Supplementary Material: Asynchronous Multi-View SLAM

Anqi Joyce Yang*[1,2], Can Cui*[1,3], Ioan Andrei Bârsan*[1,2], Raquel Urtasun[1,2], Shenlong Wang[1,2]

I. Overview

The supplementary material covers additional information on:

1) Our method, specifically the details of the linear motion model used during tracking, more mathematical details of cubic B-splines, as well as further details on the loop closing module.
2) Our dataset and its geographic splits, providing a more in-depth comparison to other related SLAM benchmarks.
3) Our experiments, showcasing additional ablation studies, quantitative tables, qualitative results and discussions.

Acknowledgments The authors would like to thank Julieta Martinez, Davi Frossard, and Wei-Chiu Ma for their valuable input on the writing of the paper, Jack Fan for his contributions to the metadata code used to select the segments in the benchmark, and Rui Hu for his help on improving the learned feature inference speed. We would also like to thank Prof. Tim Barfoot for the detailed and thoughtful feedback on the project and paper, as well as our anonymous ICRA 2021 reviewers for their thorough comments and suggestions.

II. Method

A. Asynchronous Multi-Frames

We provide an illustration for the concept of an asynchronous multi-frame compared to a synchronous multi-frame in Fig. 1.

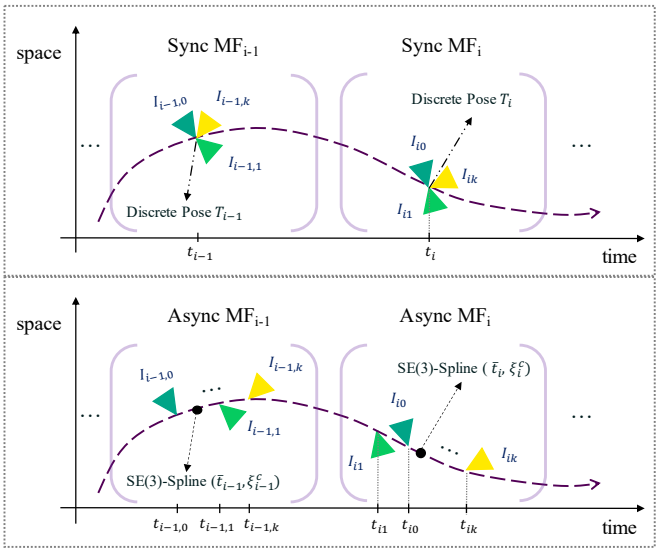

Fig. 1: Illustration of the synchronous multi-frame vs. the asynchronous multi-frame. For simplicity, in this illustration each async multi-frame is assumed to be a key multi-frame. Each key multi-frame is associated with cubic B-spline motion parameters to define the overall trajectory.

B. Linear Motion Model

During tracking, we estimate poses in the current asynchronous multi-frame with a linear motion model, denoted by the superscript ℓ. In general, given timestamps $t_1 \leq t_2$, and respective associated poses $\mathbf{T}_1$, $\mathbf{T}_2$, poses at any timestamp t could be linearly interpolated or extrapolated as

$$\begin{aligned}\mathbf{T}^{\ell}(t) &= \mathbf{T}_2(\mathbf{T}_2^{-1}\mathbf{T}_1)^{\alpha} \\ &= \mathbf{T}_2\text{Exp}(\alpha\text{Log}(\mathbf{T}_2^{-1}\mathbf{T}_1)), \quad \text{where } \alpha = \frac{t_2 - t}{t_2 - t_1}.\end{aligned} \tag{1}$$

In the context of multi-frames, for each multi-frame MF$_i$ with the representative timestamp $\bar{t}_i$, we define the linear pose parameter $\boldsymbol{\xi}_i^{\ell} \in \mathbb{R}^6$ to represent the minimal 6-DoF robot pose in the world frame at $\bar{t}_i$. It follows that poses at any timestamp t within MF$_i$ could be evaluated with

$$\begin{aligned}\mathbf{T}^{\ell}_{wb}(t) &= \mathbf{T}^{\ell}_{wb}(\bar{t}_i)\text{Exp}\left(\frac{\bar{t}_i - t}{\bar{t}_i - \bar{t}_{\text{ref}}}\text{Log}\left(\mathbf{T}^{\ell}_{wb}(\bar{t}_i)^{-1}\mathbf{T}^{c}_{wb}(\bar{t}_{\text{ref}})\right)\right) \\ &= \text{Exp}(\boldsymbol{\xi}_i^{\ell})\text{Exp}\left(\frac{\bar{t}_i - t}{\bar{t}_i - \bar{t}_{\text{ref}}}\text{Log}\left(\text{Exp}(-\boldsymbol{\xi}_i^{\ell})\mathbf{T}^{c}_{wb}(\bar{t}_{\text{ref}})\right)\right),\end{aligned} \tag{2}$$

where $\bar{t}_{\text{ref}}$ and $\mathbf{T}^{c}_{wb}(\bar{t}_{\text{ref}})$ are the respective representative timestamp and evaluated cubic B-spline pose at $\bar{t}_{\text{ref}}$ of the reference multi-frame MF$_{\text{ref}}$. In practice, new MFs are tracked against a reference *key* multi-frame, so *ref* refers to the MF id of the most recent KMF.

C. Cubic B-Spline Model

We use a cumulative cubic B-spline motion model over key multi-frames to represent the overall trajectory. We use the linear motion model parameters estimated during tracking to initialize cubic B-spline control points, and refine the spline trajectory during mapping and loop closing. In general, given $n+1$ control points $\boldsymbol{\xi}_0^c, \ldots, \boldsymbol{\xi}_n^c \in \mathbb{R}^6$, and a knot vector $\mathbf{b} \in \mathbb{R}^{n+k+1}$, the cumulative B-spline of order k is defined

*Denotes equal contribution. Work done during Can's internship at Uber.
[1]Uber Advanced Technologies Group
[2]University of Toronto,
{ajyang, iab, urtasun, slwang}@cs.toronto.edu
[3]University of Waterloo, c23cui@uwaterloo.ca

as:

$$\mathbf{T}_{wb}^{c}(t) = \text{Exp}\left(\widetilde{B}_{0,k}(t)\boldsymbol{\xi}_{0}^{c}\right)\prod_{i=1}^{n}\text{Exp}\left(\widetilde{B}_{i,k}(t)\boldsymbol{\Omega}_{i}\right), \quad (3)$$

where $\boldsymbol{\Omega}_i = \text{Log}\left(\text{Exp}(\boldsymbol{\xi}_{i-1}^{c})^{-1}\text{Exp}(\boldsymbol{\xi}_{i}^{c})\right) \in \mathbb{R}^6$ is the relative pose in Lie Algebra twist coordinate form between control poses $\boldsymbol{\xi}_{i-1}^{c}$ and $\boldsymbol{\xi}_{i}^{c}$. The superscript c is used to denote the cubic B-spline motion model. The cumulative basis function $\widetilde{B}_{i,k}(t) = \sum_{j=i}^{n} B_{j,k}(t) \in \mathbb{R}$ is the sum of basis function $B_{j,k}(t)$. Based on the knot vector $\mathbf{b} = \begin{bmatrix} b_0 & \dots & b_{n+k} \end{bmatrix}$, the basis function $B_{j,k}(t)$ is computed using the de Boor-Cox recursive formula [1], [2], with the base case

$$B_{p,1}(t) = \begin{cases} 1 & \text{if } t \in [b_p, b_{p+1}] \\ 0 & \text{otherwise} \end{cases}.$$

For $q \geq 2$,

$$B_{p,q}(t) = \frac{t - b_p}{b_{p+q-1} - b_p}B_{p,q-1}(t) + \frac{b_{p+q} - t}{b_{p+q} - b_{p+1}}B_{p+1,q-1}(t).$$

More intuitively, each $\mathbf{T}_{wb}^{c}(t)$ can be interpreted as an n-way interpolation between the control points $\boldsymbol{\xi}_{i}^{c}$ with respective interpolation weight $B_{i,k}(t)$. However, instead of directly interpolating $\mathbf{T}_{wb}^{c}(t) = \prod_{i=0}^{n}\text{Exp}(B_{i,k}(t)\boldsymbol{\xi}_{i}^{c})$, we use the cumulative formulation in Eq. 3 for accurate on-manifold interpolation in $\mathbb{SE}(3)$ [3], [4].

Since we use cubic B-splines, $n = 3$ and $k = 4$. In the context of multi-frames, we associate each *key* multi-frame KMF_i with a control point $\boldsymbol{\xi}_{i}^{c}$. In addition, since the key multi-frames are not necessarily distributed evenly in time, we cannot utilize a uniform knot vector (as typically employed for modeling rolling-shutter cameras [4] and LiDARs [5]). Instead, we associate each KMF_i with a *representative timestamp* $\bar{t}_i$ as the median of all image capture times t_{ik} (with the exception of initialization, where $\bar{t}_0$ is defined at the firing time of the overlapping camera pair). We define a non-uniform knot vector according to the representative timestamps. Specifically, we define $\mathbf{b} = \begin{bmatrix} \bar{t}_{i-3} & \bar{t}_{i-2} & \dots & \bar{t}_{i+4} \end{bmatrix} \in \mathbb{R}^8$.

D. Loop Detection

When a new KMF is selected, we run loop detection to check if a previously-seen area is being revisited. For computational efficiency, we only perform loop detection if the most recent successful loop correction took place at least a minimal number of KMFs ago. In our implementation we set this threshold to 30.

For a newly-inserted query KMF_q, the loop candidate KMF_l must pass an odometry check, a multi-camera DBoW3-based [6] similarity check, and a multi-camera geometric verification. We detail this process in the following subsections.

1) Odometry Check: To avoid false loop detection when the robot is staying in the same area, the odometry check ensures that the robot must have traveled a minimal distance since the loop candidate frame. In addition, a minimal time and a minimal number of key frames must have passed since the candidate frame as well. The traveling distance is computed based on the estimated trajectory. The time and key frame count conditions serve as complements in the case when the estimated traveling distance is unreliable. In our experiment, we set the traveling distance threshold to 30m, time to 5s, and the number of KMFs to 30. Note that the KMF threshold in odometry check is different from the KMF threshold described at the beginning of the loop detection section. The former specifies that a candidate KMF must be older than 30 KMFs ago, while the latter dictates that we will only perform loop detection for the current query KMF if the latest loop closing happened at least 30 KMFs ago.

2) Similarity check: For candidates passing the odometry check, we perform a multi-view version of the single-view DBoW3-based similarity check described in ORB-SLAM [7]. The key idea is that images in the loop candidate KMF and the query KMF should have similar appearance. We perform similarity detection with the bag-of-words techniques DBoW3 for place recognition [6]. For each key multi-frame, we concatenate features extracted from all images in the multi-frame to build the DBoW3 vocabulary and compute the DBoW3 vector. Note that the simple concatenation does not take into account of the fact that cameras can be asynchronous, but we argue that the same area should have similar appearance within the short camera firing time interval, and false positives will be filtered by the stricter geometric verification that factors in asynchronous sensors in the next step.

