2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

TTCDist: Fast Distance Estimation From an Active Monocular Camera Using Time-to-Contact

Levi Burner[1], Nitin J. Sanket[2], Cornelia Fermüller[3], Yiannis Aloimonos[3]

***Abstract*— Distance estimation from vision is fundamental for a myriad of robotic applications such as navigation, manipulation, and planning. Inspired by the mammal's visual system, which gazes at specific objects, we develop two novel constraints relating time-to-contact, acceleration, and distance that we call the τ-*constraint* and Φ-*constraint*. They allow an active (moving) camera to estimate depth efficiently and accurately while using only a small portion of the image. The constraints are applicable to range sensing, sensor fusion, and visual servoing.**

We successfully validate the proposed constraints with two experiments. The first applies both constraints in a trajectory estimation task with a monocular camera and an Inertial Measurement Unit (IMU). Our methods achieve 30-70% less average trajectory error while running 25$\times$ and 6.2$\times$ faster than the popular Visual-Inertial Odometry methods VINS-Mono and ROVIO respectively. The second experiment demonstrates that when the constraints are used for feedback with efference copies the resulting closed loop system's eigenvalues are invariant to scaling of the applied control signal. We believe these results indicate the τ and Φ constraint's potential as the basis of robust and efficient algorithms for a multitude of robotic applications.

SUPPLEMENTARY MATERIAL

The experimental data, code, extended proofs, and video are available at prg.cs.umd.edu/TTCDist.

I. INTRODUCTION

Early researchers in computer vision were fascinated by living beings' ability to control their movement in order to gather information about their environment. The process was named "Active Perception" [1]–[6], and numerous mechanisms for utilizing activeness have been developed since. In this work, we focus on the active process of fixation based time-to-contact estimation and using it to measure distance.

Because active vision systems are usually moving in some way, they must accelerate to produce changes to this movement. Despite this, utilizing observer acceleration (measured using an Inertial Measurement Unit or IMU) to facilitate visual computations has not received much attention in the Active Vision literature. To fill this gap, we introduce two mathematical constraints relating (a) time-to-contact (τ) (the ratio of camera velocity to scene distance), (b) the relative size of a planar patch over time, with (c) scene depth and (d) observer acceleration. These constraints make it possible for an active monocular camera to estimate scene depth by accelerating in any direction. We call these constraints the τ-constraint and the Φ-constraint.

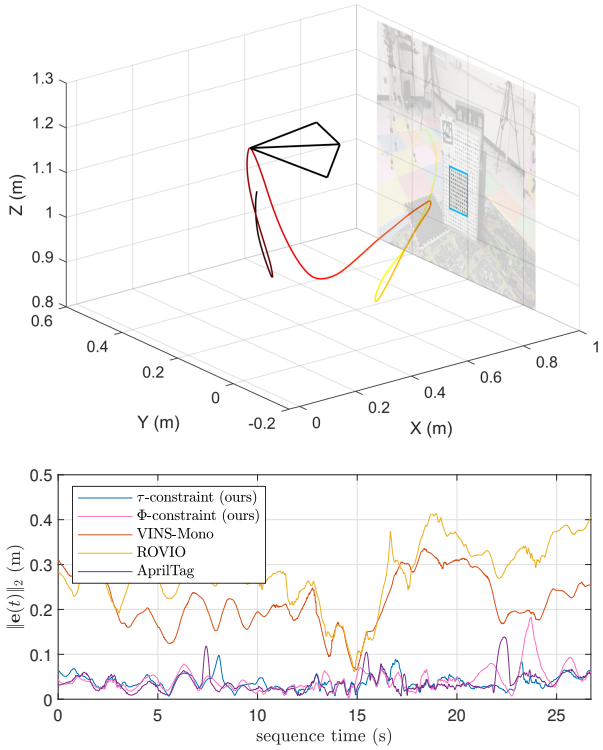

Fig. 1: Top: Five seconds of Sequence 9's camera trajectory along with the fixated scene patch used to estimate 3D distance using a monocular camera and an IMU. Sequence time is color coded with the hot colormap. Bottom: Instantaneous Euclidean error for Sequence 9 of our methods as well as VINS-Mono, ROVIO, and AprilTag 3. Ground truth is measured by a motion capture system.

We demonstrate the utility of the constraints in a series of experiments involving fixation on a single object (tracking) while a monocular camera accelerates. The method is naturally suited for object-centered representations which have been argued to be useful for a variety of tasks [3], [7]–[9].

Further, we demonstrate a novel mathematical property of the constraint. When efference copies of the control signal (defined in psychology as an internal copy of the movement producing signal) are used in the place of acceleration in the τ or Φ constraint, the closed loop dynamics become invariant to scaling of the applied control signal. Informally speaking,

The support of the NSF under awards DGE-1632976 and OISE 2020624 and the USDA NIFA sustainable agriculture system program under award number 20206801231805 is gratefully acknowledged.

[1] Corresponding author. Perception and Robotics Group, Electrical and Computer Engineering, University of Maryland, College Park, lburner@umd.edu

[2] Robotics Engineering, Worcester Polytechnic Institute nsanket@wpi.edu

[3] Perception and Robotics Group, University of Maryland Institute for Advanced Computer Studies, University of Maryland, College Park {fer, jyaloimo}@umiacs.edu

this means the weight of the robot, or the strength of the motors, can change without influencing stability.

