Supplementary Material for
HybVIO: Pushing the Limits of Real-time Visual-inertial Odometry

A. Examples and experiment details

Differentiation example Consider the case where the triangulation is performed using two poses $\boldsymbol{\pi}^{(1)}$, $\boldsymbol{\pi}^{(2)}$ in the stereo setup:

$$\begin{aligned}\mathbf{p}^* &= \mathrm{TRI}(\boldsymbol{\pi}^{(1)}, \boldsymbol{\pi}^{(2)}, \mathbf{y}^{1,L}, \mathbf{y}^{1,R}, \mathbf{y}^{2,L}, \mathbf{y}^{2,R}) \\ &= \mathrm{TRI}_{\mathrm{rays}}(\mathbf{p}_{L,1}, \mathbf{r}_{L,1}, \mathbf{p}_{L,2}, \mathbf{r}_{L,2}, \mathbf{p}_{R,1}, \mathbf{r}_{R,1}, \mathbf{p}_{R,2}, \mathbf{r}_{R,2}),\end{aligned}$$

where the ray origin $\mathbf{p}_{C,j}(\boldsymbol{\pi}^{(j)})$ and bearing $\mathbf{r}_{C,j} = \mathbf{R}_C(\mathbf{q}^{(j)})\phi_C(\mathbf{y}^{j,C})$ can be computed from Eq. (9). Then the Jacobian of the triangulated point $\mathbf{p}^*$ with respect to $\mathbf{p}^{(1)}$ can be computed using the chain rule as

$$\frac{\partial \mathbf{p}^*}{\partial \mathbf{p}^{(1)}} = \frac{\partial \mathrm{TRI}_{\mathrm{rays}}}{\partial \mathbf{p}_{L,1}} + \frac{\partial \mathrm{TRI}_{\mathrm{rays}}}{\partial \mathbf{p}_{R,1}}, \tag{A1}$$

because $\frac{\partial \mathbf{p}_{C,1}}{\partial \mathbf{p}^{(1)}} = I_3$ and $\frac{\partial \mathbf{a}}{\partial \mathbf{p}^{(1)}} = \mathbf{0}_3$ for all other arguments $\mathbf{a}$ of $\mathrm{TRI}_{\mathrm{rays}}$. The other blocks in the full Jacobian can be computed in a similar manner.

Quaternion update by angular velocity If $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ represents a constant angular velocity, then a world-to-local quaternion $\mathbf{q} = (q_w, q_x, q_y, q_z)^\top$ representing the orientation of a body transforms as

$$\mathbf{q}(t_0 + \Delta t) = \boldsymbol{\Omega}[\boldsymbol{\omega}\Delta t]\mathbf{q}(t_0) \tag{A2}$$

where

$$\boldsymbol{\Omega}[\mathbf{u}] := \exp\left[-\frac{1}{2}\begin{pmatrix} 0 & -u_x & -u_y & -u_z \\ u_x & 0 & -u_z & u_y \\ u_y & u_z & 0 & -u_x \\ u_z & -u_y & u_x & 0 \end{pmatrix}\right]. \tag{A3}$$

Note that the matrix looks different if a local-to-world quaternion representation is used (*cf.* [39]).

