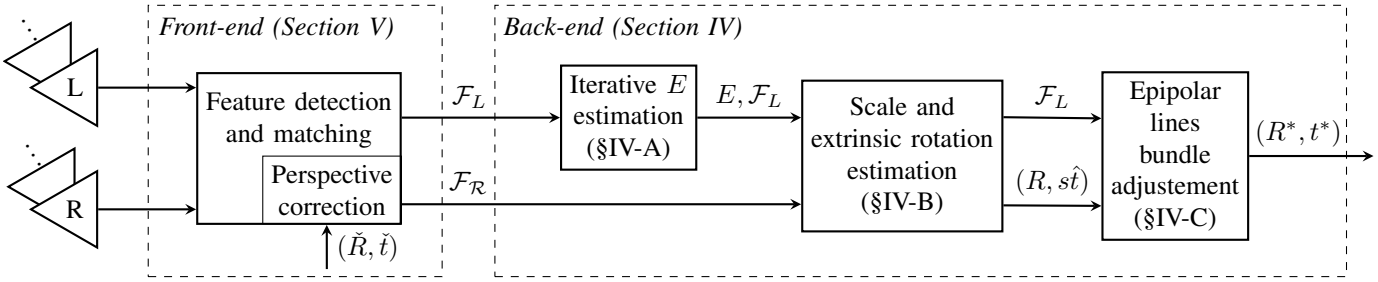


Fig. 2: Pipeline of the proposed stereo odometry approach. Triangles represent left and right cameras (L and R), respectively.

our final odometry (R^*, t^*). In all the steps we use *point-to-epipolar-line* distance minimization, rather than point-to-point distance minimization. In essence, our approach facilitates optimization in the essential matrix space and by focusing on the improvement of rotation accuracy we can expect a great boost in overall performance, since rotation error produces superlinear growth of the position error. Moreover, feature displacement due to radial distortion or blur due to increase in vehicle speed or diminishing lighting conditions, will not as much affect the optimization since the effects will coincide with the direction of the epipolar lines.

The motivation behind our approach lies in the fact that we target SOFT2 for high accuracy automotive localization and operation in real time, and consequently have decided to rely on the epipolar geometry and constraints imposed by the platform kinematics; specifically:

- ground vehicles, and similar configurations, do not exhibit pure rotation and always have a translational component, thus escaping singularity in essential matrix computation
- during forward motion, which is dominant in ground vehicles, the principal point and the epipole approximately coincide making epipolar lines direction in line with the radial distortion and blur (see Fig. 1)
- orientation and translation direction can be estimated using a single camera, thus making odometry more robust to stereo calibration errors or possible stereo rig deformations during operation
- absolute scale and second camera extrinsic rotation with respect to the first can be estimated jointly, thus increasing accuracy by continuous online calibration
- feature matching can be enhanced by taking into account appearance change due to varying perspective (both in stereo and time), thus allowing to also include close-by ground features that boost translation estimation accuracy.

The SOFT2 pipeline is similar in structure to its predecessor SOFT [9], [22]; however, with the following important differences leading to better performance:

1) SOFT computes only rotation in a 2D-2D manner, while absolute translation is determined by triangulating features via stereo and estimating odometry through reprojection error, i.e., in a 3D-2D manner. On the other hand, SOFT2 computes rotation, translation direction, scale, and extrinsic stereo camera rotation jointly and continuously by minimizing the point-to-epipolar-line metric making it a 2D-2D method.
2) SOFT uses only 5 points selected by RANSAC to estimate E via the 5-point method [1], while SOFT2 uses the dominant hypothesis to select all the inliers to compute E using an iterative method [3].
3) SOFT smooths odometry estimates by averaging over previous frames through spherical linear interpolation (Slerp) [46], while SOFT2 uses the proposed bundle adjustment based on the point-to-epipolar-line metric.
4) To enhance feature matching, SOFT2 introduces a multiple hypothesis perspective correction (MHPC) matcher that can take into account patch appearance change due to perspective.

The main motivation behind the proposed MHPC matcher is to obtain a set of appropriately distributed informative features – the key to any accurate vision-based motion estimation – and it is especially effective for ground (road) features that are close to the cameras and can aid greatly in translation estimation. However, to achieve this we already need to know the relative displacement of the camera. Given that, the "feature detection and matching" block from Fig. 2 becomes a two step process. In the first step, preliminary relative motion $(\check{R}, \check{t})$ needs to be determined, either by constant velocity assumption or by a computationally light odometry – in our case we used SOFT [22]. In the second step, we detect feature patches in the image and use the odometry information from the first step to correct the patch appearance in order to better match them in stereo and time. Note that this mostly shows improvement for stereo cameras with larger baselines, like the one used in the KITTI dataset; otherwise, the computational overhead might not be cost-beneficial. An interesting future direction would be to explore deep learning methods for obtaining features that are trained to handle such perspective changes [47]–[49].

All these enhancements lead to SOFT2 scoring 0.53% in translation error and 0.0009 deg/m in rotation error rendering it currently the highest ranking algorithm on the KITTI scoreboard. In the following, we describe each of the SOFT2 building blocks depicted in Fig. 2. We describe the feature detection and matching block last, which we can consider as the odometry *front-end*, and focus first on the other blocks that constitute the SOFT2 *back-end*, since therein most of the methodological novelties are concentrated.

IV. SOFT2 BACK-END

Our final goal is to determine rotation and translation of the stereo camera displacement by estimating the essential matrix E. Thus, we focus on estimating the rotation and translation up to scale from a single camera first. Without the loss of generality we assume that the reference camera is the left camera. Then we use the right camera to estimate jointly the scale and relative orientation of the right camera with respect to the left camera. This enables us to treat our odometry as a monocular one for as long as possible before resorting to information from the right camera – and when we do, we try to compensate for the errors in stereo extrinsics at each frame. Finally, to gain further robustness, we compute the final essential matrix from a temporal window of previous poses.

A. Iterative essential matrix estimation

In SOFT2 we base the essential matrix estimation, from which we obtain the rotation and translation up to scale, on the iterative method proposed in [3]. The state is parameterized as follows

$$E(\xi) = E(\alpha, \beta, \gamma, \theta, \phi) = R(\alpha, \beta, \gamma) \left[\hat{t}(\theta, \phi)\right]_\times, \quad (6)$$

where (α, β, γ) are the Euler angles, (θ, ϕ) are spherical coordinates of the translation $\hat{t}$ parameterized as a unit sphere vector (no scale information). We use RANSAC with multiple 5-points hypotheses, but after the selection of inliers by the dominant hypothesis, contrary to the 5-point method [1], we use all the inliers to compute the final essential matrix in order to exploit information from all the relevant points. The criterium used to define the optimization problem is based on calculating the signed distance between an image point in homogeneous coordinates x and a line $l = [l_1 \ l_2 \ l_3]^\mathsf{T}$

$$d(x, l) = \frac{l^\mathsf{T} x}{\sqrt{l_1^2 + l_2^2}}. \quad (7)$$

Given that, we can now write the objective function as the sum of symmetric squared point-to-epipolar-line distances from the current to the previous frame [44]

$$\min_\xi \sum_i d^2(x_i, l_i'(\xi)) + d^2(x_i', l_i(\xi)), \quad (8)$$

where $l_i'(\xi) = E(\xi)^\mathsf{T} x_i'$ and $l_i(\xi) = E(\xi) x_i$ are epipolar lines associated to points x_i' and x_i in the previous and current view, respectively. Note that the objective function introduces a kind of temporal symmetry, producing the same result independently of the order in which the frames are reproduced.

If we were to stack all the distances into a single column vector $r(\xi) = [d(x_1, l_1'(\xi)) \ \dots \ d(x_1', l_1(\xi)) \dots]^\mathsf{T}$, then our optimization problem can be simply written as

$$\min_\xi \ \frac{1}{2} r(\xi)^\mathsf{T} r(\xi). \quad (9)$$

Note that to compute this symmetric distance we only need to transpose the essential matrix – no matrix inversions are needed. To minimize (9), and all the subsequent optimization problems, we used the Levenberg-Marquardt algorithm.

B. Scale and extrinsic rotation estimation

In order to estimate proper absolute scale we need to complement the approach from previous section with data from the right camera. Although only the scale needs to be estimated, we jointly estimate the rotational part of the right camera extrinsics. Extrinsic values obtained during initial calibration are not necessarily valid throughout the operation, e.g., stereo camera rig is not perfectly rigid, especially when the baselines are of the size as in the KITTI dataset. Since the scale estimate is mainly dependent on extrinsics, improving the accuracy of extrinsic parameters by online estimation directly improves the scale accuracy.

We approach this problem by modeling the rig mount as a rigid beam where the left camera and the baseline are fixed, while the right camera can only rotate around its focal point. Naturally, this is an approximation, since the whole stereo rig bends, but such a simplification enables us to obtain method that can work in real time and still enhance results. At this step, our state contains the Euler angle parameterization of the extrinsic rotation $R(\zeta) = R(\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma})$ of the *right* camera around its focal point and the absolute scale s of the translation of the *left* camera due to motion that needs to be estimated. Note that we use $(\tilde{\cdot})$ to emphasize that these Euler angles differ from (α, β, γ) that are part of the left camera motion.

