Supplementary Material: Visual-Inertial Mapping with Non-Linear Factor Recovery

Vladyslav Usenko[1] Nikolaus Demmel[1] David Schubert[1] Jörg Stückler[2] Daniel Cremers[1]
[1]Technical University of Munich [2]MPI for Intelligent Systems Tübingen

I. Details on KLT Tracking

Here we provide a more detailed description of the patch-based sparse optical flow used in our system. First, we detect a sparse set of keypoints in the frame using the FAST [22] corner detector. To track the motion of these points over a series of consecutive frames we use sparse optical flow based on KLT [14]. To achieve fast, accurate and robust tracking we combine the inverse-compositional approach as described in [1] with the locally scaled sum of squared differences (LSSD) metric — a brightness-invariant norm for patch comparison as defined in [21].

We formulate the patch tracking problem as estimating the transform $\mathbf{T} \in \mathrm{SE}(2)$ between two corresponding patches in two consecutive frames that minimizes the appearance differences between the patches. At each optimization iteration we want to find an update $\boldsymbol{\xi}$ that minimizes the sum of squared residuals

$$r_i(\boldsymbol{\xi}) = \frac{I_{t+1}(\mathbf{T}\mathbf{x}_i)}{\overline{I_{t+1}}} - \frac{I_t(\mathrm{Exp}(-\boldsymbol{\xi})\mathbf{x}_i)}{\overline{I_t}} \quad \forall \mathbf{x}_i \in \Omega. \tag{27}$$

Here, contrary to the rest of the paper, Exp maps from $\mathbb{R}^3$ to $\mathrm{SE}(2)$. $I_t(\mathbf{x})$ and $I_{t+1}(\mathbf{x})$ are the image intensities of images t and $t+1$ at pixel location $\mathbf{x}$. The set of image coordinates that defines the patch is denoted Ω, and the mean intensity of the patch in image t and $t+1$ is $\overline{I_t}$ and $\overline{I_{t+1}}$, respectively.

Here we use an LSSD as a dissimilarity measure between image patches. This norm is invariant to scaling, so if we assume a linear response function of our camera, we get a tracker that is invariant to changes in the exposure time. Several authors use zero-normalized cross-correlation (ZNCC) for illumination invariant optical flow [17], [25], but as shown in [21] ZNCC is computationally more expensive compared to LSSD.

If we treat all residuals for one patch defined in (27) as residual vector function $\mathbf{r}(\boldsymbol{\xi})$, we can use the Gauss-Newton framework defined in (2) - (4) to iteratively minimize the error. The only difference here is that instead of differentiating with respect to an increment to $\mathbf{T}$, we differentiate with respect to the inverse increment applied to the image coordinates in the template frame t,but the update $\boldsymbol{\xi}^*$ (calculated according to (4)) is still applied to $\mathbf{T}$ as

$$\mathbf{T}_{i+1} = \mathbf{T}_i \mathrm{Exp}(\boldsymbol{\xi}^*)\,. \tag{28}$$

As described in [1], with this inverse-compositional approach the Jacobian $\mathbf{J}$ does not depend on the current state of $\mathbf{T}$, so $(\mathbf{J}^\top\mathbf{W}\mathbf{J})^{-1}\mathbf{J}^\top\mathbf{W}$ can be pre-computed. This way we only need to recompute the residual vector at every iteration and compute the update. To achieve robustness to large displacements in the image we use a pyramidal approach, where the patch is first tracked on the coarsest level and then on increasingly finer levels.

To filter outliers, instead of an absolute threshold on the error we track the patches from the current frame to the target frame and back. Points that did not return to the initial location after the second tracking are considered as outliers and discarded.

II. Details on IMU Preintegration

To deal with the high frequency of IMU measurements we preintegrate several consecutive IMU measurements into a pseudo-measurement. When adding an IMU factor between frame i and frame j, we compute pseudo-measurement $\Delta\mathbf{s} = (\Delta\mathbf{R}, \Delta\mathbf{v}, \Delta\mathbf{p})$ similar to [8]. For this, we compute bias-corrected accelerations $\mathbf{a}_t = \mathbf{a}_t^{\mathrm{raw}} - \bar{\mathbf{b}}_i^{\mathrm{a}}$ and rotational velocities $\boldsymbol{\omega}_t = \boldsymbol{\omega}_t^{\mathrm{raw}} - \bar{\mathbf{b}}_i^{\mathrm{g}}$ using the raw accelerometer $\mathbf{a}_t^{\mathrm{raw}}$ and gyroscope $\boldsymbol{\omega}_t^{\mathrm{raw}}$ measurements. We fix the corresponding biases $\bar{\mathbf{b}}_i^{\mathrm{a}}$ and $\bar{\mathbf{b}}_i^{\mathrm{g}}$ for the entire preintegration time and use linear approximation to account for changes in these variables.