During the similarity check, we first compute the DBoW3 similarity score between the query KMF and the neighboring KMFs that are included in the associated local bundle adjustment window, and we denote the minimum similarity score as m. Next, we compute the similarity score between the query KMF and all available candidate KMFs and denote the top score as t. Then all the candidates that pass the similarity check must have a similarity score that is greater than $\max(0.01, m, 0.9 * t)$.

3) Geometric check: For each remaining candidate KMF, we perform a geometric check by directly solving for a relative pose between cameras in the candidate KMF and cameras in the query KMF. To identify the camera pairs to be matched, let us consider a setting where the camera rig contains a set of M cameras covering a combined 360° FoV, with the cameras denoted as $\mathcal{C}_1, \dots, \mathcal{C}_M$ in the clockwise order. We also assume that the robot is on the ground plane in this setting, *i.e.*, the roll and pitch angles of the robot poses remain the same when the robot revisits the same area. Since a loop can be encountered at an arbitrary yaw angle, there are a total of M possible scenarios of how the multiple cameras between the candidate and the query frame can be matched, where in scenario i, each camera $\mathcal{C}_j$ in the candidate frame is matched to camera $\mathcal{C}_{(j+i)\%M}$ in the query frame.

For each possible matching scenario involving M pairs of cameras, we solve for a discrete relative pose between each camera pair. Specifically, for each pair of cameras, we first perform keypoint-based sparse image matching between the associated image pair by fitting an essential matrix in a RANSAC [8] loop. If the number of inlier matches passes a certain threshold, we associate the inlier matches with the existing 3D map points. Note that different from tracking,

here we draw associations in both directions: 2D keypoints in the loop candidate image are associated to 3D map points observed in the query image, and vice versa.

If the number of such keypoint-to-map-point correspondences passes a threshold, we estimate a single relative pose in $\mathbb{SE}(3)$ between the two cameras. Following [7], we perform pose estimation with Horn's method in a RANSAC loop, where within each RANSAC iteration we sample a minimal number of matches, and solve for the discrete pose by minimizing a reprojection error. The hypothesis with the most number of inliers is the final estimate.

The geometric check passes if at least a certain number of camera pairs have a minimum number of inliers for the relative pose estimation. In our full system, we perform geometric check with the $M = 5$ wide cameras covering the surroundings of the vehicle. We consider a geometric check to be successful if there exists a matching scenario where we can successfully estimate the discrete relative pose for at least 2 pairs of cameras, where for each camera pair there are at least 20 inlier correspondence pairs during sparse image matching, 20 pairs of 2D-3D associations, and 20 pairs of inlier correspondences after the relative pose estimation. If there are multiple matching scenarios that pass the check, we select the configuration with the most successfully matched camera pairs and the most inlier correspondences.

The multi-camera geometric verification outputs $\{(\mathcal{C}_{k_l}, \mathcal{C}_{k_q}, \mathbf{T}_{b_{k_q},b_{k_l}})\}$, which is a set of triplets denoting the camera indices of the matched camera pairs between the loop and the query frames, along with $\mathbf{T}_{b_{k_q},b_{k_l}}$, which is an estimated rigid-body transformation from the body frame $\mathcal{F}_b$ at the camera capture time t_{lk_l} to $\mathcal{F}_b$ at time t_{qk_q}.

E. Loop Correction

If a loop candidate KMF_l passes all loop detection checks, we perform loop correction with the geometric verification output. We first build an asynchronous multi-view version of the pose graph in ORB-SLAM [7]. Each node α of the pose graph is encoded by a timestamp t_α representing the underlying robot pose $\mathbf{T}_{wb}(t_\alpha)$. Each edge (α, β) encodes the relative pose constraint $\mathbf{T}_{\beta\alpha}$ representing the rigid transformation from $\mathbf{T}_{wb}(t_\alpha)$ to $\mathbf{T}_{wb}(t_\beta)$.

Specifically, in our pose graph, the nodes are associated with times at $\{\bar{t}_i\}_{\forall \text{KMF}_i} \cup \{t_{lk_l}, t_{qk_q}\}_{\forall (k_l,k_q)}$, *i.e.*, the representative timestamps of all existing KMFs, as well as the camera times from matched camera pairs in the geometric verification output. The edges are comprised of: (1) neighboring edges connecting adjacent KMF nodes at times $\bar{t}_{i-1}$ and $\bar{t}_i$, (2) past loop closure edges connecting nodes associated with past query and loop closure KMFs, and (3) the new loop closure edges between nodes at time t_{lk_l} and t_{qk_q}. For edges in case (1) and (2), we compute the relative pose $T_{\beta\alpha}$ by evaluating $(\mathbf{T}^c_{wb}(\bar{t}_\beta))^{-1}\mathbf{T}^c_{wb}(\bar{t}_\alpha)$ based on the current trajectory. For (3), we use the discrete poses $\mathbf{T}_{b_{k_q},b_{k_l}}$ estimated in the geometric verification step in loop detection. Please refer to Fig. 2 for an illustration of the pose graph.

We denote the local KMF windows spanning the query and loop frames as the *welding windows*. In our implementation they are the same size as the local bundle adjustment window. To correct the global drift, we perform a pose graph optimization (PGO) over the continuous-time cubic B-spline trajectory. To better anchor the trajectory, the control poses in the welding window associated to KMF_l are fixed during the pose graph optimization, where the following objective is minimized:

$$E_{\text{PGO}}(\{\boldsymbol{\xi}^c_i\}) = E_{\text{rel}}(\{\boldsymbol{\xi}^c_i\}) + E_{\text{reg}}(\{\boldsymbol{\xi}^c_i\}), \tag{4}$$

where

$$\begin{aligned} E_{\text{rel}}(\{\boldsymbol{\xi}^c_i\}) &= \sum_{(\alpha,\beta)} \rho\left(\left\|\mathbf{e}^T_{\alpha,\beta}(\{\boldsymbol{\xi}^c_i\})\right\|^2_{\boldsymbol{\Sigma}^{-1}_{\alpha,\beta}}\right), \\ \text{with } \mathbf{e}_{\alpha,\beta}(\{\boldsymbol{\xi}^c_i\}) &= \text{Log}(\mathbf{T}_{\beta\alpha}(\mathbf{T}^c_{wb}(t_\alpha))^{-1}\mathbf{T}^c_{wb}(t_\beta)) \in \mathbb{R}^6 \end{aligned} \tag{5}$$

sums over the relative pose errors of each edge weighted by an uncertainty term $\boldsymbol{\Sigma}^{-1}_{\alpha,\beta}$, and

$$\begin{aligned} E_{\text{reg}}(\{\boldsymbol{\xi}^c_i\}) &= \sum_i \rho\left(\left\|\mathbf{r}^T_i(\{\boldsymbol{\xi}^c_i\})\right\|^2_{\boldsymbol{\Lambda}^{-1}_i}\right), \\ \text{with } \mathbf{r}^T_i(\{\boldsymbol{\xi}^c_i\}) &= \text{Log}\left(\mathbf{T}^{-1}_i\mathbf{T}^c_{wb}(\bar{t}_i)\right) \in \mathbb{R}^6 \end{aligned} \tag{6}$$

is a unary regularization term weighted by uncertainty $\boldsymbol{\Lambda}^{-1}_i$ to anchor each KMF's representative pose at $\mathbf{T}_i = \mathbf{T}^c_{wb}(\bar{t}_i)$ evaluated before the optimization. Empirically, we set the diagonal entries of both $\boldsymbol{\Sigma}^{-1}_{\alpha,\beta}$ and $\boldsymbol{\Lambda}^{-1}_i$ to 1.0. The regularization term helps to better constrain the optimization especially when there is a large loop (*i.e.*, q is much bigger than l) and a large amount of drift to correct. ρ is the robust norm and we again use the Huber loss in practice. The energy term is minimized with the LM algorithm.

After PGO, we next update the map points with the adjusted trajectory. If a map point is observed in multiple images, we update the map point position with the median of all pose corrections related to the map point. Note that ideally, we would want to solve a global bundle adjustment problem that jointly refines the entire trajectory and all map points at the same time. However, with a long trajectory and many observations from multi-view cameras, global bundle adjustment becomes computational expensive or even infeasible. The two-stage process described above, where we first optimize the poses and then update the map points, is a light-weight approximation that is sufficient under most circumstances.

Furthermore, note some new map points in the query KMF window have been created during recent local mappings, but they may correspond to points already triangulated in the previous pass through the revisited area. As a result, we next deduplicate the re-triangulated map points. We first match image pairs in the candidate KMF welding window and the query KMF welding window to identify and fuse these map points. We then perform a local bundle adjustment over the motion and map points corresponding to the two welding windows. The purpose of the local bundle adjustment is to refine both map points and control poses in the query KMF window. To anchor candidate poses, we freeze the control poses corresponding to the candidate welding window in the optimization.

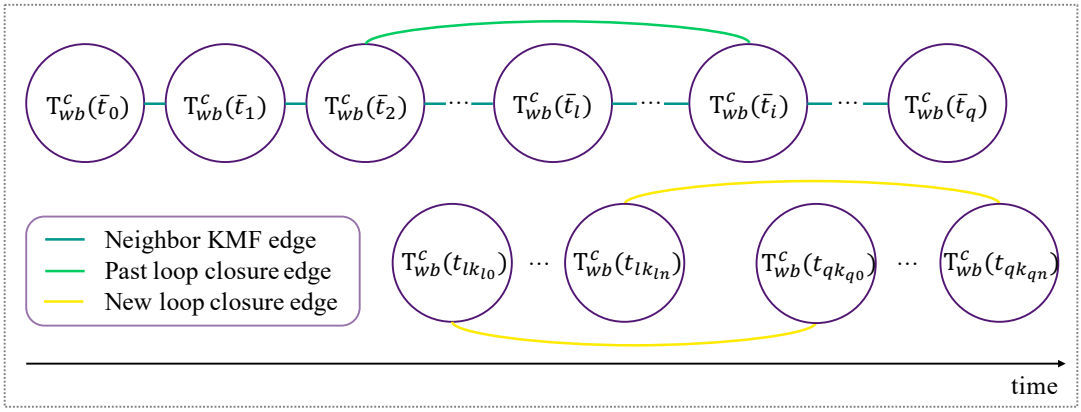

Fig. 2: Illustration of the loop correction pose graph. The nodes correspond to robot poses at the representative timestamps of all KMFs + capture times of the matched cameras in the new loop closure KMFs. The edges consist of neighboring KMF edges, past loop closing edges, and the new multi-view loop closing edges. If n camera pairs are successfully matched during the loop detection stage, then there should be n new loop closing edges.