As in the previous section, we base our approach on minimizing the symmetric point-to-epipolar-line metric, but with the following constraints. Lets assume we have a pair of stereo images – the current and the previous frame. Given that, we have at our disposal previous images from the left and right cameras, i.e, L^- and R^-, and current images from the left and right camera, i.e., L and R. Our optimization problem minimizes the point-to-epipolar-line metric between the image pairs (R, L^-), (L, R^-), and (R, R^-) and can be written as follows

$$\min_{\zeta, s} \sum_j \frac{1}{2} r_j(\zeta, s)^\mathsf{T} r_j(\zeta, s), \quad (10)$$

where the summation iterates over the following set of combinations $j \in \left\{(\mathrm{R}, \mathrm{L}^-), (\mathrm{L}, \mathrm{R}^-), (\mathrm{R}, \mathrm{R}^-)\right\}$. E.g., for the image pair $(\mathrm{R}, \mathrm{L}^-)$, the cost amounts to

$$r_{(\mathrm{R},\mathrm{L}^-)}(\zeta, s) = \left[d(x_1^R, l_1'^{L^-}(\zeta, s)) \ \dots \ d(x_1'^{L^-}, l_1^R(\zeta, s)) \dots\right]^\mathsf{T}, \quad (11)$$

where $d(x_1^R, l_1'^{L^-}(\zeta, s))$ represents the distance between the point in the current right camera and the epipolar line associated to the matched point in the previous left camera image. To obtain the required epipolar lines, the pertaining epipolar matrix is computed from the resulting rotation and translation that is obtained by concatenating the necessary $\mathsf{SE}(3)$ transforms. For example, as illustrated in Fig. 3, transform for getting from R to L^- evaluates to

$$(R_{RL^-}, t_{RL^-}) = \begin{bmatrix} R(\alpha, \beta, \gamma) & s\hat{t}(\theta, \phi) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R(\zeta) & b \\ 0 & 1 \end{bmatrix}, \quad (12)$$

where b is the stereo baseline.

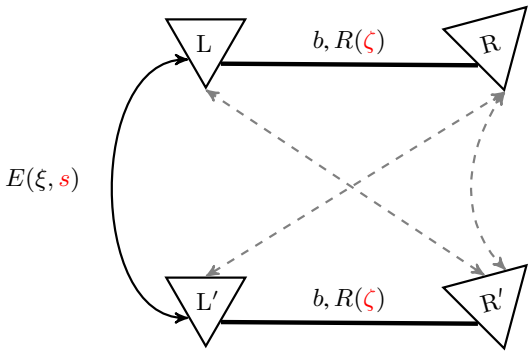


Fig. 3: Illustration of the scale and extrinsic rotation estimation. Dashed colored lines designate the epipolar lines cost pairs used in the cost function. Variables colored in red represent estimation variables.

Note that $R(\alpha, \beta, \gamma)$ and $\hat{t}(\theta, \phi)$ are known at this point since we estimated them previously from the left camera images, while ζ and s are the current optimization variables. We also assume that during the motion between the two stereo frames the extrinsic rotation of the right camera did not change. Given that, we now have the odometry result with the absolute scale. Furthermore, note that unlike the classical geometric reprojection error (5), in our equations the triangulated 3D point X_i does not appear anywhere, i.e, SOFT2 is a completely 2D-2D approach and the relative pose estimation is invariant to object depth uncertainty.

C. Epipolar lines bundle adjustement

In our previous odometry, dubbed SOFT, we improved rotation accuracy by introducing epipolar constraints between several consecutive frames. Although fast and effective, the SOFT approach is suboptimal, since it computes frame-to-frame epipolar matrices independently and fuses the results afterwards in the quaternion space. In this paper, we propose a geometrically optimal solution for determining sequence of rotations in a window of poses, thus we extend the iterative method to compute a kind of *epipolar lines bundle adjustment*.

In other words, as the result of feature tracking, multiple epipolar lines connect not only the neighboring but also other left camera poses in the temporal window, thus creating multiple constraints. However, since we are performing bundle adjustment (BA) on just a single camera, the origin and absolute scale are unobservable, but the relative scale is. For example, lets consider BA of three consecutive poses from two consecutive motions, e.g., 1-2 and 2-3, as illustrated in Fig. 4. Without BA, essential matrices E_{12} and E_{23} are estimated separately, and motions $(R_{12}, \hat{t}_{12})$ and $(R_{23}, \hat{t}_{23})$ up to scale are extracted. Conversely, when BA is used, the essential matrix E_{13} puts an additional constraint to the estimate of the two relative motions, since (R_{13}, t_{13}) is the $\mathsf{SE}(3)$ product of motions (R_{12}, t_{12}) and (R_{23}, t_{23}). However, the three motions in the essential matrix space, E_{13}, E_{12}, and E_{23}, have unit length translations, and to apply the E_{13} constraint, we must switch to the $\mathsf{SE}(3)$ space, which has one additional variable – the scale, unobservable in the essential matrix space. To make the problem observable, we first set t_{12} to be a unit vector, which fixes the overall scale of the bundle. Then, the $\mathsf{SE}(3)$ constraint will hold only if the whole bundle is scaled uniformly, and to ensure this, since in the bundle all translation vectors are unit, we need to introduce the relative scale s_r as an additional variable in the optimization process. Given that, for our working example, the initial relative scale is computed as $s_{23} = ||t_{23}||/||t_{12}||$, since at this stage we already have the absolute translations from the odometry, and the essential matrix representing motion 1-3 can be constructed by the following $\mathsf{SE}(3)$ matrix multiplication

$$(R_{13}, s_{13}\hat{t}_{13}) = \begin{bmatrix} R_{12} & \hat{t}_{12} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_{23} & s_{23}\hat{t}_{23} \\ 0 & 1 \end{bmatrix}, \quad (13)$$

where $E_{13} = R_{13}\left[\hat{t}_{13}\right]_\times$. Therefore, three poses can be represented with 11 variables, 5 for each consecutive essential matrix (6DOF minus scale), and 1 for the relative scale. Given that, our optimization problem can be written as

$$\min_{\xi, s_r} \quad \sum_j \frac{1}{2} r_j(\xi, s_r)^\mathsf{T} r_j(\xi, s_r), \quad (14)$$

where ξ and s_r contain parameters for all consecutive motions and j iterates over the set of all possible frame combinations. For example, having consecutive motions 1-2 and 2-3, ξ would contain parameters for E_{12} and E_{23}, s_r would be the relative scale of translations t_{23} and t_{12}, and $j \in \{12, 23, 13\}$. Given that, if $j = \{13\}$, the pertaining cost evaluates to

$$r_{13} = \left[d(x_1^1, l_1'^3(\xi, s_r)) \ \ldots \ d(x_1'^3, l_1^1(\xi, s_r)) \ldots\right]^\mathsf{T}, \quad (15)$$

where $d(x_1^1, l_1'^3(\xi, s_r))$ represents the distance between the point in L_1 and the epipolar line associated to the matched point in L_3. Note that we only estimate the parameters of the motion between the consecutive frame pairs. For example, as illustrated in Fig. 4, if we have a window of 4 frames we would optimize over the variables of E_{12}, E_{23}, E_{34}, i.e., 17 parameters, while E_{13}, E_{24}, and E_{14} would act as additional dependent costs in the optimization function. In general, a window of N poses can be represented by $5(N-1)+N-2$ variables. In our implementation, for computational efficiency reasons, we do not perform epipolar line BA at each step, but instead every N-th frame. Moreover, we do not correct the history of all the poses in the window – just the current pose as it would be in the case of real-time operation. To integrate the obtained correction into a real-time output, we compare the odometry trajectory segment containing the last N frames with the bundle adjusted trajectory of that segment, and obtain an accumulated $\mathsf{SE}(3)$ error over these N frames. Then, every N-th frame, we correct the odometry output to compensate for the error accumulated over the last N poses. The correction is small enough so as not to affect the odometry smoothness, but the improvement in accuracy over a longer period is noticeable. Note that our epipolar line BA corrects only the rotation and translation direction, and not the scale.

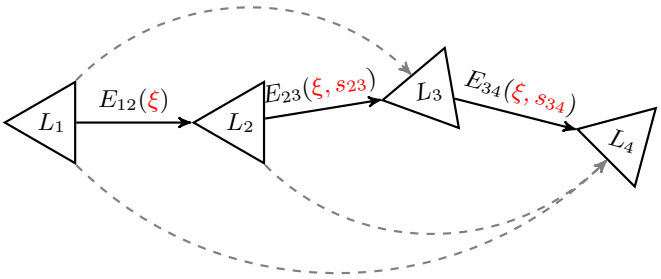


Fig. 4: Illustration of the epipolar lines BA for 4 frames. Note that we use only left camera images. Dashed lines represent the additional epipolar lines cost pairs that are used in the cost function besides the ones from consecutive frames. Variables colored in red represent estimation variables.

V. SOFT2 Front-end

The most important factor for any vision-based odometry is having a set of informative features distributed appropriately to yield accurate rotation and translation estimation. Given that, we also devoted considerable attention to further enhancing the feature matching process. Our SOFT2 pipeline requires features matching between the left and the right camera, between the current and previous stereo frame, and across several frames in the past to perform the proposed bundle adjustment. Features should also be localized with subpixel precision in order to achieve the presented accuracy. Original SOFT [22] features do satisfy these requirements and they performed well on the tested datasets. Indeed, if the scenery is visually rich and informative, majority of standard feature matchers will do the job. However, in low contrast and monotonous scenes, better matcher quality can make the difference in odometry accuracy.

