2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Madrid, Spain, October 1-5, 2018

Unit Quaternion-based Parameterization for Point Features in Visual Navigation

James Maley and Guoquan Huang

***Abstract*— In this paper, we propose to use unit quaternions to represent point features in visual navigation. Contrary to the Cartesian 3D representation, the unit quaternion can well represent features at both large and small distances from the camera without suffering from convergence problems. Contrary to inverse-depth, homogeneous points, or anchored homogeneous points, the unit quaternion has error state of minimum dimension of three. In contrast to prior representations, the proposed method does not need to approximate an initial infinite depth uncertainty. In fact, the unit-quaternion error covariance can be initialized from the initial feature observations without prior information, and the initial error-states are not only bounded, but the bound is identical for all scene geometries. To the best of our knowledge, this is the first time bearing-only recursive estimation (in covariance form) of point features has been possible without using measurements to initialize error covariance. The proposed unit quaternion-based representation is validated on numerical examples.**

I. Introduction

Visual navigation with a monocular camera typically requires estimating point features along with the states of the sensor platform. Many algorithms are available for solving this problem, for example, extended Kalman filter (EKF)-based SLAM [1]–[3], pose graph optimization [4]–[6], and visual-inertial odometry (VIO) [7]–[10]. Estimating feature positions is challenging because a monocular camera measurement provides *no* depth information without motion parallax. Prior to being observed there is typically no information about the feature location. For this reason, a covariance-form EKF is difficult to initialize without making an approximation to the prior feature covariance or performing a delayed initialization procedure [11], [12].

To address these challenges, in this paper we propose to use a unit quaternion to represent 3D feature positions. Quaternions are widely used in navigation [13], attitude estimation [14], and computer graphics [15], but only for representing rotations. It is possible to represent a $\mathbb{R}^3$ vector as a so-called "pure quaternion" for the purpose of rotating it into a different frame of reference with quaternions, but that is not what is advocated here. Instead, the unit quaternion is treated as a special case of homogeneous coordinates, subject to spherical normalization. The motivation for using quaternions for representing point feature positions is analogous to that for using quaternions to represent attitude instead of Euler angles. Euler angles are intuitive, provide a minimal error representation, and careful modeling and engineering work-arounds can avoid degenerate rotation spaces (e.g. gimbal lock). For attitude, quaternions provide a minimal error representation and work to represent any sequence of rotations without design modifications. We show in this paper that when using quaternions to represent feature positions they not only can represent features at any depth without suffering from convergence issues, but do so with minimal error representation while enabling simple undelayed initialization for robocentric geometries.

Note first that throughout this paper, we have employed the following notations:

$\mathbf{u}, \phi$	Unit vector, and quaternion angle
$\mathbf{v}$	Vectors are in bold lowercase
$\mathbf{M}$	Matrices are in bold uppercase
$\bar{f}$	Quaternions (and briefly homogeneous coordinates) denoted by $\bar{}$
$\hat{a}, \hat{\mathbf{v}}, \hat{\mathbf{M}}, \hat{\bar{f}}$	Estimated quantities denoted by $\hat{}$
$\tilde{a}, \tilde{\mathbf{v}}, \tilde{\mathbf{M}}, \tilde{\bar{f}}$	Error states denoted by $\tilde{}$
${}^C_B\mathbf{R}$	A rotation matrix from frame B to frame C
$\mathbf{R}(\boldsymbol{\alpha})$	A rotation matrix formed from Euler angles in vector $\boldsymbol{\alpha}$
${}^C\mathbf{v}_{A/B}$	Coordinates of A with respect to B viewed in frame C
${}^C\mathbf{v}_A$	No / implies coordinates of A with respect to C viewed in frame C
$\lfloor \mathbf{v} \times \rfloor$	Skew-symmetric matrix formed from $\mathbf{v}$, i.e. $\lfloor \mathbf{v} \times \rfloor \mathbf{w} = \mathbf{v} \times \mathbf{w}$ where $\times$ denotes a cross product
$\otimes$	Quaternion multiplication operator
$\mathbf{J}_{\mathbf{y}}(\mathbf{x})$	If evaluation point not specified, Jacobian of vector valued function $\mathbf{y}$ with respect to $\mathbf{x}$, evaluated at $\mathbf{x} = \mathbf{0}$
$\{G\}, \{B\}, \{I\}, \{C\}$	Global, IMU (body)-fixed, initial sighting, and camera frames
$\mathbf{p}$	Position vector
$\mathbf{p}_f$	Feature position
$\mathbf{T}$	Homogeneous transformation
${}^C_G\bar{q}$	Rotation quaternion
$\tilde{\mathbf{t}}^\top = \begin{bmatrix}\tilde{\theta}^\top & \tilde{\mathbf{p}}^\top\end{bmatrix}$	Transformation error vector
$\bar{f}^\top = \begin{bmatrix}\mathbf{v}^\top & b\end{bmatrix}$	Feature quaternion in user-chosen coordinate system
$\bar{g}^\top = \begin{bmatrix}\mathbf{w}^\top & c\end{bmatrix}$	Feature quaternion after some transformation

