Image as an IMU: Estimating Camera Motion from a Single Motion-Blurred Image

Supplementary Material

1. Pixel Motion Derivation

Suppose $\mathbf{P} \in \mathbb{R}^3$ is a 3D point in space, f is the focal length of the camera, and d is the depth. Given the standard pinhole camera perspective projection equation:

$$\mathbf{p} = f\frac{\mathbf{P}}{d}, \tag{1}$$

Trucco and Verri [2] derive Equations 2, 3 relating pixel velocity and the camera motion by taking the time derivative of both sides of Equation 1:

$$\begin{aligned}\dot{p}_x = &\frac{v_z p_{x,1} - v_x f}{d} - \omega_y f + \omega_z p_{y,1} \\ &+ \frac{\omega_x p_{x,1} p_{y,1}}{f} - \frac{\omega_y (p_{x,1})^2}{f}\end{aligned} \tag{2}$$

$$\begin{aligned}\dot{p}_y = &\frac{v_z p_{y,1} - v_y f}{d} + \omega_x f - \omega_z p_{x,1} \\ &- \frac{\omega_y p_{x,1} p_{y,1}}{f} + \frac{\omega_x (p_{y,1})^2}{f}.\end{aligned} \tag{3}$$

We assume that the velocities $\mathbf{v}$ and $\boldsymbol{\omega}$ are constant across exposure. Therefore, we multiply each side by τ, the exposure time, and reformulate the pixel velocity equations:

$$\begin{aligned}F_x = &\frac{t_z p_x - t_x f}{d} - \theta_y f + \theta_z p_y \\ &+ \frac{\theta_x p_x p_y}{f} - \frac{\theta_y (p_x)^2}{f}\end{aligned}$$

$$\begin{aligned}F_y = &\frac{t_z p_y - t_y f}{d} + \theta_x f - \theta_z p_x \\ &- \frac{\theta_y p_x p_y}{f} + \frac{\theta_x (p_y)^2}{f}.\end{aligned}$$

2. Real-world Split Ablation

We show the importance of finetuning on real-world data. Table 1 shows how fine-tuning with the real-world (RW) split improves the RMSE by 16% for angular velocities and and 30% for translational velocities, averaged across all test sequences.

Table 1. Ablation showing rotational and translational velocity RMSE w/ and w/o RW-finetuning, averaged across the test split.

Dataset	ω_x / ω_y / ω_z	v_x / v_y / v_z
Synthetic	1.56 / 1.08 / 2.01	1.92 / 1.43 / 1.47
+ RW-finetuning	**1.22 / 0.91 / 1.76**	**1.11 / 1.03 / 0.92**

3. Downstream VIO Application

We demonstrate a simple VIO example on one of our test sequences with a factor graph approach using GTSAM [1]. We compare the RMSE for 3 approaches: 1) MASt3R-only, 2) MASt3R + IMU, 3) MASt3R + ours. Table 2 shows how our method reduces the RMSE by 30%.

Table 2. Translation RMSE for three factor graph approaches. We show percent improvement from the MASt3R-only baseline.

MASt3R-only	MASt3R + IMU	MASt3R + Ours
1.89	1.68 (-11.25%)	**1.31** (-30.58%)

4. Test Sequence Visuals

Figure 1 shows several frames of each test sequence to demonstrate the severity of the motion blur.

5. Additional Results

5.1. Velocity Traces

Figures 2-4 show the velocity prediction comparisons for the rest of the test sequences: commonroom, dining, and office, respectively. Since COLMAP failed for the dining sequence, we show the output from DROID-SLAM.

5.2. Absolute Error Plots

We additionally include a time series comparison of the absolute error for all test sequences, shown in Figures 5-8. Again, we show the absolute error of DROID-SLAM instead of COLMAP in the dining sequence.