The key concept of fusing inertial measurements with camera observations has been extensively studied in the fields of Structure from Motion (SfM) and Simultaneous Localization and Mapping (SLAM). Traditionally, this fusion has been achieved in Visual Inertial Odometry (VIO) frameworks using a series of Bayesian filters or Factor graphs [10], [11].

It is important to stress that our implementation is not a fully-functional VIO implementation, and is only intended to empirically demonstrate the efficacies of the τ and Φ constraints for absolute position estimation and control. Nevertheless, our constraints naturally result in a trajectory estimate. For this reason, we compare our method with the popular state-of-the-art VIO methods such as VINS-Mono [12] and ROVIO [13] as well as the fiducial marker based pose estimation method, AprilTag 3 [14]. We also compare our results with millimeter accurate ground truth from a Vicon motion capture system.

A list of our contributions follows:

- A closed form solution for the 3D position of the camera given time-to-contact or depth-to-scale and acceleration.
- A computationally efficient position estimation method utilizing the τ or Φ constraint based on the closed form.
- Comparisons against the popular VIO methods, VINS-Mono and ROVIO, as well as AprilTag 3, to show the efficacy of the novel constraint in real-world settings.
- A corollary (resulting from the closed form) and experiment showing that our constraints, when used with efference copies in a closed loop, make the system invariant to scaling of the control signal.

II. Related Work

A multitude of prior works from both the computer vision and robotics literature deal with time-to-contact and the fusion of camera and IMU measurements to obtain relative camera pose (odometry). However, to the best of our knowledge, none of these works provided closed form solutions for the estimation of distance from time-to-contact and inertial measurements nor for distance from the apparent size of a tracked planar patch and inertial measurements.

Time-to-contact: One of the earliest works to discuss how time-to-contact τ can be used in control tasks came from psychology. [15] provided an analysis of how time-to-contact, τ, obtained from vision, could be used by drivers to control braking a vehicle. The idea was generalized to other perception modalities into a "General Tau Theory" in [16].

Because of its intuitive formulation, τ has also been the subject of many studies in robotics. [17] showed how to land a spacecraft using event cameras by computing τ from the divergence of optical flow. [18] fuses information from a depth camera and τ to compute "time-to-impact" which can in-turn be used to dodge dynamic obstacles. In robotics, most methods that perform optical flow based control use initial height estimates either implicitly or explicitly, as remarked in [19]. To this end, [19] proposed to fuse control effort and time-to-contact with an extended Kalman filter which estimated depth. Another strategy exploits the instability that manifests at certain heights when performing direct τ control with fixed gain feedback to estimate depth [20]. In the context of self-driving cars, BinaryTTC [21] proposed a network to predict per-pixel τ. Recently, EV-Catcher caught fast-moving objects using a network to predict time-to-contact and the current position of a moving object [22]. Finally, it is important to note that τ can be efficiently computed when the scene under consideration is planar [23].

Visual Inertial Odometry (VIO): One of the earliest works for real-time VIO fused sparse feature tracks with IMU measurements using a Multi-State Constraint Kalman Filter [24]. This pipeline was made more robust by ROVIO [13] which fused photometric consistency of patches with IMU measurements in an Iterative Extended Kalman Filter. ROVIO relies on tracking several planar (affine warpable) patches distributed across the field of view and uses approximations to implement the iterated EKF. To this end, VINS-Mono [12] introduced a point-based nonlinear sliding window estimator with an initialization-free formulation where the points are assumed to be distributed across the field of view. These methods are supported by a proof that tracked points and inertial measurements allow pose to be recovered in closed form [25], [26]. In contrast, our constraints admit closed form solutions for distance to a single patch of unknown size.

Recent deep learning based VIO methods have achieved better accuracy than classical approaches. VINet [27] presented the first supervised method to estimate VIO using a CNN + LSTM architecture. This was later improved by DeepVIO [28] using supervision from a stereo camera. The limiting factor of these approaches is a lack of speed and generalization across various compute platforms [29]. As will be shown, the τ-constraint can be formulated as a loss, which may be interesting for future work in deep VIO. We will present a derivation of the proposed constraints next.

III. Derivation of the τ and Φ constraint

We develop a system to use the Φ and τ constraints with a calibrated monocular camera and an aligned 6-DoF IMU that measures acceleration and angular velocity. Referring to Fig. 2, we compose the incoming frames with a rotational warp function from the IMU. Then we "fixate on" (track) a planar patch in the rotation-compensated images using an affine homography. A history of the parameters of the affine homography are then used to estimate distance with our proposed Φ and τ constraints. Finally, we filter the predictions using a Luenberger observer (explained in Sec. IV) to obtain the final trajectory estimates.

In the following derivation, we first define frequency of contact, $\mathbf{F}$, and explain its relation to Φ. Then we show how $\mathbf{F}$ and Φ can be measured from an affine homography. Finally, we related $\mathbf{F}$ and Φ to acceleration and depth which results in the τ and Φ constraints respectively.