A. Motivation behind perspective correction

Having both distant and close features in the view is of the utmost importance for any visual odometry – the larger the span, the less ambiguity between the rotational and translational part of estimated motion. Contrary to indoor operation, scenes analyzed from road vehicles in normal conditions often contain distant features – at least the horizon or even the sky. But in order to accurately resolve for translational part of the motion, having at least several features close enough to the camera is mandatory. Adequate distance depends on the stereo baseline, and as a rule of thumb, features can be considered close enough if they are less than forty stereo baselines away. However, the closer the features, the more accurate the translation, and consequently also the rotation.

While in general case one cannot always rely on having nearby features, for road vehicles the situation is different, and the source of nearby features is always present – the road. However, standard feature detector-descriptor-matcher combos usually do not result with many matches on a road surface. The reason is obvious: the road is often of low texture and self similar. Even if a standard method detects the match on such a surface, there is large probability that it is an outlier. Besides these two reasons, there is also another one, perhaps less evident – the perspective change. Usually, features seen from the left camera appear pretty much the same in the right camera, even with large stereo baselines. The difference in the appearance is small, since the majority of features reside either on a distant plane or on a plane with the normal vector pointing more or less to the viewer. However, the road plane is different: it is parallel to the camera optical axis, and close features residing on the road are subject to noticeable change in the appearance as the stereo baseline grows. Motivated by these facts, we propose a multiple hypothesis perspective correction (MHPC) patch matching for self-similar planar textures, in order to enhance odometry in weakly informative scenes with large stereo baselines, but at the cost of higher computational burden. Note that although in our case this affects mostly the road features (i.e. ground features to be more exact), it can be applied to any patch lying on a surface with a known normal. In the following, we first describe the perspective matching of features and then our approach to ground plane estimation.

B. Multi hypothesis perspective correction matcher

In Algorithm 1 we provide the pseudocode for the proposed MHPC matcher. As input, our algorithm requires current and previous stereo images, odometry and estimated ground plane from the previous step. Then, in line 1, we detect features in the left image $\mathcal{F}_L$, specifically corners, by using good-features-to-track [50]. Since we want to have strong corners equally distributed across the image, we divide the image into bins 50×50 pixels large and select the strongest corner from each bin until the preferred number of features per bin is selected or until no features are left in the bin. Since in the following steps the patch prediction requires an already computed displacement, in lines 2 through 6 we determine the initial transform $(\check{R}, \check{t})$ either from a constant velocity model or light visual odometry. In our implementation, we use SOFT [22], but only for patch prediction. The reason behind lies in speed and accuracy – with more accurate prediction, search area for the patches can be reduced to few pixels only, which lowers the search time and possibility for false matches. Processing time spent on the prediction is minor compared to the time gained due to the reduction of the search area.

Then, for each feature in the left image we do the following. In line 9, we generate two hypotheses: the patch is either on the ground or not (note that here we require ground plane estimation which is described later in Section V-C). Given the hypotheses, we generate patch predictions, i.e, hypothetical appearances of patches in the right camera. For the first hypothesis, we compute the 3D point as the intersection of the feature back-projected ray with the ground plane, that we obtained by prediction based on $(\hat{n}, d)$ and $(\check{R}, \check{t})$. Then, we project the 3D point to the right image, take 9×9 regular patch coordinates around the projected point, and find the corresponding patch pixels in the left image to generate the patch corrected for perspective. For the second hypothesis,

Algorithm 1 MHPC matcher

Require: Images L, R, L^-, R^-; odometry from the previous step (R^*, t^*); ground plane from the previous step $(\hat{n}, d)$.
Ensure: Matched stereo features $\mathcal{F}_L$, $\mathcal{F}_R$

1: Detect strong and evenly distributed features: $\mathcal{F}_L \leftarrow \text{get\_features}(\mathrm{L})$
2: **if** const_velocity_model **then**
3: $(\check{R}, \check{t}) \leftarrow (R^*, t^*)$
4: **else**
5: $(\check{R}, \check{t}) \leftarrow \text{SOFT}(\mathrm{L}, \mathrm{R}, \mathrm{L}^-, \mathrm{R}^-)$
6: **end if**
7: **for** $i = 1 : |\mathcal{F}_L|$ **do**
8: */* Matching in current stereo images */*
9: Features get two hypothetical patch transforms: $\mathcal{F}'_{L,i} \leftarrow \{\text{gnd\_tf}(\mathcal{F}_{L,i}, \mathrm{R}, \check{R}, \check{t}, \hat{n}, d), \text{norm\_tf}(\mathcal{F}_{L,i}, \emptyset)\}$
10: Compute NCC along the epipolar line: $(\mathcal{F}'_R, \text{stereo\_ncc}) \leftarrow \text{epipolar\_ncc}(\mathcal{F}'_{L,i}, \mathrm{R})$
11: $\text{stereo\_ncc} \leftarrow \text{sort}(\text{stereo\_ncc})$
12: $\text{diff\_scores} \leftarrow \text{diff}(\text{stereo\_ncc})$
13: $k \leftarrow \text{find\_index}(\text{diff\_scores} > \text{diff\_th})$
14: **if** $k > 10 \;||\; k = \emptyset$ **then**
15: $\mathcal{F}'_{L,i} \leftarrow \emptyset$, $\mathcal{F}'_R \leftarrow \emptyset$
16: **else**
17: Keep matches with k highest NCC scores: $\mathcal{F}'_R \leftarrow \mathcal{F}'_R[\&\text{stereo\_ncc}[1:\mathrm{k}]]$
18: **for** $j = 1 : k$ **do**
19: */* Matching in previous stereo images */*
20: **if** $\text{ground\_hypothesis}(\mathcal{F}'_{R,j})$ **then**
21: $(\mathcal{F}'_{\mathrm{L}^-}, \mathcal{F}'_{\mathrm{R}^-}) \leftarrow \text{gnd\_tf}(\mathcal{F}'_{R,j}, \mathrm{R}^-, \mathrm{L}^-, \check{R}, \check{t}, \hat{n}, d)$
22: **else**
23: $\delta \leftarrow \text{get\_disparity}(\mathcal{F}'_{L,i}, \mathcal{F}'_{R,j})$
24: $(\mathcal{F}'_{\mathrm{L}^-}, \mathcal{F}'_{\mathrm{R}^-}) \leftarrow \text{norm\_tf}(\mathcal{F}'_{R,j}, \mathrm{R}^-, \mathrm{L}^-, \check{R}, \check{t}, \delta)$
25: **end if**
26: Compute NCC locally around the projected point: $(\mathcal{F}'_{\mathrm{L}^-}, \text{lp\_ncc}) \leftarrow \text{local\_ncc}(\mathcal{F}'_{R,j}, \mathrm{L}^-)$ $(\mathcal{F}'_{\mathrm{R}^-}, \text{rp\_ncc}) \leftarrow \text{local\_ncc}(\mathcal{F}'_{R,j}, \mathrm{R}^-)$
27: $\text{lp\_ncc} \leftarrow \max(\text{lp\_ncc}), \text{ncc\_rp} \leftarrow \max(\text{rp\_ncc})$
28: $\text{final\_ncc}_j \leftarrow \text{ncc\_stereo} + \text{lp\_ncc} + \text{rp\_ncc}$
29: **end for**
30: Select the match with the best overall score: $\mathcal{F}_{R,i} \leftarrow \mathcal{F}'_{R,j}(\&\max(\text{final\_ncc}))$
31: **end if**
32: **end for**

we assume that the patch normal is oriented directly towards the camera and would apply the corresponding perspective transformation. However, this requires disparity and since it is not available at this point, for stereo we simply copy the patch. Then, in line 10, we correlate both patch appearance hypotheses with all the positions along the epipolar line in the right image via normalized cross correlation (NCC), and store all the NCC scores and potential matches $\mathcal{F}_{\mathrm{R}^-}$. Since the road is self-similar, the best NCC score is usually not dominant nor the highest score is necessarily the correct one. Therefore, we

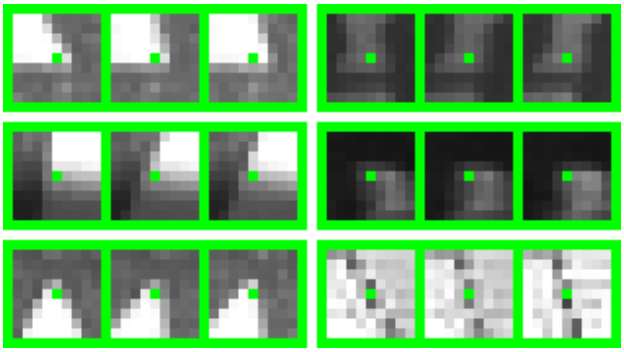


Fig. 5: Example of six road features where each triplet illustrates: road feature as seen from the left camera (left patch), prediction generated with SOFT2 (middle patch), and corresponding feature found in the right image (right patch).

sort all the scores from highest to lowest in order to analyze the differences. This is carried out in lines 11 and 12.