References

[1] Frank Dellaert and GTSAM Contributors. borglab/gtsam, 2022. 1

[2] Emanuele Trucco and Alessandro Verri. *Introductory Techniques for 3-D Computer Vision*, pages 178–184. Prentice Hall PTR, USA, 1998. 1

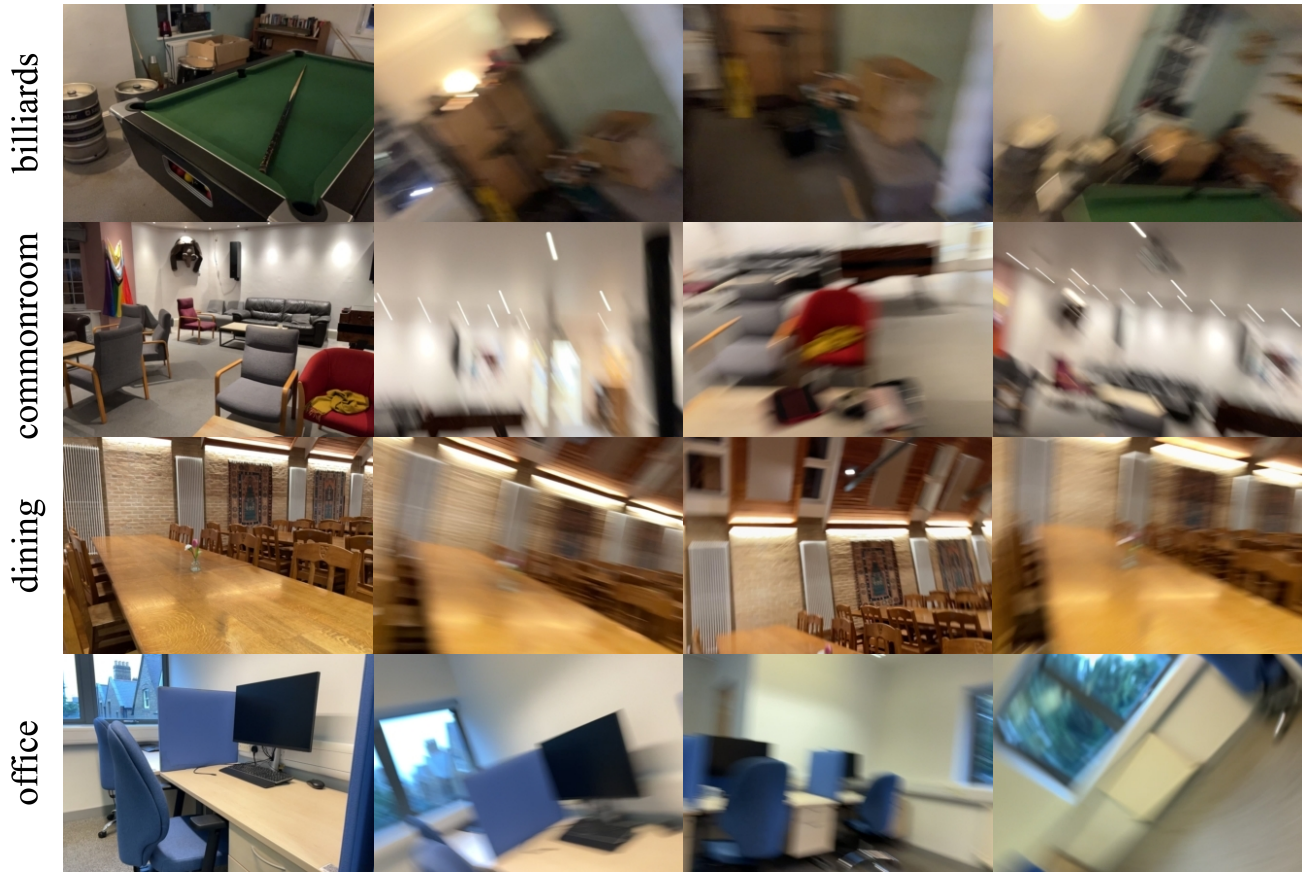

Figure 1. Frames that highlight the motion blur extent in the evaluation sequences.

