Achieving Sub-Pixel Platform Accuracy with Pan-Tilt-Zoom Cameras in Uncertain Times

Martin Vonheim Larsen, Kim Mathiassen

***Abstract*—In this paper, we present a novel method for self-calibrating a pan-tilt-zoom (PTZ) camera system model, specifically suited for long-range multi-target tracking with maneuvering low-cost PTZ cameras. Traditionally, such camera systems cannot provide accurate mappings from pixels to directions in the platform frame due to imprecise pan/tilt measurements or lacking synchronization between the pan/tilt unit and the video stream. Using a direction-only bundle adjustment (BA) incorporating pan/tilt measurements, we calibrate camera intrinsics, rolling shutter (RS) characteristics and pan/tilt mechanics, and obtain clock synchronization between the video stream and pan/tilt telemetry. We call the resulting method PTCEE (pan/tilt camera extrinsic and intrinsic estimation). In a thorough simulation study, we show that the proposed estimation scheme identifies model parameters with sub-pixel precision across a wide range of camera setups. Leveraging the map of landmarks from the BA, we propose a method for estimating camera orientation in real-time, and demonstrate pixel-level mapping precision on real-world data. Through the proposed calibration and orientation schemes, PTCEE enables high-precision target tracking during camera maneuvers in many low-cost systems, which was previously reserved for high-end systems with specialised hardware. Code is available at github.com/ffi-no/Paper-ptz-subpix-accuracy.**

I. Introduction

PAN-TILT-ZOOM cameras offer the versatile combination of high-fidelity long-range data and wide-area coverage at a low cost and with tractable processing requirements. To make full use of pan-tilt-zoom (PTZ) cameras at long ranges in precision-demanding applications, accurate calibration is essential [1]–[3]. Ideally, a calibrated system model should provide accurate mappings from each pixel in every image to viewing directions in the platform coordinate frame. When tracking closely moving targets that disappear and reappear into view, the *relative precision* of this model directly affects the system's ability to discern targets and make good predictions. Similarly, the sensitivity of a scanning PTZ camera used for change detection relies on the system model being self-consistent with pixel-level precision as the camera moves. In a multi-sensor tracking setup, on the other hand, the *absolute accuracy* of the model is critical for PTZ observations to be usable alongside observations from other sensors.

Self-calibration of rotational-only cameras observing a distant scene is a well-understood topic with many robust solutions [4]–[9]. A fundamental limitation of rotation-only camera self-calibration is that the focal length is increasingly difficult to recover as the field-of-view (FOV) becomes narrower [10]. This happens because, at narrow FOV, changes in focal length become indistinguishable from changes in the angular scale of the observed camera motion. Therefore, to calibrate rotating cameras operating at narrow FOV some external information that provides angular scale is needed. Fortunately, PTZ cameras provide built-in means to help lock down the angular scale of motion; the pan/tilt measurements.

Exploiting pan/tilt telemetry for calibration of low-cost off-the-shelf PTZ cameras is challenging for multiple reasons. Most such cameras lack proper synchronization between pan/tilt measurements and image capture [1], leading to significantly degraded pointing accuracy during camera maneuvers if not corrected [11]. It is not uncommon that low-cost cameras provide pan/tilt measurements at high resolution, but we generally cannot expect the pan/tilt actuators to be perfectly perpendicular to each other and the horizontal axis of the camera [9], nor that the pan/tilt sensors are properly calibrated [8]. In addition, these cameras generally use rolling shutter (RS) image sensors. Even during moderate maneuvers, the RS effects in a narrow FOV camera can be quite severe. To succeed in calibrating such low-cost PTZ cameras based on visual observations and pan/tilt telemetry, we argue that these issues must be handled.

In this paper, we propose PTCEE (pan/tilt camera extrinsic and intrinsic estimation): A novel method for calibrating a PTZ system model specifically tailored to handle narrow FOV operations with maneuvering low-cost PTZ cameras. Figure 1 shows an overview of the proposed method. We assume all visible parts of the scene are distant, and therefore that any non-rotational motion is negligible and the pan and tilt rotations occur in the optical center. Our proposed model includes the following parameters:

- Camera focal length f, quadratic radial distortion k and rolling shutter line duration ℓ.
- The pan and tilt rotational axes and their relation to the camera horizontal axis. We assume rotations occur around the optical center.
- Scaling parameters on the pan and tilt measurements.
- The time offset between the clocks of the image sensor and the pan/tilt unit.

We calibrate the model through a full bundle adjustment (BA) restricted to two-axis camera rotation and directional landmarks, implemented using a factor graph. To obtain the landmark observations for the BA without the need for external calibration targets, we provide a semi-direct frontend inspired by

Manuscript received July 4, 2024; revised November 1, 2024. This work was funded in part by the Autonomous Systems project (P1505) at the Norwegian Defence Research Establishment, and in part by Universitetssenteret på Kjeller (UNIK). *Corresponding author: M. Larsen*

M. Larsen and K. Mathiassen are with the Defence Systems Division at the Norwegian Defence Research Establishment, Kjeller NO-2007 Norway, and the Department of Technology Systems at the University of Oslo, Kjeller NO-2007, Norway e-mail: martin-vonheim.larsen@ffi.no; kim.mathiassen@ffi.no

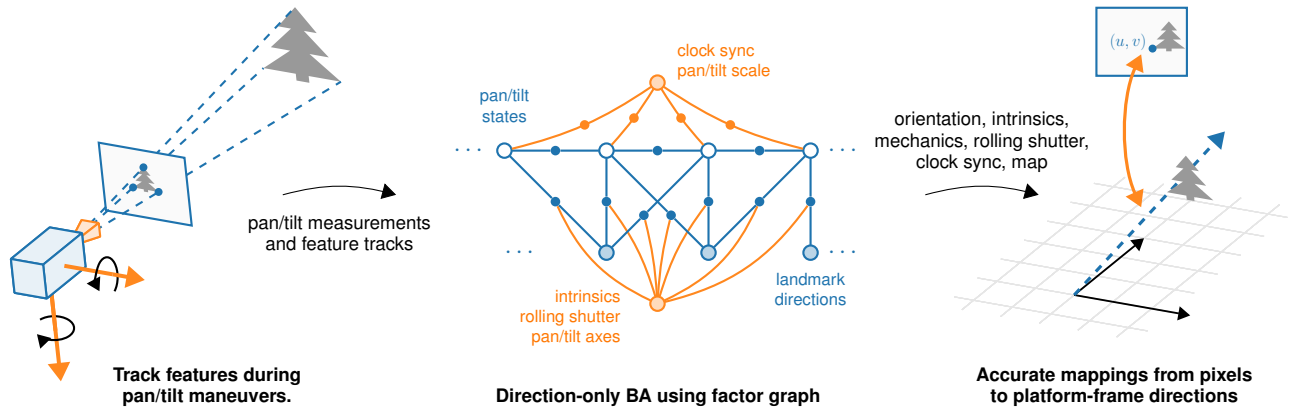


Fig. 1. Overview of our method. From feature tracks alongside pan/tilt measurements during pan/tilt maneuvers, we infer camera intrinsics, rolling shutter parameter, pan/tilt axes and clock synchronization through direction-only BA implemented using a factor graph. The resulting parameters and map of landmarks can be used to produce mappings from pixel space to the platform frame with sub-pixel accuracy in real-time.

SVO [12], modified to exploit our two-axis rotation-only setting while also handling RS images. We can use the resulting calibration together with the now-synchronized pan/tilt measurements to provide pixel-to-platform-frame-direction mappings with minimal latency. The BA also produces a map of landmarks, which we demonstrate can be used to orient incoming images with sub-pixel precision in real-time.

In the presented version, the proposed model assumes that the focal length is kept constant, which may seem restrictive for a PTZ camera. We argue that this assumption is still viable for many long-range narrow FOV applications, seeing as we can recalibrate within seconds of changing the zoom level, without using external calibration targets. The factor graph formulation of the BA can easily be extended to model camera intrinsics as a parametric function of the zoom level, but this is beyond the scope of this paper.

Some of the ideas and preliminary results behind this paper were presented in [11]. The contributions presented in this paper are as follows:

- A novel PTZ camera model for calibrating low-cost narrow FOV cameras. Previous works lack both the inclusion of pan/tilt measurements, necessary for narrow FOV, as well as the clock synchronization and RS compensation, necessary for low-cost cameras. (Section IV)
- A novel two-axis rotation-only BA capable of calibrating the proposed model. (Section V-B)
- A novel SVO-style rotation-only frontend which exploits the two-axis nature of PTZ cameras, capable of bootstrapping observations for the proposed BA. (Section V-A)
- Two different methods providing pixel-to-direction mappings for new images in real-time. (Section V-C)
- Extensive simulations and a real-world test analyzing the performance of the proposed method. (Section VI)

II. Related Work

In this section we summarize related work for calibrating PTZ cameras. Due to the varying zoom capability of PTZ cameras, traditional methods for intrinsic calibration [13], [14] that require images of a moving external calibration target every time the lens has been adjusted, are impractical. This is especially the case for long range operation at narrow FOV, where the calibration target must be placed at distance to stay in focus, and therefore also needs to be physically large. Consequently, our primary focus is on methods that leverage the scene at hand, instead of requiring specific calibration targets.

A. Self-calibration of Rotating Cameras from Images

Camera self-calibration using images alone dates back to the early 1990s, with [15] and Hartley who presented the first methods for rotating cameras in [4], [16]. Hartley's method extracts the focal length and principal point from the dual image of the absolute conic (DIAC), which can be linearly estimated from two or more homographies between pairs of images. A few years later Triggs [5] proposed to instead leverage the dual absolute quadric (DAQ), which allows for far greater flexibility in formulating constraints on the camera intrinsics and motion. These linear algebraic estimates are typically followed by a non-linear optimization of geometrical error [17], which also enables estimating lens distortion.

B. Calibration of PTZ Cameras

Since the fundamental trait of a PTZ camera is the ability to change pan, tilt and zoom level continuously, *PTZ camera calibration* typically encompasses both intrinsic and (device-internal) extrinsic calibration. PTZ camera models should either handle varying intrinsic parameters directly or at least allow for quick re-calibration when the zoom level is changed. The extrinsic calibration in a PTZ camera model essentially amounts to describing the pose between the camera and the camera base.

Building on the ideas from self-calibration to provide intrinsic calibration, Agapito et al. [6] extend Hartley's method to handle varying zoom level, but omit radial distortion. To provide camera intrinsics *with* radial distortion across the full zoom range, [7] proposes a method that uses an extended multi-resolution panorama across a discrete zoom sequence. Later, [8], [18] improve this approach by replacing the extended panorama with a discrete map of landmarks, effectively

implementing a rotation-only simultaneous localization and mapping (SLAM) with directional landmarks. These methods provides both calibrated intrinsics and camera orientation in real-time, and performs online refinement of the underlying model.

While many PTZ calibration methods use full 3-DoF orientations to represent camera mechanics, a special property of PTZ cameras is that their motion is restricted to rotations about two fixed axes. Davis et al. [9] exploit this in a model with arbitrary pan and tilt axes, which they calibrate using a tracked LED calibration target. More recently, [19] proposes a linear method for cameras that *do not* rotate about the optical center, which is suitable for bullet-type PTZ cameras operated at wide FOV.

Most of the modern PTZ calibration methods have in common that first they solve an algebraic optimization problem, which is then used to initialize a BA over geometric errors. In recent work, [20] proposes to use a dual Siamese neural network to predict focal lengths, quadratic radial distortions and relative rotation between two images. This approach has the major benefit of bypassing the often elaborate algebraic optimization problems for initialization.

C. Self-calibration of Rotating Cameras with Narrow FOV

When self-calibrating from images alone under rotation-only motion, Agapito et al. demonstrate theoretically that the observability of the focal length vanishes as the FOV is reduced [10]. In practice, exactly *when* this becomes a problem depends on the quality of the observations: The number of keypoints used and their angular precision. Although some claim to study narrow FOV, to our knowledge, no existing work verifies self-calibration accuracy at HFOVs smaller than $15^\circ - 20^\circ$ [1], [7], [8], [18], [21]. Meanwhile, modern consumer PTZ cameras[1] have lenses capable of HFOVs below 2°, at which point it is no longer clear that image-only self-calibration will work.

Methods relying on information from images alone can mitigate this issue by combining observations at wide FOV (where focal length is well-observed) with the narrow FOV observations (where focal length is weakly/not observable). Lisanti et al. [1] demonstrate qualitatively improved calibration performance when combining observations from multiple zoom levels in a joint BA. However, the limiting factor is finding good observations of landmarks across a wide range of FOVs, resulting in weak linking between the wide and narrow FOV-levels in the BA.

An alternative for overcoming the narrow FOV challenges is to exploit prior knowledge about the scene geometry. Sports broadcasting serves as a good example, where the known geometry of the playing field can be exploited to provide camera-to-field mappings [21]–[23]. Another approach is of course to introduce translation to the camera motion, or combine multiple cameras with non-negligible baseline [24].

[1]For instance, the Axis Q6318 PTZ camera has a nominal minimal HFOV of 2.4° at 4K resolution. A common usecase is to crop out the center 1920×1080px, resulting in an effective HFOV of 1.2°.

Using pan/tilt measurements as an external reference is surprisingly rare in the literature, probably due to a lack of synchronization between the measurements and the video stream [1], [25]. Assuming synchronized measurements, [26] leverages known rotations for intrinsic and extrinsic calibration in an incremental method. Wu et al. [8] incorporate pan/tilt measurements in the final stages of extrinsic calibration, but not during intrinsic calibration, and do not handle unsynchronized measurements. Frahm et al. [27] achieve improved intrinsic calibration using a method that aligns image observations with unsynchronized 3-DOF orientation measurements. Furgale et al. [28] propose a method for unified temporal and spatial calibration of a solid body system consisting of multiple cameras and an inertial measurement unit (IMU).

D. Narrow FOV Calibration with rolling shutter PTZ Cameras

In addition to unsynchronized pan/tilt measurements and video stream, a key challenge when working with consumer PTZ cameras is that they use RS. Accounting for RS is in principle no different for narrow FOV PTZ cameras than for any other camera.

A popular family of models for RS cameras assumes that the pixel rows are exposed linearly in time, and that the camera undergoes linear (in various senses) motion during exposure [29]. Some methods rectify the RS, but require parameters about the rolling readout to be known [30], [31]. Other such examples include [32], which uses a rotation-only motion model, and [33], which optimizes 6-DoF motion using BA. In [34], the authors propose a minimal formulation for estimating a linear RS model together with typical intrinsic camera parameters, under 6-DoF camera motion. [35] presents a method which only assumes smooth camera motion but requires a pre-calibrated camera. Diverging from the linear RS model, [36] estimates a mixture of homographies without needing pre-calibrated intrinsics.

All these existing methods tightly couple the focal length (either known or calibrated) to the RS parameter estimation. This coupling is problematic in the case of narrow FOV rotation-only camera calibration: We cannot estimate the focal length from images without incorporating pan/tilt measurements, and the pan/tilt measurements cannot be properly incorporated without accounting for RS, which is coupled to the focal length. Therefore, we either need an RS method independent of focal length, or must jointly include it in the PTZ calibration.

III. Background

In this section, we briefly revisit the theoretical foundations necessary for formulating our optimization setup in Sections IV and V.

