Racetrack Rolling-shutter Stereo Visual Odometry and Dataset

Igor Cvišić[1], Ivan Marković[1], Sacha Vražić[2], and Ivan Petrović[1]

***Abstract*—In this paper we present a novel vehicle localization dataset captured with a pair of stereo rolling shutter cameras mounted on top a race car while driving at speeds of more than 200 km/h. The dataset contains six sequences recorded on two different racetrack coupled with accurate ground truth trajectories recorded with a real-time kinematics GNSS system. We make the dataset publicly available to the research community along with raw calibration sequences and our estimated calibration parameters. Besides the dataset, in the paper we also propose an extended version of our previously published stereo odometry, SOFT2, to the case of rolling shutter stereo cameras. The approach successfully models the rolling shutter effect and addresses the challenges both in the front- and back-end part of the odometry due to the line-by-line scanning nature of the sensor during image capture. Specifically, for the former the challenge of feature matching in a texture-poor environment of a racetrack, while for the later the challenge of not having a unique essential matrix for all feature correspondences like in the global shutter case. The proposed odometry is evaluated on the six sequences and achieves accuracy on par with the state-of-the-art methods of the KITTI benchmark.**

The dataset is available at `https://unizgfer-lamor.github.io/rt-dataset`.

I. INTRODUCTION

Accurate autonomous vehicle localization has been a significant driving force for development of visual state estimation and localization algorithms, especially for challenging dynamic urban environments. Although the majority of the driving area is usually covered with a GNSS signal, these signals are often affected by multipath errors, can be biased due to weather conditions, or the signal can be inaccessible in GNSS denied areas like tunnels and similar. Moreover, accurate systems that also estimate vehicle orientation can be quite costly due to high-grade gyroscopic sensors. Higher localization accuracy can be obtained by using perception sensors like LiDARS, cameras, and high-resolution radars (4D radars). LiDARs have the advantage of providing ranges to obstacles with high resolution yielding indispensable information for both localization and moving object tracking, but compared to cameras can be bulkier, more costly or work on a lower frame rate. On the other hand, cameras are a rich source of information and localization-wise can provide results at the level of, or even better, than LiDARs making cameras an attractive sensor modality and visual localization has been a subject of research for multiple decades [1]–[3].

Nevertheless, for dynamic urban environments requiring accurate localization and tracking of dynamic objects, the fusion of the two sensors presents the most prudent choice for robustness and safety reasons. However, there exists another area of vehicle localization that has seldom been explored – specifically, the racetrack. Although the racetrack has some alleviating conditions compared to the city, like not having to contend with many, or any at all, traffic participants, main challenges herein arise from the fact that racetracks are often poor in features, especially those that are close to the camera, most of the time the vehicle is in open space, and the vehicle is driving at very high speeds and eventually also drifting on the road. Furthermore, for cost reasons, in the automotive industry camera setups are currently based on rolling shutter cameras that acquire the image in a line-by-line scanning fashion instead of a single snapshot like global shutter cameras. All this presents significant challenges for developing an accurate visual localization algorithm for sports vehicles driving on a racetrack.



Fig. 1: Vehicle with the mounted sensor setup that was used to collect the Rimac Technology dataset.

In the robotics and autonomous vehicles dedicated datasets on which researchers can test and evaluate localization algorithms has tremendously accelerated the development of this research topic. Such dataset usually provide measurement from a suite of sensors complemented by a high-accuracy real-time kinematics (RTK) GNNS reference system for ground truth. Starting with the KITTI dataset [4], that has served as a public benchmark for road vehicles since 2012, leading to other examples such as the KITTI-360 dataset [5], the Málaga Urban dataset [6], the Oxford Robotics Car dataset [7], and the Multivehicle Stereo Event Camera Dataset (MVSEC) [8]. Regarding visual localization, multiple stereo vision methods achieve less than 1% translation error [9]–[16] on the KITTI public benchmark, demonstrating the ability of cameras to produce highly accurate road vehicle trajectories. With rare exceptions, most datasets are recorded in urban environments at low or moderate velocities, exception being the recent RACECAR dataset [17], wherein authors have provided a

* This research has been funded by Rimac Technology Ltd. under the research agreement "Visual Localization of Sports Cars."

[1] University of Zagreb Faculty of Electrical Engineering and Computing, Laboratory for Autonomous Systems and Mobile Robotics, Croatia. {name.surname}@fer.unizg.hr

[2] DriveAhead.ai, with Rimac Technology at the time of this work. sacha.vrazic@driveahead.ai

dataset recorded by autonomous Indy race cars driving at speeds of up to 270 km/h. The sensor suite is composed of LiDARs, global-shutter cameras, and a radar sensors with RTK GNSS for ground truth, since in this case the application also includes other race ars simultaneously on the racetrack. In total, the dataset contains 11 scenarios with 6.5 hours of data recorded on the racetrack including also benchmarks for object detection and tracking.