This work was partially supported by the U.S. Army Research Laboratory, the University of Delaware College of Engineering, the NSF (IIS-1566129) and the DTRA (HDTRA 1-16-1-0039).

J. Maley is with the U.S. Army Research Lab, Aberdeen Proving Ground, MD 21005, USA. Email: james.m.maley2.civ@mail.mil

G. Huang is with the Department of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: ghuang@udel.edu

II. Representations of Point Features

A feature's mathematical representation ideally should be able to represent points that are infinitely far away from the camera, as well as points that are close to the camera. It is known that a standard $\mathbb{R}^3$ Cartesian vector is not suitable for representing points at infinity. When used in bundle adjustment (BA), Cartesian representations have poor convergence because extremely large position corrections are needed to be observable in pixel reprojection errors [6]. Cartesian representations also yield practical difficulty for covariance-form EKF-based solutions because it is hard to fully initialize a feature's error covariance to represent the complete absence of depth information. While this issue can be alleviated with initialization procedures such as [16], the convergence problem remains.

The inverse-depth parameterization was introduced to address these issues and provides a parameterization in which the transformation from feature errors to pixel residuals is more linear than the Cartesian representation [17]. Since then, inverse depth has been the standard in feature-based SLAM systems in particular when distant points are present, and Cartesian points are used when they are close to the camera relative to the motion. In visual SLAM, the inverse depth parameterization works by considering not only the current camera frame $\{C\}$ position and rotation relative to the global frame $\{G\}$, i.e., $({}^G\mathbf{p}_C, {}^C_G\mathbf{R})$, but also the position of the camera ${}^G\mathbf{p}_I$ when the feature was first observed. Prior to the perspective projection operation, the feature point position in the camera frame is given by:

$$ {}^C\mathbf{p}_f = {}^C_G\mathbf{R}\left[{}^G_I\mathbf{R}\,{}^I\mathbf{p}_f + {}^G\mathbf{p}_I - {}^G\mathbf{p}_C\right] \tag{1} $$

In [17], the feature position relative to the initial sighting is a function of the inverse depth ρ, and the azimuth and elevation angles ψ, and θ, i.e., ${}^G_I\mathbf{R}\,{}^I\mathbf{p}_f = \frac{1}{\rho}\mathbf{m}(\psi,\theta)$. The result is that the feature points are parameterized by a $\mathbb{R}^6$ vector $\mathbf{a}_f$, i.e., $\mathbf{a}_f = \begin{bmatrix}{}^G\mathbf{p}_I^\top & \psi & \theta & \rho\end{bmatrix}^\top$. The disadvantage is the additional computational complexity of including ${}^G\mathbf{p}_I$ in the state-vector, which motivates in many implementations a mechanism for switching between inverse depth and Cartesian coordinates and a metric to determine when that switch is appropriate [18], [19].

Different state-space modeling can be used to avoid including ${}^G\mathbf{p}_I$ as part of the state vector. For example, in a robocentric formulation, the feature position can be defined relative to either the current camera frame [20] or a recent keyframe [21]. Another inverse depth parameterization includes the MSCKF-based VIO [7], in which the trigonometric terms are replaced with a projective terms α, β, i.e., ${}^I\mathbf{p}_f = \frac{1}{\rho}\begin{bmatrix}\alpha & \beta & 1\end{bmatrix}^\top$. This is another instance in which it is not required to maintain ${}^G\mathbf{p}_I$ as part of the state vector because by construction the features are defined relative to camera frame in a sliding window.[1]

[1]Technically, the MSCKF and its variants do not store the feature points as states at all, while they are marginalized immediately after a window of observations are used.