In lines 13 through 17, starting with the highest score, we search for the difference between the neighboring scores that is large enough to present a noticeable distinction. If we do not find a difference larger than some threshold after the first 10 scores, e.g., all the first 10 matches have a very similar score, we discard this feature since it is too ambiguous. Otherwise, we consider all matches prior to this difference as valid matching hypotheses. Then, for each potential stereo match, we search for the corresponding matches $\mathcal{F}'_{\mathrm{L}^-}$ and $\mathcal{F}'_{\mathrm{R}^-}$ in L^- R^-, by projecting the 3D points in lines 18 through 25. If the match has emerged from the ground plane hypothesis, the patch is generated using the same principle as in the stereo matching case. Otherwise, we assume that the patch is on a plane with a normal vector pointing towards camera and we generate the corrected patch accordingly. Note that ground hypothesis does not require disparity since the 3D point is determined from the ground plane, while for the normal hypothesis we require the disparity δ. Subsequently, in lines 26 and 27, we search for the generated patch in the left and right previous images in a narrow area around the projection of the predicted point and select the highest NCC score. In the end, each hypothesis starting from the current left image will have three NCC scores: current left L relative to current right R, previous left L^-, and previous right R^- image. To consider the match, all three NCC scores must be above a user defined value and the best results were obtained by setting the threshold between 0.7 and 0.8. We compute the overall score in line 28 as the sum of the three scores. Finally, in line 30 we select the match with the best overall score. Fig. 5 shows an example of correctly predicted and matched road features.

Note that the incorrect starting hypothesis will have wrong disparity, which in turn will lead to incorrectly predicted position in the previous images with a likely lower NCC score. Consequently, correct prediction suppresses false matches in the overall score and raises the probability for a correct match. Our matches are computed with subpixel precision by fitting a parabola to the surrounding NCC values and we forward them

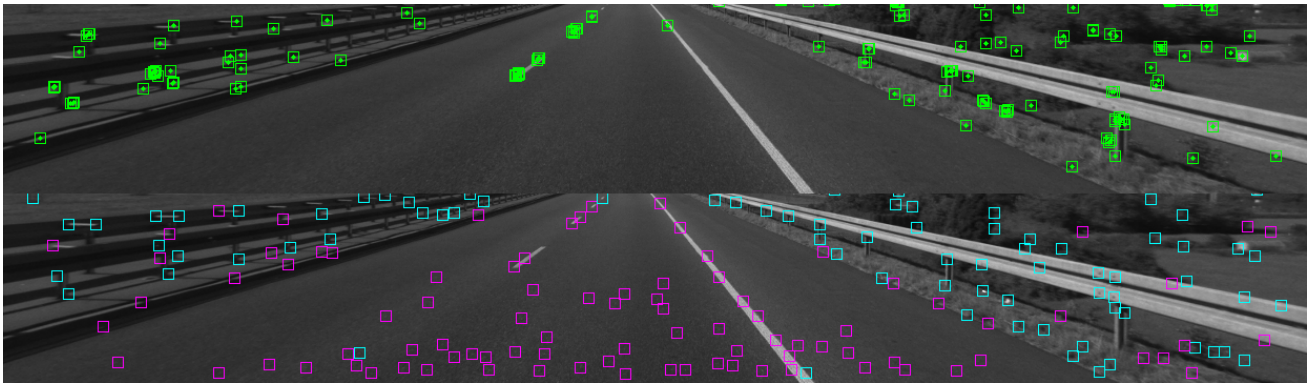


Fig. 6: Detected ORB-SLAM2 features (up) and SOFT2 features (down) in the frame 390 of KITTI sequence 01. Road features are marked in purple, while others are in cyan.

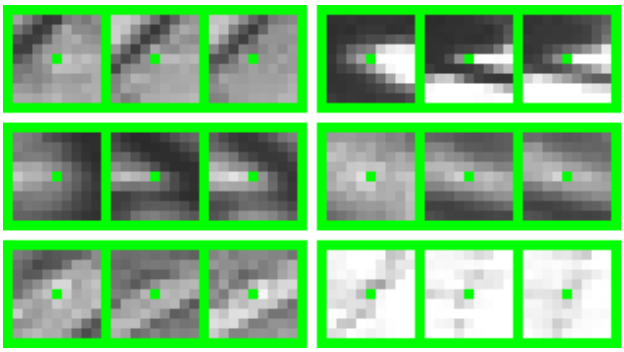


Fig. 7: Example of six road features where each triplet illustrates: feature as seen from the left camera (left patch), prediction generated with SOFT2 for 5 frames into the past (middle patch), and feature found in the left image 5 frames into the past (right patch).

to the optimization stage in floating point precision. Also, since each feature carries the information whether it resides on the ground plane, for the subsequent ground plane detection we can use only the ground features, thus making it faster and more reliable. Fig. 6 compares ORB-SLAM2 features with features detected with SOFT2. We can see that SOFT2 features are more evenly distributed across the image with plenty of close features available for accurate ego-motion estimation.

Finally, to enable bundle adjustment, we also need to establish matches with frames that are more distant in the past. For this, we employ the same mechanism as described in previous paragraphs: whenever a successful stereo match from the current to the previous frame is found, we continue the search further into the past until the size of bundle adjustment window is reached or until the overall matching score drops below a threshold. Since in our case objects in the scene dominantly move towards the camera, the predicted patches are always generated with downsampling. This is why we only generate predictions for past frames; otherwise, we would need to up-sample patches which would result in information loss and consequently lower matching precision. Fig. 7 shows matched road patches between the current left image and a 5 frames old left image.

C. Ground plane estimation

Following the SOFT2 approach, to avoid uncertainty due to 3D triangulation, the ground plane is estimated via homography. Homography states that two features residing on a planar surface in space, are related to scale λ by the homography matrix H [44]

$$\lambda x' = Hx, \tag{16}$$

where x' and x are homogeneous image coordinates in two different views. The homography matrix is then given by:

$$H = R - \frac{t\hat{n}^T}{d}, \tag{17}$$

where in our case (R, t) is the relative transformation from the previous to the current left camera frame, while $\hat{n}$ is the normal vector of the plane and d is the distance to the plane with respect to the current left camera frame. Matrix H has 8 degrees of freedom, but 4 correspondences are required for solving (16). Homography can be decomposed to obtain (R, t), $\hat{n}$, and d, but this process is very sensitive to noise and can easily become numerically unstable [24]. Therefore, we choose to estimate the ground plane via nonlinear optimization.

To reduce the probability of ending in a local minimum, the number of optimization parameters is usually reduced by decoupling the camera motion from the ground plane. We apply the same idea here and exploit the knowledge of $(R, \hat{t})$ estimated from the essential matrix. Given that, we optimize only for three parameters describing the ground plane – the ground plane normal vector $\hat{n}$ (parameterized in spherical coordinates), and the distance to ground d. There are a couple of advantages to this approach. First, with predetermined $(R, \hat{t})$, the number of optimization parameters is reduced to three, which makes optimization faster and more reliable. Second, the number of minimum correspondences required for solving the problem is also reduced to three – raising the probability that a sample will belong to the ground plane and reducing the number of RANSAC iterations. In the end, the optimization approach is encouraged by the fact that the initial solution is always close to the final solution – for ground vehicle applications the estimated camera height and

the ground plane normal reside in a narrow range of values. Given that, we estimate the ground plane via

$$\min_{\hat{n},d} \sum_i \frac{1}{2}(x_i' - Hx_i)^T(x_i' - Hx_i). \tag{18}$$

We use (18) in conjunction with a 3-point sample RANSAC to find a hypothesis with the maximum number of inliers and the final solution is estimated by using all the inliers. Since translation estimated based on E is always of unit length, the distance to the ground is up to scale, and the metric distance is obtained by simple rescaling after the scale estimation step. This way, just like the motion itself, all parameters of the ground plane up to scale are determined by a single camera.

VI. Experimental Results

We evaluated SOFT2 on five publicly available automotive datasets: KITTI-360 [34], KITTI [27], Málaga [35], Oxford [36], and MVSEC [37]. We did not consider any synthetic datasets, since we believe that these do not model accurately enough real system challenges that we are targeting in this paper. Vibrations, temperature change, initial calibration error, calibration parameters drift, synchronization issues, and possible unknown sources of error are components of every real visual odometry system. For reference, we also process every sequence with the publicly available ORB-SLAM2 [10], OV2SLAM [33], and VINS-FUSION [11], [43] implementations and compare the results to SOFT2. Each algorithm comes with a parameter file tuned for the KITTI dataset, provided by the authors, which we used on all the datasets (for OV2SLAM we selected the "accurate" option). In order to ensure fair comparison and put an emphasis on the odometry drift, we turned off the loop closing option of the competing approaches. Moreover, note that the epipolar line BA of SOFT2 does not correct the pose history within the temporal window – only the current pose every third frame.

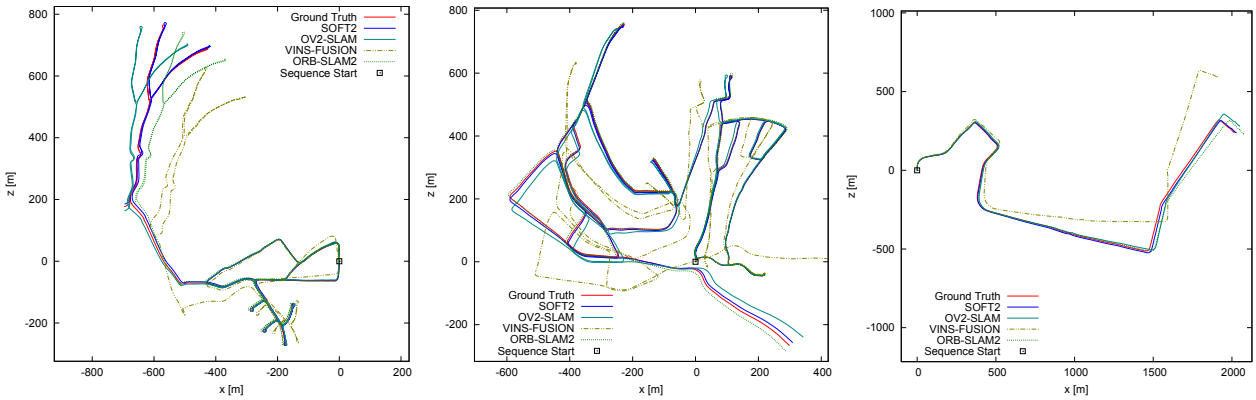
Regarding evaluation, as in the vein of [10], we used two different metrics: absolute translation root-mean-square error (ATE) t_{ate} [m] [51] and average relative translation t_{rel} [%] and rotation r_{rel} [deg/m] error [27]. The relative error, which is more tailored for odometry, is used on datasets that have accurate and reliable vehicle pose ground truth (translation and rotation), i.e., on the KITTI train sequences and manually selected segments from the Oxford dataset. ATE is used for datasets which, unfortunately, do not offer full vehicle pose ground truth, i.e., on the Málaga and MVSEC datasets.