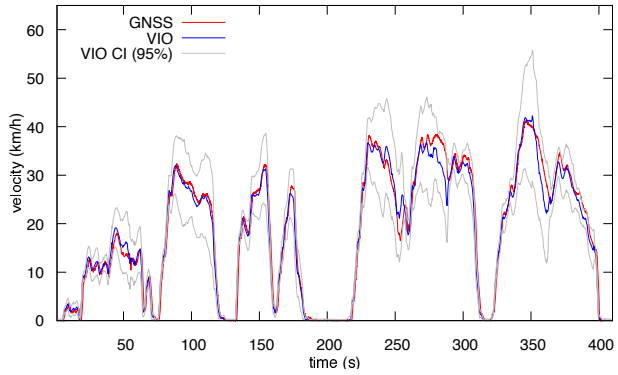


Figure A1. VIO velocity estimate for Fig. 4a, HybVIO on ARKit

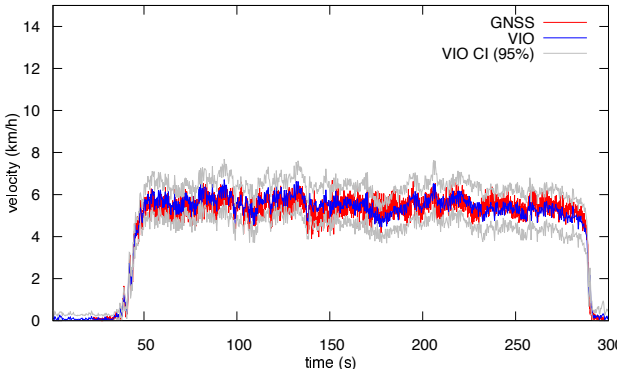


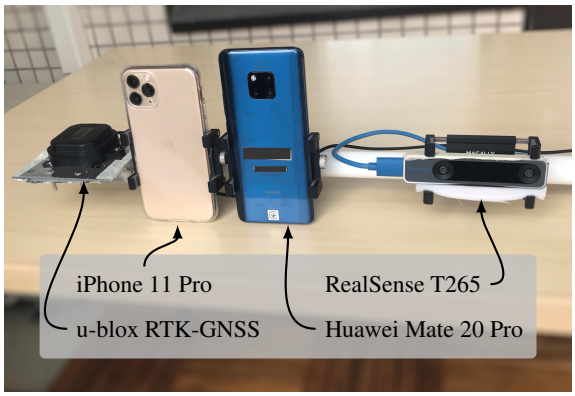
Figure A2. VIO velocity estimate for Fig. 4b, HybVIO on ARKit

(a) Intel RealSense T265 (left camera)

(b) Huawei Mate 20 Pro (through ARCore)

(c) iPhone 11 Pro (through ARKit)

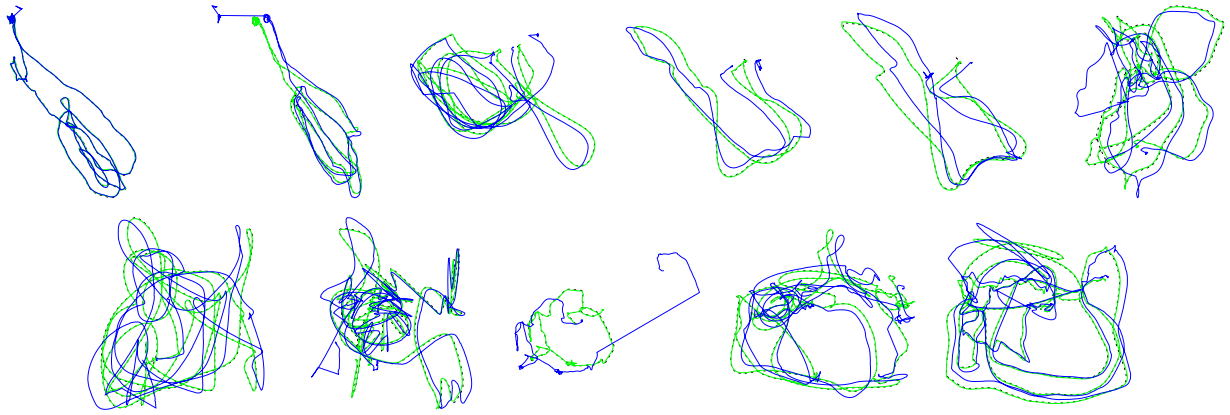
Figure A3. Example camera views in the vehicular experiment Fig. 4a. Reflections from the hood or windshield are visible in all images, and especially prominent in the RealSense fisheye camera.



(a) Devices, recorded as in Fig. 3

(b) Example camera view: Huawei Mate 20 Pro (through ARCore)

Figure A4. Walking experiment setup Fig. 4b



(a) ORB-SLAM3. Due to successful loop closures, the method eventually recovers and is able to produce accurate post-processed trajectories.

(b) BASALT

(c) HybVIO (normal SLAM / post-processed SLAM)

Figure B1. Online (blue) and post-processed (green) trajectories in EuRoC MAV stereo mode, compared to ground truth (dashed) for three different methods. Our method and BASALT produce good results in both online and post-processed modes.

B. Additional experiments

B.1. Comparison to ORB-SLAM3 and BASALT

We processed the EuRoC dataset using the publicly available source code of ORB-SLAM[1] and BASALT[2] to compare execution times and reproduce the metrics reported in [5, 41]. The ORB-SLAM3 example code was modified to output its intermediary results, namely the latest key frame pose after each input frame, without any changes to the actual algorithm. The results are presented in Table B1.