While the inverse depth and Cartesian coordinates are the two most widely used representations, other parameterizations such as homogeneous coordinates and anchored homogeneous coordinates (AHC) also exist. In particular, AHC were shown to be more consistent than the inverse depth in EKF-based SLAM [22], although they are 7-dimensional, which is over parameterized.

In summary, it is clearly possible to develop a functional visual SLAM or visual odometry system using an inverse-depth or other existing parameterization of point features. What we hope to improve upon is the generality and initialization by finding an alternative parameterization.

III. Unit Quaternion Parameterization for Point Features

A. Homogeneous Coordinates

The quaternion representation presented here can be considered a special case of homogeneous coordinates. To represent features at any depth, it is strongly suggested in [6] using homogeneous coordinates of the form:

$$ \underbrace{\bar{f} = \begin{bmatrix}\mathbf{v}\\ b\end{bmatrix}}_{\text{homogeneous}} \implies \underbrace{\mathbf{p}_f = \frac{\mathbf{v}}{b}}_{\text{Cartesian}} \tag{2} $$

with spherical normalization, i.e., $\bar{f}^\top\bar{f} = 1$. The feature position in the camera frame is given by:

$$ {}^C\mathbf{p}_f = {}^C_G\mathbf{R}\,{}^G\mathbf{p}_f - {}^C_G\mathbf{R}\,{}^G\mathbf{p}_C \tag{3} $$

$$ \Rightarrow\ {}^C\mathbf{p}_f b = \begin{bmatrix}{}^C_G\mathbf{R} & \vdots & -{}^C_G\mathbf{R}\,{}^G\mathbf{p}_C\end{bmatrix}\bar{f} \tag{4} $$

Conveniently, the scalar term b cancels out in the perspective projection that generates normalized pixel coordinates in the image plane, given by:

$$ \mathbf{y} = \frac{{}^C\mathbf{p}_{f_{x,y}} b}{{}^C\mathbf{p}_{f_z} b} \tag{5} $$

where ${}^C\mathbf{p}_{f_{x,y}}$ denotes the x and y coordinates of ${}^C\mathbf{p}_f$ and ${}^C\mathbf{p}_{f_z}$ is the z coordinate. However, the problem is that with an additive error model, the normalization constraint is violated. In addition, the feature position becomes less observable because an additional degree of freedom is added.

B. Unit Quaternion Representation

Suppose the homogeneous coordinates are further constrained to form a unit quaternion of the following form:

$$ \bar{f} = \begin{bmatrix}\mathbf{u}\sin(\phi)\\ \cos(\phi)\end{bmatrix} \tag{6} $$

In this case, $\mathbf{u}$ is a unit vector that points from the origin of the reference frame to the feature point. It is not hard to see that this satisfies the normalization constraint. From (2), the scale of the feature is given by $\tan(\phi)$, thus:

$$ \mathbf{p}_f = \mathbf{u}\tan(\phi) \tag{7} $$

Because $\tan(\phi)$ has a range of $[-\infty, \infty]$, this representation is capable of representing *any* point in $\mathbb{R}^3$.

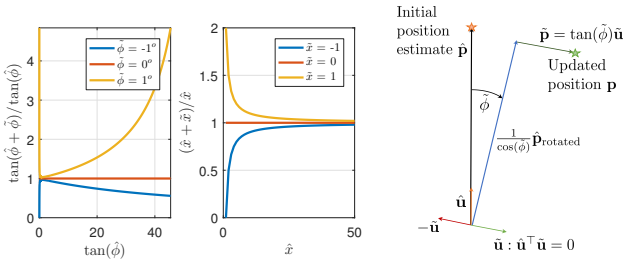

Fig. 1: Ratio of feature position scale before and after quaternion update with small $\tilde{\phi}$ values compared to that of feature position scale before and after Cartesian update with small $\tilde{x}$ values (left). Position update in the second special case (right).