A. MAP Estimation using Factor Graphs

Let $X = \{x_j\}$ denote a set of unknown state variables and $Z = \{z_k\}$ be a set of measurements involving X. The maximum a posteriori (MAP) estimate of X is given as the $\widehat{X}$ that maximizes the posterior distribution

$$\widehat{X} = \arg\max_X p(X \mid Z) = \arg\max_X p(Z \mid X)p(X). \quad (1)$$

Assuming Gaussian and independent measurements and priors, Eq. (1) simplifies to a non-linear least squares:

$$\widehat{X} = \arg\max_X \prod_i \ell_i(X_i) \tag{2}$$

$$= \arg\min_X \sum_i \|\mathbf{e}_i(X_i)\|^2_{\mathbf{\Sigma}_i}, \tag{3}$$

where i runs over the set of measurements and priors, and $\|\cdot\|_{\mathbf{\Sigma}}$ denotes the Mahalanobis distance. For each term i, X_i is the set of variables involved, ℓ_i is the measurement/prior likelihood over X_i, $\mathbf{e}_i$ is the mean error, and $\mathbf{\Sigma}_i$ is the corresponding covariance.

To perform the MAP estimation, we formulate Eq. (3) as a *factor graph*. The factor graph is a bipartite graph consisting of variable nodes for each x_j and factor nodes for each $\|\mathbf{e}_i\|^2_{\Sigma_i}$ term, with edges connecting each factor node to the variable nodes it depends on. We solve Eq. (3) using Levenberg-Marquardt (LM), which requires iterative linearization of the inner sum of Eq. (3). This linearization involves computing the Jacobian of each $\mathbf{e}_i$-factor with respect to the variables involved in that factor. In the following, we employ the notation $\mathbf{J}^{\mathbf{e}_i}_{x_j}$ to denote the Jacobian of $\mathbf{e}_i$ with respect to the variable x_j. In Section V-B, we define $\mathbf{e}_i$-factors for both priors, pan/tilt measurements and landmarks observations when constructing the factor graph for the PTZ calibration problem.

B. Manifold Representations of Orientation and Direction

In the following, we need to describe camera orientations and landmark directions, which we represent as rotation matrices $\mathbf{R} \in SO(3)$ and direction vectors $\mathbf{d} \in S^2$ respectively. Here, $\mathbf{R}$ and $\mathbf{d}$ are not defined on vector spaces but live on smooth manifolds in higher dimensional spaces. In order to apply the estimation framework above on these manifolds, we take the common approach of working in the tangent space to the manifold at the current estimate, which locally behaves as a Euclidean space.

For orientations we refer to [37] for a thorough introduction to Lie theory applied to rotations $\mathbf{R} \in SO(3)$ and poses $\mathbf{T} \in SE(3)$. Similar to [37], we use the capitalized exponential notation for angle-axis rotations via Rodrigues' formula:

$$\mathbf{R} = \mathrm{Exp}(\theta \cdot \mathbf{u}). \tag{4}$$

Using the Exp operator, we also define the $\oplus$ operator which lets us perturb a rotation $\mathbf{R}$ by a tangent vector $\boldsymbol{\xi}$:

$$\widehat{\mathbf{R}} = \mathbf{R} \oplus \boldsymbol{\xi} \triangleq \mathbf{R}\,\mathrm{Exp}(\boldsymbol{\xi}). \tag{5}$$

To describe the relative orientation between two same-origin frames $\mathcal{F}_a$ and $\mathcal{F}_b$, we use the notation $\mathbf{R}_{ab}$. The composition of two rotations $\mathbf{R}_{ab}$ and $\mathbf{R}_{bc}$ is given by the matrix product:

$$\mathbf{R}_{ac} = \mathbf{R}_{ab}\mathbf{R}_{bc}. \tag{6}$$

We also use the notation $\mathbf{v}^b$ to denote that the vector $\mathbf{v}^b \in \mathbb{R}^3$ is represented in $\mathcal{F}_b$. The relationship between a vector $\mathbf{v}^b$ represented in $\mathcal{F}_b$ and the same vector in $\mathcal{F}_a$ is given by

$$\mathbf{v}^a = \mathbf{R}_{ab}\mathbf{v}^b. \tag{7}$$

Since distance to landmarks is not observable in our orientation-only setup, we represent landmark locations as direction vectors $\mathbf{d} \in S^2 = \{\mathbf{p} \in \mathbb{R}^3 \,|\, \|\mathbf{p}\| = 1\}$. For a given direction $\mathbf{d}$, we consider the tangent space of S^2 at $\mathbf{d}$,

$$T_{\mathbf{d}}S^2 \triangleq \left\{\hat{\boldsymbol{\xi}} \in \mathbb{R}^3 \,|\, \mathbf{d}^\top \hat{\boldsymbol{\xi}} = 0\right\}. \tag{8}$$

As suggested by Dellaert et al. [38, p.94], we choose

$$\mathbf{B}_{\mathbf{d}} = \begin{bmatrix} \frac{\mathbf{d}\times\mathbf{h}}{\|\mathbf{d}\times\mathbf{h}\|} & \frac{\mathbf{d}\times(\mathbf{d}\times\mathbf{h})}{\|\mathbf{d}\times(\mathbf{d}\times\mathbf{h})\|} \end{bmatrix} \tag{9}$$

as a basis for $T_{\mathbf{d}}S^2$, where $\mathbf{h} \in \{\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z\}$ is the standard basis vector that minimizes $|\mathbf{h}^\top \mathbf{d}|$. Here, $\mathbf{B}_{\mathbf{d}} \in \mathbb{R}^{3\times 2}$ and $\mathbf{B}_{\mathbf{d}} : \mathbb{R}^2 \xrightarrow{\sim} T_{\mathbf{d}}S^2$. Stretching the Lie notation from [37], we define the $\oplus$ operator using the exponential map described in [39, p.22]:

$$\widehat{\mathbf{d}} = \mathbf{d} \oplus \boldsymbol{\xi} \triangleq \cos\left(\left\|\widehat{\boldsymbol{\xi}}\right\|\right)\mathbf{d} + \sin\left(\left\|\widehat{\boldsymbol{\xi}}\right\|\right)\frac{\widehat{\boldsymbol{\xi}}}{\left\|\widehat{\boldsymbol{\xi}}\right\|}, \tag{10}$$

where

$$\widehat{\boldsymbol{\xi}} = \mathbf{B}_{\mathbf{d}}\boldsymbol{\xi}. \tag{11}$$

This $\oplus$ operator lets us increment a direction $\mathbf{d}$ with a tangent space vector $\boldsymbol{\xi}$, which we need in order to work with directions in the factor graph estimation framework.

IV. A Model for Pan/Tilt Cameras

This section defines the components of our PTZ system model. We begin by proposing representations for direction landmarks and pan/tilt states and use these to define a model of the pan/tilt mechanics. Next, we propose a clock model for PTZ cameras, and an interpolation scheme that enables us to optimize clock synchronization continuously on discrete pan/tilt measurements. Finally, we define the intrinsic camera model, including RS modeling.

A. Manifold Representation of Pan/Tilt states and Landmarks

Pan/tilt measurements and states are two-tuples $(\phi, \psi) \in [-\pi, \pi) \times [-\pi, \pi)$, which can be viewed as points on the unit torus $T^2 = S^1 \times S^1$. To avoid issues with wraparounds when comparing states, we use a complex two-vector with unit-length components as an underlying representation. We adapt the notation from [37] to define mappings between pan/tilt angles (ϕ, ψ) and the underlying representation $\mathbf{p} \in \mathbb{C}^2$:

$$\begin{aligned} \mathbf{p} &= \mathrm{Exp}(\phi, \psi) \triangleq (e^{\phi i}, e^{\psi i}) \\ (\phi, \psi) &= \mathrm{Log}(\mathbf{p}) \triangleq (\arg(p_0), \arg(p_1)), \end{aligned} \tag{12}$$

where $\arg(p_i)$ denotes the argument of the complex number p_i. Given a pan/tilt increment $\boldsymbol{\xi}$, we define the $\oplus$ operator as

$$\widehat{\mathbf{p}} = \mathbf{p} \oplus \boldsymbol{\xi} \triangleq (p_0 \cdot e^{\xi_0 i}, p_1 \cdot e^{\xi_1 i}). \tag{13}$$

To obtain an increment $\boldsymbol{\xi}$ turning a pan/tilt $\mathbf{p}$ into a pan/tilt $\mathbf{q}$, we define the $\ominus$ operator via the complex conjugate $\overline{p}_i$, as

$$\boldsymbol{\xi} = \mathbf{q} \ominus \mathbf{p} \triangleq (\arg(q_0 \cdot \overline{p}_0), \arg(q_1 \cdot \overline{p}_1)). \tag{14}$$

In Section V-B we require the Jacobians of Exp and $\ominus$. A nice property of this representation is that all of these Jacobians are (essentially) the identity:

$$\begin{bmatrix}\mathbf{J}_{\phi}^{\mathrm{Exp}} & \mathbf{J}_{\psi}^{\mathrm{Exp}}\end{bmatrix} = \mathbf{J}_{\mathbf{q}}^{\ominus} = -\mathbf{J}_{\mathbf{p}}^{\ominus} = \mathbf{I}_{2\times 2}. \tag{15}$$

As discussed in Section III-B, we represent landmark positions as directions $\mathbf{d} \in S^2$. Since we do not measure landmark directions directly, we only need the $\oplus$ operator from Eq. (10). Here, the necessary Jacobians are provided in [40].

B. Manifold Representation of Pan/Tilt Mechanics

The goal of the extrinsic PTZ camera model is to describe the relative pose between the camera itself and the stationary part of the PTZ, as illustrated in Fig. 2. We start by defining the camera frame $\mathcal{F}_c$ with origin in the optical center of the camera and axes oriented right-down-forward (RDF). Next, we define the base frame $\mathcal{F}_b$ as fixed to the stationary part of the PTZ with the same origin and orientation as the camera at 0 pan and 0 tilt, but with forward-right-down (FRD) axes. As in [9], we model camera motion as sequential rotations about two arbitrary axes. However, as we focus on narrow FOV operation observing a very distant scene, we assume that the effects of translation in this motion discussed in [41] are negligible. Contrary to [9], we therefore assume that the rotational axes intersect the optical center, which in turn means that the pose between the camera and the base is a pure rotation $\mathbf{R}_{bc} \in SO(3)$, and the axes can be represented as directions $\mathbf{a} \in S^2$.

Given a pan/tilt $\mathbf{p}$ with $(\phi, \psi) = \mathrm{Log}(\mathbf{p})$, we explicitly write the camera orientation $\mathbf{R}_{bc}$ as a combination of pan ϕ about the pan axis $\mathbf{a}_\phi \in S^2$ and tilt ψ about the tilt axis $\mathbf{a}_\psi \in S^2$:

$$\mathbf{R}_{bc}(\mathbf{p}, \mathbf{a}_\phi, \mathbf{a}_\psi) = \mathrm{Exp}(\phi \cdot \mathbf{a}_\phi)\,\mathrm{Exp}(\psi \cdot \mathbf{a}_\psi)\mathbf{R}_{c_fc}, \tag{16}$$

where $\mathbf{R}_{c_fc}$ is the fixed rotation from RDF to FRD.

Incorporating insights from [8], we also account for potential scaling errors in the pan/tilt measurements. With β_ϕ and β_ψ denoting the pan and tilt scaling factors, we use the following measurement model:

$$\widetilde{\mathbf{p}}_i = \begin{bmatrix}\widetilde{\phi}_i \\ \widetilde{\psi}_i\end{bmatrix} = \begin{bmatrix}\beta_\phi \phi_i \\ \beta_\psi \psi_i\end{bmatrix} + \mathbf{v}_i,\ \mathbf{v}_i \sim \mathcal{N}\left(\mathbf{0}, \mathbf{\Sigma}_{\widetilde{\mathbf{p}}}\right), \tag{17}$$

where $\mathbf{\Sigma}_{\widetilde{\mathbf{p}}}$ is the covariance of the pan/tilt measurement. In the base version of our method, we assume $\beta_\phi = \beta_\psi = 1$ as a hard prior, which is suitable for most PTZ cameras. For cameras with potential pan/tilt scaling errors, we use soft priors on β_ϕ and β_ψ. We refer to these two versions as "base PTCEE" and "soft PTCEE", respectively.

The full mechanical model is illustrated in Fig. 3, with pan/tilt measured as scaled rotations about the arbitrary pan- and tilt axes.

C. A Clock Model for Pan-Tilt Cameras

When image and pan/tilt events are timestamped in software we expect there to be some nonzero offset in time between the actual event and timestamp acquisition. We assume this offset remains constant, and that any remaining jitter can be

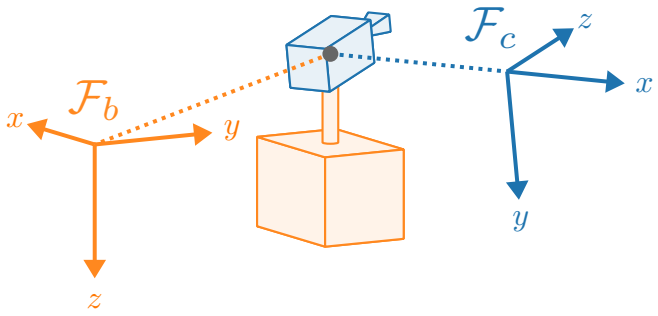


Fig. 2. In our extrinsic model of the PTZ camera, we consider the *base* (orange) and the *camera* (blue) with coordinate frames $\mathcal{F}_b$ and $\mathcal{F}_c$, respectively. We define $\mathcal{F}_b$ and $\mathcal{F}_c$ to both have origin in the optical cente (the grey dot), so they only differ in orientation. By convention, we use RDF axes for $\mathcal{F}_c$, and FRD for $\mathcal{F}_b$.

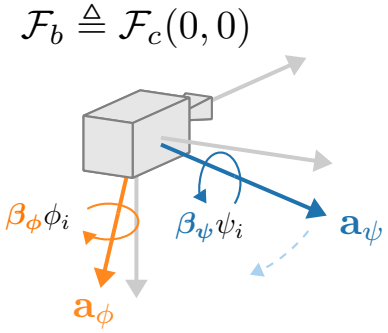


Fig. 3. The components of our extrinsic PTZ camera model. We define $\mathcal{F}_b$ to have the same orientation (albeit with FRD axes) as $\mathcal{F}_c$ when the system is at 0 pan and 0 tilt. We model panning and tilting as rotations about the axes $\mathbf{a}_\phi$ and $\mathbf{a}_\psi$, which intersect the optical center but are otherwise arbitrary. Here, $\mathbf{a}_\psi$ pans along with the camera, as described by Eq. (16). We model the true pan and tilt to be scaled by β_ϕ and β_ψ, respectively, when measured.

explained as white Gaussian noise. The model for acquiring image and pan/tilt timestamps can therefore be written as

$$\begin{aligned}\widetilde{t}_i^{(\mathrm{img})} &= t_i^{(\mathrm{img})} + d^{(\mathrm{img})} + v_i, & v_i &\sim \mathcal{N}\left(0, \sigma_t^{(\mathrm{img})}\right) \\ \widetilde{t}_j^{(\mathrm{pt})} &= t_j^{(\mathrm{pt})} + d^{(\mathrm{pt})} + v_j, & v_j &\sim \mathcal{N}\left(0, \sigma_t^{(\mathrm{pt})}\right).\end{aligned} \tag{18}$$

In practice, we cannot observe $d^{(\mathrm{img})}$ and $d^{(\mathrm{pt})}$ directly, but rather the *clock offset*, which we denote

$$d \triangleq d^{(\mathrm{img})} - d^{(\mathrm{pt})}. \tag{19}$$

D. Absolute and Relative Timing

The lack of hardware support for timestamping in many low-cost PTZ cameras often leads to disproportionately uncertain timestamps for images and pan/tilt measurements. In many cases, timestamps are acquired in software by recording the current system clock upon data reception, allowing the jittery delay of the data extraction pipeline to contaminate the timestamps. However, the actual image capture and pan/tilt measurements happen in low-level hardware and usually run at very stable fixed rates. If we can determine the data acquisition rate with reasonable accuracy, we know the period between subsequent measurements with far greater accuracy than the accuracy of their absolute timestamps.