A. The KITTI-360 dataset

The KITTI-360 dataset [34] consists of 9 sequences and was recorded across several suburbs of Karlsruhe, Germany, with a driving distance of 73.7 km. It contains recordings of a color stereo pair set at a baseline of 0.6 m with an image rate of 10 Hz and rectified images resolution of 1408×376 pixels. It also includes a Velodyne HDL-64E laser and OXTS3003 GPS/IMU sensor. Surround view of 360 degrees is accomplished with two additional fisheye cameras mounted at vehicle sides, but these images were not used in this evaluation. Although the KITTI-360 dataset was initially targeted for semantic instance labeling from annotated 3D primitives, as of 2021 accurate ground truth poses have also been provided, which enables odometry evaluation as well. Authors note that the ground truth poses are obtained from a large-scale optimization taking OXTS measurements, laser scans and multi-view images as input, thus are not identical to the raw OXTS measurements. However, ground truth poses are not available for each frame, occasionally they are missing while passing under the bridges, during backwards drive, and in general at random locations. Therefore, we had to modify the original KITTI evaluation script to skip each segment in which either the beginning or the ending ground truth pose is missing. Also, since each sequence has a block of missing poses at the beginning, for evaluation we started each sequence from the first frame that has ground truth pose available. Nevertheless, despite these minor issues, the ground truth appears more accurate and reliable than in the other datasets, which is the reason why we use this dataset first to analyze how each module of the SOFT2 pipeline enhances the odometry accuracy.

In Table I we show errors for the competing odometries and SOFT2 for which we included novel modules step-by-step to assess their contributions as well as BA with $N = 3$ (note that sequences 01 and 08 were not available at the time of writing). The visual odometry (VO) baseline column shows the error for the proposed approach when we use only the principle of minimizing the point-to-epipolar-line metric for motion and scale and extrinsic estimation with the older SOFT matcher. The following column includes the multi hypothesis perspective correction (MHPC) matcher, and we can already notice improvement on several sequences. The SOFT2 column now includes the MHPC matcher (both in stereo and time) and the epipolar lines bundle adjustment over three frames in the past (thus, VO + MHPC + BA3 = SOFT2). We can see that in this case the improvement is the strongest and that we have the lowest translational error on the majority of sequences, and on par average rotational error when compared to OV2SLAM. Regarding larger BA window size, when increased to $N = 5$ it reduced rotational and translational error, but on average only around 2%, and for larger window size the gain was even smaller. The other three odometries exhibited larger error for tracks 07 and 10 during the moments when moving traffic occupies a larger part of the scene. In urban sequences, with plenty of distinctive features, OV2SLAM results are close to SOFT2. However, SOFT2 outperforms on open road and highway sequences. In Fig. 8 we can see the alignment of several trajectories with respect to ground truth. We have also submitted SOFT2 to the recently available KITTI-360 Semantic SLAM trajectory evaluation benchmark and obtained absolute pose error of 0.7 m and relative pose error of 0.84%, which placed SOFT2 as the highest ranking approach among both visual and laser-based approaches[2].

[2]http://www.cvlibs.net/datasets/kitti-360/leaderboard_semantic_slam.php?task=trajectory



(a) KITTI-360 05 sequence (b) KITTI-360 06 sequence (c) KITTI-360 10 sequence

Fig. 8: Estimated trajectories along with the ground truth for three KITTI-360 sequences.

TABLE I: Experimental evaluation on the nine KITTI-360 sequences ($t_{\text{rel}}[\%]$, r_{rel}[deg/100 m]). VO is the base - essential matrix + scale and extrinsic with older SOFT matcher. Note that SOFT2 column represents the version that includes multi hypothesis perspective correction matcher with epipolar line BA, i.e., SOFT2 = VO + MHPC + BA3.

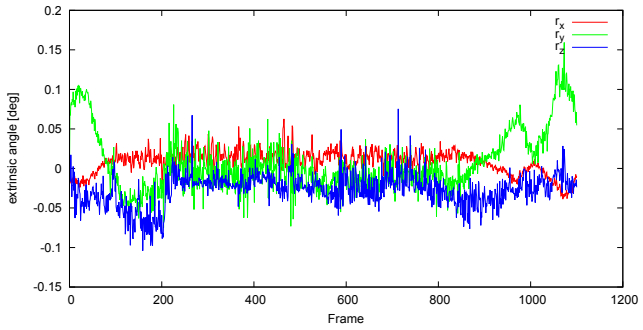
	VO		VO+MHPC		VO+MHPC+BA3		OV2SLAM		ORB-SLAM2		VINS-FUSION	
Sequence	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00	0.305	0.158	0.332	0.156	**0.294**	0.132	0.297	**0.126**	0.333	0.149	1.144	0.555
02	0.407	0.218	0.379	0.205	**0.365**	0.196	0.418	**0.174**	0.584	0.226	1.001	0.464
03	0.469	0.146	0.314	0.134	**0.227**	**0.104**	0.588	0.148	0.486	0.169	1.727	0.273
04	0.437	0.209	0.493	0.211	0.453	0.191	**0.428**	**0.166**	0.515	0.216	1.045	0.431
05	0.340	0.263	0.304	0.237	**0.304**	0.233	0.382	**0.205**	0.459	0.247	0.877	0.604
06	0.400	0.185	0.385	0.170	**0.354**	0.159	0.368	**0.151**	0.523	0.177	1.151	0.486
07	0.919	0.142	0.370	0.152	**0.306**	**0.129**	1.754	0.177	5.075	0.973	4.706	0.516
09	0.601	0.169	0.524	0.143	**0.492**	**0.125**	0.859	0.160	1.073	0.184	1.993	0.759
10	1.234	0.250	1.008	0.240	**0.871**	**0.225**	1.946	0.262	1.730	0.434	3.145	0.645
avg	0.461	0.194	0.427	0.184	**0.392**	0.168	0.559	**0.163**	0.814	0.238	1.429	0.532

We also ran an execution time analysis for this dataset on a laptop machine with an Intel CORE i7 2.7 GHz processor with 8 threads. The first step, which determines initial displacement using SOFT, takes 20 ms, where 10 ms is for matching and 10 ms for optimization. The MHPC matcher had an execution time of 40 ms on 6 threads when processing 1 feature per 50×50 pixels bin on 1408×386 KITTI-360 images. Note that on average more than 1 corner per bin is processed, because if the first corner is not successfully matched, the process repeats with the next one until one match is successful or no corners are left in the bin. Estimation of the essential matrix takes another 5 ms with 300 RANSAC iterations, while scale and extrinsic estimation finishes in 1 ms. In the end, BA with a window size of $N = 3$ takes 12 ms per frame. Note that BA can be executed in a separate thread. To sum up, the setup used for the presented results (BA with $N = 3$) had an execution time of $20 + 40 + 5 + 1 + 12 = 78$ ms per frame.

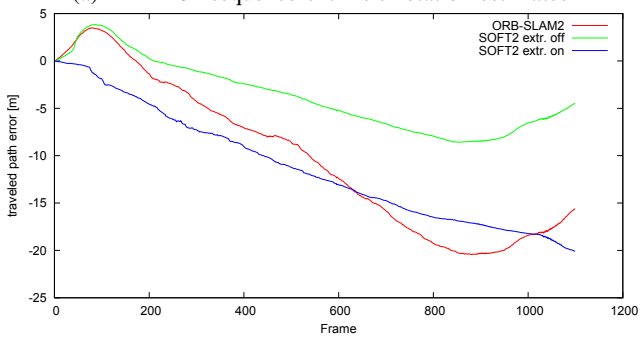
B. The KITTI dataset

The KITTI dataset was recorded in urban, rural, and highway scenarios by two stereo pairs (color and grayscale) set at a baseline of 0.54 m [52]. Image rate was 10 Hz, with a resolution somewhat smaller than the original 1392×512 pixels due to rectification. High accuracy OXTS3003 GPS/IMU unit was employed for recording the ground truth trajectory. Odometry part of the dataset contains eleven train tracks with the ground truth, plus eleven test tracks without the ground truth. Test tracks can be evaluated online by uploading resulting trajectories to a designated server. Evaluation results contain translation and rotation error and they can be published and compared on the official KITTI scoreboard. Currently, SOFT2 is the highest ranking algorithm over all sensor configurations with $t_{\text{rel}} = 0.53\,\%$ and $r_{\text{rel}} = 0.0009$ deg/m errors.