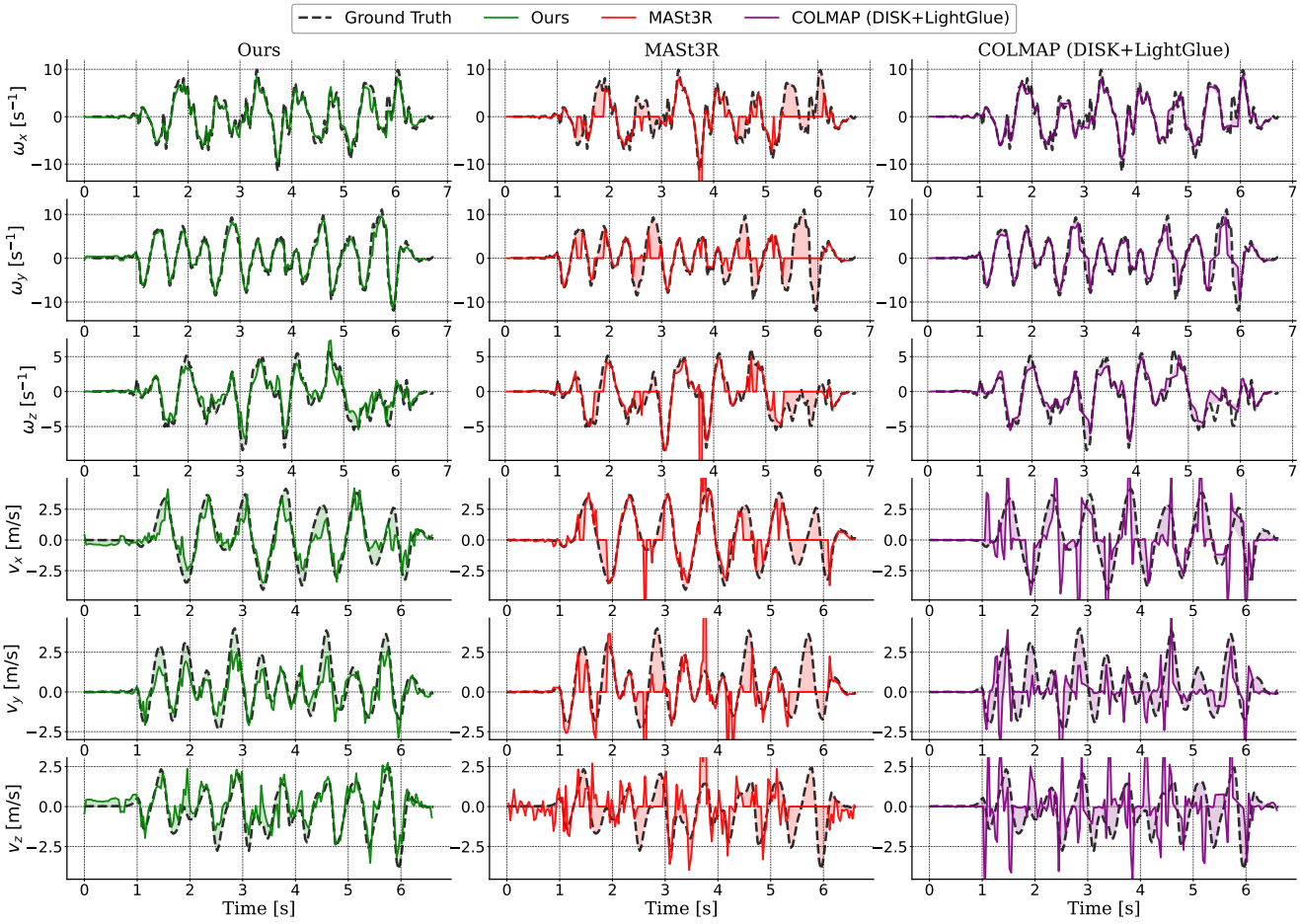

Figure 2. Velocity prediction comparison for the commonroom sequence.