For both images and pan/tilt measurements we assume that we observe white Gaussian noise on absolute timestamps as

$$\widetilde{t}_i = t_i + v_i, v_i \sim \mathcal{N}(0, \sigma_t), \tag{20}$$

and relative periods as

$$\widetilde{dt}_i = t_i - t_{i-1} + w_i, w_i \sim \mathcal{N}(0, \sigma_{dt}). \tag{21}$$

Here, σ_t and σ_{dt} denote the standard deviations of the absolute timestamps and the relative periods, respectively, where typically $\sigma_{dt} \ll \sigma_t$.

E. Continuous-Time Representations of Discrete-Time Pan-Tilt Measurements

For associating pan/tilt measurements to images via a continuous clock offset d, we opt to predict a pan/tilt measurement at each image timestamp. This approach is somewhat simplistic compared to [42], where the continuous time-shift is handled in the state space rather than in the measurements. However, due to our high rate of measurements compared to the dynamics of a calibrating PTZ camera, the effects of neglecting the system model are less severe.

We predict the pan/tilt measurement by piecewise linear interpolation on the discrete buffer of pan/tilt measurements $\{(\widetilde{t}_j^{(\mathrm{pt})}, \widetilde{dt}_j^{(\mathrm{pt})}, \widetilde{\mathbf{p}}_j)\}_{j=0}^{n}$. Given a deterministic d and a stochastic image timestamp $\widetilde{t}_i^{(\mathrm{img})}$, we find j such that $\widetilde{t}_{j-1}^{(\mathrm{pt})} \leq \widetilde{t}_i^{(\mathrm{img})} - d < \widetilde{t}_j^{(\mathrm{pt})}$, and then predict $\widetilde{\mathbf{p}}_i$ as

$$l_i(d) = \frac{\widetilde{t}_i^{(\mathrm{img})} - \left(\widetilde{t}_{j-1}^{(\mathrm{pt})} + d\right)}{\widetilde{dt}_j^{(\mathrm{pt})}} \tag{22}$$

$$\widetilde{\mathbf{p}}_i(d) = \widetilde{\mathbf{p}}_{j-1} \oplus l_i(\widetilde{\mathbf{p}}_j \ominus \widetilde{\mathbf{p}}_{j-1}). \tag{23}$$

Since we use the predicted $\widetilde{\mathbf{p}}_i$ as a measurement in our optimization were d is an optimization variable, we need $\mathbf{J}_d^{\widetilde{\mathbf{p}}_i}$ as well as $\mathbf{\Sigma}_{\widetilde{\mathbf{p}}_i}$. $\mathbf{J}_d^{\widetilde{\mathbf{p}}_i}$ simply becomes

$$\mathbf{J}_d^{\widetilde{\mathbf{p}}} = \widetilde{\boldsymbol{\omega}}_j = \frac{\widetilde{\mathbf{p}}_{j-1} \ominus \widetilde{\mathbf{p}}_j}{\widetilde{dt}_j^{(\mathrm{pt})}}. \tag{24}$$

As $\widetilde{\mathbf{p}}_i$ is nonlinear in $\widetilde{dt}^{(\mathrm{pt})}$, we employ a first order approximation of $\mathbf{\Sigma}_{\widetilde{\mathbf{p}}_i}$:

$$\begin{aligned}\mathbf{\Sigma}_{\widetilde{\mathbf{p}}_i} \approx\ & (1-l)^2 \mathbf{\Sigma}_{\widetilde{\mathbf{p}}_{j-1}} + l^2 \mathbf{\Sigma}_{\widetilde{\mathbf{p}}_j} \\ & + \left(\sigma_t^{(\mathrm{img})^2} + \sigma_t^{(\mathrm{pt})^2}\right) \widetilde{\boldsymbol{\omega}}_j^\top \widetilde{\boldsymbol{\omega}}_j \\ & + \left(\frac{\widetilde{t}_{j-1}^{(\mathrm{pt})} + d - \widetilde{t}_i^{(\mathrm{img})}}{\widetilde{dt}_j^{(\mathrm{pt})}}\right)^2 \sigma_{dt_j}^{(\mathrm{pt})^2} \widetilde{\boldsymbol{\omega}}_j^\top \widetilde{\boldsymbol{\omega}}_j.\end{aligned} \tag{25}$$

Exploiting $dt_j^{(\mathrm{pt})}$ instead of using $\widetilde{t}_j^{(\mathrm{pt})} - \widetilde{t}_{j-1}^{(\mathrm{pt})}$, which would give double absolute uncertainty, makes this approximation viable.

F. Rolling Shutter Camera Model

The camera model describes the relationship between the direction of a landmark $\mathbf{d}^c \in S^2$ in the camera frame, $\mathcal{F}_c$, and its observed pixel position $(u, v) = \mathbf{u} \in \mathbb{R}^2$ in the image. Our landmarks are static directions $\mathbf{d}^b$ in the base frame $\mathcal{F}_b$, which correspond to camera frame directions $\mathbf{d}^c = \mathbf{R}_{cb}\mathbf{d}^b$. We extend the simplified version of the perspective camera model with quadratic radial distortion used in [11] to account for rolling shutter.

We first assume that our rolling shutter camera works by exposing the image row by row linearly, with *line duration* ℓ, as in [29], [30], [32]–[34]. That is, for an image where exposure began at t_0, row v was exposed at time $t_0 + v \cdot \ell$. Further, we assume that the camera rotates approximately linearly during exposure, such that

$$\mathbf{R}_{bc}(t) \approx \mathbf{R}_{bc}(t_0) \oplus (t - t_0)\boldsymbol{\omega}_0, \tag{26}$$

is a good approximation of its orientation for t near t_0. Here $\boldsymbol{\omega}_0$ is the angular velocity of the camera at t_0, decomposed in the camera frame. Given some global shutter camera model π with parameters C, we then define our rolling shutter camera model π_{rs} through its inverse as

$$\mathbf{d}^{c_0} = \pi_{\mathrm{rs}}^{-1}(\mathbf{u}; C, \ell, \boldsymbol{\omega}_0) \triangleq \mathrm{Exp}(v\ell\boldsymbol{\omega}_0)\pi^{-1}(\mathbf{u}; C), \tag{27}$$

where $\mathbf{d}^{c_0}$ is the corresponding direction in $\mathcal{F}_c$ at t_0.

For a camera capturing images of size $w \times h$, we use the perspective camera model as the base model $\pi(\cdot)$, with a single focal parameter f, quadratic radial distortion k, and fixed optical center:

$$\begin{aligned}\mathbf{u} &= \pi(\mathbf{d}^c; f, k) \\ &\triangleq f\mathbf{x}^u \left(1 + k\|\mathbf{x}^u\|_2^2\right) + \begin{bmatrix} w/2 \\ h/2 \end{bmatrix},\end{aligned} \tag{28}$$

with

$$\mathbf{x}^u = \frac{1}{d_3^c} \begin{bmatrix} d_1^c \\ d_2^c \end{bmatrix}. \tag{29}$$

Here, neither π nor π_{rs}^{-1} are invertible, so we implement π^{-1} and π_{rs} through fixed point iteration.

V. Method

We now have all the building blocks needed to construct our method as outlined in Fig. 1. Our primary focus is constructing the factor graph and optimizing it to obtain our full PTZ camera model in Section V-B. To do so, we first develop a proof-of-concept frontend in Section V-A to provide observations to feed into the factor graph. Figure 4 provides a more detailed illustration of the frontend and backend components, and how they interact. Finally, in order to demonstrate the utility of the model on real-world data, we repurpose said frontend in Section V-C to perform orientation estimation based on the calibrated model and landmark map.

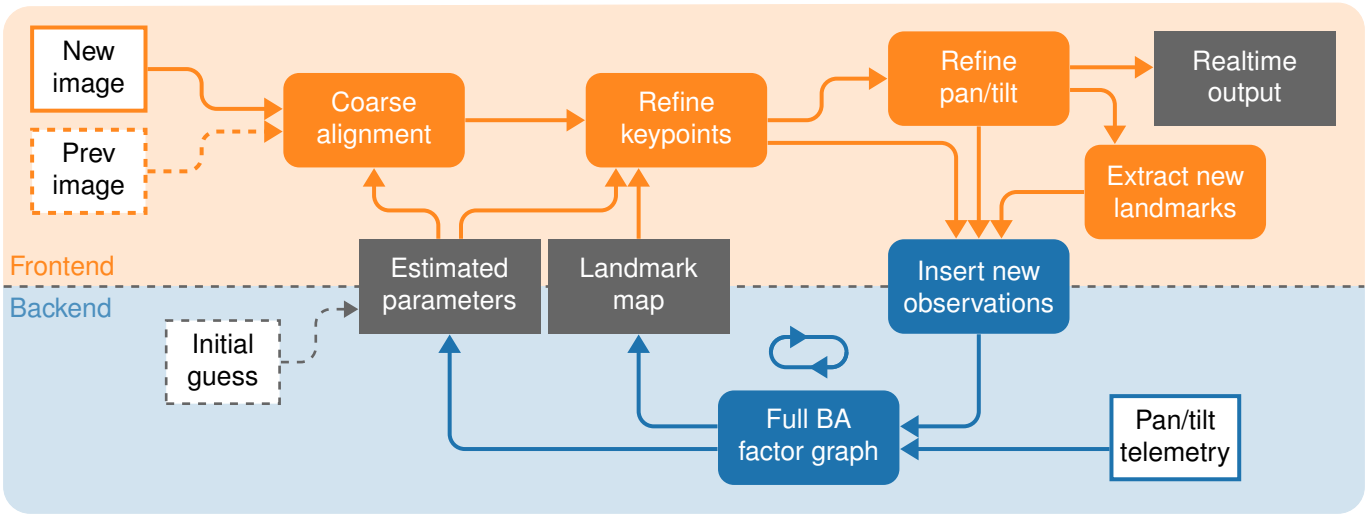


Fig. 4. The overall flow of our proof-of-concept SVO-style frontend, along with its connection to the backend. Square boxes with white background depict inputs to our method, while square boxes with gray background depict outputs. Orange rounded boxes represent frontend processing, while blue rounded boxes are part of the backend. The previous image and initial guess are drawn with a dashed border because only some already-processed data and a very rough initial guess is needed.

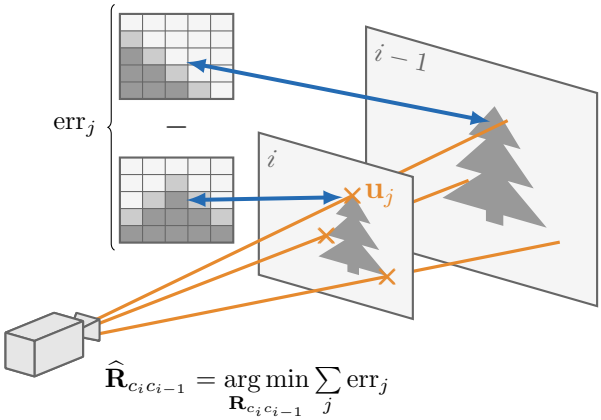


Fig. 5. Coarse alignment in the frontend. For each new image i we find the orientation between the camera at time i and the previous image $i-1$, which we write $\mathbf{R}_{c_i c_{i-1}}$. This orientation is found by minimising the total photometric error between patches around keypoints in the current image and the patches around the corresponding points reprojected about $\mathbf{R}_{c_i c_{i-1}}$ into the previous image.

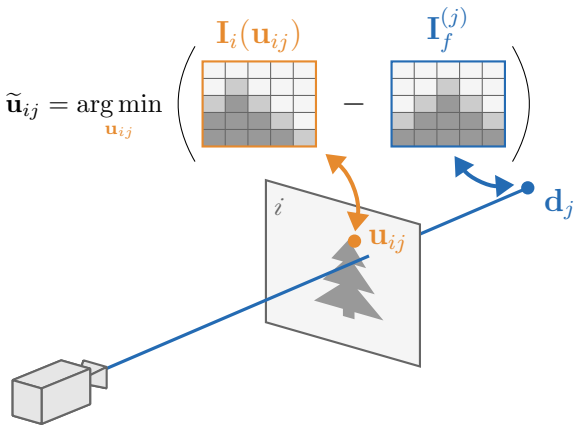


Fig. 6. Keypoint refinement in the frontend. The landmark map stores each landmark j as a direction $\mathbf{d}_j$ and a pixel patch (blue rectangle). Using the orientation of the current frame i from coarse alignment we project $\mathbf{d}_j$ as an initial guess for the pixel position $\mathbf{u}_{ij}$. We find the refined keypoint $\widetilde{\mathbf{u}}_{ij}$ as the pixel position of the patch $\mathbf{I}_i(\mathbf{u}_{ij})$ which best matches the landmark patch $\mathbf{I}_f^{(j)}$ (blue rectangle).

A. *Semi-Direct Direction-Only Frontend*

We need a method for locating landmarks across images with high precision, and preferably capable of re-finding landmarks as they move in and out of the FOV. The frontends of many visual SLAM methods such as SVO [12], [43], ORB-SLAM [44]–[46], Kimera [47] and Basalt [48] are suitable for this with minor modifications. As a proof of concept, we propose a frontend based on the semi-direct keypoint matching used in SVO, but modified to take advantage of our rotation-only setting.

For each new image i, we first perform coarse alignment between the current and previous image, as illustrated in Fig. 5. Here, we seek to find the relative camera orientation between the two images $\mathbf{R}_{c_i c_{i-1}}$. We start by extracting a set of FAST corners [49] in the current image, resulting in a set of pixel positions $\{\mathbf{u}_j\}$. Around each such keypoint we extract a small (typically 4×4) patch of pixels $\mathbf{I}_i(\mathbf{u}_j)$. Given a camera model $\pi_{(\cdot)}$ with parameters C, we use

$$\mathrm{err}_j\left(\mathbf{R}_{c_i c_{i-1}}\right) = \left\| \mathbf{I}_i(\mathbf{u}_j) - \mathbf{I}_{i-1}\left(\pi_{(\cdot)}\left(\mathbf{R}^{-1}_{c_i c_{i-1}} \pi^{-1}_{(\cdot)}\left(\mathbf{u}_j; C\right); C\right)\right)\right\| \tag{30}$$

to denote the photometric error between $\mathbf{I}_i(\mathbf{u}_j)$ and the corresponding patch in the previous image. We then find the orientation between the images as

$$\widehat{\mathbf{R}}_{c_i c_{i-1}} = \arg\min_{\mathbf{R}_{c_i c_{i-1}}} \sum_{j \in \mathcal{C}} \mathrm{err}_j^2\left(\mathbf{R}_{c_i c_{i-1}}\right), \tag{31}$$

where $\mathcal{C}$ indexes the set of keypoints from the current image. Using the inverse compositional formulation suggested in [12],

we can efficiently optimize Eq. (31) with Gauss-Newton. In order to handle larger movements between subsequent images, we perform this procedure on coarse-to-fine pyramids of the images.