A. Definition of τ, frequency-of-contact $\mathbf{F}$, and Φ

Time-to-contact or τ is defined as $Z/\dot{Z}$ (ratio of depth/distance to velocity). Below, we generalize τ to all three dimensions and define frequency of contact as

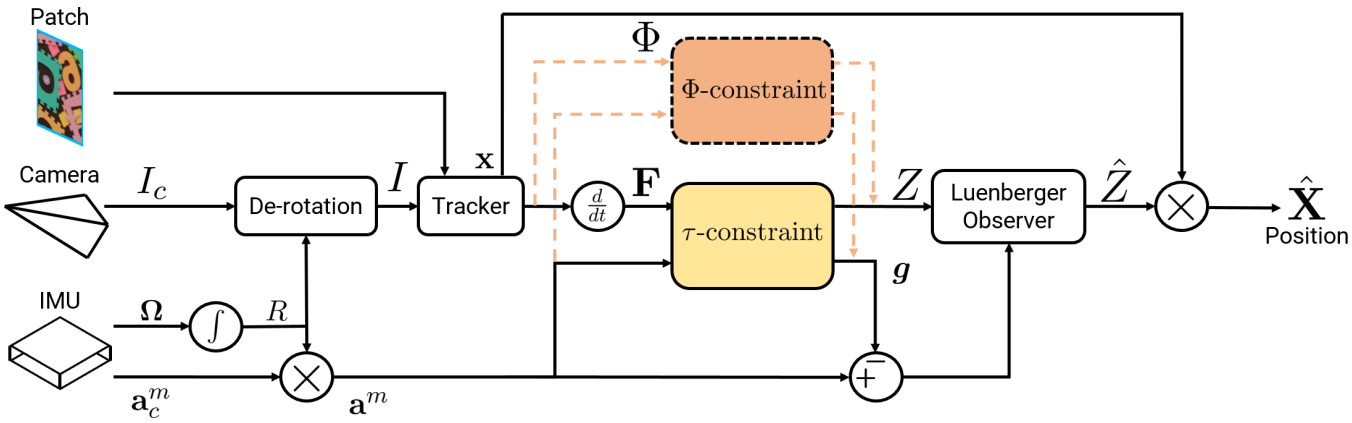

Fig. 2: System overview of our method to estimate camera position using the τ-constraint. The Φ-constraint uses an identical process except that the derivative of Φ is not taken. Only one constraint is used at a time as indicated by the dashed arrows and outlines.

$$\mathbf{F} := \frac{\dot{\mathbf{X}}}{Z}. \tag{1}$$

$\mathbf{X} = [X, Y, Z]^T$ is the position of a point as in Fig. 3.

Frequency of contact is the number of times per second a constant speed point will reach an axis. Unlike time-to-contact, frequency-of-contact is only ill-defined when $Z = 0$, which just means the object cannot be seen. The third component of $\mathbf{F}$, F_Z, is equal to $1/\tau$.

Frequency-of-contact is directly related to Φ (depth and translation to scale). To show this, consider that $\mathbf{F}$ defines the linear time varying system $\dot{\mathbf{X}} = \mathbf{F}Z$ which has the solution

$$\mathbf{X}(t) = \underbrace{\begin{bmatrix} 1 & 0 & \int_0^t F_X(\lambda)\Phi_{F_Z}(\lambda)d\lambda \\ 0 & 1 & \int_0^t F_Y(\lambda)\Phi_{F_Z}(\lambda)d\lambda \\ 0 & 0 & \Phi_{F_Z}(t) \end{bmatrix}}_{\Phi(t):=} \mathbf{X}_0, \tag{2}$$

where Φ_{F_Z} is the solution to the ODE $\dot{Z} = F_Z Z$ when $Z_0 = 1$, i.e. $\Phi_{F_Z}(t) = \exp\left(\int_0^t F_Z(\lambda)d\lambda\right)$ [30].

Next, we use a tracked planar patch to estimate $\mathbf{F}$ and Φ.

B. Estimating τ and Φ from an Affine Homography

As illustrated in Fig. 3, a point in a scene $\mathbf{X}$ is projected to a point on the image $\mathbf{x}$ according to the pinhole model $\mathbf{x} = \mathbf{X}/Z$. Without loss of generality, we assume that the camera intrinsic matrix K is identity. If we are tracking the points on a plane that is parallel to the imaging plane and the camera translates without rotating, then an affine homography relates all points on the planar patch in the current frame to their positions in the first frame and is given by

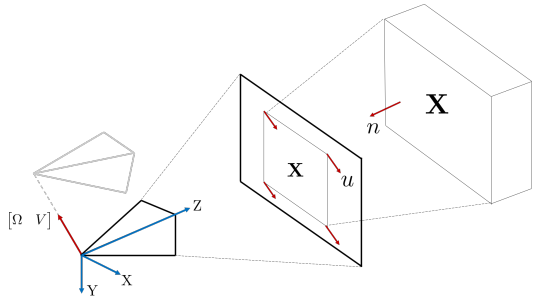

Fig. 3: By the pinhole model, a scene point $\mathbf{X}$ is projected to image point $\mathbf{x}$. The camera "perceives" movement of $\mathbf{x}$ as optical flow $\mathbf{u}$.

$$\mathbf{x}(t) = \underbrace{\begin{bmatrix} Z_0/Z & 0 & (X - X_0)/Z \\ 0 & Z_0/Z & (Y - Y_0)/Z \\ 0 & 0 & 1 \end{bmatrix}}_{A:=} \mathbf{x}(0). \tag{3}$$

The definition of Φ in Eq. (2) allows us to write $\mathbf{X} - \mathbf{X}_0 = (\Phi - I)\mathbf{X}_0$ which reveals that the components of $\Phi - I$ are determined by the elements of A. In our implementation, we estimate the affine homography A using the inverse Lucas-Kanade method. As studied in [31], the patch must have sufficient texture to ensure convergence. We also compose the affine homography with the rotation from inertial measurements (as shown in Fig. 2).