One of the challenges of the KITTI dataset lies in the values of default calibration parameters with which the images were rectified and evaluation of a particular odometry algorithm, at one point, can be interpreted as evaluation to robustness to calibration errors. Since dataset also provides raw images together with calibration board images taken every day after each recording, this challenge can be alleviated by recomputing calibration parameters and obtaining lower reprojection and odometry error. This is not necessarily a drawback per se, since certain amount of error in calibration parameters will always be inherent to real systems and motivates development of novel calibration methods. Note that the results presented in this paper were obtained using the default calibration parameters, while the result on the online scoreboard used parameters that were obtained with our calibration procedure [53].



(a) KITTI 01 sequence extrinsic rotation estimates



(b) KITTI 01 sequence error

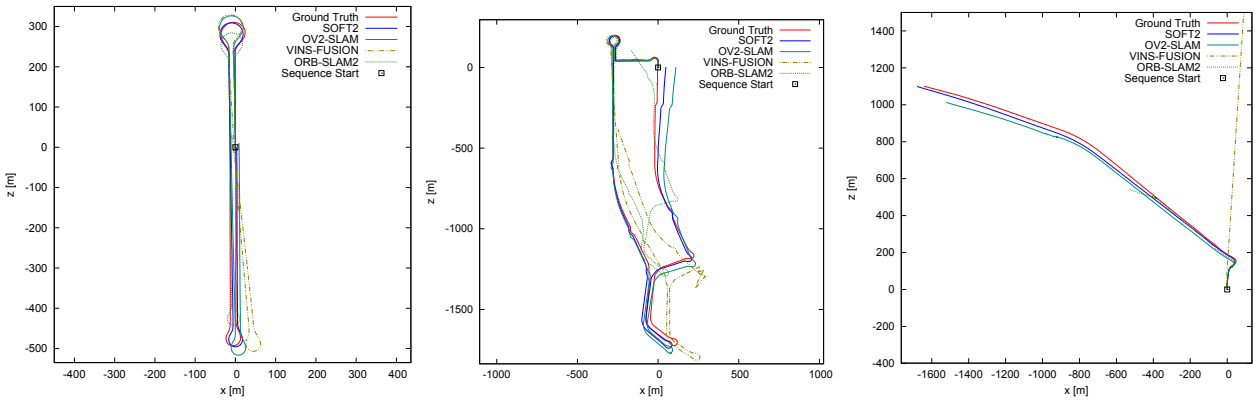
Fig. 9: Result of the extrinsic estimator showing change in relative rotation of the right camera with respect to the left camera and the effect on trajectory error with and without estimation. ORB-SLAM2 error is also shown for reference.

TABLE II: Experimental evaluation on the 11 KITTI dataset train sequences ($t_{\text{rel}}[\%]$, r_{rel}[deg/100 m])

	SOFT2		OV2-SLAM		ORB-SLAM2		VINS-FUSION	
Seq.	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00	**0.65**	0.29	0.79	**0.28**	0.88	0.31	1.29	0.63
01	**1.05**	**0.13**	3.70	0.29	1.44	0.19	2.29	0.42
02	**0.67**	0.23	0.79	**0.22**	0.77	0.28	1.37	0.50
03	**0.70**	0.24	0.98	**0.17**	0.75	0.20	1.47	0.50
04	0.52	**0.15**	1.13	0.19	**0.46**	0.19	1.35	0.72
05	**0.57**	**0.22**	0.76	0.23	0.62	0.26	1.49	0.74
06	**0.60**	**0.23**	1.13	0.28	0.89	0.27	1.35	0.71
07	**0.45**	**0.29**	1.03	0.57	0.89	0.50	1.21	0.90
08	**0.91**	**0.26**	1.11	0.31	1.03	0.31	1.83	0.72
09	**0.75**	0.22	0.96	**0.20**	0.86	0.25	1.82	0.53
10	0.74	0.22	**0.52**	**0.18**	0.62	0.29	2.64	1.01
avg	**0.71**	**0.24**	1.01	0.26	0.87	0.29	1.57	0.63

We would also like to mention the non-rigidity of the KITTI camera rig. While majority of approaches consider the extrinsic parameters constant, for large baseline rigs, such as the one used in KITTI, the assumption of constant rigid transformation between the cameras might need to be revisited. From the KITTI setup photograph, one can notice relatively large and heavy lenses. During the drive, inertial forces act on these lenses, create torque thus tilting the cameras – which is more prominent for the right camera since it does not have a vertical support like the left one. Instead, it resides on a loose end of the profile and is subject to additional dislocation due to profile flexing. As a consequence, during strong turns, the right camera tilts more than the left one. Given that, the angle between the cameras changes during the turn and all objects in view are triangulated with a bias, which propagates further to the resulting camera translation. In our case, SOFT2 addresses this issue by estimating online the extrinsic rotation parameters, and the effectiveness is most easily observable on the KITTI sequence 01 – highway. Figure 9a shows extrinsic angles estimated during the sequence with r_x, r_y, and r_z representing rotations about camera's right-hand x, y, and z axes, where the z axis is along the direction of the camera's optical axis. The most important angle is r_y, since it directly defines the triangle from the baseline to the object, which defines camera-to-object distance and consequently the magnitude of camera translation, i.e., the scale. Angle r_y increases by approximately 0.1 deg during the first turn at the beginning of the sequence, and then increases again twice at the end of the sequence corresponding to two sharp turns at the highway exit. Figure 9b shows the odometry path error. At the first road curve, frames 0 to 100, the traveled path is constantly overestimated due to increased r_y and both SOFT2 without the extrinsic estimator and ORB-SLAM2, the second best on this sequence, result in longer path traveled compared to the ground truth path. Then, over a nearly straight drive the scale is underestimated and both algorithms result in shorter path than the ground truth. This is due to errors in calibration and correction of this effect is out of the scope of this paper. Thereafter, during two strong turns at the highway exit, scale is overestimated and the traveled path length rapidly increases. On the other hand, SOFT2 with the activated extrinsic estimator is not affected by the change in r_y, resulting with nearly linear error curve during the whole sequence. Naturally, our estimator cannot correct for the initial errors in the baseline and intrinsic parameters, hence the sloped error line.

Table II shows SOFT2 results for each train sequence together with ORB-SLAM2, OV2SLAM, and VINS-FUSION. Note that VINS-FUSION results are not in proportion to the result achieved on the official KITTI rank list – in our tests, the error was about 50% larger than expected. We tried to improve the results by modifying the default parameters: increasing the number of features, solver time and solver iterations, but the results were similar. Also, VINS-FUSION often has initialization problems, which causes larger trajectory error at the beginning *(This is reported as Issue #84: Reproducing KITTI Stereo result)*. Although OV2SLAM is the most accurate algorithm out of the three on the KITTI rank list, in our tests it is outperformed by ORB-SLAM2 on the KITTI training sequences. However, on the KITTI-360 training sequences, OV2SLAM performs much better than ORB-SLAM2. The reason behind this could be different robustness of algorithms to challenges of the KITTI dataset calibration. In conclusion, although different, all the approaches exhibited qualitatively similar errors on each KITTI train sequence; however, as can be seen from Table II, our approach achieved the lowest error on average. SOFT2 has the most advantage over the competing algorithms on the highway scene – sequence 01.



(a) Málaga extracted sequence 05 (b) Málaga extracted sequence 08 (c) Málaga extracted sequence 11

Fig. 10: Estimated trajectories along with the ground truth for three Málaga sequences.

TABLE III: Experimental evaluation on the Málaga dataset

				SOFT2	OV2-SLAM	ORB-SLAM2	VINS-FUSION
Seq.	start frame	stop frame	length [m]	t_{ate} [m]	t_{ate} [m]	t_{ate} [m]	t_{ate} [m]
05	1261229234.473050	1261229424.206370	1696.45	**4.13**	13.95	28.41	15.21
06	1261229737.377720	1261229936.379589	1133.92	**9.00**	10.46	14.44	18.59
07	1261229995.330167	1261230073.780891	643.33	**4.35**	4.97	8.14	8.52
08	1261230087.681000	1261230569.035530	4673.49	**15.41**	38.34	162.28	186.42
10	1261230734.837048	1261231591.245032	5814.70	**14.57**	19.45	18.79	26.83
11	1261231654.045614	1261231796.996934	2148.00	**10.49**	59.91	444.85	111.50