Next, suppose we have a map of landmarks with locations represented in some frame $\mathcal{F}_f$, and that for the previous image $(i-1)$ we have an estimate of its orientation relative to the map $\widehat{\mathbf{R}}_{c_{i-1}f}$. For each landmark j, we have stored a direction $\mathbf{d}_j^f$ and an image patch $\mathbf{I}_f^{(j)}$ from the first image it was observed. In image i, we use

$$\mathbf{u}_{ij} = \pi_{(\cdot)}\left(\widehat{\mathbf{R}}_{c_i c_{i-1}}\widehat{\mathbf{R}}_{c_{i-1}f}\mathbf{d}_j^f; C\right) \tag{32}$$

as an initial guess for the current pixel position of landmark j. As illustrated in Fig. 6, we then perform the feature alignment procedure from [12] by finding

$$\widetilde{\mathbf{u}}_{ij} = \underset{\mathbf{u}_{ij}}{\arg\min} \left\| \mathbf{I}_i(\mathbf{u}_{ij}) - \mathbf{I}_f^{(j)} \right\|, \tag{33}$$

where $\mathbf{I}_i(\mathbf{u}_{ij})$ is the extracted patch around $\mathbf{u}_{ij}$ in the new image. We use the resulting set of $\{\widetilde{\mathbf{u}}_{ij}\}$ as our refined keypoint observations from image i. Similarly to [12], we use the approximated Hessian to obtain an estimate for the observation covariance $\Sigma_{\widetilde{\mathbf{u}}_{ij}}$, but assume pixel intensity noise to be distributed as $\mathcal{N}(0, \sigma_I/2^k)$, where k is the current pyramid level.

Given pan/tilt axes $\mathbf{a}_\phi$ and $\mathbf{a}_\psi$, we now want to obtain a refined pan/tilt state $\mathbf{p}_i$ for the camera at image i based on our refined keypoint observations. Using Eq. (16) we write

$$\mathbf{R}_{c_i f}\left(\mathbf{p}_i\right) = \mathbf{R}_{bc}^{-1}\left(\mathbf{p}_i, \mathbf{a}_\phi, \mathbf{a}_\psi\right). \tag{34}$$

By minimising the reprojection error of the set of all visible landmarks $\mathcal{J}$ we then obtain

$$\widetilde{\mathbf{p}}_i^f = \underset{\mathbf{p}_i}{\arg\min} \sum_{j\in\mathcal{J}} \left\| \widetilde{\mathbf{u}}_{ij} - \pi_{(\cdot)}\left(\mathbf{R}_{c_i f}\left(\mathbf{p}_i\right)\mathbf{d}_j^f; C\right)\right\|_{\mathbf{\Sigma}_j}^2 \tag{35}$$

as the refined pan/tilt state at time i and the corresponding refined orientation $\mathbf{R}_{c_i f}\left(\widetilde{\mathbf{p}}_i^f\right)$.

If the current image contains suitable FAST corners [49] in empty regions of the map, we want to create corresponding landmarks and insert them into the map. To ensure a somewhat uniform spread of landmarks, we organize the map by dividing the unit sphere around the camera into equally sized cells in terms of latitude/longitude. In each such cell we allow atmost one landmark per pyramid octave. Given a new keypoint $\mathbf{u}_j$ in the current image which falls in an empty combination of cell/octave, we insert

$$\mathbf{d}_j^f = \mathbf{R}_{c_i f}^{-1}\left(\widetilde{\mathbf{p}}_i^f\right)\pi_{(\cdot)}^{-1}\left(\mathbf{u}_j; C\right) \tag{36}$$

and the corresponding pixel patch $\mathbf{I}_i\mathbf{u}_j$ into the map.

When the frontend is applied to the very first image, we omit all of these steps and use $\mathbf{R}_{c_1 f} = \mathbf{I}$ to insert landmarks directly. Upon startup, we use $\mathbf{a}_\phi = \mathbf{e}_z$ and $\mathbf{a}_\psi = \mathbf{e}_y$ as initial guesses for the pan/tilt axes. For the camera model, we initialize using the global shutter camera model $\pi(\cdot)$ from Eq. (28) with quadratic radial distortion $k = 0$. As long as the camera has a narrow FOV, our experiments indicate that the initial focal length f can be off by several orders of magnitude. Therefore, using an initial f corresponding to an FOV of 2° seems to work for any true FOV $\in (0^\circ, 5^\circ]$.

Continuing to operate the frontend with these initial values works in the sense that it produces consistent new observations $\mathbf{u}_{ij}$ of landmarks j. However, the refined pan/tilt states $\widetilde{\mathbf{p}}_i^f$ relate to the arbitrary frame $\mathcal{F}_f$, and their scale is dictated by the initial focal length f. At some point the backend should provide bundle adjusted estimates of all the camera parameters and axes, as well as refined landmark directions $\widehat{\mathbf{d}}_j^b$ and pan/tilt states $\widehat{\mathbf{p}}_i^b$ for previous images, this time in the base frame $\mathcal{F}_b$. With these in hand, we transition the frontend to operate in the base frame by replacing each $\mathbf{d}_j^f$ with $\widehat{\mathbf{d}}_j^b$ and $\mathbf{R}_{c_{i-1}f}$ with $\mathbf{R}_{bc}^{-1}(\widehat{\mathbf{p}}_i^b, \widehat{\mathbf{a}}_\phi, \widehat{\mathbf{a}}_\psi)$. We also replace the initial camera model $\pi(\cdot)$ with the rolling shutter compensated camera model $\pi_{\text{rs}}(\cdot)$ using estimated focal length $\widehat{f}$, quadratic radial distortion $\widehat{k}$ and line duration $\widehat{\ell}$. From this point on, the refined pan/tilt states from Eq. (35) can be interpreted directly as corresponding to orientations in the base frame.

B. Constructing the Factor Graph

We use GTSAM [40] to implement our estimation problem as a factor graph, but several other optimization frameworks would also be suitable. Our graph contains eight global variables: The clock offset d as defined in Eq. (19), the camera focal length f, quadratic radial distortion k and rolling shutter line duration ℓ, the pan and tilt axes $\mathbf{a}_\phi$, $\mathbf{a}_\psi \in S^2$ and the corresponding pan/tilt scaling factors β_ϕ and β_ψ. For each image i, we include a variable $\mathbf{p}_i \in T^2$ representing the true pan/tilt of the camera at the time the first row was exposed $t_i^{(\text{img})}$. Additionally, for each landmark track j we include a variable $\mathbf{d}_j^b \in S^2$ representing its true direction in the base frame. In the following, we define the two factors incorporating our pan/tilt measurements and landmark observations. Figure 7 shows the resulting cluster for a single image in the final factor graph.

1) The Pan/Tilt Buffer Factor: As suggested in Section IV-E, we enforce the pan/tilt measurements on each pan/tilt state $\mathbf{p}_i$ by using the buffer of timestamped pan/tilt measurements to predict a measurement $\widetilde{\mathbf{p}}_i$ at $t_i^{(\text{img})}$. With $\mathbf{p}_i = \text{Exp}(\phi_i, \psi_i)$, we define the pan/tilt factor as

$$\phi_i^{(\text{pt})}(\mathbf{p}_i, d, \beta_\phi, \beta_\psi) \triangleq \|\widetilde{\mathbf{p}}_i(d) \ominus \text{Exp}(\beta_\phi\phi_i, \beta_\psi\psi_i)\|_{\mathbf{\Sigma}_{\widetilde{\mathbf{p}}_i}}^2, \tag{37}$$

where $\widetilde{\mathbf{p}}_i(\cdot)$ is given in Eq. (23), and $\mathbf{\Sigma}_{\widetilde{\mathbf{p}}_i}$ in Eq. (25).

2) The Directional Projection Factor: For each landmark j that has been observed by image i at pixel position $\widetilde{\mathbf{u}}_{ij}$, we ideally want to compare the observation to a projection $\mathbf{u}_{ij}^{(\text{proj})}$ of the landmark, essentially as

$$\left\| \widetilde{\mathbf{u}}_{ij} - \mathbf{u}_{ij}^{(\text{proj})} \right\|_{\mathbf{\Sigma}_{\widetilde{\mathbf{u}}_{ij}} + \mathbf{\Sigma}_{\mathbf{u}_{ij}^{(\text{proj})}}}^2. \tag{38}$$

Here, $\mathbf{\Sigma}_{\widetilde{\mathbf{u}}_{ij}}$ and $\mathbf{\Sigma}_{\mathbf{u}_{ij}^{(\text{proj})}}$ are the covariances of the landmark observation and projection, respectively.

Given an estimate of the landmark direction in the base frame $\mathbf{d}_j^b$ as well as the orientation and angular velocity of the camera as the first row of image i was exposed $\mathbf{R}_{bc_i}$ and $\boldsymbol{\omega}_i$, we could in principle obtain the landmark projection as

$$\mathbf{u}_{ij}^{(\text{proj})} = \pi_{\text{rs}}\left(\mathbf{R}_{bc_i}^{-1}\mathbf{d}_j^b; \cdot, \boldsymbol{\omega}_i\right). \tag{39}$$

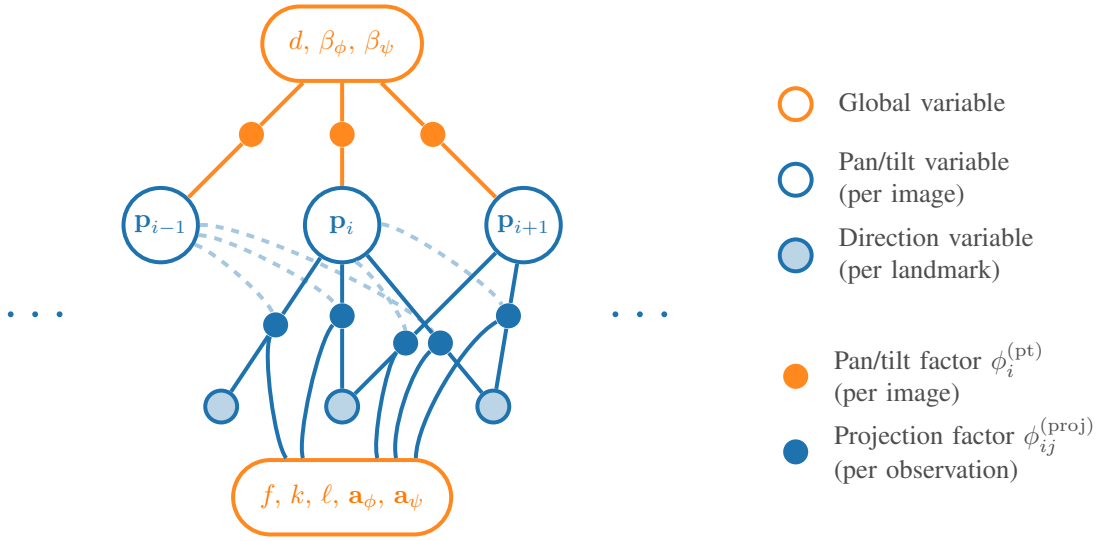


Fig. 7. Overview of the factor graph representing our estimation problem, showing the factors generated for image i. The graph contains global variables for the clock offset d, pan/tilt scales β_ϕ and β_ψ, camera focal length f, quadratic radial distortion k, rolling shutter line duration ℓ, and pan/tilt axes $\mathbf{a}_\phi$ and $\mathbf{a}_\psi$. For each image, we estimate a variable for the current camera pan/tilt $\mathbf{p}_i \in T^2$, and for each landmark track, we estimate the landmark direction $\mathbf{d}_j^b \in S^2$ in the base frame.

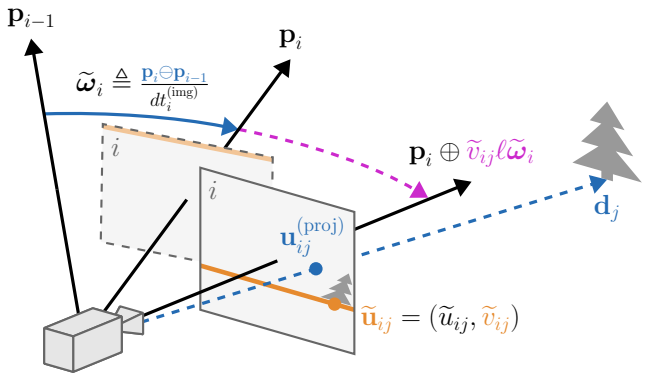


Fig. 8. An exaggerated visualization of the components involved in the projection factor from Eq. (43). The dashed frame shows image i at the exact moment when the first row is exposed. The solid frame shows image i a few moments later as landmark j at pixel position $\widetilde{\mathbf{u}}_{ij}$ is exposed. The pan-tilt states $\mathbf{p}_i$ and $\mathbf{p}_{i-1}$ correspond (via Eq. (16)) to the orientation of the camera as the first row of the current and previous frames were captured. Based on the rolling shutter line duration ℓ, we extrapolate (the magenta arc) the orientation of the camera to when row $\widetilde{v}_{ij}$ (the orange line) was captured, which in turn allows us to project the landmark direction (via Eq. (42)) as $\mathbf{u}_{ij}^{(\mathrm{proj})}$ and compare it with $\widetilde{\mathbf{u}}_{ij}$.

The first issue with this approach is that we do not have an estimate for $\boldsymbol{\omega}_i$, as it is not part of the factor graph. For simplicity, we instead approximate the angular velocity using the pan/tilt state of the previous image as

$$\widetilde{\boldsymbol{\omega}}_i \triangleq \frac{\mathbf{p}_i \ominus \mathbf{p}_{i-1}}{dt_i^{(\mathrm{img})}}. \tag{40}$$

Secondly, the forward evaluation of $\pi_{\mathrm{rs}}(\cdot)$ involves the costly fixed point iteration on the resulting pixel row. The key insight here is that the pixel observation $\widetilde{\mathbf{u}}_{ij} = (\widetilde{u}_{ij}, \widetilde{v}_{ij})$ has relatively low uncertainty, and that we expect our projection to be close to it. We can therefore instead use the orientation corresponding to the time of exposure for row $\widetilde{v}_{ij}$ together with the base camera model $\pi(\cdot)$ directly. To this end, we employ Eq. (16) and use

$$\widetilde{\mathbf{R}}_{ij} \triangleq \mathbf{R}_{bc}^{-1}\left(\mathbf{p}_i \oplus \widetilde{v}_{ij}\ell\widetilde{\boldsymbol{\omega}}_i,\ \mathbf{a}_\phi,\ \mathbf{a}_\psi\right) \tag{41}$$

to denote our approximation to the camera orientation as landmark j was exposed by image i. As illustrated in Fig. 8, we now apply the base camera model $\pi(\cdot)$ to arrive at our approximation of the landmark projection

$$\mathbf{u}_{ij}^{(\mathrm{proj})} \approx \pi\left(\widetilde{\mathbf{R}}_{ij}\mathbf{d}_j^b; f, k\right). \tag{42}$$

We then finally formulate our projection factor as

$$\begin{aligned}\phi_{ij}^{(\mathrm{proj})}&\left(\mathbf{p}_i, \mathbf{p}_{i-1}, \mathbf{d}_j^b, f, k, \mathbf{a}_\phi, \mathbf{a}_\psi\right)\\ &\triangleq \left\|\widetilde{\mathbf{u}}_{ij} - \pi\left(\widetilde{\mathbf{R}}_{ij}\mathbf{d}_j^b; f, k\right)\right\|^2_{\boldsymbol{\Sigma}_{\mathbf{u}_{ij}} + \boldsymbol{\Sigma}_{\mathbf{u}_{ij}^{(\mathrm{proj})}}}.\end{aligned} \tag{43}$$

Occasionally, the frontend might produce an observation $\widetilde{\mathbf{u}}_{ij}$ which is associated with the wrong landmark j, or simply does not belong to any landmark. To reduce the effect of such outlier observations, we apply a Huber kernel [50] to each $\phi_{ij}^{(\mathrm{proj})}$ before adding it to the factor graph.