III. Dataset

A. Existing SLAM Benchmarks

As described in the main paper, existing SLAM datasets fall short in terms of geographic diversity, modern sensor layouts, or scale. In this section, we describe the most relevant modern SLAM datasets together with their primary limitations.

The KITTI Odometry Benchmark [9] covers 40km of driving through Karlsruhe, Germany using a vehicle equipped with a stereo camera pair, a 64-beam LiDAR, IMU, and RTK-based ground truth. However, most sequences in the dataset have relatively small numbers of dynamic objects, and all the data is captured in sunny or overcast weather, which is not representative of the variety of conditions which can be encountered by commercial AVs. The NCLT dataset [10] covers a larger distance in the University of Michigan campus using a custom-built robotic Segway equipped with three LiDARs an IMU and an omnidirectional camera. While the scale of the dataset is large, its geographic diversity is lacking, being constrained to a university campus, while at the same time not capturing the same challenging motion patterns which would be encountered by an SDV.

The Oxford RobotCar [11] dataset covers over 1000km of driving in challenging conditions containing a large number of dynamic objects as well as strong weather and lighting variation. However, since it is focused on a single primary trajectory it lacks geographic diversity. Moreover, it does not provide 360° camera data in HD, which is critical for SDVs. Similarly, the Ford Multi-AV Dataset [12] contains a large volume of data but is focused on a relatively limited 60km route which is traversed repeatedly by multiple SDVs. The A2D2 Dataset [13] contains a high-resolution multi-camera multi-LiDAR setup optimized for 3D reconstruction. However, the approximate total scale of A2D2 is still on the order of a few hours of driving, which is insufficient for evaluating robust SLAM system in a wide variety of challenging conditions.

Finally, the recent 4Seasons dataset [14] covers diverse

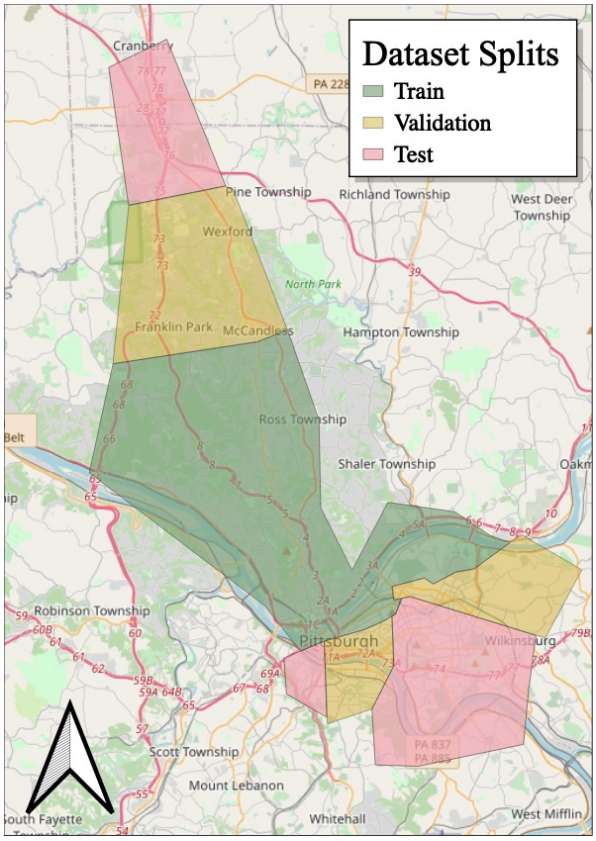

Fig. 3: The geographic splits of the proposed datasets. They are designed such that train, val, and test each covers a balanced blend of environment types (highway, urban, industrial, residential, etc.). The splits are also selected to have similar distributions of weather, loop closures, etc.

TABLE I: The proposed dataset, AMV-Bench, in numbers.

Split	Sequences	Distance (km)	Time (h)
Train	65	281	12
Validation	25	103	4
Test	26	98	5
Total	116	482	21

areas in a wide range of environments (highway, industrial, residential, etc.) over a long time period, but lacks the HD multi-view sensor array common in commercial SDVs.

Note that in the dataset overview table in the main paper we label A2D2 and Ford Multi-AV as non-asynchronous dataset on the basis that their cameras are not described as following the LiDAR or any custom firing pattern causing more than 2–3ms of delay. While Ford Multi-AV does have cameras firing at different frame rates, with the higher-resolution front stereo operating at 5Hz and the other cameras at 15Hz, we do not consider this as a true asynchronous setting.

Additionally, even though datasets such as Waymo Open [15] and nuScenes [16] include asynchronous cameras, they are focused on perception tasks and contain short sequences (e.g., less than a minute each). Therefore, we do not consider them in our evaluation as they are too short to robustly evaluate SLAM algorithms.

B. Dataset Details

The dataset has been selected using a semi-automatic curation process to ensure all splits cover similar categories of environments, while at the same time having substantial appearance differences between the splits.

Table I shows the high-level statistics of the train, validation, and test partitions of the dataset. Figure 3 shows the geographic regions of the splits.

The cameras are all RGB and have a resolution of 1920×1200 pixels, a global shutter; the (asynchronous) shutter timestamps are recorded with the rest of the data. The wide-angle and narrow-angle cameras have rectified FoVs of 76.6°× 52.6°and 37.8°× 24.2°, respectively.

Furthermore, the 6-DoF ground-truth was generated using an offline HD-map based localization system, which enables SLAM systems to be evaluated at the centimeter level.

IV. Additional Experiments

A. Full Implementation Details

All images are downsampled to 960×600 for both our method as well as all baselines. In our system, we extract 1000 ORB [17] keypoints from each image, using grid-based sampling [7] to encourage homogeneous distribution. Image matching is performed with nearest neighbor + Lowe's ratio test [18] with a ratio threshold of 0.7. The initial 2D matches between each image pair are additionally filtered by fitting an essential matrix with RANSAC. The inlier correspondences are used (1) as input to the multi-view PnP during tracking, (2) for new map point triangulation during mapping, (3) for geometric verification during loop closing.

In our system, we use the synchronous stereo camera pair during initialization. During tracking, we match image pairs captured by the same camera. During new map point creation, we match images captured by the same camera between the new KMF and four previous KMFs, and triangulate new map points based on the 2D matches. We additionally triangulate new map points from the stereo cameras within the new key multi-frame. Note that we do not match between different wide cameras in the same multi-frame due to little overlap between them and ORB's reduced performance in wide baseline image matching settings. Please see Sec. IV-D.1 for details.

During tracking, we randomly sample 7 pairs of correspondences within each PnP RANSAC loop to solve for a hypothesis. The pose estimate hypothesis with the most number of inliers becomes the final estimate.

During key frame selection, a new KMF is inserted either (1) when the estimated local translation against reference KMF is over 1m, or local rotation is over 1°, or (2) when under 35% of the map points are re-observed in at least two camera frames, or (3) when a KMF hasn't been inserted for 20 consecutive MFs. Note that (3) is necessary to model the spline trajectory when the robot stays stationary.

We perform bundle adjustment over a recent window of size $N = 11$. We cull map points with reprojection error over 1.5 pixels.

Following [7], the uncertainty weighting Σ in both tracking and bundle adjustment is based on the scale level where ORB features are extracted. Keypoints extracted from larger/coarser scale levels are less precise and therefore correspond to higher uncertainty and lower weighting during pose estimation.

During the pose graph optimization in loop closure, the uncertainty weighting for the relative pose error terms and the regularization terms are all set to 1.0.

In our system and all our ablation implementations, we declare a tracking failure if the estimated pose parameters yield under 12 inlier PnP correspondences in total. We declare bundle adjustment failure and not apply the bundle adjustment update if after bundle adjustment any of the pose parameters is changed by more than 6 meters or 20 degrees. We selected these values empirically based on training set performance. We stop the system in the middle of processing a sequence if there are at least 5 successive tracking failures, or if there are at least 5 successive bundle adjustment failures.

We use the Ceres Solver [19] for modeling and solving the non-linear optimization problems arising in tracking, bundle adjustment, and loop closure.

B. Third-Party Baseline Experiment Details

For all ORB-SLAM [7], [20] experiments, we lowered the default tracking failure threshold from 30 matching inliers to 10 matching inliers. This is to increase the tolerance for tracking failures, as the system with 10–30 matching inliers was able to complete larger portions of the training sequences without much compromise of local tracking errors. Apart from the inlier threshold,, we use all other default hyperparameters

provided for the KITTI experiments by the authors[1], which extract 2000 ORB keypoints per image, while we only extract 1000 ORB keypoints for our system and all our baseline and ablation study implementations.

We use the default KITTI hyperparameters for LDSO [21] provided by the authors[2].

For the keypoint extractor ablation study, we extract 1000 keypoints and associated features from each image for all methods. We match features with Lowe's ratio test. The ratio is tuned on the training set. We use 0.7 for ORB [17] and RootSIFT [18], [22], and 0.8 for SuperPoint [23] and R2D2 [24]. For SuperPoint, we run the provided pre-trained model [3]. For R2D2, we run the provided pre-trained `r2d2_WASF_N16` model [4].

C. Metrics

In the additional experiments, aside from reporting the median and AUC for the aggregated ATE and RPE results, we also report the ATE and RPE errors at the 90th percentile, *i.e.* x with $f(x) = 0.9$ where f is the cumulative error curve. Compared to median (the 50th percentile), the 90th percentile metric better characterizes outlier behavior, and the AUC metric gives a better characterization of the overall performance, while being able to model system failures.

D. Quantitative Results

In this subsection we show additional details on the main paper results, as well as additional ablation studies.

1) Detailed main paper results: Table II and Table III compare (1) third-party baselines, (2) our implementation of the synchronous baselines, and (3) our main system, in the SLAM mode and visual odometry (VO) mode respectively. In the VO mode we disable loop closing, and relocalization in ORB-SLAM. Figs. 4 and 5 plot the respective cumulative error curves.

Note that some metrics of our full system with loop closure in Table II are slightly worse than those without loop closure in Table III. The difference can also be observed in the DSO experiments. The difference in our system is due to the stochasticity of the trials, instead of loop closing failures. To support our claim, in the main paper we plot the drift relative to the ground truth with and without loop closure to show that loop closing successfully reduced global drifts at all key multi-frames where loop closing was performed. Furthermore, Table IV compares all metrics on the 8/25 validation sequences where loop closure was performed, and shows that the 8 loop closing sequences did not contribute to the metrics differences over all 25 validation sequences.

Table V showcases the motion model ablation study results. For simplicity, loop closure is disabled. The asynchronous linear motion model setting represents the trajectory with a linear motion model parameterized by a 6-DoF pose $\boldsymbol{\xi}_i^\ell$ at each representative timestamp $\bar{t}_i$. The motion model is explained in detail in Sec. II. Similar to the main system, we estimate linear motion model parameter during tracking, and we jointly refine the linear motion model parameters along with the map points during bundle adjustment, with the reprojection energy similar to that of tracking. Our experiments show that modeling the vehicle motion using cubic B-splines leads to improved performance due to the splines' ability to impose a realistic motion prior on the estimated trajectories.