However, most existing datasets, especially urban and racetrack versions, are recorded with global shutter cameras (or special event-based cameras), while majority of consumer devices, including also off-the-shelf automotive cameras, incorporate CMOS sensors with electronic rolling shutter. CMOS sensors are more economic and have effectively larger sensitivity as they can continue gathering photons while reading the sensor data, which allows them to use a smaller sensor and lens. This makes them attractive not only for consumer, but also for automotive applications, as a suite of such sensors can be inconspicuously installed on vehicles taking into account the aesthetic constraints of vehicles as well. Regarding rolling-shutter datasets with ground-truth for localization, in [18] authors recorded a monocular rolling shutter dataset synchronized with an IMU and a global shutter camera in a handheld device. In [19] authors presented an IMU-assisted correction of the rolling-shutter effect within the direct sparse odometry (DSO) framework using a deep neural network to predict depth and row-wise pose. The paper also provides an indoor dataset that was used for evaluating the algorithm. A synthetic indoor rolling shutter camera dataset was presented in [20] that includes both a rolling shutter camera and an IMU simulated measurements, as well as time synchronized global shutter images. In [21] authors extended the indoor dataset presented in [22] for a dense RGB-D rolling shutter odometry and mapping algorithm. The CityScapes dataset [23] contains urban recordings with industrial-level rolling shutter cameras, but is more oriented towards visual understanding of complex urban scenes, and although contains GPS measurements, their accuracy is not intended to benchmark localization algorithms [24]. Nevertheless, none of the existing dataset provide a stereo rolling shutter setup with RTK GNSS for vehicles with high speeds and highly dynamic driving, such as encountered on a racetrack.

Regarding the rolling shutter effect and visual odometry, in [24] author discuss challenges in monocular visual odometry including the rolling shutter effect, among others, as image pixels are exposed at different timestamps and can introduce non-trivial errors to a visual localization system. Therein, authors have tested direct, feature-based, and semi-direct visual odometry methods. The work in [25] presents a semantic SLAM approach with a rolling shutter camera and an IMU in urban environments with emphasis on leveraging semantic information, such as lane boundaries, traffic signs, and road markings to enhance localization accuracy. In [26] authors proposed a visual-inertial odometry based on EKF that uses an IMU to correct point features in the rolling-shutter images to new pixel coordinates in the global shutter view (middle of rolling-shutter read-out time) by transfer along the epipolar line. Direct monocular visual odometry was extended to the rolling shutter in [27] exhibiting significant improvement in both the estimated trajectory and reconstructed 3D environment. In [28] authors analyzed the problem of estimating the essential matrix for the case of rolling shutter cameras and proposed pertaining linear and nonlinear solvers.

In this paper we present and make publicly available the Rimac Technology (RT) dataset that was recorded on two different professional racetracks (one being a FIA certified track) with a stereo rolling shutter camera setup and an RTK GNSS ground truth as shown in Fig. 1. The dataset contains six sequences, three per racetrack, with maximal speeds exceeding 200 km/h. With the dataset we also provide our calibration parameters, but also calibration sequence images that enable users to conduct their own calibration if needed. Besides the dataset, we also propose an extension of our previously published stereo odometry SOFT2 [29], currently ranking first on the KITTI benchmark, to the stereo rolling shutter case and evaluate it on the presented dataset. The results show on par performance with top ranking solutions on the KITTI benchmark.

II. The Rimac Technology Dataset

The RT dataset is a subset of recordings during sports driving by a professional driver on racetracks NAVAK Centre, Serbia [30] and Automotodrom Grobnik, Croatia [31]. There were no other vehicles on the tracks during the data collection. Images were captured with a pair of NVidia rolling shutter automotive cameras at a frame rate of 30 Hz, synchronized up to several microseconds placed just above the rear-view mirror as can be seen in Fig. 1. We assume existence of a slight inter frame delay, i.e., an image was captured during approximately 98% of the image line scanning period. In the setup, stereo baseline was 12 cm, images were of 1920x1208 pixel resolution, while the lenses had a field of view of 60 degrees. The ground truth was recorded with the SBG Ellipse-D RTK inertial navigation system with a dual antenna GNSS device running at 200 Hz.

The Automotodrom Grobnik track. Automotodrom Grobnik is located near Rijeka, Croatia [31]. The length of the track is 4168 meters with a width of 10 meters. It has 18 turns, 8 left and 10 right ones. The dataset contains three laps, the first lap starting from the pit-stop and ending at the start line, the second lap starting and ending at the start line, enabling loop closing, while the last lap start at the start line and ends at the pit-stop. Fig. 2 shows the aerial photograph of the track.

The NAVAK Centre track. The NAVAK Centre is located 40 km away from Belgrade, Serbia [30]. The Dataset was recorded in 2020 on a configuration today known as the Old NAVAK (see Fig. 3), since the racetrack has been extended with a new segments in the meantime, and today contains several possible configurations for driving. The Old NAVAK had a length of 2950 meters with a minimum track width of 10 meters, allowing cars to reach the speed of more than 200 km/h on some track segments with our vehicle. The dataset



Fig. 2: The Automotodrom Grobnik track.