C. Achieving High Platform Accuracy in Real-Time

Once the backend has obtained an estimate of the full pan/tilt camera model estimated in Section V-B, we saw in Section V-A how we can tweak the frontend to make it output the refined pan/tilt state $\mathbf{p}_i^b$ for each image in the base frame. This process still involves the coarse alignment, keypoint refinement, and the final pan/tilt refinement steps before the full estimated orientation is obtained. Operating our method with this proposed orientation estimation (OE) we refer to as "OE mode". As we shall see in Section VI-C, the OE mode is relatively fast but incurs a few ms of latency before the estimate is available.

As a minimal latency alternative to the OE mode, we can skip the image processing and use the estimates from the backend together with pan/tilt measurements directly. To do this, we maintain a buffer of recent pan/tilt measurements. When a new image i arrives with timestamp $t_i^{(img)}$, we use the estimated clock offset $\widehat{d}$ to interpolate the pan/tilt buffer at time $t_i^{(img)} + \widehat{d}$. This gives us a pan/tilt measurement

$$\widetilde{\mathbf{p}}_i = \mathrm{Exp}\left(\widetilde{\phi}_i, \widetilde{\psi}_i\right) \tag{44}$$

for the moment the first row was exposed. Using the estimated pan/tilt axes $\widehat{\mathbf{a}}_\phi$ and $\widehat{\mathbf{a}}_\psi$, and pan/tilt scales $\widehat{\beta}_\phi$ and $\widehat{\beta}_\psi$, we can obtain the orientation of the camera as

$$\mathbf{R}_{bc_i} = \mathrm{Exp}\left(\frac{\widetilde{\phi}_i}{\widehat{\beta}_\phi} \cdot \widehat{\mathbf{a}}_\phi\right) \mathrm{Exp}\left(\frac{\widetilde{\psi}_i}{\widehat{\beta}_\psi} \cdot \widehat{\mathbf{a}}_\psi\right) \mathbf{R}_{c_f c}, \tag{45}$$

where $\mathbf{R}_{c_f c}$ is the fixed rotation from RDF to FRD. From the pan/tilt buffer we can also obtain the camera angular rate as

$$\boldsymbol{\omega}_i = \mathbf{J}_\phi^{\mathbf{R}_{bc}} \frac{\delta\phi}{\delta t} + \mathbf{J}_\psi^{\mathbf{R}_{bc}} \frac{\delta\psi}{\delta t}, \tag{46}$$

where $\left(\frac{\delta\phi}{\delta t}, \frac{\delta\psi}{\delta t}\right) = \widetilde{\boldsymbol{\omega}}_j$ from Eq. (24). Together with the backend-estimated focal length $\widehat{f}$, quadratic radial distortion $\widehat{k}$ and line duration $\widehat{\ell}$, the estimates for $\mathbf{R}_{bc_i}$ and $\boldsymbol{\omega}_i$ give us everything we need to use the RS camera model $\pi_{\mathrm{rs}}(\cdot)$. We refer to this manner of pairing pan/tilt measurements with backend-calibrated parameters as using our method in "PT mode".

Where the OE mode incurs a delay of a few ms running each image through the frontend, the PT mode requires only a few μs to perform the necessary look-ups in the pan/tilt buffer. Another benefit of PT mode is that it does not rely on maintaining landmark tracks, which makes it more robust in featureless regions and during full camera occlusions. The major benefit of OE mode is that its accuracy is limited by the camera resolution, which enables us to achieve sub-pixel accuracy even in the most narrow FOV cases. At these narrow zoom levels, the accuracy in PT mode is typically bounded by the inferior angular precision of the pan/tilt encoders.

VI. Experiments and Results

To analyze the performance of our method, we perform both simulations and real-world experiments. We begin in Section VI-A with a simplified simulated case where we can study how existing image-only methods compare to our pan/tilt-aided backend. Next, in Section VI-B, we expand the simulation to cover our full model with pan/tilt mechanics, radial distortion, rolling shutter and unsynchronized measurements. This setup is well suited for revealing estimation biases, verifying the covariance estimates and studying the effects of the system parameters on the estimation performance in the backend. Finally, we run the full system on data from a real camera, which also puts the frontend and real-time estimation to the test. The overall goal of the real-world test is to verify that the system behaves similarly to the backend-only simulations in the face of observation-to-landmark mismatching and realistic measurement noise.

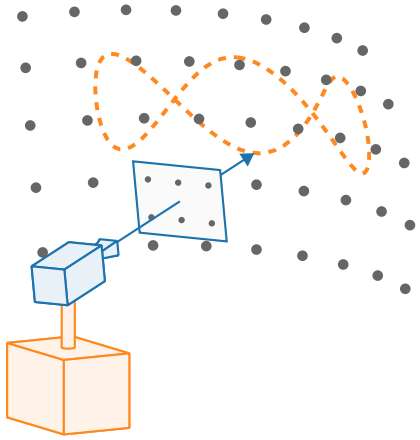


Fig. 9. Simulation setup with measurements generated from observations of idealized landmarks. The landmarks are placed in a uniform azimuth/elevation grid on the unit sphere in the base frame. We then simulate the camera maneuvering along the orange dashed path and generate observations by adding noise to projected landmarks as the camera maneuvers.

A. The effects of narrow FOV

From the theoretical discussion in [10] we expect that information from images alone is insufficient to calibrate the focal length of a camera undergoing rotation-only motion at narrow FOV. However, it remains unclear whether this effect is significant at relevant FOVs and levels of keypoint measurement noise. To investigate this, we conducted a simulation experiment comparing the performance of our proposed method against conventional image-only approaches at decreasing HFOVs of 32°, 16°, 8°, 4°, 2° and 1°. As a baseline, we used a DAQ-derived direct linear transform (DLT) followed by BA, as described in [5]. Additionally, we compared our method with the multi-level BA approach proposed in [1].

We simulate the PTZ camera in a scene with landmarks distributed in a uniform azimuth/elevation grid on the unit sphere, as illustrated in Fig. 9. The distance between landmarks is set to be $1/10$th of the horizontal FOV, resulting in 50-60 visible landmarks in each image. In each simulation run, we pan and tilt the camera along a *"triple infinity"* path, scaled to be three times the width and the same height as the FOV, illustrated by the orange curve in Fig. 9. As the camera moves along, we generate pan/tilt measurements, landmark observations and accompanying timestamps by adding measurement noise. For each FOV setting, we perform 128 simulation runs and evaluate each method on the generated measurements. Code to reproduce the results is available at github.com/ffi-no.

As detailed in Appendix B, we have chosen simulation parameters that favor the image-only methods (low keypoint measurement noise, high pan/tilt and timestamp noise). Since neither of the DLT-based methods handle rolling shutter, we assume global shutter and no radial distortion in these simulations. When calibrating for a given FOV, for instance 8°, PTCEE and the single-level DLT and BA methods only use observation data from the current FOV. The multi-level BA method also uses the observation data captured at the higher FOVs, which means data from 8°, 16° and 32° in this case.

TABLE I
MEAN ABSOLUTE ERROR OF ESTIMATED FOCAL LENGTH AT PROGRESSIVELY NARROWER FOV, COMPARING OUR PROPOSED METHOD (PTCEE) TO CONVENTIONAL METHODS.

true HFOV	**Mean absolute error in estimated HFOV**			
$\mathbf{32}^\circ$	0.028°	0.007°	0.007°	$\mathbf{0.004}^\circ$
$\mathbf{16}^\circ$	0.054°	0.015°	0.005°	$\mathbf{0.003}^\circ$
$\mathbf{8}^\circ$	0.120°	0.035°	$\mathbf{0.003}^\circ$	$\mathbf{0.003}^\circ$
$\mathbf{4}^\circ$	0.229°	0.061°	0.020°	$\mathbf{0.003}^\circ$
$\mathbf{2}^\circ$	0.480°	0.262°	0.077°	$\mathbf{0.003}^\circ$
$\mathbf{1}^\circ$	0.618°	0.194°	0.166°	$\mathbf{0.005}^\circ$
	DLT only	DLT + BA single-level	DLT + BA multi-level	PTCEE (ours)

Results for absolute error in estimated FOV are shown in Fig. 10 and Table I. Here we see that the conventional DLT + BA approaches that only employ images at the current zoom level quickly perform worse as the FOV is reduced. The multi-level BA approach from [1] remedies much of this effect by using images both at the current and higher FOVs. However, at FOVs 4° and below a growing portion of estimates diverge from the true FOV. Our proposed method (PTCEE) performs consistently well across the full range of FOVs. At wide FOV PTCEE outperforms the multi-level BA because of the extra information from the pan/tilt measurements. At the intermediate FOVs (8° down to 2°), the multi-level BA performs better on a significant portion of the estimation runs because it has access to far more data, as discussed. At narrow FOV (2° and below) the pan/tilt measurements are needed to resolve the focal length, causing PTCEE to outperform the image-only methods.

B. Backend-only simulations

To analyze the performance of the backend, we expand the simulations in Section VI-A to a wider range of focal lengths, and include distortion, clock-offset, RS line duration, and imperfect pan/tilt axes and pan/tilt measurement scaling. Since we expect uncertain priors for the pan- and tilt scaling factors β_ϕ and β_ψ to greatly affect the absolute estimation accuracy, we perform two experiments: One with hard priors ("base PTCEE") and one with soft priors ("soft PTCEE") for β_ϕ and β_ψ. We conduct 10,000 simulation runs for both experiments, while sampling a range of ground-truth parameters as detailed in Appendix C. The goal of these experiments is two-fold: 1) verify that the estimates are stable across a wide range of typical camera parameters, and 2) learn how the various system properties affect estimation accuracy.

Since the focal length f greatly varies in magnitude across these simulations, we use the mean relative error (MRE), as defined in Eq. (49). Meanwhile, the ground-truths for the quadratic radial distortion k, clock-offset d and line duration ℓ come very close to 0 in some of the runs, making the mean absolute error (MAE) a more suitable metric, as defined in Eq. (48). To assess the consistency of the predicted uncertainty (the $\widehat{\sigma}$s), we also consider the average normalized estimation error squared (ANEES), which is defined in Eq. (50).

Figure 11 shows the distribution of the $\widehat{\sigma}$-normalized estimation error for each of the parameters in the base PTCEE experiment. Here we see that PTCEE produces estimates with little bias and only minor under-reporting of $\widehat{\sigma}$-values for each of the four parameters across the wide range of parameter values considered. The soft PTCEE experiment produced similar graphs, but these are not reported for brevity.

The estimation results for both the base PTCEE and soft PTCEE experiments are shown in Table II. Here we see that base PTCEE is able to estimate the focal length f to four significant digits on average, and both the clock-offset d and the RS line duration ℓ to two-to-three significant digits. The MAE for the clock-offset d of 0.15ms should be sufficient for most applications, as it is smaller than typical shutter times. Interestingly, we determine the line duration ℓ to within tens of nanoseconds, which suggests that the joint optimization is able to determine ℓ much more accurately than what the millisecond-precision of the timestamps might imply. The quadratic radial distortion k, which we sample in the range $[-0.3, 0.3]$, is estimated to barely one significant digit, making its utility questionable for such small values. As expected, we see a significant drop in precision in the estimates of the focal length f for soft PTCEE, which uses weak priors for the pan/tilt scales β_ϕ and β_ψ. The MAE of β_ϕ and β_ψ of 0.006 is also not much better than the prior of 0.01, indicating that we are not able to extract much more information about the focal length vs the pan/tilt scale. The estimates of the quadratic radial distortion k, the clock-offset d and the line duration ℓ appear relatively unaffected by the soft priors for β_ϕ and β_ψ, which indicate that these quantities are less dependent on the absolute focal length.

In Fig. 13 we plot the MRE of the focal length f for soft PTCEE as a function of the ground truth f. For small ground truth focal lengths f, we are able to estimate f very accurately. However, as f grows from the smallest values (which correspond to large FOV), the relative estimation error increases rapidly. For higher values of f, corresponding to an FOV less than approx. 7° (right of the green line), the relative error plateaus out. Our interpretation of this effect is as follows. For small f, the estimation is dominated by the visual observations. For large values of ground truth f, increasing $\widehat{f}$ is indistinguishable in the visual observations from decreasing $\widehat{\beta}_\phi$ and $\widehat{\beta}_\psi$, and vice versa. At some point, the weak priors we set on β_ϕ and β_ψ begin to dominate the estimation. In this particular case, this seems to happen when f is corresponding to 7° FOV.