In the additional motion model ablation results, we also compare with linear and cubic B-spline models that perform interpolation in $\mathbb{SO}(3)$ and $\mathbb{R}^3$ separately, instead of jointly in $\mathbb{SE}(3)$. Previous works [26]–[29] have shown that the split interpolation formulation is generally better in terms of both computation time and trajectory representation.

The results show our main system with cubic B-spline model and full interpolation in $\mathbb{SE}(3)$ has the best performance overall. The split-interpolation motion models have slightly worse performance in our experiments, most likely because full interpolation in $\mathbb{SE}(3)$ is more apt at modeling curvy trajectories (*e.g.*, during turns).

Table VI and Fig. 7 compare with additional camera configurations during tracking and mapping. In the additional camera configurations, during local mapping we additionally match and triangulate new map points between the wide front left and wide front middle cameras, and between the wide front middle and wide front right cameras within the same key multi-frame. Note that in the settings without stereo cameras, we still use stereo cameras (only) for initialization.

Configurations without stereo cameras have worse performance, and we argue this is due to ORB's poor performance in the much harder wide-baseline image matching problem posed by little overlap between the wide front cameras during new map point creation. The table shows that with keypoint extractors such as SuperPoint [23] in place of ORB, this performance gap is significantly narrowed.

Table VII and Fig. 8 show the full results of the keypoint extractor ablation study. Compared to ORB, SIFT and SuperPoint finish more sequences and have better ATE, with the caveat that feature extraction takes more time.

2) Per sequence results: Table VIII shows per-sequence errors comparing baselines and our method, all with loop closure. We report the mean over all three trials. If at least one trial did not complete the sequence successfully, we do not report results for that sequence.

3) KMF heuristics ablation: To study the effect of the combined KMF selection heuristics that factors in both map point reobservability and motion, we perform an ablation study where we run our stereo + synchronous discrete-time motion model implementation with a reobservability-only KMF heuristic. Table IX shows that our stereo implementation with a reobservability-only heuristic performs worse than the stereo system with the more robust combined heuristic. Table X compares the number of key frames inserted by ORB-SLAM2, our stereo sync with a reobservability-only heuristic, and our stereo sync with a reobservability+motion heuristic. The key frame numbers are taken from a randomly-selected trial. The table shows that our reobservability-only heuristic

[1] https://github.com/raulmur/ORB_SLAM2
[2] https://github.com/tum-vision/LDSO
[3] https://github.com/rpautrat/SuperPoint
[4] https://github.com/naver/r2d2

TABLE II: Baseline methods. M=monocular, S=stereo, A=all cameras.

Method	RPE-T (cm/m)			RPE-R (rad/m)			ATE (m)			SR (%)
	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	
LDSO-M [21]	42.72	-	28.08	8.02E-05	-	54.23	594.39	-	44.67	62.67
ORB-M [7]	34.00	-	25.66	5.49E-05	-	63.77	694.37	-	42.65	64.00
ORB-S [20]	1.85	-	65.70	3.29E-05	-	70.47	30.74	-	74.31	77.33
Sync-S	1.30	-	77.54	2.91E-05	-	78.37	24.53	-	77.44	84.00
Sync-A	2.15	-	68.46	3.47E-05	-	70.47	58.18	-	75.01	74.67
Ours-A	**0.35**	**1.99**	**88.63**	**1.13E-05**	**6.50E-05**	**88.17**	**6.13**	**322.95**	**88.82**	**92.00**

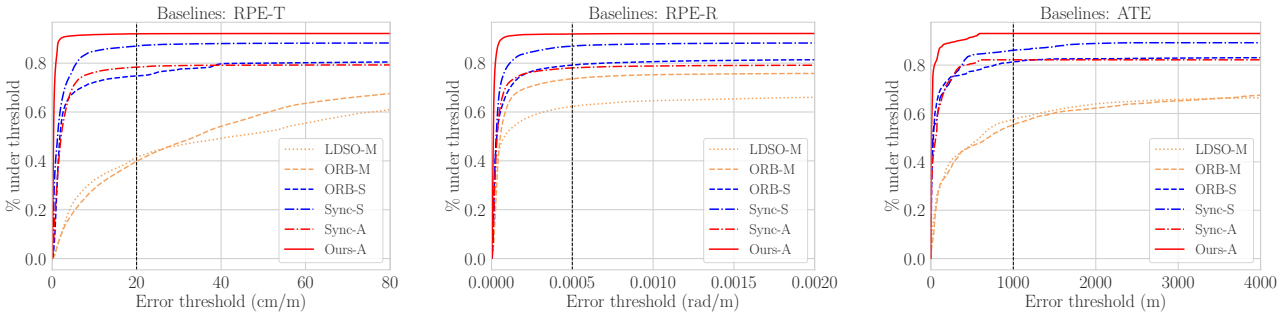

Fig. 4: Cumulative error curves comparing all baseline methods and our full system, with loop closure.

TABLE III: Baseline methods, all in the visual odometry (VO) mode with loop closing (and relocalization in ORB-SLAM) disabled.

Method	RPE-T (cm/m)			RPE-R (rad/m)			ATE (m)			SR (%)
	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	
DSO-M [25]	30.99	-	32.93	3.88E-05	-	58.98	801.99	-	41.87	64.00
ORB-M (VO) [7]	45.22	-	20.33	4.93E-05	-	64.91	849.64	-	40.62	58.67
ORB-S (VO) [20]	2.24	-	64.91	2.99E-05	-	72.30	45.28	-	74.61	66.67
Sync-S (VO)	1.27	-	77.77	2.80E-05	-	78.72	24.07	-	77.64	85.33
Sync-A (VO)	1.97	-	69.46	2.96E-05	-	73.39	55.24	-	75.11	70.67
Ours-A (VO)	**0.35**	**2.14**	**88.79**	**1.11E-05**	**6.30E-05**	**88.47**	**6.53**	**299.30**	**89.04**	**92.00**

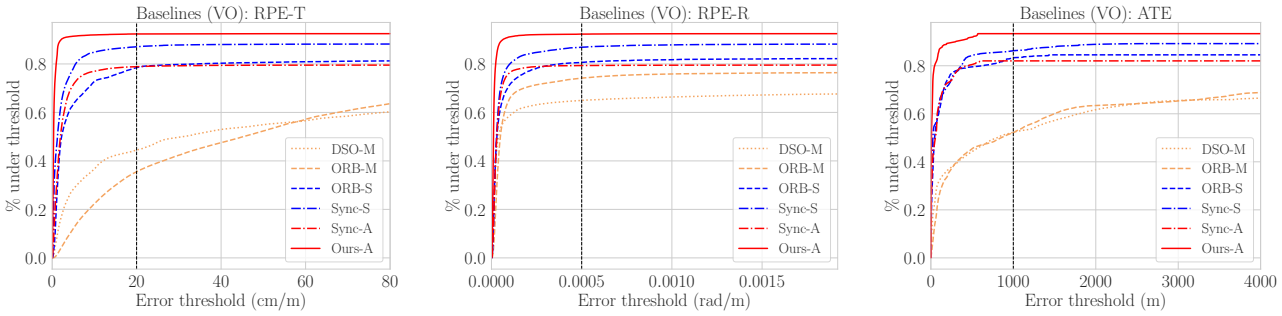

Fig. 5: Cumulative error curves comparing all baseline methods and our full system, without loop closure.

TABLE IV: Ablation study on loop closure on 8 validation sequences where loop closing was performed.

Method	RPE-T (cm/m)			RPE-R (rad/m)			ATE (m)			SR (%)
	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	
Ours-A (VO)	**0.27**	0.99	96.67	**9.51E-06**	**3.07E-05**	95.73	3.87	25.08	97.67	**100.00**
Ours-A	0.28	**0.92**	**96.78**	9.83E-06	3.19E-05	**95.88**	**2.97**	**20.58**	**97.73**	**100.00**

TABLE V: Ablation study on motion models. All experiments in visual odometry (VO) mode with loop closing disabled. Split indicates interpolation in $\mathbb{SO}(3)$ and $\mathbb{R}^3$ separately [26] instead of jointly in $\mathbb{SE}(3)$.

Method	RPE-T (cm/m)			RPE-R (rad/m)			ATE (m)			SR (%)
	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	
Sync Assumption	1.97	-	69.46	2.96E-05	-	73.39	55.24	-	75.11	70.67
Linear (Split)	0.36	2.51	88.08	**1.10E-05**	7.96E-05	87.79	6.25	580.23	87.43	**92.00**
Linear	0.41	2.71	87.76	1.11E-05	**5.99E-05**	88.39	6.09	429.86	88.31	89.33
Cubic B-Spline (Split)	0.38	3.34	87.86	1.12E-05	9.01E-05	87.69	**5.15**	588.96	87.34	**92.00**
Cubic B-Spline	**0.35**	**2.14**	**88.79**	1.11E-05	6.30E-05	**88.47**	6.53	**299.30**	**89.04**	**92.00**

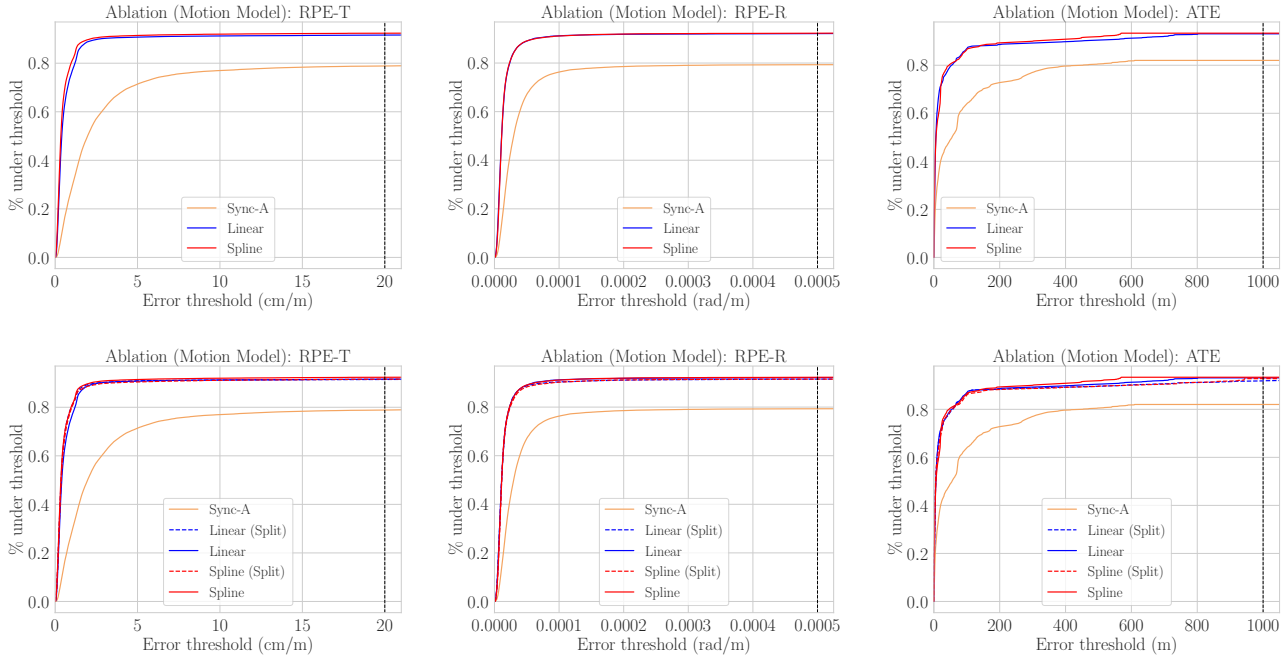

Fig. 6: Cumulative error curves of the motion model ablation study. (Top) The three comparisons in the main paper. (Bottom) All comparisons in the additional experiments.