Fig. 3: The "Old" NAVAK Center track.

contains three with somewhat varying speed profiles, but at the maximum of speed allowed by the enabled by the track. Fig. 3 shows the aerial photograph of the track.

A. Sensor setup calibration

Prior to each drive, a standard checkerboard with 7 by 10 squares of size 9×9 cm is recorded with the stereo camera setup mounted on the race car. For camera calibration we used the method presented in our previous work [29], with radial-tangential five-parameter distortion model. The radial-tangential model was selected due to relatively narrow lens view angle of 60 degrees. We also considered the influence of the rolling shutter effect to camera calibration. Since the checkerboard was moved in front of the camera, corners in the lower part of the image moved relative to the upper ones, so the board geometry and a priori known inter-corner distances were violated. To account for a moving checkerboard, beside the pose of the each board, we also estimated the twist – translational and rotational velocity of the board during the image capture, increasing the degrees of freedom from 6 to 12 per board. By modeling the rolling-shutter effect during the calibration procedure, we managed to reduce average pixel reprojection error from 0.15 pixels per corner to 0.10 pixels per corner. However, calibration parameters obtained with this method did not enhance the odometry accuracy in the end. It is possible that the data was overfitting to the motion parameters. Given that, we provide the checkerboard images with the dataset, so that users can also test their own calibration approach, besides the calibration parameters obtained with our method for reference.

The relative translation of the camera with respect to the RTK GPS frame was determined by manual measurements, while the relative rotation was obtained by aligning the RTK GPS trajectory with the trajectory of the proposed odometry method. Note that the resulting transform depends on the accuracy of the used odometry and also on the obtained calibration parameters. In the dataset, we also provide the transform that we used for odometry evaluation.

III. SOFT2 with Rolling Shutter Cameras

In the following we describe challenges that rolling shutter cameras pose for stereo motion estimation and describe extensions needed to be implemented within our previous global shutter based approach SOFT2 [32]. Notably, challenges are present both in the front-end, relating to feature matching, and the back-end, related to essential matrix and motion estimation.

A. Feature matching

One of the largest challenges for developing highly-accurate odometry for the RT dataset was feature matching, not just because of the rolling shutter effect, but also because of the feature-poor and homogeneous texture environment. Specifically, all features that are close to the camera, which are crucial for resolving the translational part of the estimated motion, reside on the road. There are no distinctive nor unique features, and the texture is low and self-similar. Fig. 4 shows images of the asphalt on the Grobnik Autodrom track (top panel) and the NAVAK track (bottom panel). The Grobnik Autodrom track asphalt was much smoother, had almost no texture, with traces of tires evolving from lap to lap making feature detection and matching very difficult. The NAVAK track is slightly more textured, but still presents challenges for the classic detection and matching techniques. To ensure enough reliable features for motion estimation, we adopted techniques of perspective projection and ground estimation presented in [32], and extended them for rolling shutter case. To begin with, in a stereo setup, feature matching ambiguity can be significantly reduced by finding correspondences in the left and right camera images first, since the pose between the stereo pair is accurately obtained through the calibration procedure.

1) Left-to-right camera frame feature matching: Since the relative pose between the calibrated stereo pair is a priori known, the epipolar constraint can be used to significantly narrow down the search area. It is well know, that the correspondence for the feature detected in the left image must lay on the corresponding epipolar line in the right image. For the case of a calibrated stereo pair, this is the horizontal line passing through the row-coordinate equal to the row-coordinate of the feature in the other image. We have used this approach in our previous work [32], and therein additionally



Fig. 4: Asphalt texture on the Grobnik (top panel) and NAVAK (bottom panel) tracks. The near homogenity makes feature matching in stereo and time extremely challenging (can also occur on new roads in urban areas).

augmented it with a perspective patch correction and ground plane estimation (for details please confer the original paper, but this is not relevant for the rolling shutter modeling). However, this approach cannot be directly transferred to the rolling shutter case. The difference between the global shutter and rolling shutter image capture is illustrated in Fig. 5, where we can see that global shutter cameras have a unique projection center and timestamps for all features in the image, while in the rolling shutter case as the camera moves each scanline, and features that fall within, will have their own projection center and different timestamps.