For the timestamp t_i of frame i, we assign the initial state delta $\Delta\mathbf{s}_{t_i} = (\mathbf{I}, \mathbf{0}, \mathbf{0})$. Then, for each IMU timestamp t satisfying $t_i < t \le t_j$ the following updates are calculated.

$$\Delta\mathbf{R}_{t+1} = \Delta\mathbf{R}_t \mathrm{Exp}(\boldsymbol{\omega}_{t+1}\Delta t)\,, \tag{29}$$

$$\Delta\mathbf{v}_{t+1} = \Delta\mathbf{v}_t + \Delta\mathbf{R}_t \mathbf{a}_{t+1}\Delta t\,, \tag{30}$$

$$\Delta\mathbf{p}_{t+1} = \Delta\mathbf{p}_t + \Delta\mathbf{v}_t\Delta t\,. \tag{31}$$

This defines $\Delta\mathbf{s}_{t+1}$ as a function of $\Delta\mathbf{s}_t$, $\mathbf{a}_{t+1}$, and $\boldsymbol{\omega}_{t+1}$,

$$\Delta\mathbf{s}_{t+1} = f(\Delta\mathbf{s}_t, \mathbf{a}_{t+1}, \boldsymbol{\omega}_{t+1})\,, \tag{32}$$

with corresponding Jacobian $\mathbf{J}_f = [\mathbf{J}_f^{\mathrm{s}}, \mathbf{J}_f^{\mathrm{a}}, \mathbf{J}_f^{\mathrm{g}}]$. Furthermore, all previous iterations of f up to $t+1$ define $\Delta\mathbf{s}_{t+1}$ as a function of the biases,

$$\Delta\mathbf{s}_{t+1} = g_{t+1}(\mathbf{b}_i^{\mathrm{a}}, \mathbf{b}_i^{\mathrm{g}})\,. \tag{33}$$

Starting with zero-initialization, the corresponding Jacobian $\mathbf{J}_{g_{t+1}} = [\mathbf{J}_{g_{t+1}}^{\mathrm{a}}, \mathbf{J}_{g_{t+1}}^{\mathrm{g}}]$ can be computed recursively using $\mathbf{J}_f$,

$$\mathbf{J}_{g_{t+1}}^{\mathrm{a}} = \mathbf{J}_f^{\mathrm{s}}\mathbf{J}_{g_t}^{\mathrm{a}} - \mathbf{J}_f^{\mathrm{a}}\,, \tag{34}$$

$$\mathbf{J}_{g_{t+1}}^{\mathrm{g}} = \mathbf{J}_f^{\mathrm{s}}\mathbf{J}_{g_t}^{\mathrm{g}} - \mathbf{J}_f^{\mathrm{g}}\,, \tag{35}$$

which results from the chain rule. Eventually, the Jacobians of g_{t_j} are denoted $\mathbf{J}^{\mathrm{g}}$ and $\mathbf{J}^{\mathrm{a}}$. Small changes in biases can be represented as increments to the linearization point $\mathbf{b}_i^{\mathrm{a}} = \bar{\mathbf{b}}_i^{\mathrm{a}} + \boldsymbol{\epsilon}^{\mathrm{a}}$ and $\mathbf{b}_i^{\mathrm{g}} = \bar{\mathbf{b}}_i^{\mathrm{g}} + \boldsymbol{\epsilon}^{\mathrm{g}}$. Then, $\Delta\mathbf{s}$ is approximated as

$$\Delta\tilde{\mathbf{s}}(\mathbf{b}_i^{\mathrm{a}}, \mathbf{b}_i^{\mathrm{g}}) = \Delta\mathbf{s}(\bar{\mathbf{b}}_i^{\mathrm{a}}, \bar{\mathbf{b}}_i^{\mathrm{g}}) \oplus (\mathbf{J}^{\mathrm{a}}\boldsymbol{\epsilon}^{\mathrm{a}} + \mathbf{J}^{\mathrm{g}}\boldsymbol{\epsilon}^{\mathrm{g}}) \,, \tag{36}$$

with components $\Delta\tilde{\mathbf{s}} = (\Delta\tilde{\mathbf{R}}, \Delta\tilde{\mathbf{v}}, \Delta\tilde{\mathbf{p}})$. The residuals are then calculated as

$$\mathbf{r}_{\Delta\mathbf{R}} = \mathrm{Log}\left(\Delta\tilde{\mathbf{R}}\mathbf{R}_j^\top \mathbf{R}_i\right) \,, \tag{37}$$