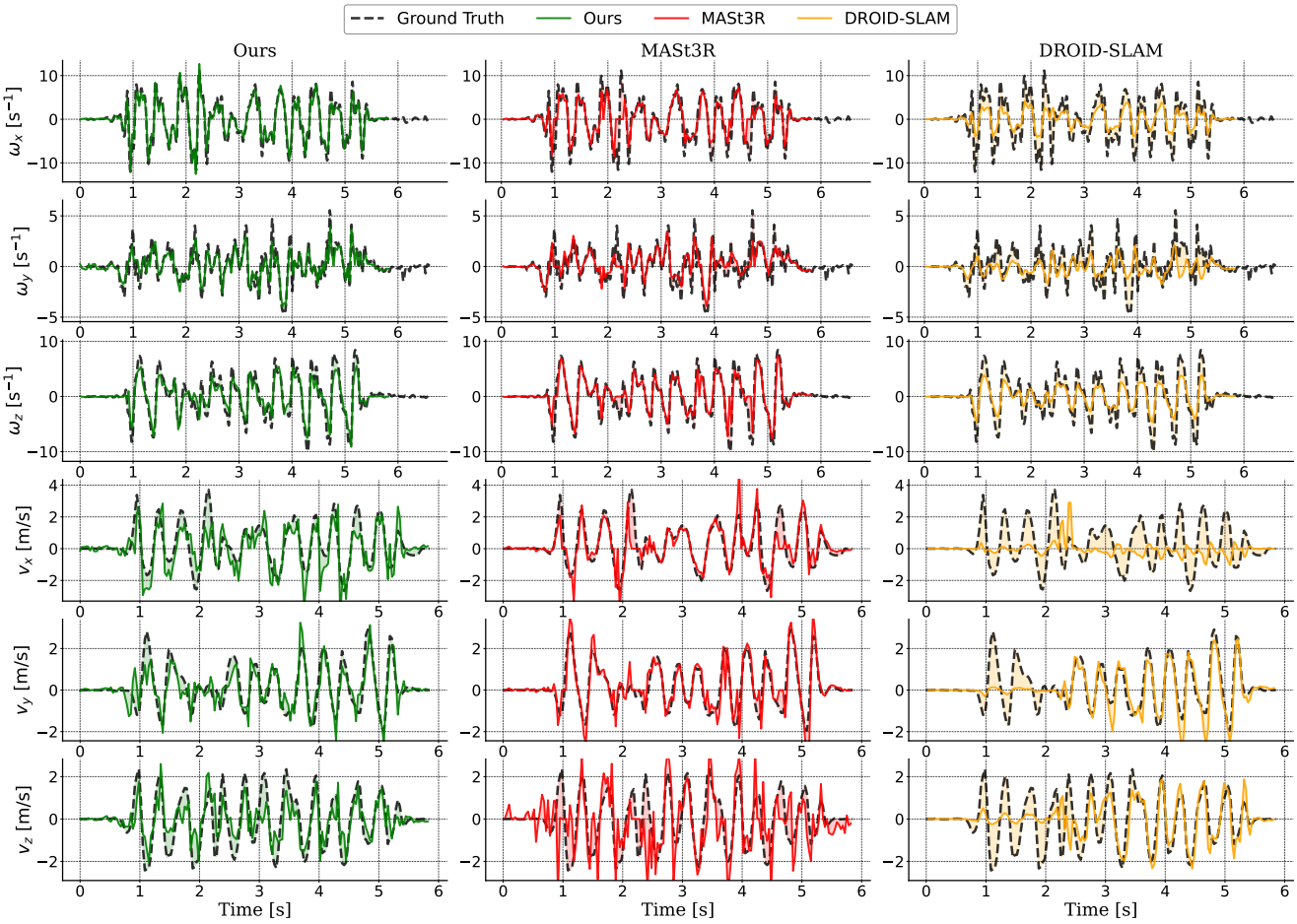

Figure 3. Velocity prediction comparison for the dining sequence.