When calibrating a stereo camera, besides the lens distortion rectification, the procedure corrects for geometrical alignment of the stereo pair and aligns the image planes. When this is applied, the two rectified image planes are perfectly aligned, and epipolar lines simplify search for correspondence as the corresponding features lay on the same horizontal line and have the common row-coordinate. However, they do not reside on the same line in the original unrectified image, i.e., the row of pixels in the left sensor that captured the feature has a different row than the row of pixels in the right sensor. While this has a zero effect on visual odometry in the global shutter case, in the rolling shutter case that means that the same feature will be captured at different points in time in the left and right camera. And as the camera motion increases, the difference in times when the feature was captured in the left camera and when the corresponding feature was captured in the right camera cannot be neglected any more. Fig. 6 shown an example of this effect on road features during a high-speed drive. The figures shows rectified and aligned left and right images of the stereo pair. Given that, one would expect that the tip of the number '3' in the left image would lie on the horizontal epipolar line in the right image. However, due to a slight physical misalignment in camera mounting, that feature is captured at sensor row 1122 in the left camera, and sensor row 1084 in the right camera, meaning that the feature was captured by the right camera 1.05 ms before it was captured in the left camera. During that time, due to high speed, the race car moved forward and the feature is captured at a different position in the right image than expected and at a different timestamp.

Let us define as t_1 the time the feature is captured in left

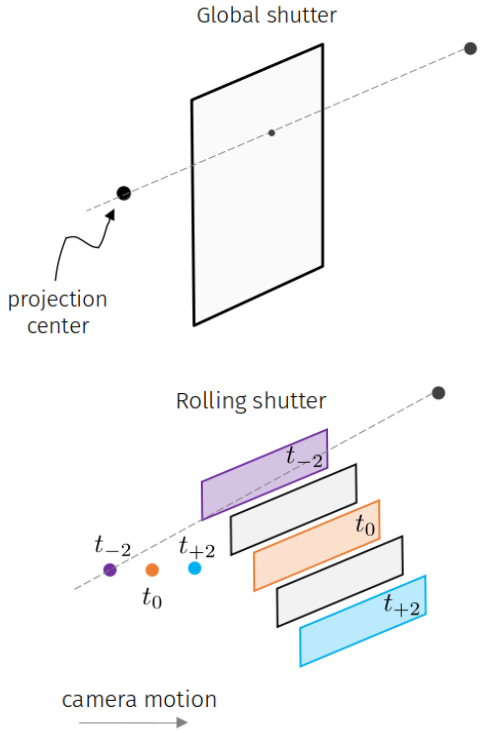


Fig. 5: Global shutter compared to rolling shutter image capture. When the sensor is in motion, each row, i.e., scanline, and pertaining features have different projection centers and different timestamps.

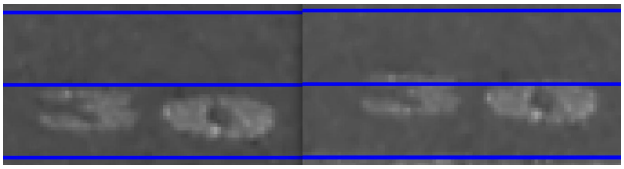


Fig. 6: Images of the left and right rolling shutter camera taken at the same time step. Even though images are rectified and aligned the features exhibit large deviation from the theoretically equal epipolar line.

camera `CamL`, and as t_2 the time when the feature is captured in the right camera `CamR`. The expected feature location in the `CamR` image plane depends on the object's 3D position X in the `CamL` coordinate system. Assuming the constant translational and rotational velocity, here modeled as a twist ξ, one can obtain the expected location of the corresponding feature as

$$x' = \pi(X, \exp(\xi \Delta t) \cdot \texttt{CamL_to_CamR}), \tag{1}$$

where π represents the reprojection function, exp is the matrix exponential mapping the motion displacement to homogeneous tranformation matrix, `CamL_to_CamR` is the relative pose between the cameras, while

$$\Delta t = t_2 - t_1 = \frac{\texttt{row}_2 - \texttt{row}_1}{\texttt{imHeight}} \cdot T, \tag{2}$$

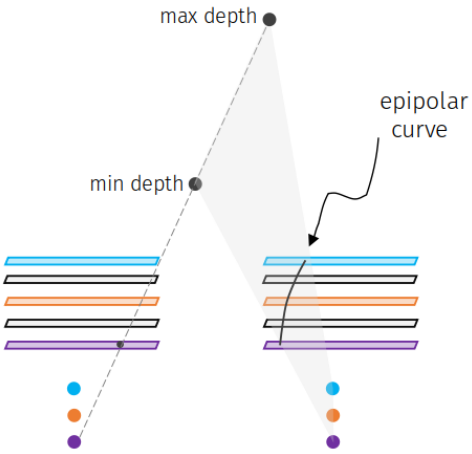


Fig. 7: Possible feature correspondence locations are computed by reprojecting 3D points along the ray emanating from the left camera's feature location, to the right camera – note that each scanline has a different projection center, i.e., it is projected at different points in time.

is the time difference between the two rolling shutter scanlines, computed as the ratio of the row difference and the number of scanlines in an image, and then multiplied by the image sampling period T.

Since the feature depth X is not known a priori, we cannot compute the expected location x' directly from (1), but instead we construct the corresponding *epipolar curve* in the right image by reprojecting 3D points along the ray emanating from the feature in the left image starting from some minimal expected distance to infinity. This process is illustrated in Fig. 7. Note that such a rigorous procedure is not really necessary for correctly matching a distinctive feature such as the number 3 marking on the road. However, such markings are very sparse on the race track, and for correctly matching features on self-similar textures such as the asphalt, epipolar curves constructed in the proposed manner significantly improve the number of correct matches. Fig. 8 shows different shapes of epipolar curves for an asphalt image example taken at different vehicle speeds.