To obtain $\mathbf{F}$ on the tracked patch from the affine homography we consider the optical flow $\mathbf{u}_\mathbf{x}$ as a function of $\mathbf{x}$

$$\mathbf{u}_\mathbf{x} = \frac{d\mathbf{x}(t)}{dt} = \dot{A}A^{-1}\mathbf{x} = -\underbrace{\begin{bmatrix} \dot{Z}/Z & 0 & \dot{X}/Z \\ 0 & \dot{Z}/Z & \dot{Y}/Z \\ 0 & 0 & 0 \end{bmatrix}}_{B:=} \mathbf{x}. \tag{4}$$

Thus, B's terms are equal to frequency-of-contact $\mathbf{F}$. In practice, we fit an affine homography to a slightly slanted plane. In that case, it is more appropriate to consider the affine flow parameters in $B = [b_{i,j}]$ as corresponding to the linear terms of optical flow due to a planar surface. Then, if $\eta := ((b_{2,1}b_{1,3}/b_{2,3}) - b_{1,1})$, the frequency-of-contact is

$$\frac{\dot{\mathbf{X}}}{Z} = \begin{bmatrix} b_{2,1}b_{1,3}/b_{2,3} & b_{1,2} & b_{1,3} \\ b_{2,1} & b_{1,2}b_{2,3}/b_{1,3} & b_{2,3} \\ \eta(b_{2,1}/b_{2,3}) & \eta(b_{1,2}/b_{1,3}) & \eta \end{bmatrix} \mathbf{x}. \tag{5}$$

Next, our constraints are derived from the above relations.

C. The τ-constraint and the Φ-constraint

Now, we will relate $\mathbf{F}$ and Φ to depth and acceleration which results in the τ and Φ constraints respectively. The

constraints relate the initial conditions of the linear time varying system defined by time-to-contact and the linear time invariant system defined by acceleration.

By the fundamental theorem of calculus, $\mathbf{X}$ is given by

$$\mathbf{X}(t) - \mathbf{X}_0 = t\dot{\mathbf{X}}_0 + \underbrace{\int_0^t \left(\int_0^\lambda \ddot{\mathbf{X}}(\lambda_2) d\lambda_2 \right) d\lambda}_{\mathcal{J}\{\ddot{\mathbf{X}}\}(t) :=} . \quad (6)$$

Where $\mathcal{J}\{\mathbf{f}\}(t) : L_p^n \times \mathbb{R} \to \mathbb{R}^n$ is an operator returning the double integral of a vector of functions [32].

Since $\mathbf{X} - \mathbf{X}_0 = (\Phi - I)\mathbf{X}_0$ we can make a substitution which results in the Φ constraint.

Theorem 3.1 (Φ-constraint): If Φ and $\ddot{\mathbf{X}}$ are known $\forall t$ in a closed time interval and $Z \geq \epsilon > 0, \forall t$, then the following linear constraint between initial depth Z_0, Φ, and acceleration $\ddot{\mathbf{X}}$ holds for each point on the planar scene patch

$$(\Phi(t) - I) \begin{bmatrix} 0 \\ 0 \\ Z_0 \end{bmatrix} - t\dot{\mathbf{X}}_0 = \mathcal{J}\{\ddot{\mathbf{X}}\}(t). \quad (\Phi\text{-constraint})$$

Using the Φ-constraint for position estimation requires determining four unknowns, Z_0 and $\dot{\mathbf{X}}_0$. Using the fact that $(\Phi(t) - I) \propto t$ if and only if $\ddot{\mathbf{X}} = 0, \forall t$ makes it possible to show that when $\Phi(t)$ and $\mathcal{J}\{\ddot{\mathbf{X}}\}(t)$ are known over a time interval, then the linear system defined by the Φ-constraint determines Z_0 and $\dot{X}_0$ if and only if $\exists\ t$ between the two times s.t. $\ddot{\mathbf{X}} \neq 0$. A proof is in the supplementary material.

Since $\dot{\mathbf{X}}_0 = \mathbf{F}(0)Z_0$, the τ-constraint follows directly.

Theorem 3.2 (τ-constraint): If $\mathbf{F}$ and $\ddot{\mathbf{X}}$ are known $\forall t$ in a closed interval and $Z \geq \epsilon > 0, \forall t$, then the following linear constraint between depth Z_0, frequency-of-contact $\mathbf{F}$, and acceleration $\ddot{\mathbf{X}}$ holds for each point on the planar scene patch

$$\underbrace{\left(\Phi(t) - I - t \begin{bmatrix} 0 & 0 & \mathbf{F}(0) \end{bmatrix}\right)}_{E(t) :=} \begin{bmatrix} 0 \\ 0 \\ Z_0 \end{bmatrix} = \mathcal{J}\{\ddot{\mathbf{X}}\}(t). \quad (\tau\text{-constraint})$$

$E(t)$ is the ratio between positional change due to acceleration and the depth Z_0. Intuitively, it is the *"action's effect"*.

It is proven in the supplementary material that determining Z_0 is well posed if and only if $\ddot{\mathbf{X}}$ is non-zero at some time.