TABLE VI: Ablation study on camera rigs in the VO mode, all initialized with the stereo cameras. s = stereo, wf = wide-front, wb = wide-back, ✓ is used for intra-frame new map point creation during mapping. The last row in the ORB table represents the main system.

	Camera Config			RPE-T (cm/m)			RPE-R (rad/m)			ATE (m)			SR (%)
	s	wf	wb	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	
ORB [17]	✓			0.70	-	79.86	1.93E-05	-	80.48	11.44	-	75.75	88.00
	✓	✓		0.41	8.52	84.88	1.21E-05	2.48E-04	85.86	9.00	802.88	84.92	90.67
		✓		6.07	-	49.00	4.58E-05	-	52.96	53.76	-	55.05	57.33
		✓	✓	1.16	-	63.94	1.65E-05	-	74.34	18.63	-	72.10	74.67
	✓	✓	✓	0.36	3.43	88.43	1.12E-05	**5.35E-05**	**88.60**	**5.95**	**298.08**	**89.05**	**92.00**
	✓	✓	✓	**0.35**	**2.14**	**88.79**	**1.11E-05**	6.30E-05	88.47	6.53	299.30	89.04	**92.00**
SuperPoint [23]	✓			0.64	19.44	82.97	1.59E-05	5.77E-04	83.09	16.40	463.41	85.09	82.67
	✓	✓		0.44	1.62	92.83	**1.01E-05**	3.48E-05	93.43	8.33	106.31	93.92	97.33
		✓		1.06	-	78.32	2.88E-05	-	76.73	21.08	-	73.48	88.00
		✓	✓	0.54	1.85	89.42	1.84E-05	6.51E-05	88.31	10.62	413.52	87.67	96.00
	✓	✓	✓	**0.38**	**1.22**	**95.66**	1.04E-05	2.63E-05	95.86	**5.38**	86.67	**96.38**	**100.00**
	✓	✓	✓	0.41	1.28	95.28	1.03E-05	**2.54E-05**	**95.90**	6.83	**78.06**	96.34	98.67

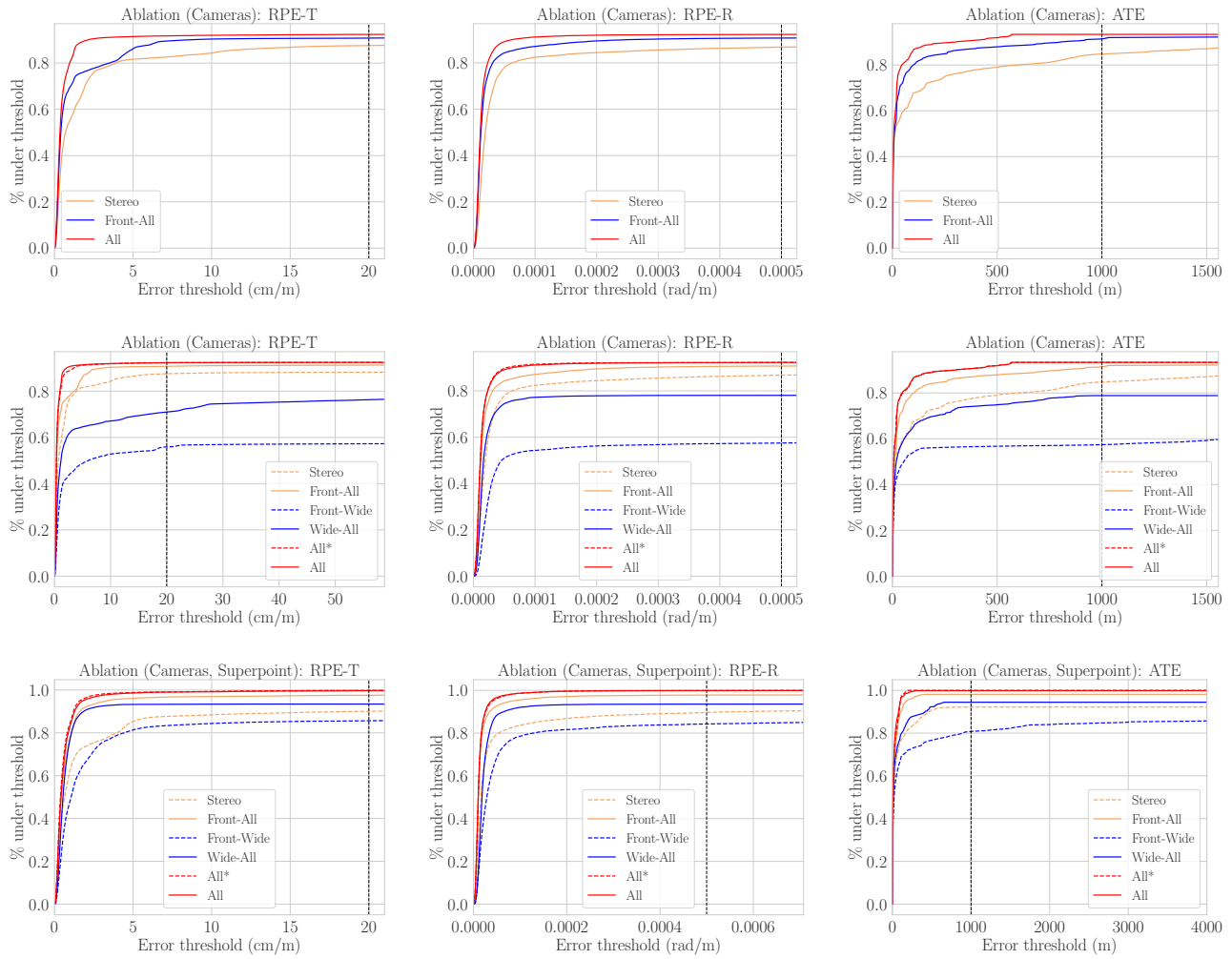

Fig. 7: Cumulative error curves of the camera ablation study. (Top) The three configurations in the main paper. (Middle) All camera configurations in the additional experiments. (Bottom) All camera configurations with SuperPoint in place of ORB as the keypoint extractor in the additional experiments. Camera configuration legend order corresponds to the order in the table.

TABLE VII: Ablation study for keypoint extractors in the VO mode. Time is the average feature extraction time per image using 24 CPU cores.

Method	Time (s)	RPE-T (cm/m)			RPE-R (rad/m)			ATE (m)			SR (%)
		@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	
RootSIFT [22]	0.10	0.41	2.26	94.06	1.11E-05	2.69E-05	95.84	6.66	123.72	91.50	**98.67**
SuperPoint [23]	0.35	0.41	**1.28**	**95.28**	**1.03E-05**	**2.54E-05**	**95.90**	6.83	**78.06**	**96.34**	**98.67**
R2D2 [24]	20.50	0.41	-	84.09	1.40E-05	-	83.72	7.42	-	86.62	88.00
ORB [17]	**0.01**	**0.35**	2.14	88.79	1.11E-05	6.30E-05	88.47	**6.53**	299.30	89.04	92.00

in general inserts fewer key frames than ORB-SLAM2, and that the combined heuristic selects almost twice as many key frames during highway sequences, when the vehicle is driving very fast in a highly repetitive scene.

E. Qualitative Results

1) Qualitative Trajectories:

a) Full Results for Ours-A vs. ORB-SLAM2: Fig. 9 and Fig. 11 showcase the trajectories in all 25 validation sequences, comparing ORB-SLAM2 using only the stereo cameras to our full system using all 7 asynchronous cameras. Fig. 10 depicts the trajectories ins all 65 training sequences.

In the following paragraphs, we qualitatively showcase the failure cases of our main paper ablation study baselines.

b) Motion Model Ablation: Fig. 12 plots failure cases of the linear motion model and the discrete-time motion model with a wrong synchronous assumption. The linear motion model trial failed early due to repeated mapping failures in a challenging case with dynamic objects, and the synchronous model had huge estimation errors during complex maneuvers

TABLE VIII: Per sequence errors of all baselines and our method. Errors averaged over all three trials at all evaluated timestamps. - denotes at least one trial did not successfully complete the sequence. RPE-T(cm/m), RPE-R(rad/m), ATE(m).