C. *Málaga dataset*

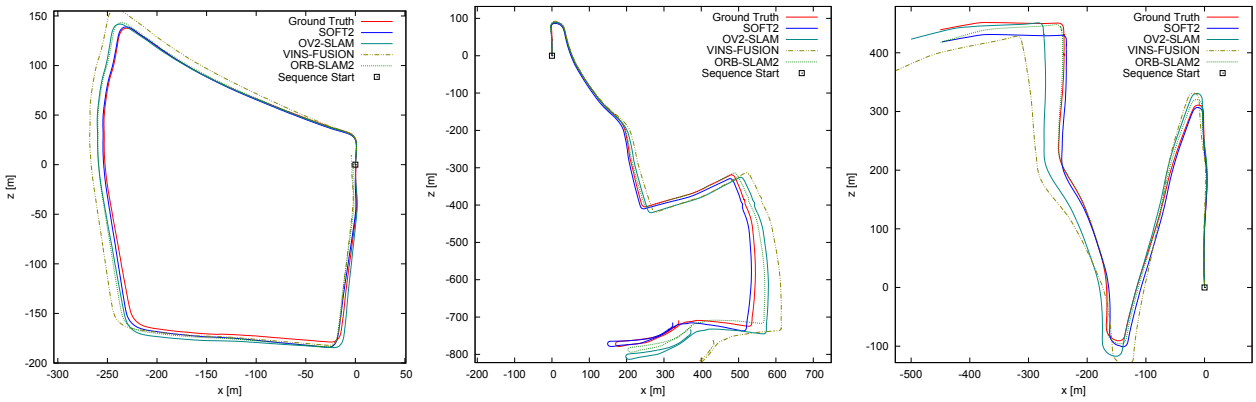
The Málaga dataset was recorded in an urban environment of Málaga with a Point Grey Bumblebee 2 stereo camera having resolution of 1024×768 pixels and frame rate of 20 Hz. Bumblebee 2 ensures precise synchronization between the stereo pair with automatic gain and white balance control [54]. However, Bumblebee 2 dynamic range in conjunction with auto-exposure algorithm were inadequate for outdoor conditions, leaving the image often over or under exposed with blooming and vertical smear artifacts. These effects significantly deteriorate odometry accuracy and in extreme cases they leave no room to detect features. The ground truth trajectory was recorded with a low cost Xsens MTi-28A53G35 yielding position data at 1 Hz with errors of up to 20 meters, which is nowadays inadequate for estimating odometry error with KITTI-like metrics. While the position data could still provide good reference over longer trajectory distance, the orientation data is below the needed accuracy, especially the heading angle which drifts ± 45 deg from the actual value during the drive. That prevented us from using the relative metric for odometry evaluation, since the results would reflect only the sensor heading erratic behavior.

Given that, for the Málaga dataset we evaluated the odometry with the absolute metric which does not depend on the vehicle orientation. We used the dataset extracts that contain loop closure, and carefully selected starting and ending frame for each extract, thus choosing the pair with approximately the same pose. We also evaluated sequence extract 11, since it is the only one that contains highway with the vehicle driving at 70 km/h. This extract is very challenging, since there are almost no close features except for the ones residing on moving traffic. Large amount of outliers were present in SOFT2, while ORB-SLAM2 completely lost track at one moment. In general, the best results for ORB-SLAM2 were obtained with setting *minThFAST* to 3 due to low contrast on the road. We also believe that the lack of near FAST features is the reason for under-scaled ORB-SLAM2 trajectory. In general, all algorithms had issues dealing with sudden change in exposure due to direct sunlight and moving vehicles occupying larger part of the image. These effects are the cause behind the unexpectedly large error values in Table III (VINS-FUSION also had problems with initialization on sequence 11). Evaluation was performed on rectified 1024×568 images – we cut out the upper part of the image since it contained mainly the sky and just slowed down the evaluation process.

The main challenge in obtaining state-of-the-art results on the Málaga dataset lies in the low dynamic range of the camera. This is further exacerbated by the fact that cameras are slightly tilted upwards (in order to remove the car chassis from the view), making the sky dominate in the image and dictating the overall exposure, thus frequently leaving road, the only close and static object in the scene, completely dark. Table III shows the results obtained on the Málaga dataset, while Fig. 10 plots the results for three extracted sequences.

D. *Oxford Robotics Car dataset*

The Oxford dataset was recorded in central Oxford with a Point Grey Bumblebee XB3 trinocular stereo camera having resolution of 1280×960 pixels and frame rate of 16 Hz [55].



(a) Segment from 2014-05-14-13-59-05 (b) Segment from 2014-05-19-13-20-57 (c) Segment from 2014-11-14-16-34-33

Fig. 11: Estimated trajectories along with the ground truth for three Oxford segments.

TABLE IV: Experimental evaluation for the Oxford dataset ($t_{\text{rel}}[\%]$, r_{rel}[deg/100 m])

				SOFT2		OV2-SLAM		ORB-SLAM2		VINS-FUSION	
dataset	start frame	stop frame	offset [sec]	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
2014-05-14-13-59-05	1400075963932033	1400076150344330	12.20	**1.35**	**0.97**	2.74	1.03	2.15	0.98	4.74	1.33
2014-05-19-12-51-39	1400503987511809	1400504417267370	13.18	**1.52**	0.71	2.86	**0.70**	2.69	0.96	6.06	1.97
2014-05-19-13-20-57	1400505700597042	1400506098919265	13.25	**1.84**	0.92	3.62	**0.91**	2.18	1.05	8.63	2.77
2014-11-14-16-34-33	1415985043842007	1415985331240621	0.0	**2.60**	1.10	5.73	**0.85**	7.81	3.73	7.23	1.51
			avg	**1.82**	0.89	3.68	**0.83**	3.54	1.57	6.93	2.04

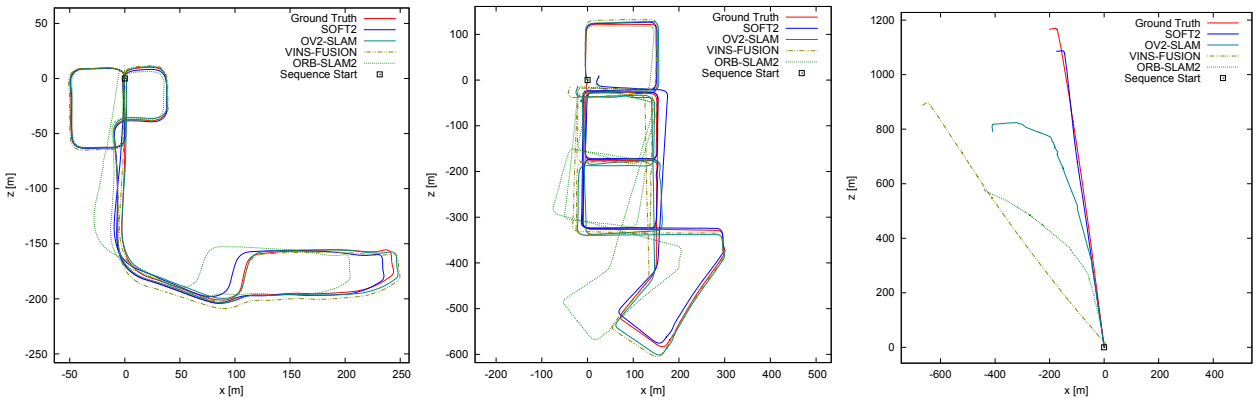
Therein a good contrast of the road and nearby objects was achieved by ignoring the sky and the authors implemented a custom auto-exposure algorithm. However, the resulting exposure can be very unstable with constant oscillations and overexposed images, which is sometimes so strong that several subsequent frames turn-out completely white. Unfortunately, this impedes evaluation using complete Oxford individual sequence, since it is impossible for any visual odometry algorithm to reconstruct the trajectory during the oversaturated segments. The pose of the vehicle was recorded with the NovAtel SPAN-CPT ALIGN inertial and GPS navigation system at 50 Hz. Although the measured poses are more accurate than on the Málaga dataset, the quality of GPS reception and the accuracy of the fused INS solution varied significantly, and authors do not recommend using them directly as ground truth for benchmarking localization algorithms [55]. Furthermore, many segments contain time offset between GPS/INS and camera which have to be determined by the user, and some contain invalid GPS/INS timestamps (e.g. 2014-8-11-10-22-21). Consequently, we had to search a subset of relatively complete segments containing valid data, good GPS/INS trajectory, and good image quality. We extracted two out of the first twenty sequences and used three segments for evaluation interrupted by two periods of oversaturation and one subsegment from a night drive.

Images were recorded in raw format and rectification look-up tables were provided. For our experiments, we used the outer left-right pair with the baseline of 24 cm. We cut out 140 pixels both from the top and bottom of the image to remove the sky and the car chassis. By examining Fig. 11a we can notice overestimated trajectory scale in all the competing algorithms, which might might suggest inaccuracies in calibration. SOFT2 extrinsic estimator reveals slight tilt between the stereo pair and keeps the scale relatively accurate. This is one of the reasons why SOFT2 outperforms other algorithms in translation accuracy, although OV2SLAM has slightly better rotation accuracy. By comparing to KITTI, we can notice that the error is several times larger, suggesting probably less accurate GPS/INS poses, calibration or both. Table IV shows the final results obtained on the selected segments, while Fig. 11 plots the results for three specific segments.

E. MVSEC dataset

The MVSEC dataset is primarily targeted at stereo event cameras, but it also includes the Skybotix VI-sensor which we used for our evaluation [23]. The VI-sensor is an integrated stereo camera with image resolution of 752×480 pixels, 11 cm baseline, and frame rate of 20 Hz. Ground truth was generated by fusing IMU data with lidar loop closings from the Cartographer in 2D. Since full 6DOF pose is not available, we evaluate odometry with the absolute metric only.

Images were recorded in raw format and calibration parameters are provided. The VI-sensor is not ideal for outdoor conditions, due to small resolution and baseline. Part of the image is also in this case occupied by the car chassis, making the usable image area even smaller. We cut out this part of the image since the chassis is glossy and reflects features from the environment introducing large amount of outliers. Similar to the Málaga dataset, sky dictates the auto-exposure algorithm often leaving the road and all nearby objects completely dark, thus significantly lowering the odometry accuracy. Table V shows the absolute translation errors for MVSEC outdoor



(a) Outdoor day 1 (b) Outdoor day2 (c) Outdoor night 1

Fig. 12: Estimated trajectories along with the ground truth for three MVSEC sequences.

tracks, while Fig. 12 shows an example of estimated trajectories. We can notice reduced accuracy for the night drives due to lack of features. Fig. 12c shows that VINS-FUSION had problems with initialization on that track, but nevertheless achieved the most accurate score in the end.