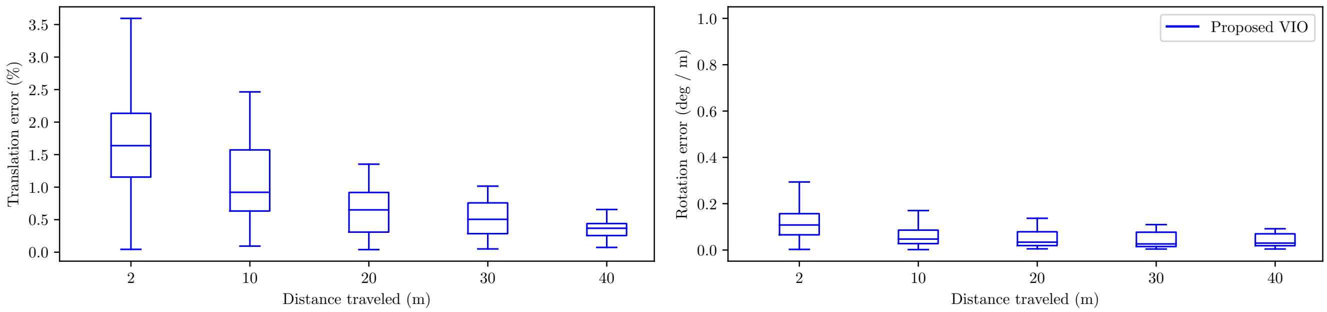
$$\mathbf{r}_{\Delta\mathbf{v}} = \mathbf{R}_i^\top(\mathbf{v}_j - \mathbf{v}_i - \mathbf{g}\Delta t) - \Delta\tilde{\mathbf{v}} \,, \tag{38}$$

$$\mathbf{r}_{\Delta\mathbf{p}} = \mathbf{R}_i^\top(\mathbf{p}_j - \mathbf{p}_i - \frac{1}{2}\mathbf{g}\Delta t^2) - \Delta\tilde{\mathbf{p}} \,, \tag{39}$$

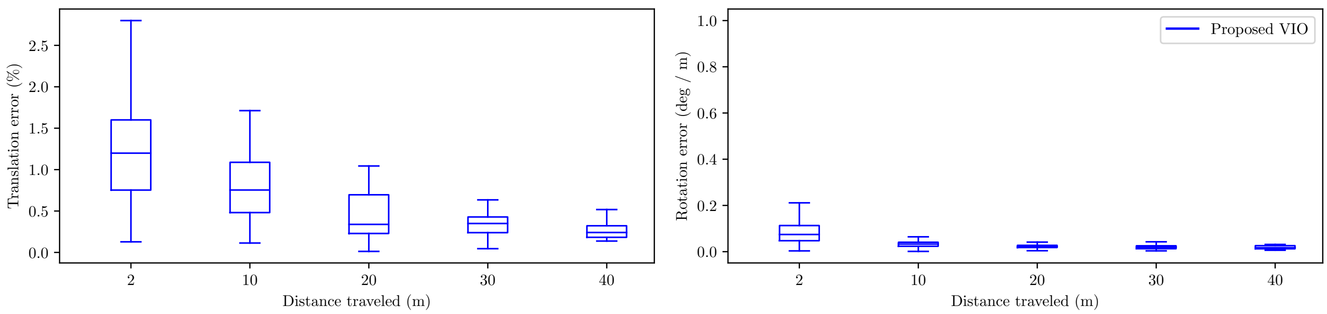
where $\mathbf{g}$ is the gravity vector and $\mathbf{R}$ and $\mathbf{p}$ denote the rotation and translation components of $\mathbf{T}_{\mathrm{WI}}$, respectively. These residuals have to be weighted with an appropriate covariance matrix, which can be also calculated recursively. Starting from $\boldsymbol{\Sigma}_{t_i} = \mathbf{0}$, updates are calculated as