2) Current-to-previous camera frame feature matching: After the left-to-right match is established we can triangulate the 3D point in the `CamL` frame. Again, using the twist ξ, the expected feature location in the previous frames can be predicted. However, we still cannot know the exact time when the feature was captured in the previous frame, i.e, in which scanline the feature could be. We approach this in an iterative manner. In the first iteration, we set $\Delta t = 0$, and the twist is multiplied by T, i.e., the duration of one whole frame, and the expected position in previous left image is determined:

$$x' = \pi(X, \exp(\xi \cdot (T + \Delta t))). \tag{3}$$

From the scanline, i.e., the row, where feature ends up being projected, we read out the time of the row and set the new Δt to be the difference between the duration of one whole frame and the newly read out time. With the newly computed Δt the feature is reprojected again, and the method converges after several iterations providing a precise prediction of the feature location in the previous left camera frame.

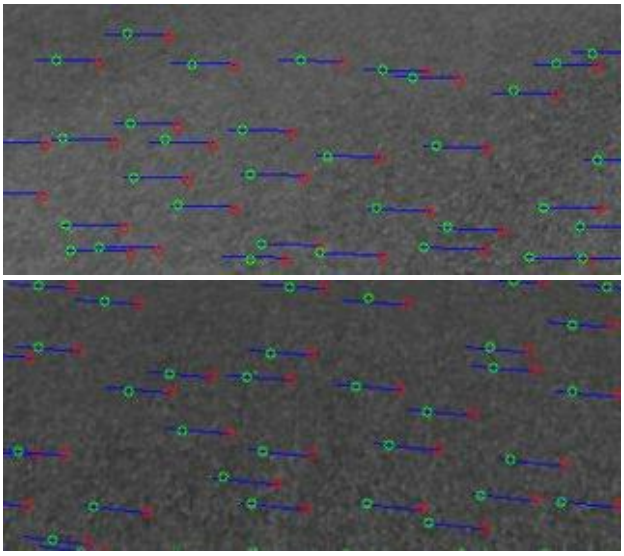


Fig. 8: Rolling shutter feature matching at 30 km/h (top panel) and 200 km/h (bottom panel). Red circles represent left image feature coordinates, while green features are corresponding matches. Epipolar curves are shown in blue. At lower speeds, epipolar curves are nearly horizontal lines and resemble the global shutter case, while at higher speeds the epipolar curves exhibit a much greater slope.

B. Motion estimation

To estimate the motion, we rely on the approach from SOFT2 presented in [32] and extend it to contend with the rolling shutter effect. As in the original version, we estimate first the rotation and translation up to scale by relying only on the left camera images. Then, we use the right camera to estimate the scale. That way, any potential error emerging from a misalignment between the stereo pair will propagate only to scale. Additionally, to reduce the scale error, we jointly estimate in an online fashion relative extrinsic angles between the cameras of the stereo pair, where we consider the left camera to be fixed, and the right camera is freely rotating around its focal point about all the three axis. For simplicity, and without the loss of generality, we align our reference coordinate system with the left camera. Online extrinsics calibration has showed better accuracy than the rigid stereo-rig model, especially for the case of the KITTI dataset as the wide-baseline camera mount can slightly bending during the drive and larger speeds.

Although the RT dataset baseline is relatively small and the mount is rigid, our extrinsic estimator still detected different extrinsic angles during the drive then the ones present during the calibration. The largest discrepancy was along the camera's optical axis, i.e., the rotation around the z-axis of the local right-handed frame, which differed for 0.17 degrees between the calibration time and the drive time. Also, extrinsic angles

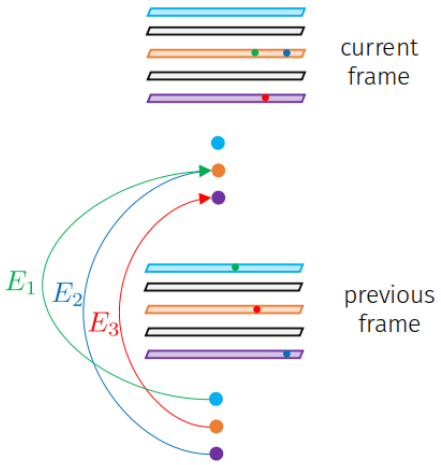


Fig. 9: Rolling shutter motion estimation. For example, the green feature match defines motion from the blue scanline to the orange scanline, i.e., E_1, while the blue feature match defines motion from the purple scanline to the orange scanline, i.e., E_3. Thus for a pair of left camera rolling shutter images taken at different times and during motion, different feature pairs can correspond to different displacements.

slightly change during the drive correlating with the race car speed and and the trajectory, i.e., with external forces acting more on he mount during sharper turns. Note that online estimation of extrinsic angles not only improves the scale accuracy, but it also enables proposed matcher to work correctly, since the search from left to right camera matches is performed only in the near neighborhood of the epipolar curve, whose location highly depends on the extrinsic angles.