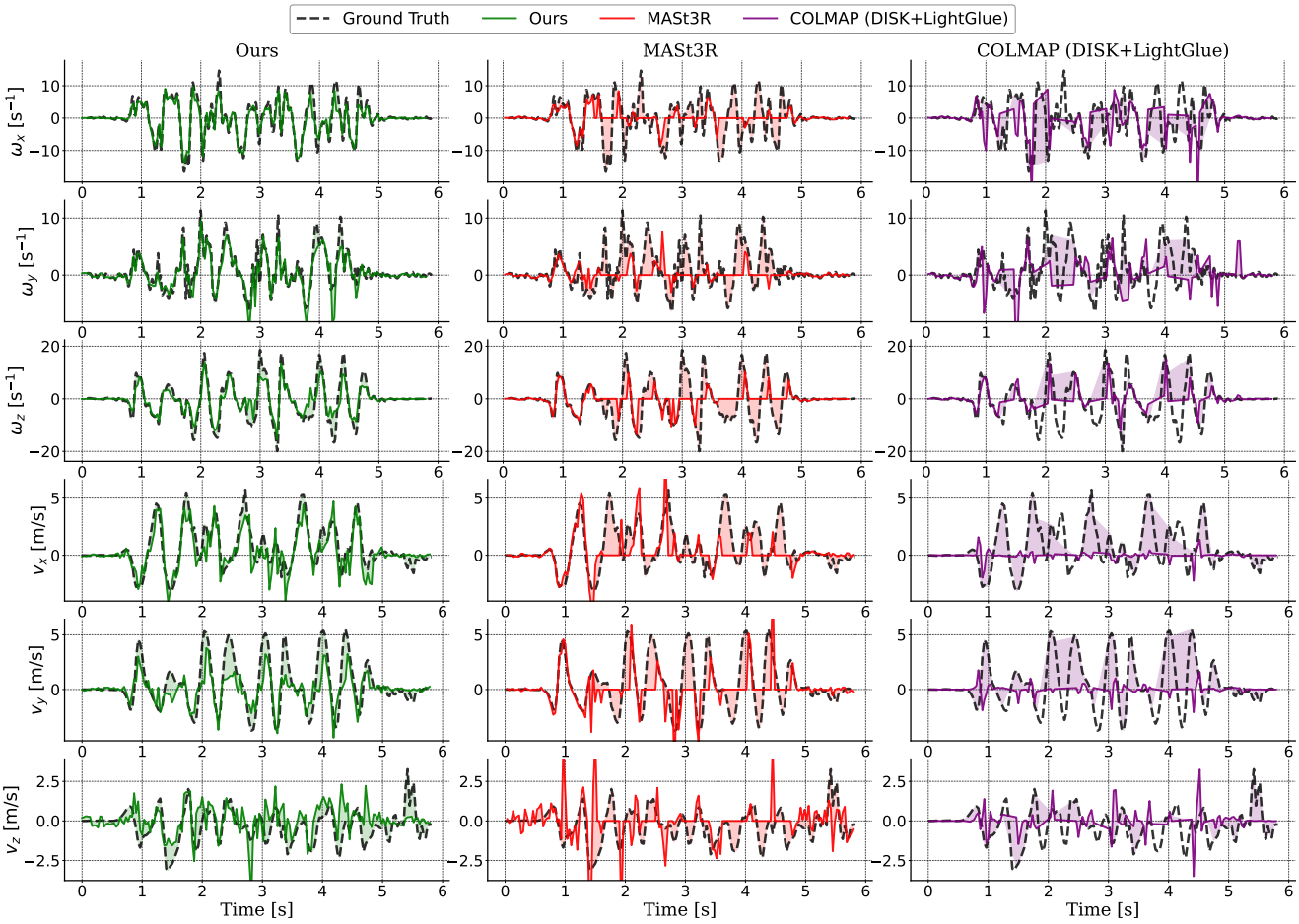

Figure 4. Velocity prediction comparison for the office sequence.