$$\boldsymbol{\Sigma}_{t+1} = \mathbf{J}_f^{\mathrm{s}}\boldsymbol{\Sigma}_t{\mathbf{J}_f^{\mathrm{s}}}^\top + \mathbf{J}_f^{\mathrm{a}}\boldsymbol{\Sigma}^{\mathrm{a}}{\mathbf{J}_f^{\mathrm{a}}}^\top + \mathbf{J}_f^{\mathrm{g}}\boldsymbol{\Sigma}^{\mathrm{g}}{\mathbf{J}_f^{\mathrm{g}}}^\top \,, \tag{40}$$

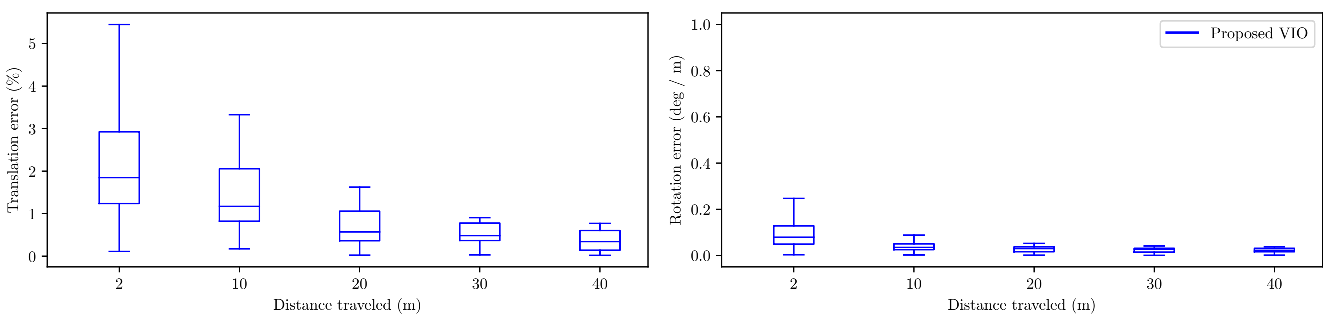
with diagonal matrices $\boldsymbol{\Sigma}^{\mathrm{a}}$ and $\boldsymbol{\Sigma}^{\mathrm{g}}$ that contain the hardware-specific IMU noise parameters for accelerometer and gyroscope. For more detailed information about the underlying physical model of the IMU and preintegration theory we refer the reader to [8].