All tests were performed on the same machine (the *Ryzen* setup described in Sec. 4.1) using two configurations: In the first, unrestricted configuration, the methods were allowed to utilize all 12 CPU cores in the system in parallel. BASALT was most efficiently parallelized and its processing time per frame was significantly lower than in the other methods. In the second configuration, we restricted the entire process (including input decoding) to use only 2 parallel CPU cores. In this mode, the processing times of the three methods were comparable.

The common results in Table B1 are similar to those reported in [5, 41] (and reproduced in Table 2). None of the methods (including ours) yield strictly equal results on different machines, which can explain the small remaining discrepancies. ORB-SLAM3 outputs also varied significantly between runs, but the online instability seen in Fig. B1 was consistently observed.

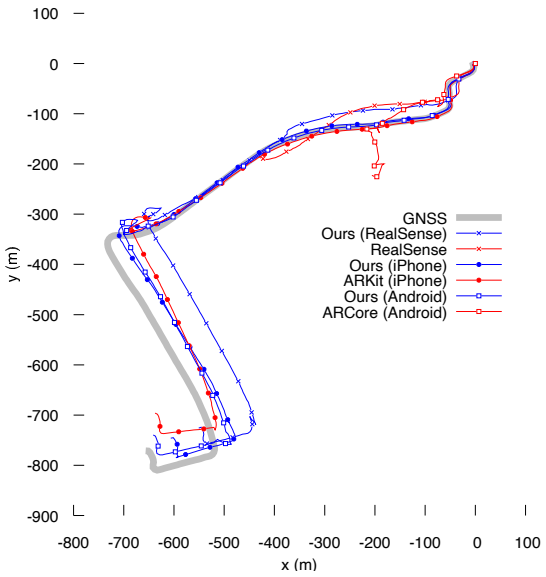
B.2. Ablation studies

An ablation study equivalent to Table 3 for the TUM Room dataset is presented in Table B2. In the monocular case, the results are mixed: disabling individual novel features does not consistently improve the metrics, and the simple post-processing actually degrades the SLAM results. However, most of our configurations, notably also *Fast stereo VIO*, outperform all previous methods except ORB-SLAM3 (*cf.* Table 4)

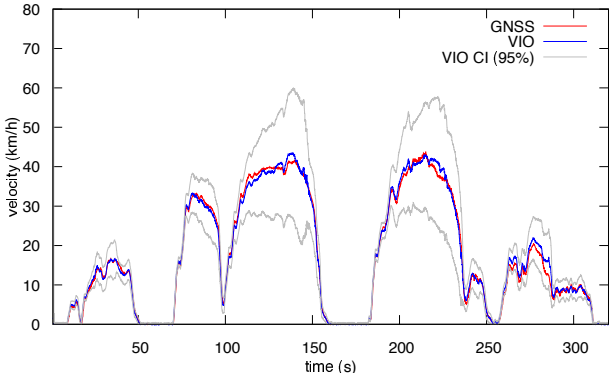
The SenseTime benchmark results in Table B3 are more consistent and similar to Table 3: the novel features are all beneficial and the baseline PIVO implementation is not stable. However, our simple post-processing method is not able to improve the results compared to the online case.

Table B4 studies the effect of varying the parameters presented in Table 1 individually. Deviations from the selected parameters caused degraded metrics, except increasing n_{BA} improves the baseline results. However, this larger bundle adjustment problem is too heavy for the real-time use case and therefore we only use it in the post-processed setting.

[1] https://github.com/UZ-SLAMLab/ORB_SLAM3 (V0.4)
[2] https://github.com/VladyslavUsenko/basalt-mirror (June 7, 2021)



(a) Position



(b) VIO velocity estimate, HybVIO on ARKit

Figure B2. Vehicular experiment 2, using the setup in Fig. 3

B.3. Vehicular

This section includes additional vehicular experiments (Fig. B2 and Fig. B3) using the setup shown in Fig. 3, as well as results from a slightly modified setup (shown in Fig. B5a), where we have added ZED 2 as a new device.

ZED 2 also has a proprietary VISLAM capability, but it did not perform well in the vehicular test cases (see Fig. B4 for an example) and we omitted it in the other sequences to avoid frame drop issues experienced when recording ZED 2 input and tracking output data simultaneously. The ZED 2 camera data was recorded at 60FPS but utilized at 30FPS.