The Φ-constraint and τ-constraint are closely related. The Φ-constraint considers the position and area of an object in the image, and the τ-constraint considers its velocity and rate of change of size. Since $\mathbf{F}$ determines Φ by integration, it is reasonable to use either the τ or Φ constraint when $\mathbf{F}$ is available. However, using the τ-constraint when only Φ is available is difficult because Φ must be numerically differentiated to get $\mathbf{F}$, which can introduce significant noise.

Next, we discuss estimating depth using our constraints.

IV. Fusing Inertial Measurements with τ and Φ

To tightly couple IMU measurements and frequency-of-contact from a camera in a sliding window manner, we set up a least squares problem over the constraints. The solution is the depth of a point in the scene and the gravity direction.

As illustrated in Fig. 2, we estimate a rotation matrix, R, using the gyroscope. Using this matrix allows all measurements to be rotated into the orientation of the initial camera frame, and thus over a short time period, the method can be considered rotation invariant. Thus, we use R to rotate the measured acceleration $\mathbf{a}_c^m(t)$ back to a fixed frame, $\mathbf{a}^m(t) = R(t)\mathbf{a}_c^m(t)$. The IMU measures the resultant of gravitational acceleration and linear acceleration. Thus, $\mathbf{a}^m(t) = -\ddot{X} + \boldsymbol{g}$.

A. Efficient Computation of Depth (Z Distance)

Let us suppose a history of $\mathbf{a}^m(t)$ and $\mathbf{F}(t)$ measurements are available over a time interval $[0, T]$. Now, without loss of generality, we consider the problem along only the Z axis. The problem for the τ-constraint can be written as

$$\underset{Z_0, g_Z}{\operatorname{argmin}} \int_0^T \left(E_Z Z_0 + \mathcal{J}\{a_Z^m + g_Z\}\right)^2 dt. \quad (7)$$

The problem for the Φ-constraint is given by

$$\underset{Z_0, \dot{Z}_0, g_Z}{\operatorname{argmin}} \int_0^T \left(\left(\Phi_{F_Z} - 1\right) Z_0 - r\dot{Z}_0 + \mathcal{J}\{a_Z^m + g_Z\} \right)^2 dt, \quad (8)$$

where $r(t) \coloneqq t$ is the ramp function.

In both cases, the problem is the same for the X or Y axis up to a change of subscripts and the initial velocity.

It is proven in the supplementary material that Eq. (7) and Eq. (8) are well-posed if and only if the measured acceleration, a_z^m, is not constant for the entire time interval.

Thus, given motion along an axis, depth and gravitational acceleration can be recovered efficiently by solving a linear least squares problem based on either constraint. The estimates have noise and so we discuss filtering them next.

B. Filtering of Depth

Now, we extend the above derivation for trajectory estimation. Since Eq. (7) estimates g_z and Z, we can set up a Luenberger observer [30] to filter the trajectory estimate. Let $\hat{Z}$ and $\hat{\dot{Z}}$ be the estimated quantities, then

$$\begin{bmatrix} \dot{\hat{Z}} \\ \dot{\hat{\dot{Z}}} \end{bmatrix} = \begin{bmatrix} \hat{\dot{Z}} \\ a_z^m - g_z \end{bmatrix} + L \begin{bmatrix} Z - \hat{Z} \\ \dot{Z} - \hat{\dot{Z}} \end{bmatrix}. \quad (9)$$

When multiple estimates for Z are available from motion along multiple axes, the results are averaged. When $\hat{Z}$ and g_z are not available, due to lack of acceleration, then dead reckoning is used by applying $\mathbf{F}$ or Φ to the latest $\hat{Z}$.

Finally, we recover the three dimensional trajectory by multiplying $\hat{Z}$ with the current image location of the center of the planar patch: $\begin{bmatrix} \hat{X} & \hat{Y} & \hat{Z} \end{bmatrix}^T \coloneqq \mathbf{x}\hat{Z}$. Next, we explain how to substitute control effort for acceleration.

V. Closed Loop Control using Efference Copies

Since the control effort in robotics often corresponds to acceleration up to some scale, it is interesting to consider what happens when control effort, $\mathbf{u}$, is substituted for $\ddot{\mathbf{X}}$

when solving Eq. (7). Control effort and optical flow are often referred to as $\mathbf{u}$. So we use $\mathbf{u_x}$ for the flow at pixel $\mathbf{x}$.

This substitution leads to an interesting property.

Corollary 5.1: If $\ddot{\mathbf{X}} = b\mathbf{u}, b \neq 0$, where $\mathbf{u}$ is a linear control determined by $\mathbf{u} = K\hat{\mathbf{X}}$, then if $\mathbf{u}$ is substituted for $\mathbf{a}^m$ in Eq. (7), the dynamics become invariant to b.

Proof: By definition $\ddot{\mathbf{X}}/b = \mathbf{u}$, and the solution to Eq. (7) is linear in $\mathbf{a}^m$. Thus, using $\mathbf{u}$ instead of $\mathbf{a}^m$ results in a state estimate scaled by the inverse of b, i.e. $\hat{\mathbf{X}} = \mathbf{X}/b$.