1) Iterative essential matrix estimation: Rolling shutter SOFT is based on the essential matrix estimation as well. However, since the essential matrix is constructed from the rotation matrix and up-to-scale translation vector, a single rolling shutter image, unless camera is stationary, does not yield a single essential matrix. This is due to the fact that each scanline has different timestamps and projection centers, thus during motion combinations of features on varying scanlines represent different motion, i.e., essential matrix. This effect is illustrated in Fig. 9. In the following we describe how this effect is taken in to account during motion estimation.

Similaly to SOFT2, we start by minimizing sum of squared point-to-epipolar-line distances to estimate motion parameters ξ:

$$\min_{\xi} \sum_{i} d^2(x_i, l'_i(\xi)) + d^2(x'_i, l_i(\xi)), \tag{4}$$

where

$$d(x, l) = \frac{l^\mathsf{T} x}{\sqrt{l_1^2 + l_2^2}}, \tag{5}$$

and $l'_i(\xi) = E_i(\xi)^\mathsf{T} x'_i$ and $l_i(\xi) = E_i(\xi) x_i$ are epipolar lines associated to points x'_i and x_i in the previous and current view, respectively. What makes (4) differ from the global-shutter case is that each point pair x'_i and x_i has its own corresponding essential matrix E_i. To avoid solving an underconstrainted system, we assume that during the rolling capture process the camera is moving with constant-velocity, as such a motion model showed to yield a good performance-simplicity ratio. Additionally, instead of estimating the essential matrices directly, e.g., by the 5-point method [1], we propose to construct each E_i for optimization in (4) by explicitly computing the E_i as a vector product of the rotation matrix and the translation vector where the state vector now contains the angular velocities and direction of motion:

$$\begin{aligned} E_i(\xi) &= E_i(\omega_x, \omega_y, \omega_z, \theta, \phi) \\ &= \left[\hat{t}(\theta, \phi)\right]_\times R_i(\exp(\omega_x, \omega_y, \omega_z, dt_i)), \end{aligned} \tag{6}$$

where $\left[\hat{t}(\theta, \phi)\right]_\times$ is the skew-symmetric matrix representation of the translation vector. Note that under the constant velocity model each E_i has equal angular velocities but different rotation angles as the timestamps t_i between different scanlines differ. Thus, to obtain the rotation matrix R_i for each feature pair x'_i and x_i, we simply integrate estimated velocities over the time period t_i, which is defined by the matches feature positions in the image, i.e., their respective scanlines. The translation direction is the same for all feature correspondences in the constant velocity model.

C. Scale and extrinsic angles estimation

For scale estimation, we use the same approach as in SOFT2 and estimate the scale jointly with the relative rotation of the right camera with respect to the left camera, i.e., relative stereo extrinsics. In principle we concatenate transforms between each possible camera pairing, i.e., previous right to current left, previous left to current right, and previous right to current right, and compute the final transforms which are then converted to essential matrices that are used to compute point-to-epipolar-line distances. Again, difference to SOFT2 is that each feature pair has its own essential matrix, dependent on the corresponding features positions in the image. The state we estimate contains the scale and the three extrinsic angles.

Fig. 10 show extrinsic angles estimation during a single lap, from which we can notice that they seem to be dependent on the race car speed. Since the camera rig is mounted outside the vehicle on the rooftop, this effect can be due to higher drag forces or stronger cooling due to wind during the higher speeds. Dynamic changes of the extrinsic angles while driving the laps are not as pronounced; however, static change from the time of calibration to the time of drive is more notable. Therefore, the provided calibration files are corrected for mean extrinsic angles estimated with our approach, which better fit the driving part of the dataset.

IV. Experimental Results

We evaluate the proposed solution on three laps from the Grobnik dataset and three laps from the Navak dataset. Since there are no open-source rolling shutter stereo odometry algorithms available, we relate our results via the KITTI metric [4] and compare to state-of-the-art results on the KITTI benchmark. Since KITTI contains segments recorded on a highway,

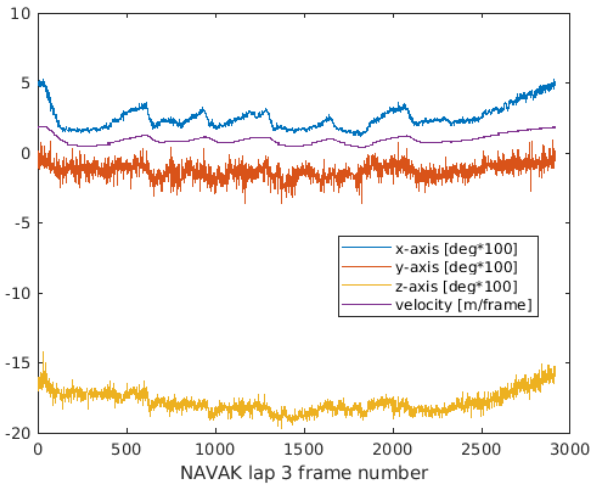


Fig. 10: Stereo extrinsic angles estimation during a single lap of the Navak sequence. We can notice the correlation between the profile of the vehicle speed and the extrinsic angles of the stereo pair. The z axis points in the direction of the camera's optical axis.