Using the estimated pan/tilt state for each image $\widehat{\mathbf{p}}_i$ and position of the landmarks $\widehat{\mathbf{d}}_j^b$ we can compute the mean estimated projection error (MEPE) of each run as

$$\hat{e}^{(\text{proj})} = \frac{1}{n} \sum_i \sum_j \|\tilde{\mathbf{u}}_{ij} - \pi(\widehat{\mathbf{R}}_{ij} \widehat{\mathbf{d}}_j, \widehat{f}, \widehat{k})\|, \qquad (47)$$

where $\widehat{\mathbf{R}}_{ij}$ is given in Eq. (41). Keep in mind that these are fully controlled experiments, where we add perfectly Gaussian noise to the landmark observations. If the method was able to perfectly estimate every pan/tilt state $\widehat{\mathbf{p}}_i$ and landmark $\widehat{\mathbf{d}}_j^b$, we would observe an $\hat{e}^{(\text{proj})}$ (nearly) equal to the pixel measurement noise σ_{px}. Therefore, Fig. 12 shows the σ_{px}-normalized MEPE, $\hat{e}^{(\text{proj})}/\sigma_{\text{px}}$, which is approximately lower-

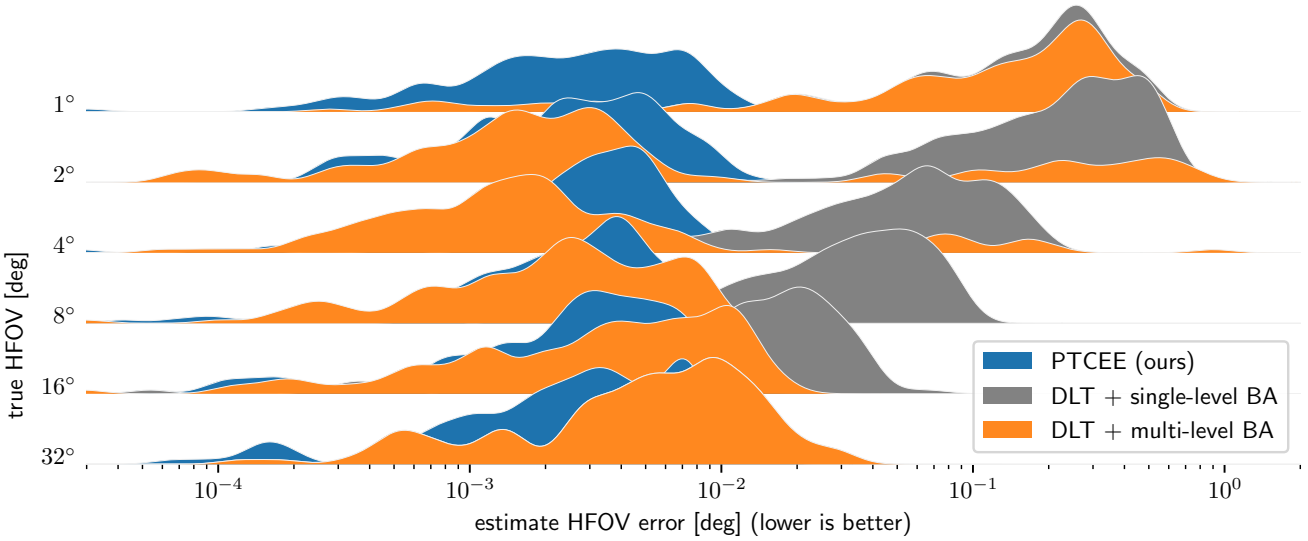


Fig. 10. Comparison of the estimation error of methods at progressively narrower FOV. Each ridgeline shows the distribution of the absolute estimation error for each method in 128 simulations at the corresponding ground-truth FOV. At the widest FOVs all three methods perform similarly. As we shrink the FOV, the performance of the single-level BA quickly degrades, while the multi-level BA mostly performs well down to 2° FOV in this setup. Our proposed method, PTCEE, maintains consistent performance across all FOVs.

bounded by 1. Here, we see that the MEPE is never far away from σ_{px}, which indicates that the estimates are self-consistent. The average $\widehat{e}^{(\text{proj})}/\sigma_{\text{px}}$ was 1.20 both for base- and soft PTCEE. Having this average so close to 1 also suggests that we are close to fully exploiting the landmark observations. Assuming a typical measurement noise of 0.5px, these results lead us to expect sub-pixel accuracy with an average reprojection error of 0.6px.

To study the effects of the various system parameters on estimation performance for base PTCEE, we report correlations between selected error metrics and system parameters in Table III. We evaluated other combinations of metrics and parameters as well, but these were the only ones showing significant correlation. We see that the average reprojection error $\widehat{e}^{(\text{proj})}$ has almost perfect correlation with pixel measurement noise σ_{px}. This correlation is expected, as its absence would be a strong indicator of the model failing to explain the variability in the observations. The accuracy in the estimate of the focal length f is highly correlated with the ground-truth f and the pan/tilt measurement noise σ_{pt}. These correlations suggest that as we approach narrower FOVs, we rely more on the absolute accuracy of the pan/tilt measurements to get the estimated focal length $\widehat{f}$ right. This result supports the theoretical discussion from [10], which highlights the ambiguity between focal length and rotational scale at narrow FOVs. The estimation error in the quadratic radial distortion k is strongly correlated with the focal length f, weakening the case for k as part of the model in this setup. Unsurprisingly, the estimation accuracy of the clock-offset d is correlated with the accuracy of the timestamps and the pan/tilt measurements. Perhaps more remarkable is that the line duration ℓ seems to be fully estimated from the landmark observations alone, and that the full range of the relative timestamp measurement noise $\sigma_{\text{dt}}^{(\text{img})}$ is sufficient to produce reasonable estimates of ℓ.

C. Real-world Experiment

We have performed an experiment on real-world data from an Axis Q6115-E PTZ camera. This camera produces Full-HD video (1920×1080) in up to either 50fps (H.264 over RTP) or 25fps (MJPEG over HTTP), and has a nominal horizontal FOV range from $63.4°$ to $2.3°$. Using a custom onboard driver, we obtain pan/tilt telemetry at approximately 30Hz through poll-sampling with timestamping upon readout. The client computer receives images in 1920×1080 at 25fps using the MJPEG stream and records timestamps upon image reception. We operate the camera with the base at the bottom, which is upside-down from its designed use, and therefore rotate the images $180°$ after reception.

We record against a rural background with a single $50\text{cm} \times 50\text{cm}$ april tag at approximately 550m distance. Figure 14 shows a typical frame. The dataset is a single continuous recording divided into three sections, as illustrated in Fig. 15. First, the dataset contains 10s (250 frames) where the camera completes a full loop around the calibration maneuver pattern. Next, there is a short 2s (50 frames) section where the camera is stationary while the april tag is in view. Finally, we pan the camera back-and-forth across the april tag while slowly tilting downwards for 10s (250 frames).

The dataset was recorded on a sunny day with damp soil and little wind, causing the scene to ”boil” due to turbulence when operating the camera at long range. To quantify the effects of this, we assess the april tag detector and frontend performance on the 2s section where the camera is stationary. In these frames, the april tag was detected with a stddev. of 1.26px, while the frontend had a stddev. across all landmarks of 1.05px. The high variation in the detection of april tag gives an impression of the atmosferic difficulties this day, and forms a lower bound for what we can expect to see in reprojection errors.

TABLE II
ESTIMATION RESULTS FOR BASE PTCEE AND SOFT PTCEE IN THE BACKEND-ONLY EXPERIMENTS. FOR EACH PARAMETER, WE REPORT EITHER THE MEAN ABSOLUTE ERROR (MAE) FROM EQ. (48) OR THE MEAN RELATIVE ERROR (MRE) FROM EQ. (49) IN ADDITION TO THE AVERAGE NORMALIZED ESTIMATION ERROR SQUARED (ANEES) FROM EQ. (50).

	f		k		d		ℓ	
	MRE	**ANEES**	**MAE**	**ANEES**	**MAE** [ms]	**ANEES**	**MAE** [ns]	**ANEES**
Base PTCEE	6.46×10^{-5}	1.04	7.68×10^{-2}	0.81	0.148	1.22	6.53	0.87
Soft PTCEE	6.05×10^{-3}	1.77	7.69×10^{-2}	0.81	0.147	1.21	6.51	0.86
	β_ϕ		β_ψ		$\mathbf{a}_\phi$ [mrad]		$\mathbf{a}_\psi$ [mrad]	
	MAE	**ANEES**	**MAE**	**ANEES**	**MAE**	**Mean $\widehat{\sigma}$**	**MAE**	**Mean $\widehat{\sigma}$**
Base PTCEE	–	–	–	–	0.39	0.43	0.42	0.56
Soft PTCEE	6.04×10^{-3}	1.77	6.04×10^{-3}	1.77	0.41	0.45	0.43	0.57

TABLE III
CORRELATIONS BETWEEN SELECT ERROR METRICS AND SYSTEM PARAMETERS IN THE BASE PTCEE BACKEND-ONLY EXPERIMENT. CORRELATIONS ABOVE 0.1 ARE WRITTEN IN **BOLD** FACE. OTHER PARAMETERS WERE CONSIDERED BUT DID NOT SHOW STRONG CORRELATIONS.

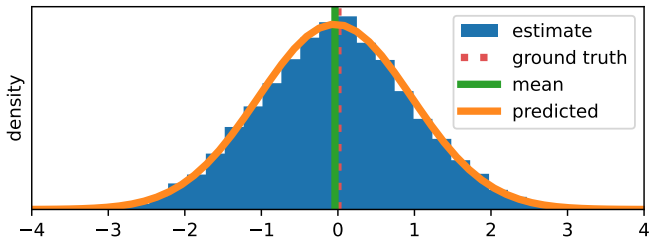
	f	σ_{pt}	σ_{px}	$\sigma_{\text{t}}^{(\text{img})}$	$\sigma_{\text{t}}^{(\text{pt})}$
$\widehat{e}^{(\text{proj})}$	-0.004	-0.009	**0.999**	0.022	-0.016
$\frac{\lvert f-\widehat{f}\rvert}{f}$	**0.391**	**0.390**	0.004	0.046	0.022
$\lvert k-\widehat{k}\rvert$	**0.593**	0.006	**0.184**	0.009	-0.023
$\lvert d-\widehat{d}\rvert$	**0.194**	**0.198**	0.003	**0.385**	**0.231**
$\lvert \ell-\widehat{\ell}\rvert$	0.011	-0.006	**0.293**	0.007	-0.001

We perform calibration on every other frame of the first 10s (125 of 250 frames) of the dataset, which gives us a similar setup to the one we used in Section VI-B. The camera was operated at 75% of maximum focal length, which according to the datasheet should correspond to a horizontal FOV of approximately $3.0°$ (52mrad), which we also use to pick our initial guess for the focal length f. By visually inspecting the power pole near the right edge of the image in Fig. 14, it is clear that the camera has severe radial distortion in this setting, so we use an initial guess of $k = 70$. To assess the effects of the pan/tilt scale, we perform estimation with both base- and soft PTCEE. In general, the optimization did not seem to be very sensitive to the initial values, but the frontend required a fairly high value for k to work reasonably well.

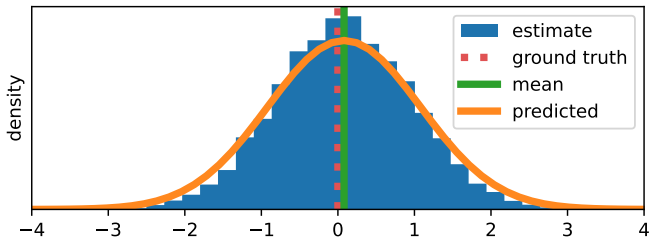
The estimation results are shown in Table IV. We observe relatively high values for the quadratic radial distortion k. However, based on our manual test in Fig. 16, such values for k are plausible in this setting. The estimate for the RS line duration ℓ is negative, which is expected since the camera is mounted upside-down, but exhibits a larger absolute value than anticipated. Upon manual inspection, the RS effects are clearly visible in the dataset. The image stretches vertically when the camera tilts upward and compresses when tilting downward. In the dataset, tilting speeds reach $30\text{mrad}/\text{s}$, at which point the image visibly warps by several pixels. At these speeds, and with an FOV of $3.44°$ ($\sim 60\text{mrad}$), the estimated $\ell = -13.5\mu\text{s}$ corresponds to a displacement at the extreme edge of approx. 8px, which aligns with our observations. A clock offset of $d = -84.7\text{ms}$ suggests a delay of 84.7ms in receiving the MJPEG images relative to the readout of pan/tilt measurements, which also seems reasonable.

As discussed, we should expect overly confident results for the focal length f with hard priors on β_ϕ and β_ψ, so there is no reason to expect the reported estimate to correspond to the true FOV of the camera. The MEPE across all landmarks was for both estimation runs 1.57px. However, the low MEPE suggests that we still find parameter estimates consistent with the estimated landmarks. Comparing the estimation results for base PTCEE vs soft PTCEE in Table IV, we see that f is poorly constrained when we only have soft priors for the βs. The predicted joint covariance (not reported) between f and the βs shows strong correlation between these estimates, which causes the reported marginal σ-values to be inflated. The other parameters seem to be consistently estimated regardless of the β-prior. In contrast to [8], we cannot conclude that there are evident scaling errors in the pan/tilt measurements. Our experiment shows that βs are poorly constrained and do not significantly differ from 1. This weak fit for the βs suggests that our setup with narrow FOV and no external calibration target is not suited to reveal such scaling bias in the pan/tilt measurements, which is consistent with the theoretical discussion from [10].

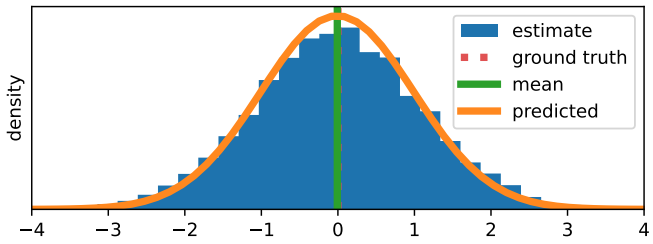
Using the last 10s (250 frames) of the dataset, we empirically evaluate the performance of our method operating in a total of four different modes (configurations) as follows. For each frame, we detect the april tag using AprilTag 3 [51] and then reproject it using each operating mode into a selected frame of the dataset. In order to not inflate the error in the uncalibrated case, we chose to reproject into frame 160 of the test dataset, which is centered around the april tag. Here, one must keep in mind that this reprojection effectively counts our model error twice: Once when backprojecting a pixel observation $\widetilde{\mathbf{u}}_{ij}$ from image i to a direction $\widetilde{\mathbf{d}}_{ij}^{b}$ in the base frame. And once more when projecting $\widetilde{\mathbf{d}}_{ij}^{b}$ back into image 160. As the baseline configuration, we use the initial guess calibration from Table IV together with (unsynchronized) pan/tilt measurements. For the PT mode, we use all the estimated calibration parameters, including focal length f, quadratic radial distortion k and RS line duration ℓ together with pan/tilt measurements corrected with the clock-offset d. For OE mode, we use the refined pan/tilt output of the frontend together with estimated calibration parameters, and compare using the estimated line duration ℓ to assuming $\ell = 0$.



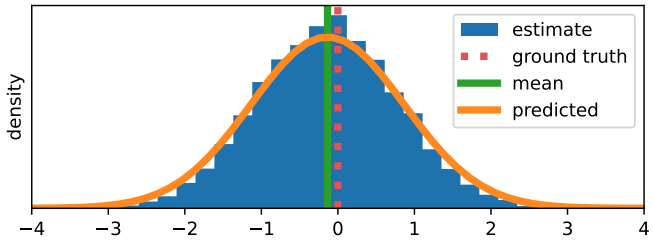
(a) $(\widehat{f} - f)/\widehat{\sigma}_f$



(b) $(\widehat{k} - k)/\widehat{\sigma}_k$



(c) $(\widehat{d} - d)/\widehat{\sigma}_d$



(d) $(\widehat{\ell} - \ell)/\widehat{\sigma}_\ell$

Fig. 11. Histograms of the estimation error in the base PTCEE backend-only experiment normalized by the predicted stddev. of the estimate. The dashed vertical line indicates the ground truth at zero. The solid vertical line indicates the empirical mean of the normalized estimation errors. The orange curve shows a normal distribution of unit variance around the estimated mean value.

The reprojections into frame 160 are shown in Fig. 16, and Table VI lists the std.dev. of the reprojections. Figure 16 shows the reprojections for each of these four modes, and the stddev. of the reprojections are listed in Table VI. Here we see that the full OE-based method achieves impressive reprojection accuracy. Seeing as the april detector in itself had stddev. of 1.26px on the stationary section of the dataset, the observed reprojection error of 1.57px (which counts the model error twice) shows that the model operating in the OE mode has sub-pixel accuracy. For OE mode, we see that the effect of including the estimate for RS line duration ℓ is also

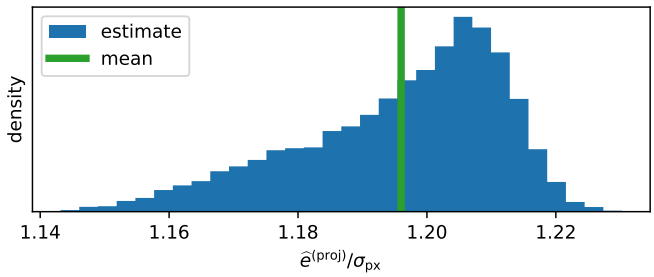


Fig. 12. Histogram over the mean estimated projection error $\widehat{e}^{(\mathrm{proj})}$ normalized by the nominal stddev. of the pixel measurement noise σ_{px} from each run of the base PTCEE backend-only experiment. Since σ_{px} is the lower bound for $\widehat{e}^{(\mathrm{proj})}$, 1 is the lower bound in this histogram. The green line indicates the average across all runs.