We used the *normal VIO* mode (see Table 1) for all vehicular experiments. Stereo mode was used with both stereo camera devices.

Table B1. EuRoC computational times and RMSE in different methods (stereo SLAM). Also shown in Fig. B1.

	Method	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Mean	Ryzen frame time (ms) all CPUs	Ryzen frame time (ms) 2 CPUs
online	Ours[(1)]	0.088	0.080	0.038	0.071	0.108	0.044	0.035	0.040	0.075	0.041	0.052	0.061	32	47
	ORB-SLAM3	0.094	1.229	1.124	1.887	2.177	0.698	2.036	0.529	3.488	1.498	0.445	1.382	56	78
	BASALT	0.080	0.052	0.078	0.106	0.120	0.045	0.058	0.088	0.035	0.073	0.897	0.148	5	36
post-pr.	Ours[(3)]	0.048	0.028	0.037	0.056	0.066	0.038	0.035	0.037	0.031	0.029	0.044	0.041	52	95
	ORB-SLAM3	0.033	0.030	0.031	0.056	0.100	0.036	0.014	0.025	0.037	0.016	0.019	0.036	56	78
	BASALT	0.085	0.065	0.056	0.105	0.099	0.046	0.033	0.035	0.041	0.028	0.175	0.070	14	66

Table B2. Different configurations of HybVIO on the TUM Room dataset (*cf.* Table 3 and Table 4).

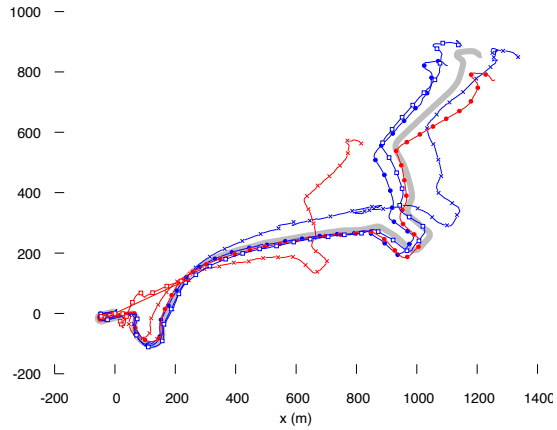
		Method	R1	R2	R3	R4	R5	R6	Mean
online	stereo	Normal SLAM	0.016	0.013	0.011	0.013	0.02	0.01	0.014
		Normal VIO	0.05	0.053	0.041	0.042	0.082	0.033	0.050
		⧹ Eq. (5)	0.072	0.052	0.037	0.042	0.11	0.058	0.062
		Fast VIO	0.075	0.064	0.074	0.041	0.07	0.037	0.060
		⧹ Eq. (18)	0.09	0.11	0.055	0.052	0.065	0.083	0.076
	mono	Normal SLAM	0.02	0.02	0.17	0.018	0.019	0.017	0.044
		Normal VIO	0.08	0.06	0.17	0.036	0.079	0.06	0.080
		⧹ RANSAC	0.065	0.072	0.092	0.058	0.06	0.049	0.066
		⧹ Eq. (4)	0.089	0.062	0.23	0.057	0.094	0.07	0.101
		⧹ Eq. (6)	0.09	0.066	0.21	0.05	0.069	0.049	0.089
		⧹ Sec. 3.9	0.087	0.06	0.14	0.046	0.079	0.06	0.078
		⧹ Eq. (5)	0.083	0.083	0.081	0.07	0.066	0.068	0.075
		PIVO baseline	0.075	0.077	0.11	0.051	0.14	0.071	0.088
		Fast VIO	0.086	0.061	0.066	0.077	0.061	0.07	0.070
		⧹ Eq. (18)	0.09	0.062	0.12	0.082	0.08	0.051	0.082
post-pr.		Stereo SLAM	0.042	0.041	0.028	0.025	0.061	0.02	0.036
		Mono SLAM	0.039	0.033	0.16	0.032	0.039	0.026	0.055

Table B3. Different configurations of HybVIO in the SenseTime benchmark (*cf.* Table 3 and Table 5).