VII. Conclusion

In this paper we have presented SOFT2 – a stereo visual odometry approach targeted for high accuracy and real-time road vehicle localization. SOFT2 is based on several principles, namely it exploits constraints imposed by the epipolar geometry and vehicle kinematics. Specifically, throughout the whole pipeline we rely on minimizing the point-to-epipolar line distance and rely on single camera images for as much as possible (in our case the left camera). This way, we alleviate any issues related to stereo calibration that are present in the system or occur during operation. Second, we utilize right camera images to jointly estimate the scale and extrinsic rotation with respect to the left camera, thus performing also online extrinsic calibration at each step. Third, we smooth motion estimates in a temporal window of frames by using the proposed epipolar line bundle adjustment procedure, leading to more accurate and robust odometry. Note that these assumptions also introduce a limitation for cases when cameras can exhibit pure or close-to-pure rotations, i.e., when the trajectory does not have a large translational component in the overall motion. In such cases and datasets, e.g., hand-held or aerial vehicle applications, SOFT2 is not expected to perform well. We decided to sacrifice generality in order to obtain gain in accuracy for road vehicle applications. Furthermore, the back-end of SOFT2 omits 3D projections to avoid depth uncertainty and consequently does not offer mapping and loop closing – it is strictly an odometry algorithm. In the front-end part, we introduce a multiple hypothesis feature matching method for self-similar planar textures that accounts for changes in feature appearance due to varying perspective – this has the most effect on the road features affecting the accuracy of translation estimation. In the experiments we tested and compared SOFT2 to ORB-SLAM2, OV2SLAM, and VINF-

TABLE V: Experimental evaluation for the MVSEC dataset

	SOFT2	OV2-SLAM	ORB-SLAM2	VINS-FUSION
outdoor	t_{ate} [m]	t_{ate} [m]	t_{ate} [m]	t_{ate} [m]
day1	4.22	**2.62**	13.93	5.38
day2	**8.55**	10.95	17.58	16.04
night1	25.71	87.84	136.26	**23.13**
night2	**14.43**	17.98	34.50	48.44
night3	**14.21**	15.71	37.47	30.85

FUSION on five different public datasets: KITTI-360 dataset, KITTI train sequences, the Málaga Urban dataset, Oxford Robotics Car dataset, and Multivehicle Stereo Event Camera dataset. Finally, we evaluated SOFT2 on the KITTI benchmark and SOFT2 is currently the highest ranking algorithm on the scoreboard achieving $t_{\text{rel}} = 0.53\,\%$ and $r_{\text{rel}} = 0.0009\,\text{deg/m}$ errors, outperforming even 3D laser based approaches.

References

[1] D. Nister, O. Naroditsky, and J. Bergen, "Visual odometry," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[2] M. A. Fischler and R. C. Bolles, "Random sample consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[3] J. Hedborg and M. Felsberg, "Fast iterative five point relative pose estimation," in *IEEE Workshop on Robot Vision (WORV)*, 2013.

[4] D. Scaramuzza and F. Fraundorfer, "Visual odometry: Part I: The First 30 Years and Fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[5] Y. Ma, S. Soatto, J. Košecka, and S. Satry, *An Invitation to 3D Vision*. Springer, 2004.

[6] F. Fraundorfer and D. Scaramuzza, "Visual odometry part II: matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[7] K. Konolige and M. Agrawal, "FrameSLAM: From bundle adjustment to real-time visual mapping," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1066–1077, 2008.

[8] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-Based Visual-Inertial SLAM Using Nonlinear Optimization," in *Robotics: Science and Systems*, 2013, p. 8.

[9] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," *2015 European Conference on Mobile Robots, ECMR 2015 - Proceedings*, pp. 0–5, 2015.

[10] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

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

[12] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Direct monocular SLAM," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 834–849.

[13] J. Zubizarreta, I. Aguinaga, and J. M. Montiel, "Direct sparse mapping," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1363–1370, 2020.

[14] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semi-direct visual odometry for monocular and multi-camera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 1–17, 2016.

[15] Y. Zhao and P. A. Vela, "Good Feature Matching: Toward Accurate, Robust VO/VSLAM with Low Latency," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 657–675, 2020.

[16] D. Scaramuzza, "1-point-RANSAC structure from motion for vehicle-mounted cameras by exploiting non-holonomic constraints," *International Journal of Computer Vision*, vol. 95, no. 1, pp. 74–85, 2011.

[17] J. Delmerico and D. Scaramuzza, "A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 2502–2509.

[18] M. Klodt and A. Vedaldi, "Supervising the new with the old: Learning SFM from SFM," in *Lecture Notes in Computer Science*, 2018, pp. 713–728.

[19] S. Wang, R. Clark, H. Wen, and N. Trigoni, "End-to-End, Sequence-to-Sequence Probabilistic Visual Odometry through Deep Neural Networks," *International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 513–542, 2018.

[20] N. Yang, L. Von Stumberg, R. Wang, and D. Cremers, "D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 1278–1289.

[21] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1885–1892.

[22] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "SOFT-SLAM: Computationally Efficient Stereo Visual SLAM for Autonomous UAVs," *Journal of Field Robotics*, vol. 35, no. 4, pp. 578–595, 2018.

[23] A. Z. Zhu, Y. Chen, and K. Daniilidis, "Realtime time synchronized event-based stereo," in *European Conference on Computer Vision (ECCV)*, 2018.

[24] Y. Zhou, G. Gallego, and S. Shen, "Event-based Stereo Visual Odometry," *arXiv*, 2020.

[25] W. Huang, H. Liu, and W. Wan, "An Online Initialization and Self-Calibration Method for Stereo Visual-Inertial Odometry," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1153–1170, 2020.

[26] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Transactions on Robotics*, vol. 35, no. 2, pp. 314 – 333, 2020.

[27] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[28] R. Wang, M. Schwörer, and D. Cremers, "Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras," in *International Conference on Computer Vision (ICCV)*, 2017, pp. 3903–3911.

[29] J. Zhu, "Image gradient-based joint direct visual odometry for stereo camera," in *IJCAI International Joint Conference on Artificial Intelligence*, 2017, pp. 4558–4564.

[30] K. Lenac, J. Ćesić, I. Marković, and I. Petrović, "Exactly Sparse Delayed State Filter on Lie groups for Long-term Pose Graph SLAM," *International Journal of Robotics Research*, vol. 37, no. 6, pp. 585–610, 2018.

[31] M. Buczko, V. Willert, J. Schwehr, and J. Adamy, "Self-Validation for Automotive Visual Odometry," in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 573–578.

[32] P. Bénet and A. Guinamard, "Robust and Accurate Deterministic Visual Odometry," in *International Technical Meeting of the Satellite Division of The Institute of Navigation (ION + GNSS)*, 2020, pp. 2260 – 2271.

[33] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, and G. Lebesnerais, "OV^{2}SLAM : A Fully Online and Versatile Visual SLAM for Real-Time Applications," *IEEE Robotics and Automation Letters*, 2021.

[34] J. Xie, M. Kiefel, M. T. Sun, and A. Geiger, "Semantic Instance Annotation of Street Scenes by 3D to 2D Label Transfer," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016-Decem, pp. 3688–3697, 2016.

[35] J.-L. Blanco-Claraco, F.-Á. Moreno-Duenas, and J. González-Jiménez, "The Malaga Urban Dataset : High-rate Stereo and Lidars in a realistic urban scenario," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 207–214, 2013.

[36] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[37] A. Z. Zhu, D. Thakur, T. Özaslan, B. Pfrommer, V. Kumar, and K. Daniilidis, "The Multi Vehicle Stereo Event Camera Dataset: An Event Camera Dataset for 3D Perception," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2032–2039, 2018.

[38] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3213–3223.

[39] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 5000–5009.

[40] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape Open Dataset for Autonomous Driving and Its Application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2702–2719, 2020.

[41] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "Nuscenes: A multimodal dataset for autonomous driving," in *EEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 618–11 628.

[42] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2633–2642.

[43] T. Qin and S. Shen, "Online Temporal Calibration for Monocular Visual-Inertial Systems," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 3662–3669.

[44] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

[45] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2013, pp. 1280–1286.

[46] E. B. Dam and M. Lillholm, "Quaternions, Interpolation and Animation," in *Technical Report DIKU-TR-98/5*, 1998.

[47] D. Detone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 337–349.

[48] Y. Ono, P. Fua, E. Trulls, and K. M. Yi, "LF-Net: Learning local features from images," in *Advances in Neural Information Processing Systems*, 2018, pp. 6234–6244.

[49] M. Dusmanu, I. Rocco, T. Pajdla, M. Pollefeys, J. Sivic, A. Torii, and T. Sattler, "D2-net: A trainable CNN for joint description and detection of local features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8084–8093.

[50] J. Shi and C. Tomasi, "Good features to track," in *IEEE International Conference on Computer Vision and Patter Recognition (CVPR)*, 1994.

[51] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.

[52] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[53] I. Cvisic, I. Markovic, and I. Petrovic, "Recalibrating the KITTI dataset camera setup for improved odometry accuracy," in *European Conference on Mobile Robots (ECMR)*, 2021.

[54] G.-J. J. Blanco-Claraco J-L, Moreno-Dueñas F-Á, "The málaga urban dataset: High-rate stereo and lidar in a realistic urban scenario," *The International Journal of Robotics Research*, no. 33(2), pp. 207–214, 2014.

[55] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.