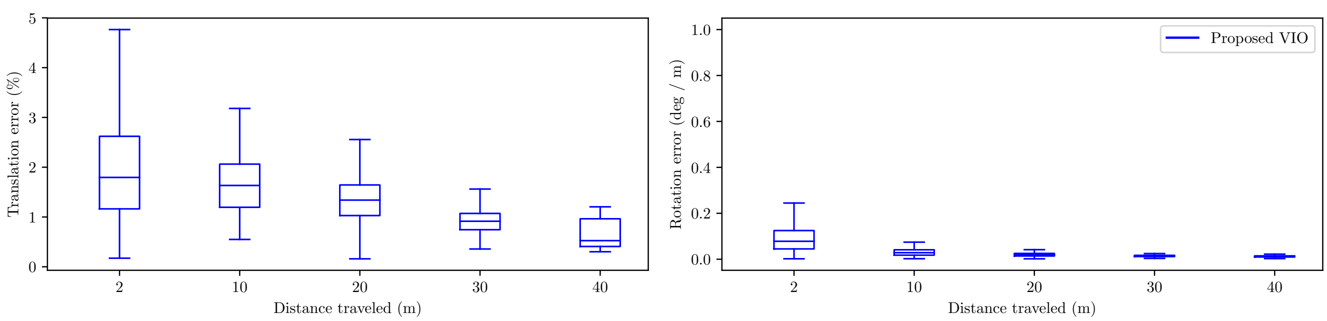
(a) MH_01



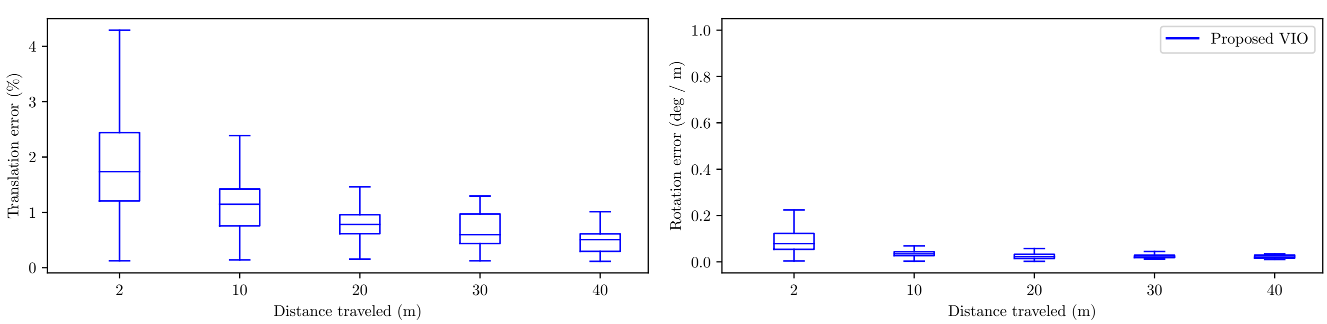
(b) MH_02



(c) MH_03

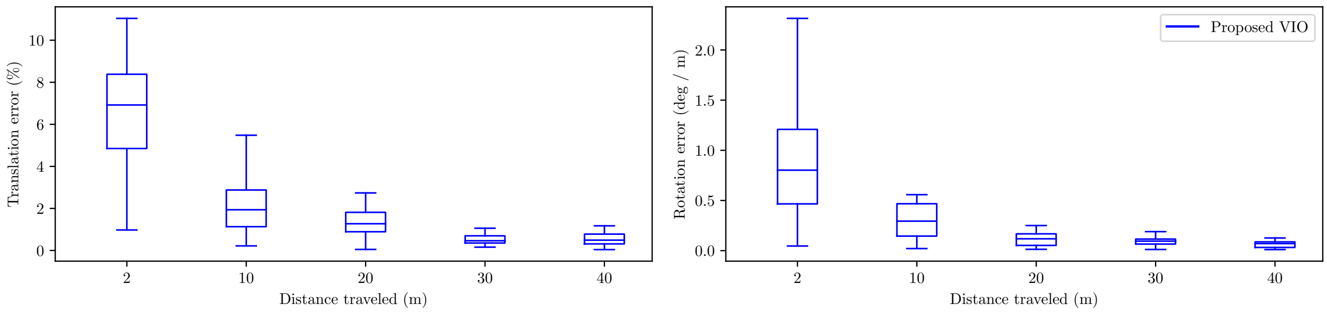


(d) MH_04

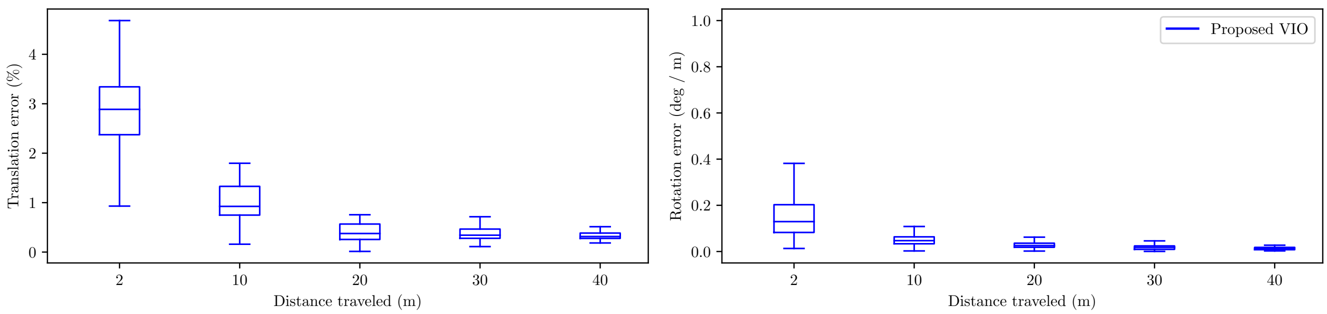


(e) MH_05

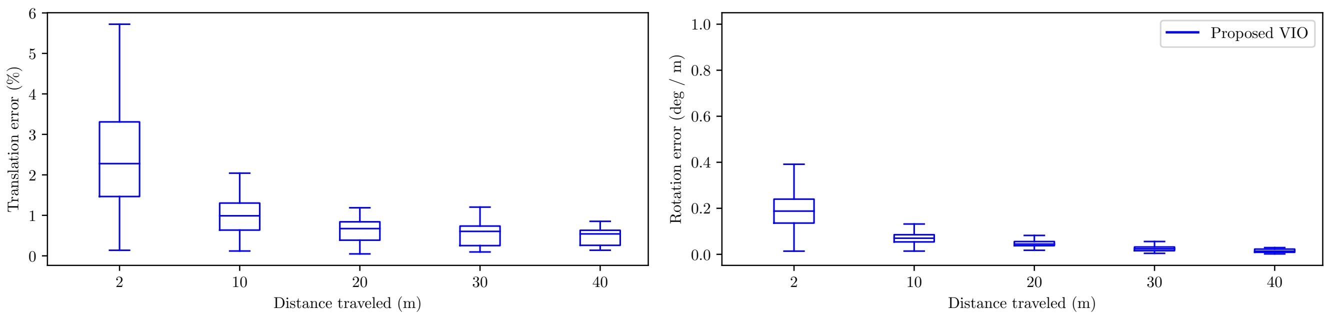