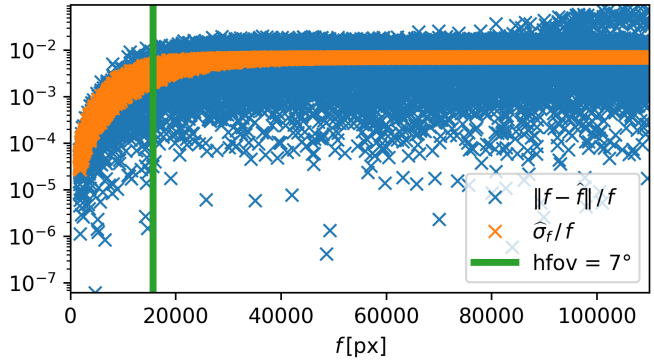


Fig. 13. The relative estimation error of the focal length f as a function of ground-truth f, from the soft PTCEE experiment. The relative estimation error appears to be constant for large focal lengths, but decreases drastically as $f \to 0$. The green line corresponding to 7° FOV is included for reference, as it approximately indicates the boundary between these two estimation behaviours.

Fig. 14. A typical image from the real-world dataset. The 50cm × 50cm april tag is approx. 550m away. The orange curve shows the distortion of a straight line with $k = 70$ at a horizontal FOV of 3.0°. The power pole next to the curve is straight in reality, and appears to be even more distorted than the curve in this image.

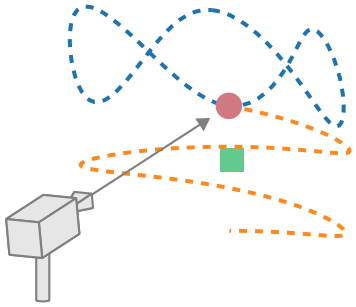


Fig. 15. The maneuvering pattern used in the real-world recording. The april tag was located approximately at the green square. For the first 10s (250 frames), the camera completes a full loop around the calibration pattern (blue curve). Then the camera stops at the red dot and remains stationary with the april tag in view for 2s (50 frames). Finally, the camera follows the orange curve, panning across the april tag while slowly tilting downwards for 10s (250 frames).

TABLE IV
INITIAL VALUES AND CALIBRATION RESULTS WITH PREDICTED MARGINAL STDDEV. FROM THE REAL-WORLD EXPERIMENTS. FOR SOFT PTCEE WE USED PRIORS WITH $\sigma = 0.1$ FOR BOTH β-PARAMETERS. THE INITIAL GUESS FOR f AND THE ESTIMATES CORRESPOND TO FOVS OF $3.0°$, $3.44°$ AND $3.36°$, RESPECTIVELY.

		Base PTCEE	**Soft PTCEE**
	Initial guess	**Estimate**	
f [px]	36661	32008 ± 24	32743 ± 2061
k	70	55.4 ± 0.11	57.9 ± 7.4
d [ms]	0	-84.4 ± 1.0	-84.1 ± 1.0
ℓ [μs]	0	-14.2 ± 0.07	-14.2 ± 0.07
β_ϕ	$1\,(\pm 0.1)$	1	1.02 ± 0.07
β_ψ	$1\,(\pm 0.1)$	1	1.06 ± 0.06

TABLE V
ESTIMATED OFFSETS FROM THE INITIAL GUESSES FOR THE PAN/TILT AXES IN THE REAL-WORLD EXPERIMENTS.

Initial guess			**Offset** [mrad]	
$\mathbf{a}_\phi$	$\mathbf{e}_z$	forward	-23.7 ± 0.82	-22.4 ± 0.73
		rightward	2.38 ± 1.8	4.64 ± 2.4
$\mathbf{a}_\psi$	$\mathbf{e}_y$	forward	4.92 ± 3.1	4.71 ± 3.2
		downward	16.4 ± 2.5	12.2 ± 2.2
			Base PTCEE	**Soft PTCEE**

considerable, with a 3x reduction in reprojection error, which should be even greater under faster camera maneuvers.

As expected, the PT mode, which uses the estimated calibration together with time-synchronized pan/tilt measurements outperforms using the initial guess calibration and raw pan/tilt measurements. However, the reprojection errors for this method are still surprisingly large and seem to have some structure. This structure indicates that our model fails to capture some properties of the PTZ camera, forcing the optimization to tweak the landmarks to account for the remaining variation. One possible explanation is that the camera has insufficiently square pixels, rendering our model with a single focal length parameter f inadequate. The difference in the estimates for β_ϕ and β_ψ in Table IV might also suggest non-square pixels, as these parameters would potentially be strongly correlated with an f_x and f_y, respectively.

The estimates for the pan/tilt axes are shown in Table V. Based on this, the pan and tilt axes are respectively rotated $\sim 12\text{mrad}$ backwards and $\sim 14\text{mrad}$ downward, relative to the camera at $(0, 0)$ pan/tilt. The predicted σ-values in Table V indicate that the estimated axes differ significantly from the initial guesses. However, when running the estimation with hard priors for both axes and βs (not reported), we see little change in reprojection error on the april tag experiment. Further investigation is therefore needed to conclude whether non-square pixels (or some other unmodeled effect) deludes our method into believing the camera is slantly mounted relative to the pan/tilt axes.

We also evaluated the running time of the full setup on a desktop CPU, a Jetson Xavier NX and a Raspberry Pi 4, with results shown in Table VII. The OE mode, which uses the frontend to refine the pan/tilt state for every frame, runs at $> 100\text{fps}$ on FullHD images on the embedded Jetson Xavier NX in low power mode. This performance should be sufficient for most real-time applications. The per-frame processing time of 1.7ms on the desktop CPU should also be adequate for most latency-sensitive applications.

VII. CONCLUSION AND DISCUSSION

We have proposed PTCEE (pan/tilt camera extrinsic and intrinsic estimation): A real-time method for self-calibrating a pan-tilt-zoom (PTZ) camera system model specifically tailored to handle narrow field-of-view (FOV) operations with maneuvering low-cost PTZ cameras. Self-calibration for rotation-only cameras is fundamentally limited by the diminishing observability of the focal length as the FOVs becomes small [10]. We overcome this limitation by including angular information from the pan/tilt measurements in the calibration. However, low-cost PTZ cameras lack clock synchronization between the pan/tilt-unit and the image sensor, have imperfect pan/tilt mechanics, and use rolling shutter (RS) image sensors. Through a rotation-only bundle adjustment (BA), we perform self-calibration of camera intrinsics, rolling shutter parameter and pan/tilt mechanics, in addition to synchronizing the video feed and the pan/tilt unit. The resulting model maps pixels to directions in the platform frame with pixel-level precision for each frame in the video stream. Our method relies only on timestamped images and pan/tilt measurements and does not require special calibration targets.

In our initial simulation study, we demonstrated that existing methods using only observations from images fail to estimate focal length even in simplified narrow FOV scenarios. Meanwhile, PTCEE, which also includes pan/tilt measurements, performed well. Expanding the simulation to include distortion, rolling shutter and imperfect pan/tilt mechanics, we showed that PTCEE provides highly precise calibration estimates and landmark maps when used with a good frontend. By pairing the calibration with orientation estimation (OE) that exploits the landmark map, the resulting full PTZ camera model can achieve sub-pixel orientation precision in the platform frame.

Under challenging conditions, we also demonstrated pixel-level precision on real-world data, while using a fast but simple frontend. Our choice of using a semi-direct frontend is in contrast to existing PTZ and RS calibration methods,

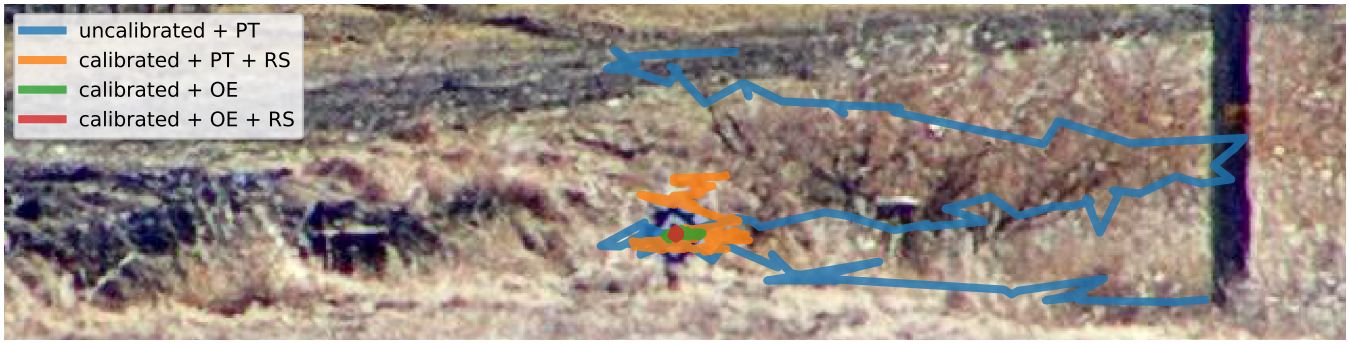


Fig. 16. Reprojection results on the april tag test for four different modes of our method. For each frame, we use each mode to reproject the detected april tag back into the pictured frame. The "PT"-modes lookup orientation for each frame using pan/tilt measurements, while the "OE"-modes use the orientation estimation based on landmark observations discussed in Section V-C. "RS"-modes leverage the estimated ℓ to correct for rolling shutter effects, while the other modes assume global shutter.

TABLE VI
REPROJECTION RESULTS FROM THE APRIL TAG TEST. THE "UNCALIBRATED" RUN USES THE INITIAL GUESSES FOR f AND k, AND WITH $d = \ell = 0$. THE OTHER RUNS USE THE OPTIMIZED VALUES, EXCEPT FOR THE "CALIBRATED + OE" RUN, WHICH USES $\ell = 0$.

	Reproj stddev. [px]	
Uncalibrated + PT	101	101
Calibrated + PT + RS	16.8	15.2
Calibrated + OE	4.20	4.18
Calibrated + OE + RS	**1.24**	**1.32**
	Base PTCEE	**Soft PTCEE**

TABLE VII
TIMING RESULTS FOR THE APRIL TAG TEST. THE FIRST COLUMN REPORTS THE TIME SPENT PREPROCESSING AND OPTIMISING THE FACTOR GRAPH, WHILE THE SECOND COLUMN REPORTS THE TIME SPENT ESTIMATING ORIENTATION IN THE EVALUATION SECTION OF THE DATASET.

	Calib (125 frames)	**OE (per frame)**
Intel i5 12600K	1.14s	$1.79\text{ms} \pm 0.5\text{ms}$
Jetson Xavier NX (20W)	4.21s	$7.14\text{ms} \pm 1.9\text{ms}$
Jetson Xavier NX (10W)	5.29s	$8.82\text{ms} \pm 2.4\text{ms}$
Raspberry Pi 4B	11.3s	$14.3\text{ms} \pm 4.4\text{ms}$

which use keypoint descriptor-based frontends [1], [8], [34], [35]. The semi-direct approach alleviates the need to develop minimal solvers to be used in RANSAC schemes [43], and enables us to enforce geometric constraints through simple forward models.

On cameras where the pan/tilt measurements are suspected to have (an unknown) scaling bias, our simulation experiments indicate that the observability of focal length at narrow FOV is once again lost. In such cases PTCEE loses *absolute* accuracy, but still estimates a *consistent* calibration and map of landmarks. Since the estimated model accounts for clock synchronization and rolling shutter compensation, the method remains well suited for tracking moving targets during pan/tilt maneuvers, which requires self-consistent calibration and OE. To provide better absolute accuracy in such scenarios, some type of external measurement, such as known geometry or multiple views, must be included. The factor graph formulation of our method makes incorporating these types of external information sources straightforward.

A natural improvement to the proposed backend is to include a model for varying zoom levels. While this can be easily integrated into the factor graph formulation, it is beyond the scope of this paper. Similarly, expanding the intrinsic model to perform photometric calibration as in [52] seems like a useful addition for consumer-grade cameras. Our real-world tests revealed that our method would benefit from a frontend which is more robust during initialization. Since the frontend performs no estimation of RS or distortion, the first images are processed under the assumption of an undistorted global shutter camera, until the first BA estimates are available. Ideally, the frontend should handle severe RS effects and motion blur during fast maneuvers from the beginning. Additionally, a good frontend should handle scenes where large parts of the background are non-stationary, such as trees blowing in the wind, waves in coastal areas, or moving clouds.

APPENDIX A
ERROR METRICS USED ON SIMULATION RESULTS

For a given parameter, let θ_i denote its ground truth value at time i, $\widehat{\theta}_i$ its estimated value, and $\widehat{\sigma}_i$ the predicted stddev. of the estimate. We compute the mean absolute error (MAE) as

$$\mathrm{MAE}(\widehat{\theta}) \triangleq \frac{1}{n}\sum_{i=1}^{n} |\widehat{\theta}_i - \theta_i|, \tag{48}$$

and the mean relative error (MRE) as

$$\mathrm{MRE}(\widehat{\theta}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \frac{|\widehat{\theta}_i - \theta_i|}{\theta_i}. \tag{49}$$

We compute the average normalized estimation error squared (ANEES) as

$$\mathrm{ANEES}(\widehat{\theta}) \triangleq \frac{1}{n}\sum_{i=1}^{n} \frac{(\widehat{\theta}_i - \theta_i)^2}{\widehat{\sigma}_i^2}. \tag{50}$$

TABLE VIII
TO-BE-ESTIMATED GROUND TRUTH PARAMETERS USED IN THE NARROW FOV SIMULATED EXPERIMENT. THE CLOCK OFFSET IS SAMPLED UNIFORMLY AS GIVEN BELOW FOR EACH SIMULATION RUN.

Focal length	f	$f(\{32°, 16°, 8°, 4°, 2°, 1°\})$px
Pan/tilt vs frame clock offset	d	$[-100, 100]$ms

TABLE IX
A PRIORI KNOWN PARAMETERS USED IN THE NARROW FOV SIMULATED EXPERIMENT.

Frame rate	$r^{(img)}$	12.5Hz
Pan/tilt rate	$r^{(pt)}$	30Hz
Pan axis	$\mathbf{a}_\phi$	$\mathbf{e}_z$
Pan measurement scale	β_ϕ	1
Tilt axis	$\mathbf{a}_\psi$	$\mathbf{e}_y$
Tilt measurement scale	β_ψ	1
Quadratic radial distortion	k	0
RS line duration	ℓ	0
Focal length initial guess		$\left[\frac{2}{3}f, \frac{3}{2}f\right]$

APPENDIX B
DETAILS FOR THE NARROW FOV EXPERIMENT

We simulate the PTZ camera in a scene with landmarks distributed in a uniform azimuth/elevation grid on the unit sphere, as illustrated in Fig. 9. In this experiment, we assume a global shutter FullHD camera with no distortion and perfect pan/tilt mechanics, corresponding to the parameters given in Table IX. The noise parameters are chosen to be fairly realistic, but also to favor the existing image-only methods over our proposed method PTCEE. Specifically, we have chosen a low pixel noise on landmark observations, which is beneficial to the image-only methods. Also, we use a higher-than-typical noise on pan/tilt measurements and timestamps, which only (negatively) affects PTCEE.