Sequence	Monocular						Stereo						All-Camera					
	LDSO-M			ORB-M			ORB-S			Sync-S			Sync-A			Ours-A		
	RPE-T	RPE-R	ATE	RPE-T	RPE-R	ATE	RPE-T	RPE-R	ATE	RPE-T	RPE-R	ATE	RPE-T	RPE-R	ATE	RPE-T	RPE-R	ATE
day_no_rain_0	-	-	-	-	-	-	2.95	8.44E-05	40.96	0.51	1.70E-05	**6.17**	-	-	-	**0.46**	**1.25E-05**	7.38
day_no_rain_1	46.45	1.61E-03	225.75	1.25	2.78E-05	5.43	1.29	3.07E-05	5.09	0.90	6.57E-05	2.74	1.66	6.55E-05	8.43	**0.22**	**1.00E-05**	**0.55**
day_no_rain_2	-	-	-	5.46	4.89E-05	52.75	1.72	7.37E-05	12.04	0.41	2.45E-05	5.54	-	-	-	**0.29**	**1.20E-05**	**3.53**
day_no_rain_3	21.63	2.63E-05	382.04	-	-	-	1.53	3.33E-05	13.03	0.33	2.00E-05	2.95	0.85	1.71E-05	5.18	**0.17**	**1.16E-05**	**2.71**
day_no_rain_4	-	-	-	31.93	4.33E-05	426.90	**0.64**	**2.17E-05**	8.85	0.96	3.61E-05	10.17	-	-	-	0.96	2.57E-05	**3.74**
day_no_rain_5	-	-	-	-	-	-	1.06	2.77E-05	6.64	1.25	5.11E-05	6.49	2.61	5.28E-05	7.83	**0.44**	**1.55E-05**	**1.18**
day_no_rain_6	4.11	1.13E-04	**17.04**	1.74	4.91E-05	47.99	-	-	-	0.79	4.01E-05	24.63	3.27	2.00E-04	73.84	**0.24**	**1.24E-05**	20.11
day_no_rain_7	0.72	3.35E-05	1.75	2.74	1.52E-04	50.96	0.64	**1.58E-05**	1.41	0.28	1.84E-05	**0.38**	0.50	1.61E-05	1.80	**0.23**	1.64E-05	0.49
day_no_rain_8	-	-	-	44.84	7.88E-05	411.75	-	-	-	1.82	4.80E-05	26.81	1.63	3.13E-05	23.36	**0.40**	**1.28E-05**	**5.30**
day_no_rain_9	17.47	4.31E-04	74.46	26.15	1.47E-04	571.95	-	-	-	0.37	1.82E-05	2.70	-	-	-	**0.33**	**1.70E-05**	**2.07**
day_no_rain_10	31.34	4.76E-05	110.64	14.70	3.54E-05	73.36	1.12	3.73E-05	7.43	0.33	1.66E-05	2.91	0.41	1.55E-05	2.66	**0.27**	**9.05E-06**	**1.96**
day_no_rain_11	1.18	2.50E-05	11.79	2.47	3.16E-05	82.23	0.73	2.50E-05	3.12	0.26	1.53E-05	1.70	0.28	1.49E-05	2.86	**0.24**	**1.12E-05**	**1.38**
day_rain_0	23.37	2.41E-05	652.45	-	-	-	-	-	-	1.59	5.73E-05	65.56	4.06	2.88E-05	73.63	**0.31**	**1.33E-05**	**12.56**
day_rain_1	-	-	-	19.60	3.43E-05	228.22	-	-	-	0.49	3.33E-05	7.85	-	-	-	**0.31**	**1.52E-05**	**3.01**
day_rain_2	8.45	2.75E-05	34.99	19.52	8.92E-05	82.56	11.85	6.23E-04	130.85	-	-	-	-	-	-	**0.77**	**2.05E-05**	**4.91**
day_rain_3	-	-	-	31.31	1.11E-04	219.47	-	-	-	-	-	-	-	-	-	**0.37**	**1.15E-05**	**14.46**
day_rain_4	23.91	3.19E-04	138.17	29.49	1.94E-04	173.28	5.19	1.79E-04	27.15	6.59	4.39E-05	83.15	-	-	-	**0.72**	**2.14E-05**	**7.21**
day_rain_5	11.88	2.40E-05	25.51	4.65	2.71E-05	8.62	10.32	1.92E-04	18.70	0.44	3.60E-05	0.86	1.86	3.28E-05	2.90	**0.16**	**1.23E-05**	**0.51**
hwy_no_rain_0	-	-	-	-	-	-	-	-	-	2.21	4.02E-05	218.65	2.51	3.25E-05	190.31	**0.41**	**1.52E-05**	**52.94**
hwy_no_rain_1	-	-	-	-	-	-	2.92	3.26E-05	49.79	3.21	9.19E-05	888.29	2.91	4.01E-05	92.70	**0.59**	**1.49E-05**	**22.54**
hwy_no_rain_2	-	-	-	-	-	-	-	-	-	3.08	3.30E-05	**324.44**	3.34	3.56E-05	395.12	**0.45**	**1.40E-05**	377.00
hwy_no_rain_3	-	-	-	-	-	-	9.04	7.69E-05	852.20	5.53	7.38E-05	185.56	3.38	5.60E-05	84.41	**0.48**	**1.50E-05**	**36.58**
hwy_rain_0	-	-	-	-	-	-	-	-	-	0.74	2.57E-05	171.41	1.38	1.86E-05	168.77	**0.31**	**9.69E-06**	**63.98**
hwy_rain_1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
hwy_rain_2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

TABLE IX: Ablation study for the impact of the KMF-selection heuristics on the system performance.

Method	**KMF Heuristics**		RPE-T (cm/m)			RPE-R (rad/m)			ATE (m)			SR (%)
	reobservability	motion	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	@0.5	@0.9	AUC(%)	
Sync-S	✓		2.90	-	65.24	3.28E-05	-	71.39	68.56	-	68.02	82.67
Sync-S	✓	✓	**1.30**	-	**77.54**	**2.91E-05**	-	**78.37**	**24.53**	-	**77.44**	**84.00**

TABLE X: Number of KMFs selected per validation sequence, comparing ORB-SLAM2, ours stereo with reobservability-only heuristics, and ours stereo with a combined heuristics. Empty cells correspond to unfinished sequences.

sequence	ORB-S [20]	Ours-S (r-only)	Ours-S (combined)
day_no_rain_0	1718	1322	1759
day_no_rain_1	2249	1645	2054
day_no_rain_2	3795	2370	2690
day_no_rain_3	1245	969	1399
day_no_rain_4	2364	1734	2143
day_no_rain_5	-	1455	2034
day_no_rain_6	3373	-	2983
day_no_rain_7	501	464	633
day_no_rain_8	1724	1121	2104
day_no_rain_9	-	1598	1744
day_no_rain_10	1263	982	1505
day_no_rain_11	1593	1107	1559
day_rain_0	-	1522	2838
day_rain_1	-	1902	3043
day_rain_2	1097	-	-
day_rain_3	1952	2181	-
day_rain_4	2520	2339	3119
day_rain_5	383	408	531
hwy_no_rain_0	3646	3507	6532
hwy_no_rain_1	2400	2094	3994
hwy_no_rain_2	3358	2495	4632
hwy_no_rain_3	2124	2242	4484
hwy_rain_0	1835	2316	5132
hwy_rain_1	-	-	-
hwy_rain_2	-	-	-

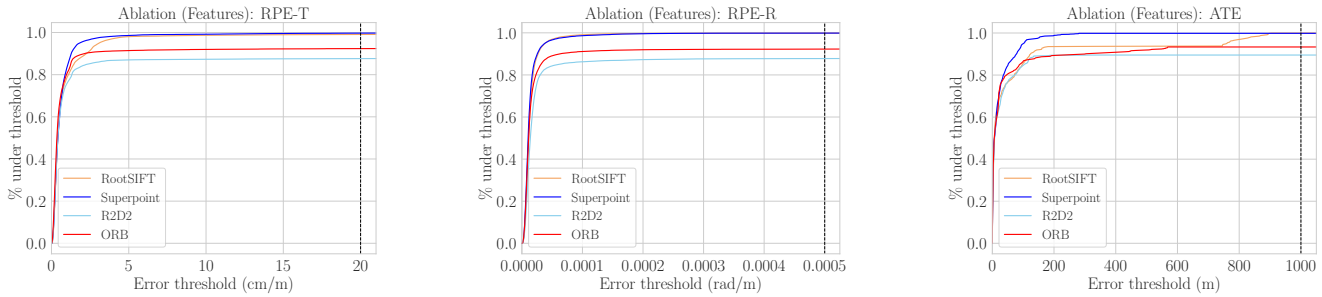

Fig. 8: Cumulative error curves of the keypoint extractor ablation study. Our experiments show that while ORB features still remain competitive, SuperPoint features lead to the best overall performance, especially in terms of translational error. However, this comes at a much higher computational cost.

such as reversing and parking.

c) Camera Ablation: Fig. 13 plots trajectories estimated with different camera configurations, highlighting failure cases resulted from camera configurations with a narrower field of view in challenging conditions like view obstruction, low light, rainy environments and low-textured highway driving.

d) Keypoint Extractor Ablation: Fig. 14 plots trajectories estimated by SLAM systems that use ORB, RootSIFT [22] and SuperPoint [23] respectively as the keypoint extractor. RootSIFT and SuperPoint trajectories visually align better with the ground truth and are able to complete more challenging rainy highway sequences.

2) Loop Closure: Figure 15 shows a failure case consisting in a false positive loop detection in the stereo setting. The large bus dominates the field of view of both cameras while also having rich texture due to the lights, ad, *etc*., causing a loop to be incorrectly closed. Multi-view loop closure correctly rejects this case and many similar others. This highlights the importance of multiple cameras for robust SLAM in the real world. For more qualitative results on the loop closure in the main system, please refer to the supplementary video.

3) Qualitative Map: Figures 16, 17, 18, and 19 showcase visualizations of some of the maps produced by AMV-SLAM. Please refer to our supplementary video for additional qualitative results.

References

[1] C. De Boor, "On calculating with B-splines," *Journal of Approximation Theory*, vol. 6, no. 1, pp. 50–62, 1972. 2

[2] M. G. Cox, "The numerical evaluation of B-splines," *IMA Journal of Applied Mathematics*, vol. 10, no. 2, pp. 134–149, 1972. 2

[3] M.-J. Kim, M.-S. Kim, and S. Y. Shin, "A general construction scheme for unit quaternion curves with simple high order derivatives," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 369–376. 2

[4] S. Lovegrove, A. Patron-Perez, and G. Sibley, "Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras," in *BMVC*, vol. 2, no. 5, 2013, p. 8. 2

[5] D. Droeschel and S. Behnke, "Efficient continuous-time SLAM for 3D lidar-based online mapping," in *ICRA*, 2018. 2

[6] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012. 2

[7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015. 2, 3, 5, 7

[8] 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. 2

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *IJRR*, vol. 32, no. 11, pp. 1231–1237, 2013. 4

[10] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and lidar dataset," *IJRR*, vol. 35, no. 9, pp. 1023–1035, 2016. 4

[11] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *IJRR*, vol. 36, no. 1, pp. 3–15, 2017. 4

[12] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford Multi-AV seasonal dataset," *arXiv preprint arXiv:2003.07969*, 2020. 4

[13] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn *et al.*, "A2D2: Audi autonomous driving dataset," *arXiv preprint arXiv:2004.06320*, 2020. 4

[14] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers, "4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving," *arXiv preprint arXiv:2009.06364*, 2020. 4

[15] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo Open Dataset," in *CVPR*, June 2020. 5

[16] 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 *CVPR*, June 2020. 5

[17] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *ICCV*, vol. 11, no. 1, 2011, p. 2. 5, 6, 8, 9

[18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, Nov. 2004. 5, 6

[19] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver.org. 5

[20] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source slam system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, 2017. 5, 7, 10

[21] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *IROS*. IEEE, 2018, pp. 2198–2204. 6, 7

[22] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012, pp. 2911–2918. 6, 9, 11

[23] D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperPoint: Self-supervised interest point detection and description," in *CVPR Workshops*, Jun. 2018. 6, 8, 9, 11