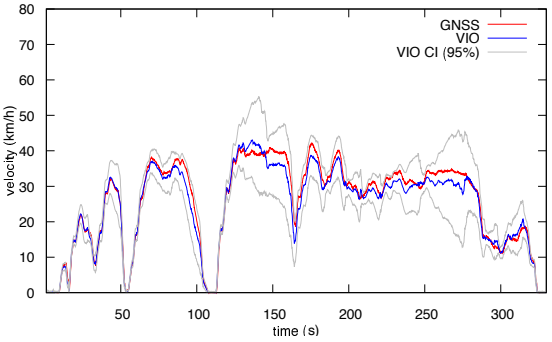
	Method	A0	A1	A2	A3	A4	A5	A6	A7	Mean
online	Normal SLAM	49.9	30	36	22.2	19.6	37.8	29.3	17.3	30.3
	Normal VIO	63.5	53.1	53.9	28.1	26.3	75.6	29.7	26.6	44.6
	⧹ RANSAC	73.1	109	59	27.6	24.2	65.8	30	22	51.3
	⧹ Eq. (4)	171	272	208	40.8	83.6	170	116	120	148
	⧹ Eq. (6)	69.5	70.9	47.8	33.9	31.4	74.9	32.2	43.1	50.5
	⧹ Sec. 3.9	87.7	764	149	75.6	158	351	310	328	278
	⧹ Eq. (5)	75.2	46.3	53	27.7	34	72.9	31.2	27.6	46
	PIVO baseline	166	1150	225	219	242	472	109	239	353
	Fast VIO	47.2	59.6	43.1	25.8	46	60.5	30.6	40	44.1
	⧹ Eq. (18)	80	73.4	89.1	26.8	50	119	38	50.2	65.8
Post-pr. SLAM		63.9	28.4	28.5	23.2	42.7	23.3	22.2	18.1	31.3

Table B4. Effect of individual parameters in Table 1 on the mean RMSE and frame time in EuRoC. The baseline is *Normal SLAM*.

Altered parameter	Value	RMSE	Frame time (ms)
baseline[(1)]		0.061	35
feature detector	FAST	0.067	33
subpix. adjustment	off	0.065	33
max. features	70	0.066	19
max. features	100	0.065	23
max. features	300	0.061	48
max. itr.	8	0.064	35
max. itr.	40	0.066	36
window size	13	0.062	34
window size	51	0.07	37
n_{a}	30	0.062	43
n_{target}	5	0.063	33
n_{target}	10	0.064	37
n_{target}	30	0.061	34
n_{FIFO}	20	0.068	36
n_{FIFO}	14	0.31	35
n_{BA}	50	0.053	42
n_{BA}	100	0.055	79
n_{matching}	35	0.06	36
n_{matching}	50	0.06	36



(a) Position



(b) VIO velocity estimate, HybVIO on ARKit

Figure B3. Vehicular experiment 3, using the setup in Fig. 3

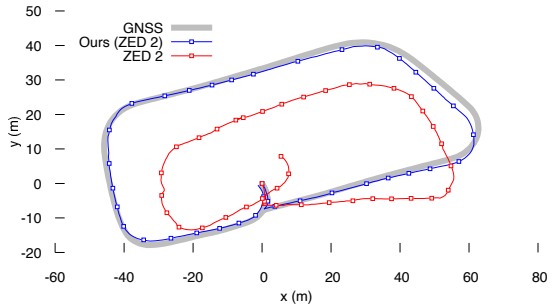
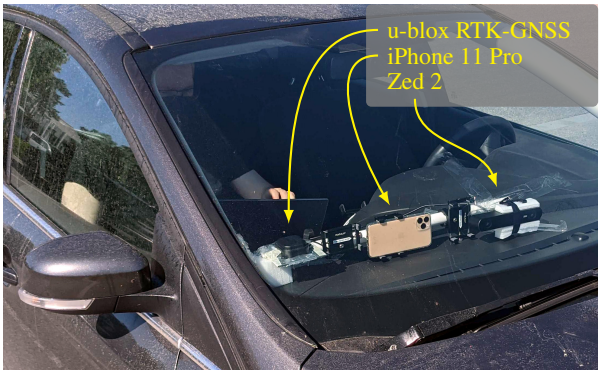
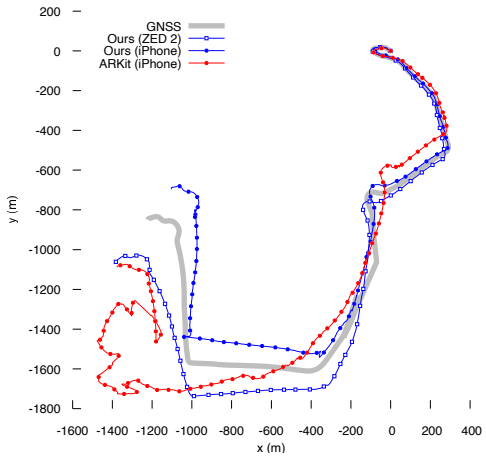