Further, since $\mathbf{u} \coloneqq K\hat{\mathbf{X}}$, we can see that $\ddot{\mathbf{X}} = bK(\mathbf{X}/b) = K\mathbf{X}$. Thus, the dynamics are invariant to b. ∎

Practically speaking, this allows changing the strength of a motor, or the weight of a robot, without drastically changing the stability properties of the system. Typically such a change would require re-tuning the control gains K.

VI. Experiments

We designed the experiments to evaluate the τ and Φ constraints as well as the invariance property to determine if the constraints are a promising method for future research and applications.

A. Metric Trajectory Estimation

To test the constraints' ability for trajectory estimation, we created ten sequences with five distinct scenes in an indoor setting. Each scene contains a planar object to fixate on as shown in Fig. 1. For each scene, two recordings were made, one with an AprilTag and one without. The trajectories feature acceleration often over 2 m/s^2.

The sequences were recorded with an Intel® RealSense™ D435i camera using the built-in IMU and the left grayscale camera [33]. We used the grayscale camera because the D435i's IMU is hardware time-stamped to the grayscale imager. The D435i captures images at 90 frames per second at 848 × 480 px. resolution. The IMU records gyroscope measurements at 400 Hz and the acceleration at 250 Hz.

We tracked the fixated planar patch using gyroscope measurements for rotation stabilization and by continually fitting an affine homography using the ubiquitous patch tracking method from [31]. The frequency-of-contact $\mathbf{F}$ was then recovered using Eq. (5). The affine tracker was initialized by tracking a 100×100 pixel patch sub-sampled to 4000 pixels. While the patch size changes dramatically during fixation, only 4000 pixels are drawn from each frame.

Then, the τ-constraint was fused with the IMU measurements using a 2 second signal history and a 100 Hz sampling rate using linear interpolation. If the average power of the bias corrected acceleration along an axis was below 2 m/s^2 the resulting depth from the τ-constraint was not used. If no observations of depth were available the depth estimate was forward propagated using dead reckoning with $\mathbf{F}$ or Φ. The Luenberger's gain was set to $L = \text{diag}(2, 20)$ for all the sequences. *Also, it is important to note, that these parameters were not tuned using the sequences considered in this work.*

Our implementation is written in Python 3.8 using standard scientific Python libraries. Numba is used to accelerate critical sections [34]. One thread on an Intel® Core™ i7-6820HQ laptop processor was used to perform computations.

TABLE I: Sequence duration (seconds), path length (meters), and each method's accuracy in centimeters of Average Trajectory Error (ATE). Since AprilTag 3 is always the best when it is available we also bold the second best result in such sequences.

Seq.	1	2	3	4	5
Duration (s)	15.06	26.15	32.28	36.23	16.41
Length (m)	15.73	29.65	22.21	34.75	15.63
Method	ATE (cm)				
AprilTag 3	**2.80**	-	**2.67**	-	**3.76**
VINS-Mono	5.41	8.80	14.21	15.37	-
ROVIO	7.77	9.89	11.88	33.23	29.96
Φ-constraint (ours)	**3.77**	**5.79**	**7.60**	**7.32**	**7.40**
τ-constraint (ours)	8.07	6.91	12.33	10.21	16.82
Seq.	6	7	8	9	10
Duration (s)	16.27	8.02	32.15	26.73	40.08
Length (m)	15.78	7.30	26.75	21.39	35.37
Method	ATE (cm)				
AprilTag 3	-	**0.65**	-	**2.48**	-
VINS-Mono	6.10	1.15	18.45	13.07	4.34
ROVIO	**2.84**	**0.69**	3.93	16.62	3.79
Φ-constraint (ours)	5.71	1.42	**3.28**	2.86	**2.42**
τ-constraint (ours)	7.21	10.70	4.32	**2.38**	3.24

The open-source VINS-Mono and ROVIO implementations were used for comparison [12], [13], [35]. VINS-Mono was configured to output poses at 90 Hz (the camera frame rate), however, it produced poses only at 80 Hz. ROVIO was configured to produce poses at 90 Hz.

As another source of comparison, we used the AprilTag 3 [14] library to detect 36h11 tags. The corners were used to solve the Perspective-n-Point problem to recover a tag's location in the current frame. The gyroscope measurements were used to rotate each AprilTag pose measurement back to the fixed orientation used by the constraints.

The ground truth trajectory was measured at 200 Hz using a Vicon motion capture system with 8 Vantage V8 cameras. We align all trajectories to Vicon ground truth and compute the Average Trajectory Error (ATE) as described in [36].

$$\text{ATE}(\mathbf{X}, \mathbf{X}^v) = \left(\frac{1}{N} \sum_{n=0}^{N-1} \|\mathbf{X}_n - \mathbf{X}_n^v\|_2^2 \right)^{1/2}. \quad (10)$$

Here $\mathbf{X}_n^v$ is themotion capture systems n'th position estimate and $\mathbf{X}_n$ is the n'th estimated position.