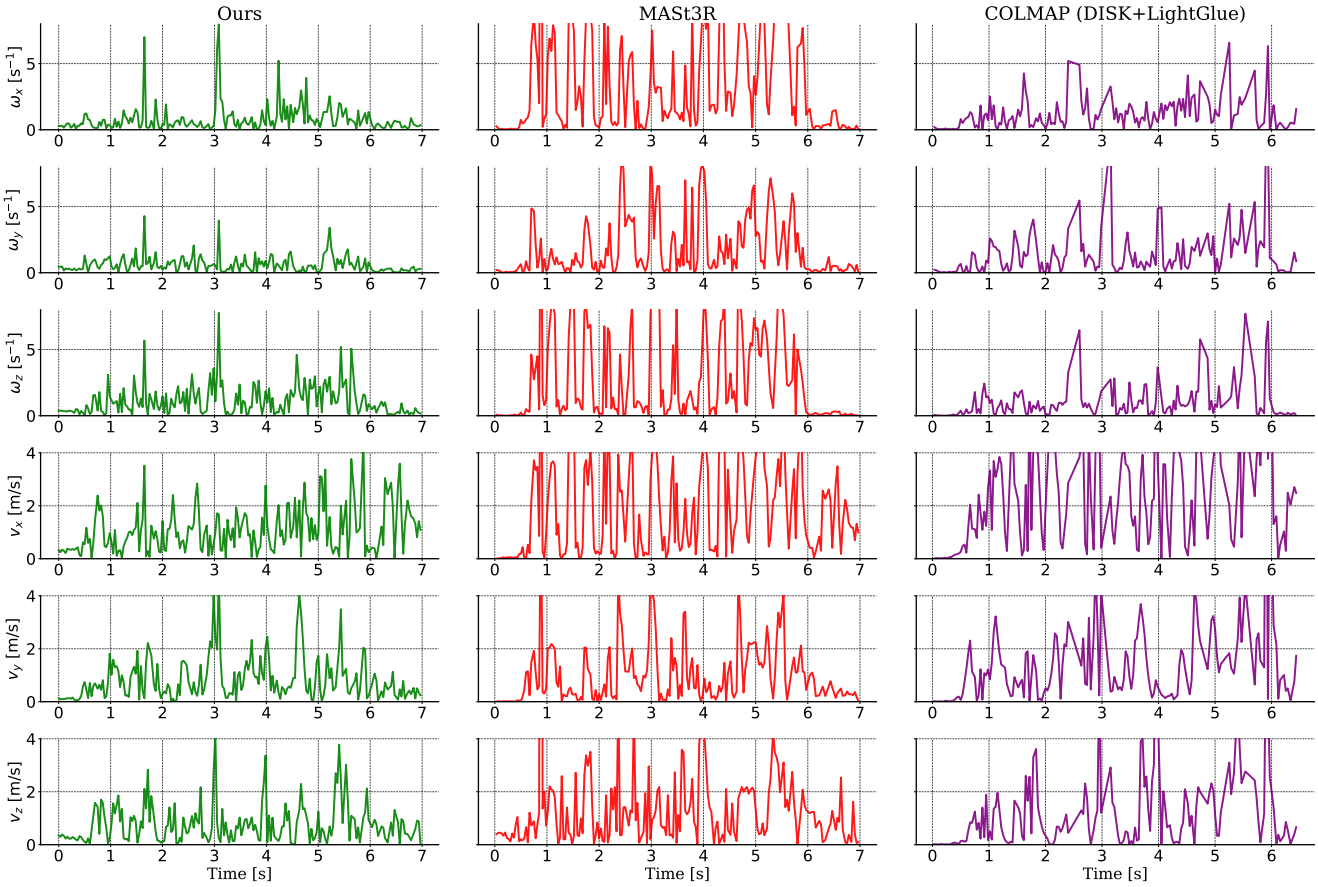

Figure 5. Absolute error for the billiards sequence.