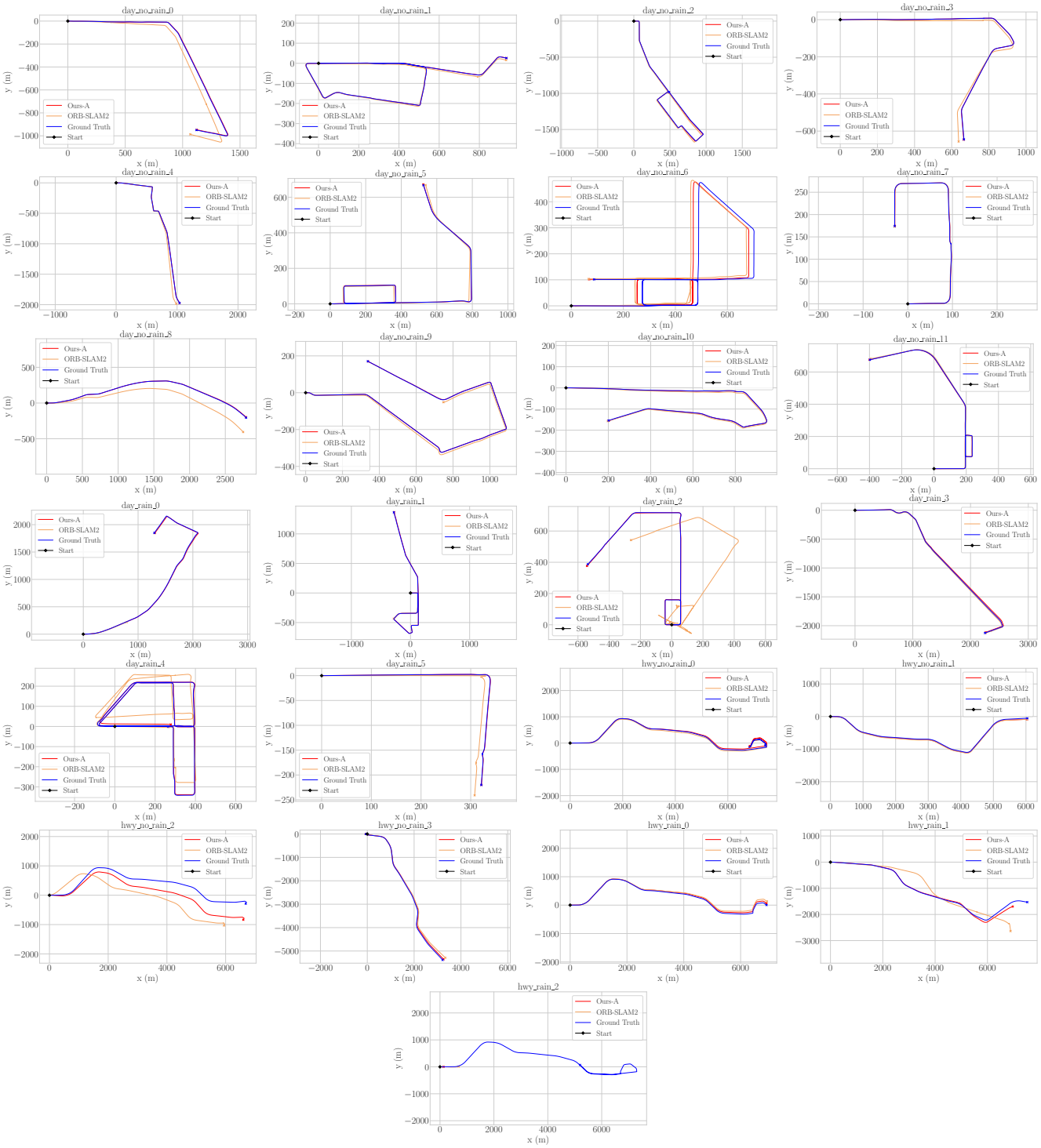

Fig. 9: Estimated trajectories in all 25 validation sequences, comparing ORB-SLAM2 (stereo) with our full system with all 7 cameras.

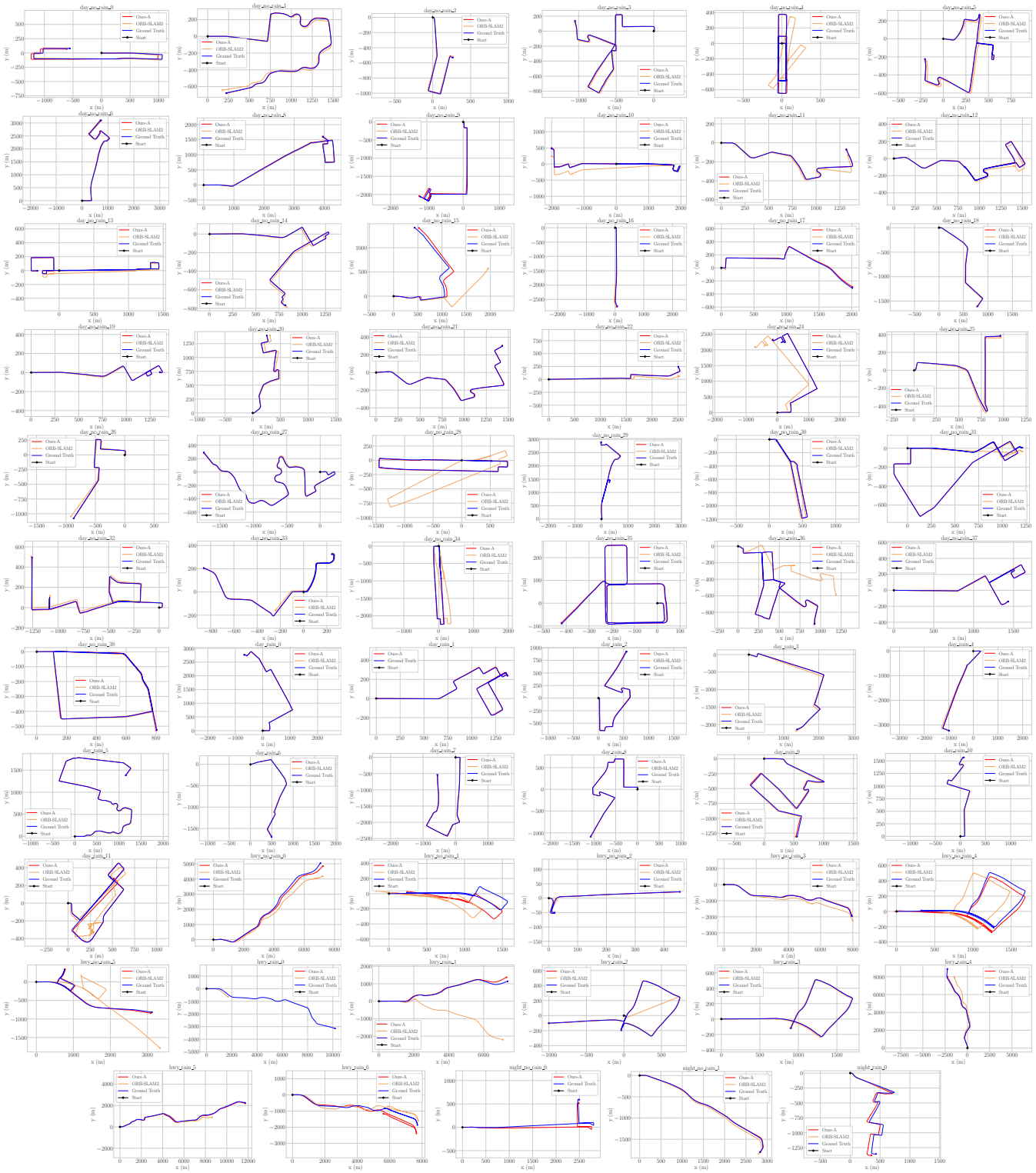

Fig. 10: Estimated trajectories in all 65 train sequences, comparing ORB-SLAM2 (stereo) with our full system (all seven cameras).

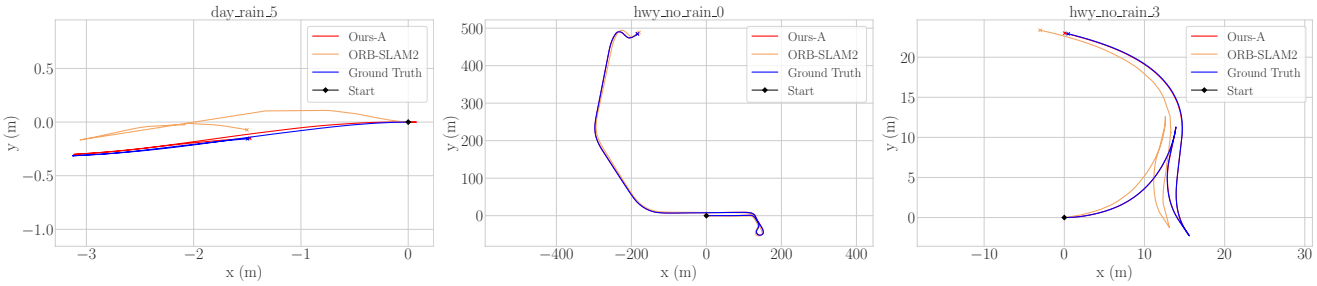

Fig. 11: Example maneuvers in the validation set, comparing ORB-SLAM2 and our full system with all 7 cameras. (Left) Reversing into a parallel parking spot. (Middle & Right) Maneuvers in a parking lot.

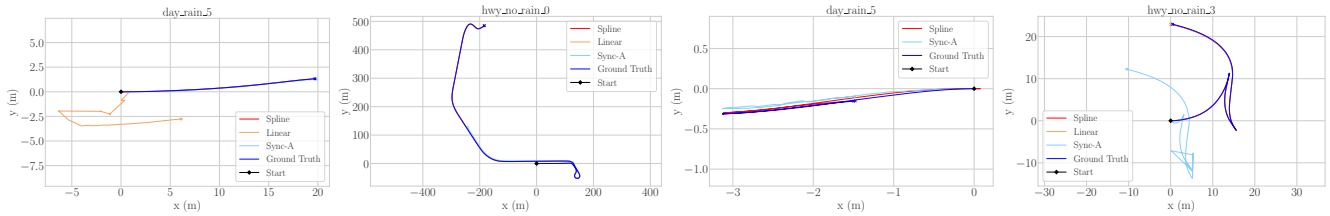

Fig. 12: Motion model ablation trajectories comparing our asynchronous cubic B-spline model, an asynchronous linear motion model, and a discrete-time motion model falsely assuming all cameras are synchronous. (Leftmost) Zoomed-in view on a segment where the linear motion model failed. The vehicle was at an intersection with many dynamic objects. (Right) Maneuvers in the validation set. The linear motion model is missing in the middle sequence because it failed before reaching the maneuver.