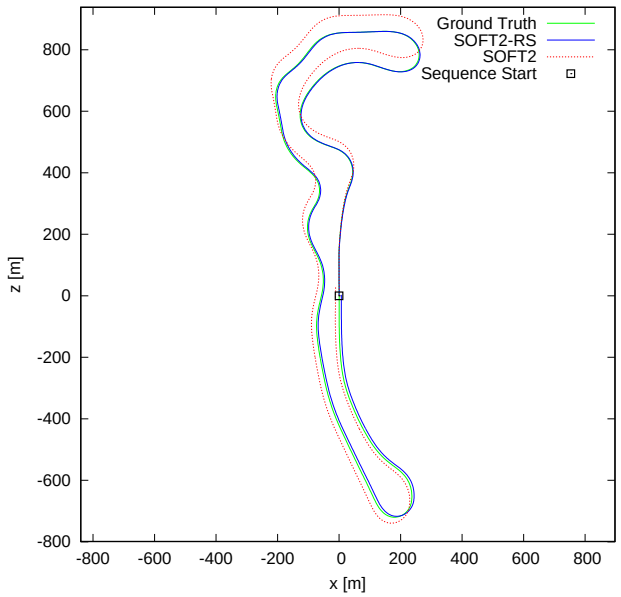


Fig. 11: Odometry evaluation on Lap 2 of the Grobnik sequences.

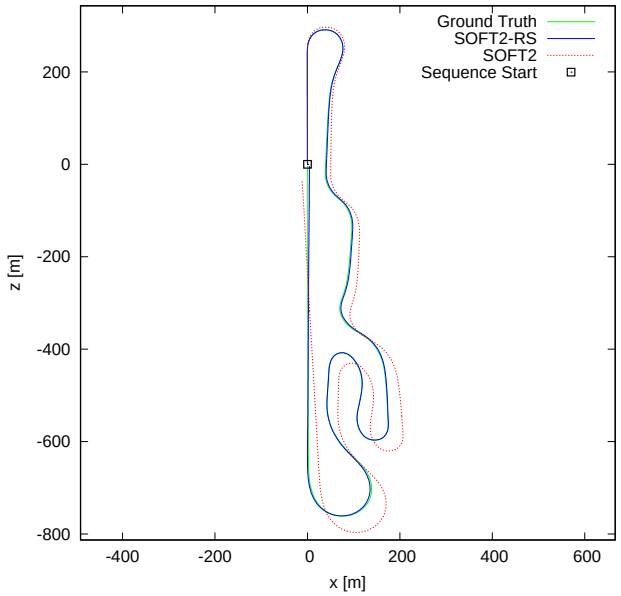


Fig. 12: Odometry evaluation on Lap 2 of the Navak sequences.

one could expect similar translational and rotational errors; nevertheless, odometry requirements for the RT dataset are more challenging due to the rolling-shutter cameras combined with vehicle very high speeds and the lack of distinctive features on the race track. On the other hand, there are no other actors, no dynamic objects in the RT dataset, which is a slight relaxation compared to the KITTI dataset. In Figs. 11 and 12 we can see the original SOFT2, that does not take the rolling shutter effect into account and thus exhibits significant drift, and the extended version along with the ground truth. We also provide the absolute translation root-mean-square error t_{abs} [33] and the results are shown in Table I.

The final average translation result of 0.4% suggests greater accuracy than the odometries listed on the KITTI benchmark. However, one should note that errors are computed respective to a ground truth trajectory recorded by an RTK GNSS systems, which in the case of the KITTI dataset was more prone to the effects of the urban area, while the racetracks of our dataset are in open space enabling less sensitive GNSS localization. Furthermore, ground truth trajectories in the RT dataset used additionally a post-processing kinematics procedure bringing the accuracy to the range of 2-3 cm. Given that, we can assume that the provided ground truth trajectories are quite reliable. Although we cannot make a direct comparison to the above mentioned reasons, an error of 0.4% is in the range of general global shutter state-of-the-art odometries.