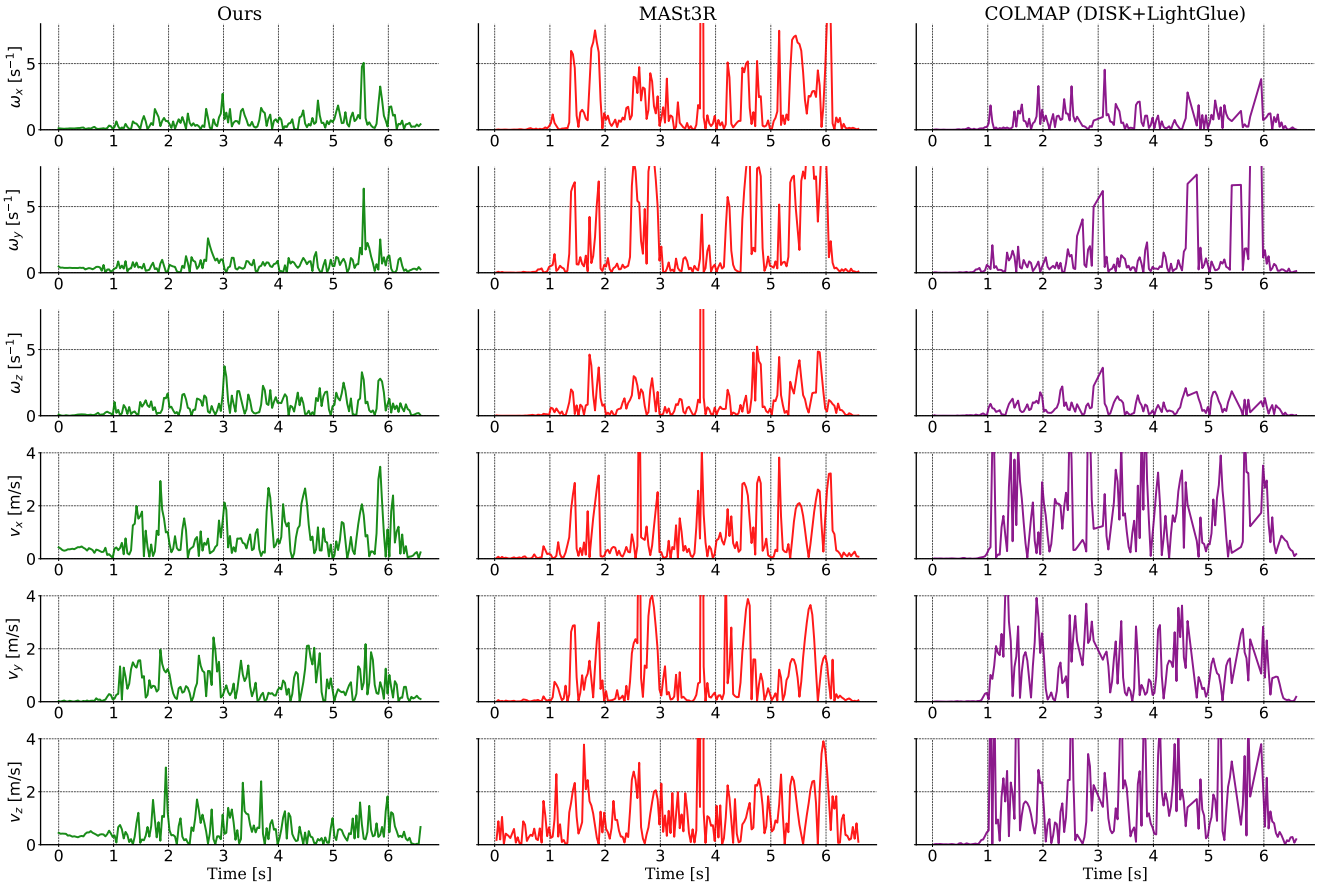

Figure 6. Absolute error for the commonroom sequence.

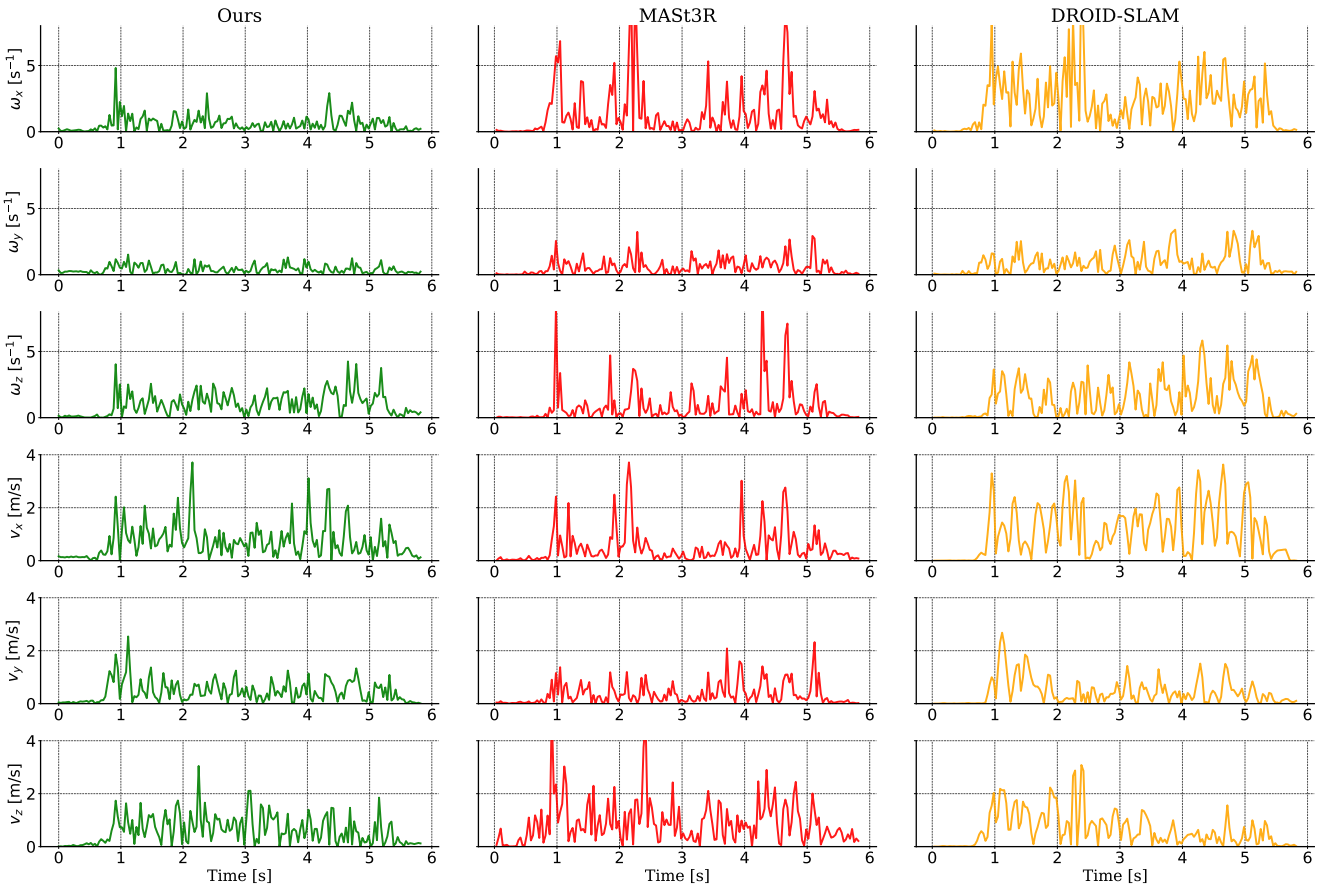

Figure 7. Absolute error for the dining sequence.