For each ground-truth FOV of $32°$, $16°$, $8°$, $4°$, $2°$ and $1°$, we perform 128 simulation runs. In each run we simulate images at the given rate from $t = 0$s to $t = 10$s. To enable interpolating pan/tilt measurements around each image, we generate pan/tilt measurements in a slightly larger interval from $t = -1$ to $t = 11$. We also draw clock offset d as given in Table VIII, which of course only affects PTCEE.

In each simulation run we provide PTCEE an initial guess for f, randomly drawn as specified in Table X. The DLT-based methods do not require an initial guess for f in the BA, as they generate one through the DLT. To avoid any advantage for PTCEE, we use the closer value to the ground truth between the DLT estimate and the randomly drawn guess for f as the input to the BA for the existing methods. Meanwhile, PTCEE is always given the randomly drawn guess.

APPENDIX C
DETAILS FOR BACKEND-ONLY EXPERIMENT

In the backend-only experiment, we use the same setup as described in Appendix B, but also include quadratic radial distortion k, RS line duration ℓ, pan/tilt axes and pan/tilt measurement scale (for soft PTCEE). Both for base PTCEE and soft PTCEE we perform a total of 10,000 Monte-Carlo simulation runs. Instead of using fixed system parameters as in Appendix B, we sampled parameters for each simulation run as described in Tables XII and XIII. Additionally, the initial guesses assume zero clock offset d, zero quadratic radial distortion k, zero RS line duration ℓ, perfect pan/tilt axes, and

TABLE X
PARAMETERS USED FOR SIMULATION NOISE IN THE NARROW FOV SIMULATED EXPERIMENT.INITIAL GUESS FOR FOCAL LENGTH IS DRAWN UNIFORMLY, WHILE ALL OTHER NOISE IS DRAWN FROM RESPECTIVE ZERO-MEAN NORMAL DISTRIBUTIONS.

Pixel noise	σ_{px}	0.5px
Pan/tilt noise	σ_{pt}	1mrad
Frame timestamp noise	$\sigma_{\text{t}}^{(img)}$	5ms
Pan/tilt timestamp noise	$\sigma_{\text{t}}^{(pt)}$	5ms
Frame timestep noise	$\sigma_{\text{dt}}^{(img)}$	0.1ms
Pan/tilt timestep noise	$\sigma_{\text{dt}}^{(pt)}$	0.1ms

TABLE XI
TO-BE-ESTIMATED GROUND TRUTH PARAMETERS USED IN THE BACKEND-ONLY SIMULATED EXPERIMENT.EACH PARAMETER IS SAMPLED UNIFORMLY AS GIVEN BELOW FOR EACH SIMULATION RUN.(*) IN THE HARD β EXPERIMENT WE USE $\beta_\phi = \beta_\psi = 1$, WHILE THE SOFT β EXPERIMENT USES THE SAMPLED VALUE.

Focal length	f	$[f(1°), f(60°)]$px
Quadratic radial distortion	k	$[-0.3, 0.3]$
Pan/tilt vs frame clock offset	d	$[-100, 100]$ms
RS line duration	ℓ	$[0, 1.85$us$]$
Pan axis	$\mathbf{a}_\phi$	$\mathbf{e}_z \oplus [-50\text{mrad}, 50\text{mrad}]^2$
Tilt axis	$\mathbf{a}_\psi$	$\mathbf{e}_y \oplus [-50\text{mrad}, 50\text{mrad}]^2$
Pan measurement scale	β_ϕ	1 or* $[0.98, 1.02]$
Tilt measurement scale	β_ψ	1 or* $[0.98, 1.02]$

TABLE XII
PARAMETERS USED FOR SIMULATION NOISE IN THE BACKEND-ONLY SIMULATED EXPERIMENT.FOR EACH SIMULATION RUN, THE NOISE PARAMETERS ARE DRAWN UNIFORMLY AS GIVEN BELOW.THE CORRESPONDING *noise* IS LATER DRAWN FROM RESPECTIVE ZERO-MEAN NORMAL DISTRIBUTIONS.

Pixel noise	σ_{px}	$[0.2, 0.5]$px
Pan/tilt noise	σ_{pt}	$[0.01, 0.1]$mrad
Frame timestamp noise	$\sigma_{\text{t}}^{(img)}$	$[0.1, 5]$ms
Pan/tilt timestamp noise	$\sigma_{\text{t}}^{(pt)}$	$[0.1, 5]$ms
Frame timestep noise	$\sigma_{\text{dt}}^{(img)}$	$[0.01, 0.1]$ms
Pan/tilt timestep noise	$\sigma_{\text{dt}}^{(pt)}$	$[0.01, 0.1]$ms

TABLE XIII
A PRIORI KNOWN PARAMETERS USED IN THE BACKEND-ONLY SIMULATED EXPERIMENT. EACH PARAMETER IS DRAWN UNIFORMLY AS GIVEN BELOW FOR EACH SIMULATION RUN.

Frame rate	$r^{(img)}$	$[10, 30]$Hz
Pan/tilt rate	$r^{(pt)}$	$[3 \cdot r^{(img)}, 100]$Hz
Focal length initial guess		$\left[\frac{2}{3}f, \frac{3}{2}f\right]$

perfect (unit) pan/tilt measurement scale. The ground truth values for the to-be-estimated parameters were sampled as described in Table XI.

REFERENCES

[1] G. Lisanti, I. Masi, F. Pernici, and A. Del Bimbo, "Continuous localization and mapping of a pan–tilt–zoom camera for wide area tracking," *Machine Vision and Applications*, vol. 27, no. 7, pp. 1071–1085, 10 2016.

[2] P. Salvagnini, F. Pernici, M. Cristani, G. Lisanti, I. Masi, A. Del Bimbo, and V. Murino, "Information theoretic sensor management for multi-target tracking with a single pan-tilt-zoom camera," in *2014 IEEE Winter Conference on Applications of Computer Vision, WACV 2014*. IEEE Computer Society, 2014, pp. 893–900.

[3] R. P. S. Mahler, *Advances in statistical multisource-multitarget information fusion*. Artech House, 2014.

[4] R. I. Hartley and G. E. Crd, "Self-Calibration of Stationary Cameras *," *ICA International Journal of Computer Vision International Journal of Computer Vision*, vol. 22, no. 1, pp. 5–23, 1997.

[5] B. Triggs, "Autocalibration and the absolute quadric," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Comput. Soc, 1997, pp. 609–614. [Online]. Available: http://ieeexplore.ieee.org/document/609388/

[6] L. d. Agapito, E. Hayman, and I. Reid, "Self-Calibration of a Rotating Camera with Varying Intrinsic Parameters," *BMVC*, pp. 1–10, 1998.

[7] S. N. Sinha and M. Pollefeys, "Pan-tilt-zoom camera calibration and high-resolution mosaic generation," *Computer Vision and Image Understanding*, vol. 103, no. 3, pp. 170–183, 9 2006.

[8] Z. Wu and R. J. Radke, "Keeping a pan-tilt-zoom camera calibrated," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1994–2007, 2013.

[9] J. Davis and X. Chen, "Calibrating pan-tilt cameras in wide-area surveillance networks," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 1. Institute of Electrical and Electronics Engineers Inc., 2003, pp. 144–149.

[10] L. Agapito, E. Hayman, and I. Reid, "Self-Calibration of Rotating and Zooming Cameras," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 107–127, 2001.

[11] M. V. Larsen, T. Haavardsholm, and K. Mathiassen, "In-operation calibration of clock-bias and intrinsic parameters for pan-tilt-zoom cameras based on keypoint tracking," *Electro-Optical Remote Sensing XIV*, vol. 11538, no. September 2020, p. 11, 2020.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," *Icra14*, 2014.

[13] R. Tsai, "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 8 1987. [Online]. Available: http://ieeexplore.ieee.org/document/1087109/

[14] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000. [Online]. Available: http://ieeexplore.ieee.org/document/888718/

[15] O. D. Faugeras, Q. T. Luong, and S. J. Maybank, "Camera self-calibration: Theory and experiments," in *European Conference on Computer Vision '92*, 1992, pp. 321–334. [Online]. Available: http://link.springer.com/10.1007/3-540-55426-2_37

[16] R. I. Hartley, "Self-calibration from multiple views with a rotating camera," in *European Conference on Computer Vision '94*, 1994, pp. 471–478. [Online]. Available: http://link.springer.com/10.1007/3-540-57956-7_52

[17] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 3 2004. [Online]. Available: https://www.cambridge.org/core/product/identifier/9780511811685/type/book

[18] J. Lu, J. Chen, and J. J. Little, "Pan-tilt-zoom SLAM for Sports Videos," *arxiv preprint*, 7 2019. [Online]. Available: http://arxiv.org/abs/1907.08816

[19] Y. Liu and H. Zhang, "Linear Auto-calibration of Pan-Tilt-Zoom Cameras With Rotation Center Offset," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2023-May. Institute of Electrical and Electronics Engineers Inc., 2023, pp. 11 461–11 467.

[20] C. Zhang, F. Rameau, J. Kim, D. M. Argaw, J.-C. Bazin, and I. S. Kweon, "DeepPTZ: Deep Self-Calibration for PTZ Cameras," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 3 2020, pp. 1030–1038. [Online]. Available: https://ieeexplore.ieee.org/document/9093629/

[21] J. Chen, F. Zhu, and J. J. Little, "A Two-Point Method for PTZ Camera Calibration in Sports," *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*, vol. 2018-Janua, pp. 287–295, 2018.

[22] J. Chen and J. J. Little, "Sports camera calibration via synthetic data," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2019-June. IEEE Computer Society, 6 2019, pp. 2497–2504.

[23] L. Sha, J. Hobbs, P. Felsen, X. Wei, P. Lucey, and S. Ganguly, "End-to-End Camera Calibration for Broadcast Videos," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 6 2020, pp. 13 624–13 633. [Online]. Available: https://ieeexplore.ieee.org/document/9157432/

[24] K. Mao, Y. Xu, R. Wang, and S. Pan, "A General Calibration Method for Dual-PTZ Cameras Based on Feedback Parameters," *Applied Sciences (Switzerland)*, vol. 12, no. 18, 9 2022.

[25] R. Wang, R. Huang, and J. Yang, "Facilitating PTZ camera auto-calibration to be noise resilient with two images," *IEEE Access*, vol. 7, pp. 155 612–155 624, 2019.

[26] M. Kim, S. Kim, and J. Choi, "Robust and incremental stitching and calibration with known rotation on pan-tilt-zoom camera," in *2013 IEEE International Conference on Image Processing*. IEEE, 9 2013, pp. 2247–2251. [Online]. Available: http://ieeexplore.ieee.org/document/6738463/

[27] Frahm and Koch, "Camera calibration with known rotation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, vol. 5, no. 3. IEEE, 9 2003, pp. 1418–1425. [Online]. Available: http://ieeexplore.ieee.org/document/1238656/

[28] P. Furgale, J. Rehder, and R. Siegwart, "Unified temporal and spatial calibration for multi-sensor systems," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1280–1286, 2013.

[29] M. Meingast, C. Geyer, and S. Sastry, "Geometric Models of Rolling-Shutter Cameras," *arxiv preprint*, 3 2005. [Online]. Available: http://arxiv.org/abs/cs/0503076

[30] L. Oth, P. Furgale, L. Kneip, and R. Siegwart, "Rolling shutter camera calibration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1360–1367.

[31] B. Zhuang, L.-F. Cheong, and G. Hee Lee, "Rolling-Shutter-Aware Differential SfM and Image Rectification," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[32] E. Ringaby and P. E. Forssén, "Efficient video rectification and stabilisation for cell-phones," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 335–352, 2 2012.

[33] J. Hedborg, P.-E. Forssen, M. Felsberg, and E. Ringaby, "Rolling shutter bundle adjustment," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 6 2012, pp. 1434–1441. [Online]. Available: http://ieeexplore.ieee.org/document/6247831/

[34] Z. Kukelova, C. Albl, A. Sugimoto, K. Schindler, and T. Pajdla, "Minimal Rolling Shutter Absolute Pose with Unknown Focal Length and Radial Distortion," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 698–714. [Online]. Available: https://link.springer.com/10.1007/978-3-030-58558-7_41

[35] F. Bai, A. Sengupta, and A. Bartoli, "Scanline Homographies for Rolling-Shutter Plane Absolute Pose," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[36] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," in *2012 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 4 2012, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/6215213/

[37] J. Solà, J. Deray, and D. Atchuthan, "A micro Lie theory for state estimation in robotics," *arXiv preprint arXiv:1812.01537*, 2018. [Online]. Available: http://arxiv.org/abs/1812.01537

[38] F. Dellaert and M. Kaess, "Factor Graphs for Robot Perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[39] F. Dellaert, "Derivatives And Differentials," 2020. [Online]. Available: https://github.com/borglab/gtsam/blob/develop/doc/math.pdf

[40] GTSAM, "GTSAM." [Online]. Available: http://gtsam.org

[41] E. Hayman and D. W. Murray, "The effects of translational misalignment when self-calibrating rotating and zooming cameras," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 1015–1020, 2003.

[42] P. Furgale, C. H. Tong, T. D. Barfoot, and G. Sibley, "Continuous-time batch trajectory estimation using temporal basis functions," *International Journal of Robotics Research*, vol. 34, no. 14, pp. 1688–1710, 2015.

[43] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect Visual Odometry for Monocular and Multicamera

Systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2017.

[44] R. Mur-Artal, J. M. Montiel, and J. D. Tardos, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 10 2015.

[45] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 10 2017.

[46] C. Campos, R. Elvira, J. J. Rodriguez, J. M. Montiel, and J. D. Tardos, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[47] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1689–1696, 2020.

[48] V. Usenko, N. Demmel, D. Schubert, J. Stuckler, and D. Cremers, "Visual-Inertial Mapping with Non-Linear Factor Recovery," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 422–429, 2020.

[49] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *Computer Vision – ECCV 2006*, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443.

[50] Z. Zhang, "Parameter estimation techniques: a tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, 1 1997. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0262885696011122

[51] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible Layouts for Fiducial Tags," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1898–1903, 2019.

[52] J. Engel, V. Koltun, and D. Cremers, "Direct Sparse Odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 3 2018.

Martin Vonheim Larsen is a scientist and PhD student at the Norwegian Defence Research Establishment (FFI). He received his M.S. in mathematics from the University of Oslo (UiO) in 2015. After starting as a scientist at FFI in 2016, he has since been working with computer vision and target tracking for autonomous systems. Applications range from using radar/lidar/camera fusion to prevent the unmanned surface vessel Odin from running ashore, to providing situational awareness for counter-UAS.

Kim Mathiassen received a Master's degree in engineering cybernetics from the Norwegian University of Science and Technology in 2010 and a PhD in robotics from the University of Oslo in 2017. His PhD thesis was on a semi-autonomous system for diagnostics and treatment using medical ultrasound. In 2015 he started working at FFI and is currently Research Manager for the Combat vehicles and dismounted soldiers research program. In addition to his research position, he is an associate professor at the University of Oslo, teaching advanced robotics.