Fig. 1: Translation and rotation RPE on the Machnie Hall sequences of the Euroc dataset.



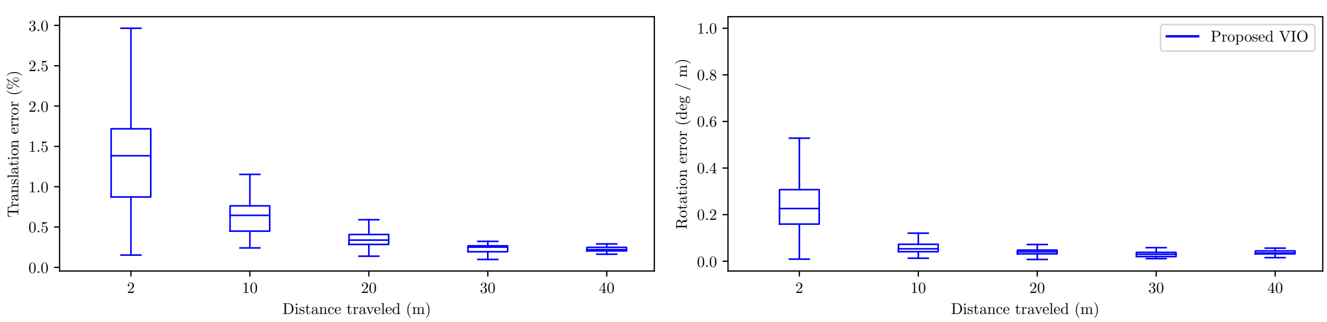
(a) V1_01



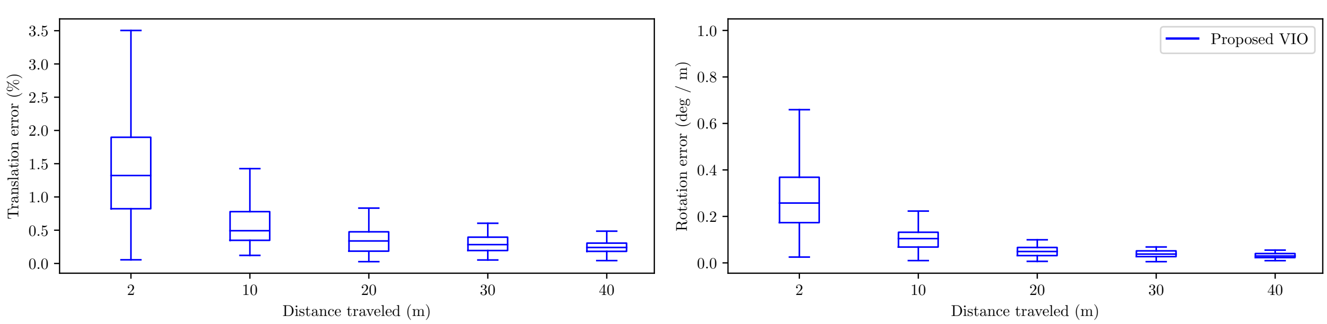
(b) V1_02



(c) V1_03



(d) V2_01



(e) V2_02

Fig. 2: Translation and rotation RPE on the Vicon Room sequences of the Euroc dataset.