Figure B4. Vehicular experiment 4. A slow drive around a parking lot, recording the setup shown in Fig. B5a. Unlike the following experiments with ZED 2, the proprietary tracking output from ZED 2 is compared to HybVIO using the same input data.

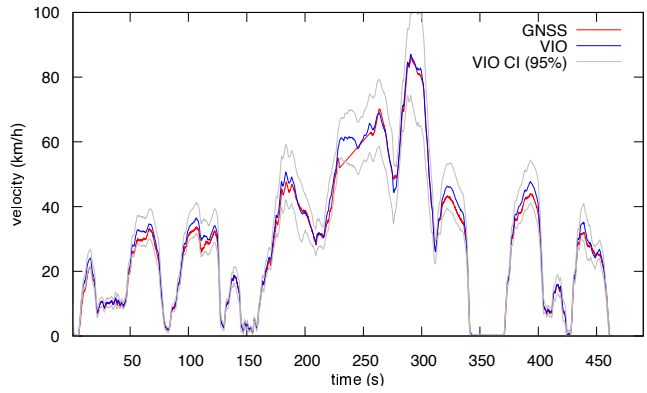


(a) Second car experiment setup: GNSS is used as ground truth. The iPhone records ARKit and its input data simultaneously. ZED 2 records camera (stereo rolling shutter at 60FPS) and IMU data.

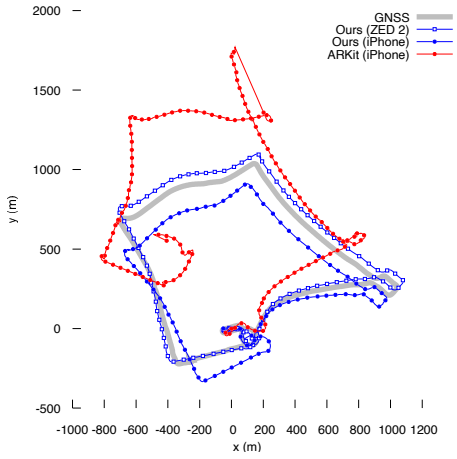
(b) Example ZED 2 left camera view corresponding to (c)



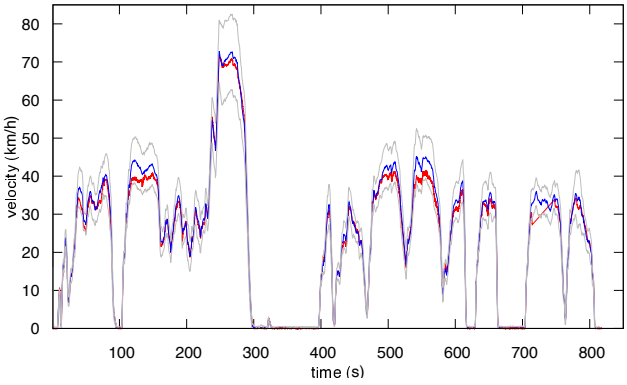
(c) Vehicular experiment 5



(d) VIO velocity estimate for (c), HybVIO on ZED 2. A GNSS outage is visible as a straight line segment near the 250 seconds mark.



(e) Vehicular experiment 6



(f) VIO velocity estimate for (e), HybVIO on ZED 2. A GNSS outage is visible after 700 seconds.

Figure B5. Additional vehicular experiments with higher velocities ($\sim$ 80km/h), in which ARKit also fails. In this case, HybVIO performs better on the same data.