V. CONCLUSION

In this paper we have introduced a novel Rimac Technology stereo rolling-shutter camera dataset recorded on a racing track at speeds over 200 km/h. The dataset is publicly available to research community and contains six sequences from two different racetrack coupled with high-accuracy ground truth trajectory for evaluation. Furthermore, we have also presented an extended version of our previously published visual odometry SOFT2 to the case of rolling shutter stereo cameras. The accuracy is evaluated on the proposed dataset with relative and absolute localization metrics and demonstrate on par results with state-of-the-art odometries evaluted on the KITTI benchmarkt, despite the RT dataset being more challenging due to lack of distinctive features and working with rolling shutter images recorded at very high speeds. It the end, we provide means for other researchers to process the dataset, evaluate results, and compare using standard odometry

TABLE I: Visual odometry results for the RT dataset. We used the KITTI metric for the translational $t_{\text{rel}}[\%]$ [deg/100 meters] and rotational r_{rel}[deg/100 m] errors, while the absolute translation root-mean-square error t_{abs} was computed as in [33]. Relative average errors were obtained from the KITTI metric, while the average absolute error is simply the arithmetic mean.

Seq.	length	v_max	v_mean	start frame	end frame	$t_{\text{rel}}[\%]$	r_{rel}[deg/100 m]	t_{abs}
GROBNIK lap 1	3.63 km	203 km/h	121 km/h	1870	5105	0.44	0.15	2.58
GROBNIK lap 2	4.12 km	214 km/h	137 km/h	5105	8351	0.39	0.14	2.12
GROBNIK lap 3	3.98 km	204 km/h	108 km/h	8351	12337	0.39	0.16	2.64
NAVAK lap 1	2.77 km	189 km/h	70 km/h	94740	99039	0.47	0.25	1.35
NAVAK lap 2	2.77 km	201 km/h	95 km/h	99039	102185	0.40	0.23	1.74
NAVAK lap 3	2.78 km	206 km/h	103 km/h	102185	105099	0.39	0.21	1.24
avg						**0.43**	**0.20**	**1.95**

evaluation metrics.

ACKNOWLEDGMENT

The authors would like to thank the RT Autonomous Driving Department members who collected the data and collaborated on the project. The authors would also like to thank the Automotodrom Grobnik, Croatia and the NAVAK Centre, Serbia for granting the permission to make the dataset sequences publicly available.

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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] I. Cvišić and I. Petrović, "Stereo odometry based on careful feature selection and tracking," in *European Conference on Mobile Robots (ECMR)*, 2015, p. 6.

[10] 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.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] A. Kulkarni, J. Chrosniak, E. Ducote, F. Sauerbeck, A. Saba, U. Chirimar, J. Link, M. Behl, and M. Cellina, "Racecar - the dataset for high-speed autonomous racing," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 11 458–11 463.

[18] D. Schubert, N. Demmel, L. v. Stumberg, V. Usenko, and D. Cremers, "Rolling-shutter modelling for direct visual-inertial odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 2462–2469.

[19] J. Mo, J. Islam, and J. Sattar, "Imu-assisted learning of single-view rolling shutter correction," in *Proceedings of the 5th Conference on Robot Learning*, 2022, pp. 861–870. [Online]. Available: https://github.com/IRVLab/unrolling.

[20] L. Cao, J. Ling, and X. Xiao, "The whu rolling shutter visual-inertial dataset," *IEEE Access*, vol. 8, pp. 50 771–50 779, 2020.

[21] C. Kerl, J. Stückler, and D. Cremers, "Dense continuous-time tracking and mapping with rolling shutter rgb-d cameras," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[22] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *IEEE International Conference on obotics and Automation (ICRA)*. IEEE, 2014.

[23] 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.

[24] N. Yang, R. Wang, X. Gao, and D. Cremers, "Challenges in monocular visual odometry: Photometric calibration, motion bias, and rolling shutter effect," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2878–2885, 10 2018.

[25] Y. Zhang, M. Fan, S. Xu, X. Liu, and H. Xiong, "Semantic slam with rolling-shutter cameras and low-precision ins in outdoor environments," in *IEEE Intelligent Vehicles Symposium*, 2025.

[26] K. Zhang and M. Zhang, "Point feature correction based rolling shutter modeling for ekf-based visual-inertial odometry," *Measurement Science and Technology*, vol. 35, 1 2024.

[27] D. Schubert, V. Usenko, N. Demmel, J. Stueckler, and D. Cremers, "Direct sparse odometry with rolling shutter," in *European Conference on Computer Vision (ECCV)*, Sept. 2018, oral presentation.

[28] Y. Dai, H. Li, and L. Kneip, "Rolling shutter camera relative pose: Generalized epipolar geometry," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[29] I. Cvišić, I. Marković, and I. Petrović, "Enhanced calibration of camera setups for high-performance visual odometry," *Robotics and autonomous systems, 155 (2022), 104189, 12.*, 2022.

[30] "Navak." [Online]. Available: https://navak.rs/en/

[31] "Automotodrom grobnik." [Online]. Available: https://grobnik.hr/track/

[32] I. Cvišić, I. Marković, and I. Petrović, "SOFT2: Visual Stereo Odometry for Ground Vehicles based on a Point-to-Epipolar-Line Metric," *IEEE Transactions on Robotics, in review*, 2021.

[33] 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.