B. Closed Loop Stability Invariance Property

To test the invariance property, we implement a closed loop controller on a DJI® RoboMaster™ robot pictured in Fig. 4. The robot's goal is to reach a fixed distance from a pre-determined visual target, where distance is in meters for

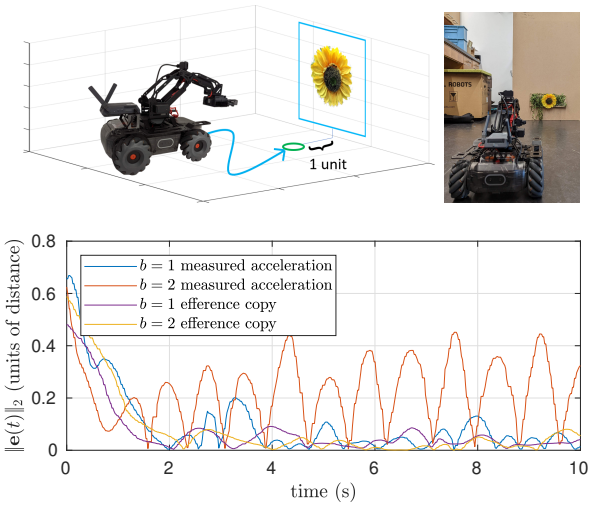

Fig. 4: Top left: The invariance property is tested by using the constraint to navigate to a unit distance from a visual target. Top right: The experimental setup. Bottom: The normed error between the robot's position and the target position. When efference copies are used, the system is stable despite changes to the unknown gain b. When using measured acceleration, the system destabilizes when b is set to 2. Distance is in meters for trials using measured acceleration and units of effort when using efference copies.

trials using measured acceleration and units of effort when using efference copies. By adjusting a static gain b applied to the control signal we can test if using efference copies in the Φ-constraint prevents the control system from exhibiting unstable behavior when b is increased dramatically. Four trials were run. In two of these acceleration from the onboard IMU was fed to the Φ-constraint whose estimates were then used for closed loop control. In the other two trials, efference copies were fed to the Φ-constraint, meaning that $\mathbf{u}$ was used instead of the measured acceleration. For each group of two experiments, one was run with the actuator gain set to 1 (which was the value the K matrix was tuned for). In the second run, the actuator gain was set to 2, which doubled all control signals without the control algorithm's knowledge. The control gains were chosen as $K = \text{diag}(2, 2)$.

VII. Results And Discussion

A. Metric Trajectory Estimation

The ATE results across all the 10 sequences are given in Tab. I along with the path length and duration of the sequences. The proposed Φ-constraint achieves lower ATE than VINS-Mono and ROVIO in all but two sequences. In those two sequences, it remains competitive. The τ-constraint achieves a lower ATE than VINS-Mono in six out of nine comparable sequences and a lower ATE than ROVIO in 5 out of 10 comparable sequences. ATE averaged over all sequences was 5.4 cm for the Φ-constraint, 8.5 cm for the τ-onstraint, 16.9 cm for ROVIO, and 2.8 cm for AprilTag 3. The average ATE for VINS-Mono, averaged over all sequences except sequence 5, was 12.2 cm. The VINS-Mono result is omitted for sequence 5 because its estimate diverged.

It is no surprise that the AprilTag 3 method routinely achieved the best ATE. This is because the AprilTag system uses the known size of the visual fiducial to estimate depth. Regardless, the τ and Φ-constraints perform comparably to the AprilTag 3 method in some sequences. This is particularly noticeable in Sequence 9. In Fig. 1 the instantaneous l_2 error of each method in Sequence 9 is plotted for comparison.

While the ATE errors are promising, they do not indicate that our method is better than existing VIO methods. Such a claim would require developing a full VIO stack around the τ or Φ-constraint and comparisons on existing datasets.

Our Python implementation achieved 6.5 $\times$ realtime or 588 frames per second (fps). VINS-Mono's C++ implementation ran at 0.26 $\times$ realtime or 23.6 fps. ROVIO's C++ implementation ran at 1.05 $\times$ realtime or 94.5 fps.

B. Closed Loop Stability Invariance Property

As shown in Fig. 4, all achieved trajectories are similar, and approach the target, except for the case where the actuator gain was doubled and measured linear acceleration was used in the Φ-constraint. This was expected. Indeed, in this case, the poles and zeros of the closed loop system are dramatically shifted because the control gain matrix is effectively doubled. As a result, the robot began to oscillate around its stopping point as is typical of a "poorly tuned" controller. However, the control scheme using efference copies had no such limitation, as predicted by Corollary 5.1.

VIII. Conclusion and Future Work

In our work, we developed two novel constraints called the τ and Φ-constraint which allow a moving camera to estimate depth using a small part of the image. Applying these constraints to trajectory estimation achieved better results while being orders of magnitude faster than some state-of-the-art VIO approaches. Further, we presented a method to perform closed-loop control with the constraints while using efference copies which is invariant to scaling of the control signal. We will talk about some future directions next.

Both constraints require that there is acceleration in order to measure distance. In practice, we found accelerations with approximately 2 m/s^2 of power were necessary to get good measurements. However, when using efference copies only a small nominal acceleration was required for reasonable performance. Further theoretical analysis would be useful for gaining more insight into this behaviour.

VIO methods commonly estimate IMU biases and so it would be interesting to add such terms to our constraints. Similarly, it is of interest to extend Corollary 5.1 to account for a transfer function relating control effort and acceleration.

For our method to be deployable as a full VIO system it would need to be extended to fixate on multiple patches and actively switch between them. Based on the significant speed up, and competitive accuracy presented in our preliminary results, we believe that further development of the τ and Φ constraint, in theory and practice, is a promising direction for VIO, VI-SLAM, active perception, and robotics.