C. Supplementary videos

The following video files are included in the supplementary material

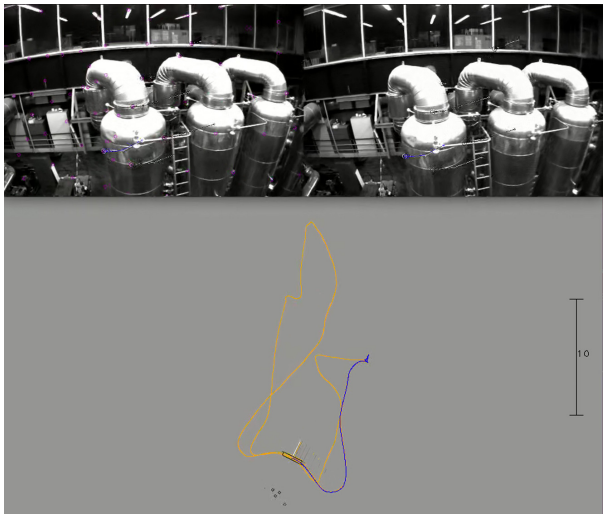
(a) **euroc-mh-05-difficult_fast-VIO** (**https://youtu.be/ou1DrtjPx1Q**) A screen recording from a laptop running HybVIO in *fast VIO* mode (*cf.* Table 1) on the EuRoC MAV sequence MH-05. The final trajectory also appears in Fig. 1.

The visual tracking and update status are visualized on the left and right camera frames, similarly to Fig. 2 but with different colors: reprojections are in white, successfully updated tracks in black and failed tracks in blue. The lower part of the video shows the online track (x and y coordinates) in blue and ground truth in orange. Triangulated points are shown as small black circles. The online track is automatically rotated to optimally match the ground truth, after approximately 10 seconds.

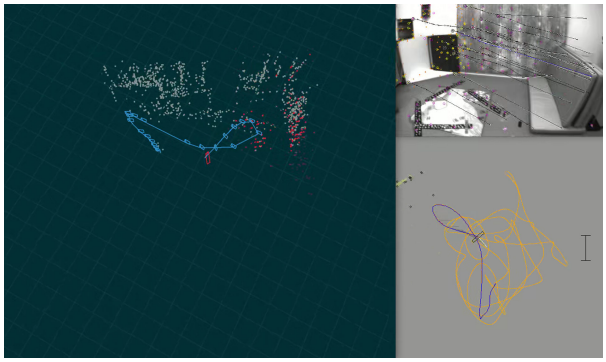
(b) **euroc-v1-02-medium_normal-SLAM** (**https://youtu.be/7j1rYoD_pPc**) Similar to the previous video, but HybVIO is running in the *normal SLAM* mode on the sequence V1-02. The triangulated SLAM map points in the current local map are shown as yellow in the lower right subimage, and their reprojections as orange on the (left) camera image. The LK-tracked features (*cf.* Alg. 1) that correspond to SLAM map points are shown as yellow circles on the camera image.

The left part of the video shows the SLAM map. Key frame camera poses are shown in light blue. In the beginning, the triangulated points in the local map are shown in red. At time 00:27, the colors are changed to show the observation direction of the map point. The *covisibility graph* is shown first at 00:15, in a yellow-green color. We consider a pair of key frames adjacent in this graph if they observe at least $N_{\text{neigh}} = 10$ common map points.

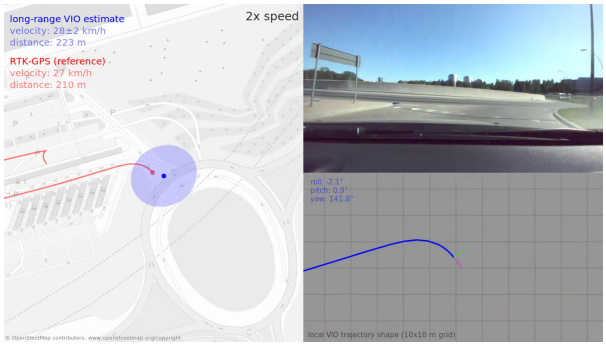
(c) **vehicular-experiment-6** (**https://youtu.be/iVNicL_S14Y**) Visualizes the vehicular experiment in Fig. B5e on a map (HybVIO on ZED 2). The VIO trajectory is aligned using a fixed angle and offset. The *local VIO trajectory* is formed using the pose trails (*cf.* Sec. 3.1) in the VIO state. The traffic light stops are automatically cut from the video (based on the VIO velocity estimate). Despite generally good RTK-GNSS coverage in the area, the sequence includes a GNSS outage in a tunnel, starting at time 02:32 in the video.



(a) euroc-mh-05-difficult_fast-VIO



(b) euroc-v1-02-medium_normal-SLAM



(c) vehicular-experiment-6

Figure C1. Screenshots from the supplementary videos