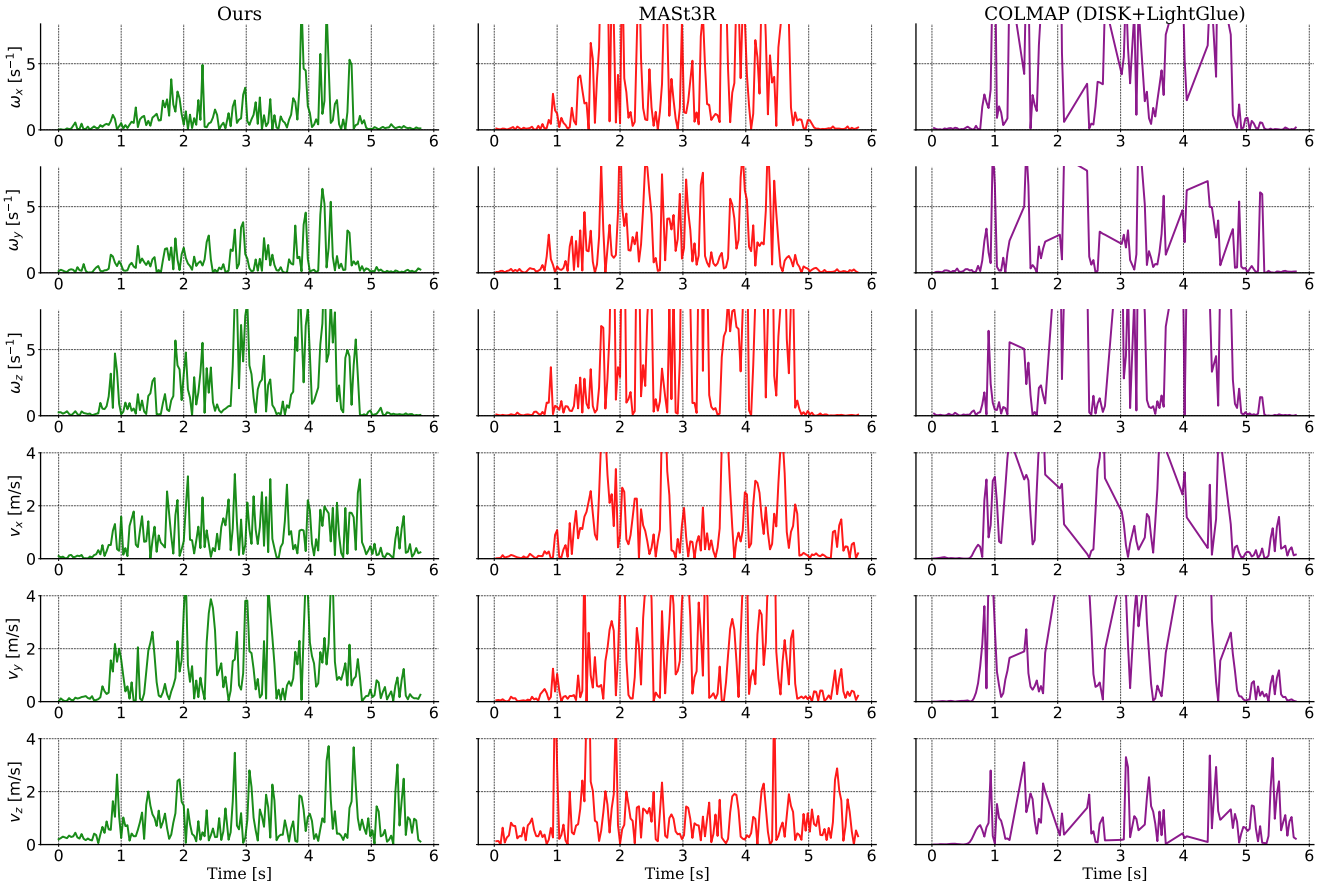

Figure 8. Absolute error for the office sequence.