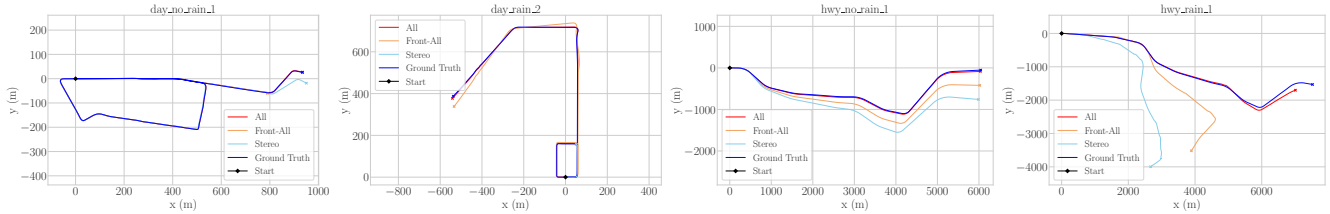

Fig. 13: Camera ablation trajectories estimated with (1) all 7 cameras, (2) all 3 wide front cameras + the stereo pair, and (3) the stereo pair only. The leftmost scenario happened at an intersection where the front view was obstructed by a huge truck that was making a turn. Front-all failed due to repeatedly inconsistent bundle adjustment results, while stereo persisted with a visible rotation error. The second-left scenario is a rainy dusk environment with high volume of traffic. The two scenarios on the right correspond to fast highway driving.

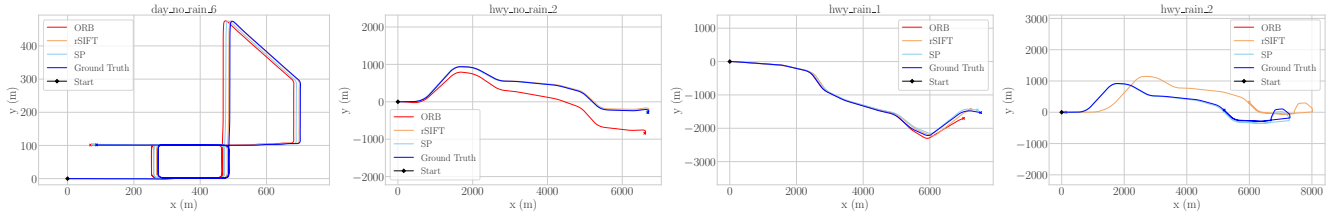

Fig. 14: Keypoint extractor ablation trajectories estimated with ORB, RootSIFT and SuperPoint. RootSIFT and SuperPoint have smaller absolute errors overall and finish a higher percentage of the challenging rainy highway sequences.

[24] J. Revaud, C. De Souza, M. Humenberger, and P. Weinzaepfel, "R2D2: Reliable and repeatable detector and descriptor," in *NIPS*, 2019, pp. 12 405–12 415. 6, 9

[25] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *PAMI*, vol. 40, no. 3, pp. 611–625, 2017. 7

[26] C. Sommer, V. Usenko, D. Schubert, N. Demmel, and D. Cremers, "Efficient derivative computation for cumulative b-splines on lie groups," in *CVPR*, 2020, pp. 11 148–11 156. 6, 8

[27] A. Haarbach, T. Birdal, and S. Ilic, "Survey of higher order rigid body motion interpolation methods for keyframe animation and continuous-time trajectory estimation," in *2018 International Conference on 3D Vision (3DV)*, 2018, pp. 381–389. 6

[28] H. Ovrén and P.-E. Forssén, "Spline error weighting for robust visual-inertial fusion," in *CVPR*, 2018, pp. 321–329. 6

[29] ——, "Trajectory representation and landmark projection for continuous-time structure from motion," *IJRR*, vol. 38, no. 6, pp. 686–701, 2019. 6

Fig. 15: Example where stereo-only loop detection fails due to the presence of the same large bus in two geographically distant frames. This sample is from the training set sequence titled `day_rain_7`.

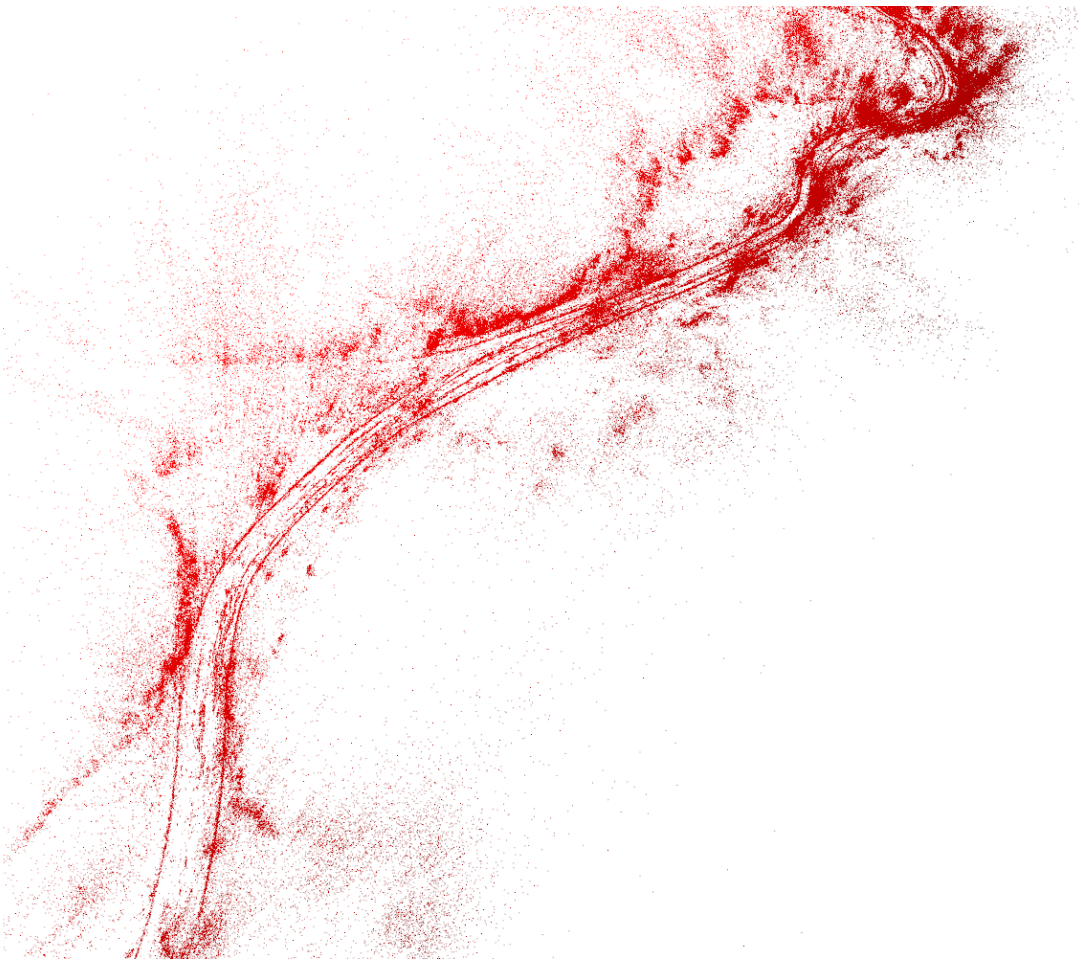

Fig. 16: Qualitative example of the 3D structure produced by the AMV-SLAM system. Note the system's ability to sharply reconstruct the road boundaries, in addition to the surrounding vegetation. This example is from the training set sequence titled `day_rain_5`.

Fig. 17: Reconstructed point cloud from the training sequence `day_rain_5`. Post-processed to include colors from the original camera images. Best viewed in electronic format.

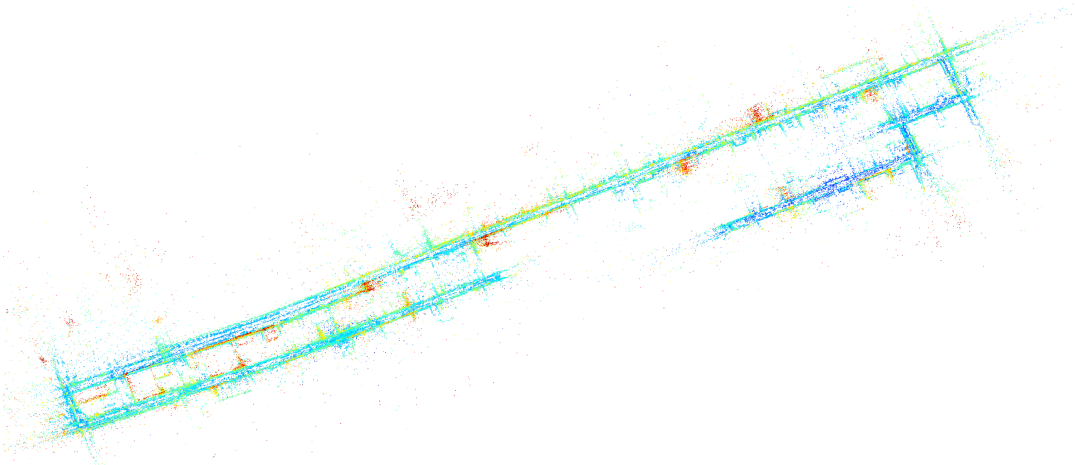

Fig. 18: Reconstructed point cloud from the training sequence `day_no_rain_0`.

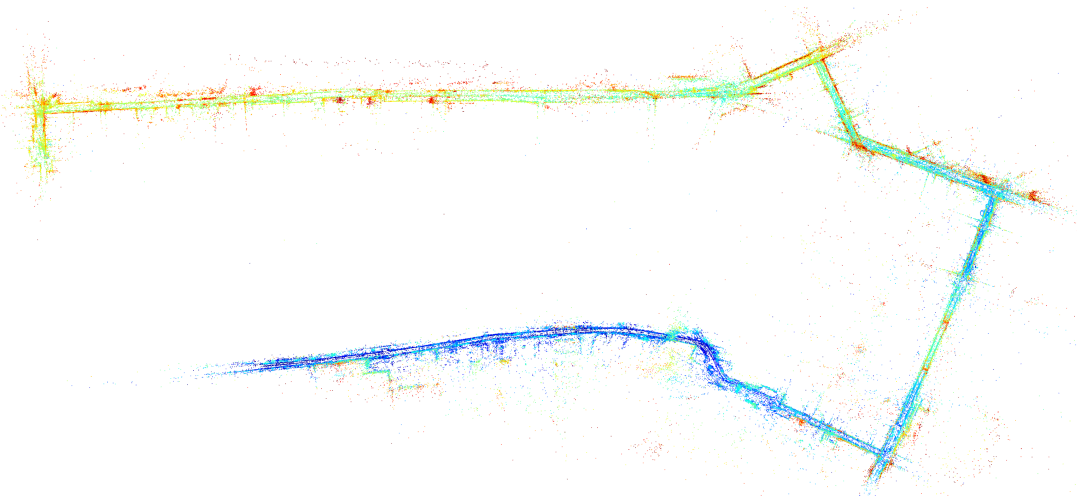

Fig. 19: Overview of a reconstruction produced from the training set sequence `day_rain_7` by our system. The point cloud is colored by the height (Z) of each point, in the map reference frame.