References

[1] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988. 1

[2] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988. 1

[3] D. H. Ballard, "Animate vision," *Artificial Intelligence*, vol. 48, no. 1, pp. 57–86, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0004370291900804 1

[4] C. Fermüller, "Navigational preliminaries," in *Active Perception*, Y. Aloimonos, Ed. Hillsdale, NJ: Lawrence Erlbaum Assoc., 1993, ch. 3, pp. 103–150. 1

[5] C. Fermüller and Y. Aloimonos, "Vision and action," *Image and Vision Computing*, vol. 13, no. 10, pp. 725–744, 1995. 1

[6] N. Jagannatha Sanket, "Active vision based embodied-ai design for nano-uav autonomy," Ph.D. dissertation, University of Maryland, College Park, 2021. 1

[7] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *Autonomous Robots*, vol. 42, pp. 177–196, 2018. [Online]. Available: https://doi.org/10.1007/s10514-017-9615-3 1

[8] C. Fermüller and Y. Aloimonos, "Tracking facilitates 3-D motion estimation," *Biological Cybernetics*, vol. 67, no. 3, pp. 259–268, 1992. [Online]. Available: https://doi.org/10.1007/BF00204399 1

[9] A. Mishra, Y. Aloimonos, and C. Fermüller, "Active segmentation for robotics," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3133–3139. 1

[10] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Information fusion in navigation systems via factor graph based incremental smoothing," *Robotics and Autonomous Systems*, vol. 61, no. 8, pp. 721–738, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S092188901300081X 2

[11] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Visual–inertial navigation systems for aerial robotics: Sensor fusion and technology," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 260–275, 2017. 2

[12] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018. 2, 5

[13] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 298–304. 2, 5

[14] M. Krogius, A. Haggenmiller, and E. Olson, "Flexible layouts for fiducial tags," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1898–1903. 2, 5

[15] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976. [Online]. Available: https://doi.org/10.1068/p050437 2

[16] D. N. Lee, R. J. Bootsma, M. Land, D. Regan, and R. Gray, "Lee's 1976 paper," *Perception*, vol. 38, no. 6, pp. 837–858, 2009. 2

[17] O. Sikorski, D. Izzo, and G. Meoni, "Event-based spacecraft landing using time-to-contact," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 1941–1950. 2

[18] C. Walters and S. Hadfield, "EVReflex: Dense time-to-impact prediction for event-based obstacle avoidance," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1304–1309. 2

[19] H. W. Ho, G. C. de Croon, and Q. Chu, "Distance and velocity estimation using optical flow from a monocular camera," *International Journal of Micro Air Vehicles*, vol. 9, no. 3, pp. 198–208, 2017. 2

[20] G. C. H. E. de Croon, "Monocular distance estimation with optical flow maneuvers and efference copies: a stability-based strategy," *Bioinspiration & Biomimetics*, vol. 11, no. 1, p. 016004, 2016. [Online]. Available: https://doi.org/10.1088/1748-3190/11/1/016004 2

[21] A. Badki, O. Gallo, J. Kautz, and P. Sen, "Binary TTC: A temporal geofence for autonomous navigation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12 946–12 955. 2

[22] Z. Wang, F. C. Ojeda, A. Bisulco, D. Lee, C. J. Taylor, K. Daniilidis, M. A. Hsieh, D. D. Lee, and V. Isler, "EV-Catcher: High-speed object catching using low-latency event-based neural networks," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8737–8744, 2022. 2

[23] B. K. Horn, Y. Fang, and I. Masaki, "Time to contact relative to a planar surface," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 68–74. 2

[24] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3565–3572. 2

[25] A. Martinelli, "Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012. 2

[26] ——, "Closed-form solution of visual-inertial structure from motion," *International Journal of Computer Vision*, vol. 106, no. 2, pp. 138–152, 2014. [Online]. Available: https://doi.org/10.1007/s11263-013-0647-7 2

[27] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, "VINet: Visual-inertial odometry as a sequence-to-sequence learning problem," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, vol. 31, 2017, pp. 3995–4001. 2

[28] L. Han, Y. Lin, G. Du, and S. Lian, "DeepVIO: Self-supervised deep learning of monocular visual inertial odometry using 3D geometric constraints," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6906–6913. 2

[29] N. J. Sanket, C. D. Singh, C. Fermüller, and Y. Aloimonos, "PRGFlow: Unified SWAP-aware deep global optical flow for aerial robot navigation," *Electronics Letters*, vol. 57, no. 16, pp. 614–617, 2021. 2

[30] J. P. Hespanha, *Linear Systems Theory*, 2nd ed. Princeton University Press, 2018. [Online]. Available: https://doi.org/10.23943/9781400890088 3, 4

[31] S. Baker and I. Matthews, "Lucas-Kanade 20 years on: A unifying framework," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004. 3, 5

[32] M. Sh, Birman, and M. Solomyak, "Lectures on double operator integrals," 2003. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.550.8049 4

[33] L. Keselman, J. I. Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel(R) RealSense(TM) stereoscopic depth cameras," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1267–1276. 5

[34] S. K. Lam, A. Pitrou, and S. Seibert, "Numba: A LLVM-based Python JIT compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 2015, pp. 7:1–7:6. [Online]. Available: https://doi.org/10.1145/2833157.2833162 5

[35] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *The International Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017. [Online]. Available: https://doi.org/10.1177/0278364917728574 5

[36] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 7244–7251. 5