Quaternions have been used extensively for representing orientation in most navigation applications. Usually when a quaternion is being estimated, the initial estimate $\hat{\bar{f}}$ is corrected by multiplying it by an error quaternion $\tilde{\bar{f}}$ [23]. One reason for this is unit quaternions are closed under quaternion multiplication, but not under addition (see [24]). For reference, this operation is given by:

$$\bar{f} = \tilde{\bar{f}} \otimes \hat{\bar{f}} = \begin{bmatrix}\tilde{\mathbf{v}}\\ \tilde{b}\end{bmatrix} \otimes \begin{bmatrix}\hat{\mathbf{v}}\\ \hat{b}\end{bmatrix} \tag{8}$$

$$\begin{bmatrix}\mathbf{v}\\ b\end{bmatrix} = \begin{bmatrix}\mathbf{I}\hat{b} + \lfloor\hat{\mathbf{v}}\times\rfloor & \hat{\mathbf{v}}\\ -\hat{\mathbf{v}}^\top & \hat{b}\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{v}}\\ \tilde{b}\end{bmatrix} \tag{9}$$

To understand the physical interpretation of this operation when applied to feature positions, we examine the equivalent updated Cartesian position by substituting (6) into (8):

$$\begin{cases}\mathbf{v} = & \cos(\tilde{\phi})\sin(\hat{\phi})\hat{\mathbf{u}} + \cos(\hat{\phi})\sin(\tilde{\phi})\tilde{\mathbf{u}} + \\ & \sin(\hat{\phi})\sin(\tilde{\phi})\lfloor\hat{\mathbf{u}}\times\rfloor\tilde{\mathbf{u}}\\ b = & \cos(\hat{\phi})\cos(\tilde{\phi}) - \sin(\hat{\phi})\sin(\tilde{\phi})\hat{\mathbf{u}}_f^\top\tilde{\mathbf{u}}\end{cases}$$

$$\Rightarrow \mathbf{p}_f = \frac{\tan(\hat{\phi})\hat{\mathbf{u}} + \tan(\tilde{\phi})\tilde{\mathbf{u}} - \tan(\tilde{\phi})\tan(\hat{\phi})\lfloor\tilde{\mathbf{u}}\times\rfloor\hat{\mathbf{u}}}{1 - \tan(\hat{\phi})\tan(\tilde{\phi})\hat{\mathbf{u}}^\top\tilde{\mathbf{u}}} \tag{10}$$

1) Special Cases: Two special cases provide intuition about the updated Cartesian vector (10). First, if $\tilde{\mathbf{u}}$ is in the exactly same direction as $\hat{\mathbf{u}}$, i.e., $\hat{\mathbf{u}}^\top\tilde{\mathbf{u}} = 1$ and $\lfloor\tilde{\mathbf{u}}\times\rfloor\hat{\mathbf{u}} = \mathbf{0}$, then (10) reduces to:

$$\mathbf{p}_f = \frac{\tan(\hat{\phi}) + \tan(\tilde{\phi})}{1 - \tan(\hat{\phi})\tan(\tilde{\phi})}\hat{\mathbf{u}} = \tan(\hat{\phi} + \tilde{\phi})\hat{\mathbf{u}} \tag{11}$$

This becomes convenient, because the change in the magnitude of $\mathbf{p}_f$ imparted by a small $\tilde{\phi}$ scales with the magnitude of $\hat{\mathbf{p}}_f$. Fig. 1(left) illustrates this effect. With the quaternion parameterization, small $\tilde{\phi}$ corrections produce larger and larger position scale changes as the original estimate increases. This is in contrast to the Cartesian parameterization, in which corrections to any 3D coordinate (x in this example) have diminishing returns as the prior estimate scale increases. This is why Cartesian representations have convergence problems for distant feature points. But what about if $\hat{\phi}$ is close to $\pi/2$? In this case, it is indeed possible for a correction with a large enough $\tilde{\phi}$ to push the magnitude of the resulting Cartesian position scale past ∞ to a large negative number. Although this results in a reconstructed Cartesian coordinate that is behind the camera, this does not effect the normalized pixel coordinates because the scale factor cancels in the perspective projection (5). In addition, an indirect (EKF) SLAM estimator is not jointly estimating Cartesian coordinates of the features with the robot states but estimating the feature quaternion error states, and these estimates are what must be consistent. This is similar to pushing ρ below 0 in an inverse depth parameterization.

In the second special case, if $\hat{\mathbf{u}}^\top\tilde{\mathbf{u}} = 0$, we have:

$$\begin{aligned}\mathbf{p}_f &= \tan(\hat{\phi})\hat{\mathbf{u}} + \tan(\tilde{\phi})\tilde{\mathbf{u}} - \tan(\tilde{\phi})\tan(\hat{\phi})\lfloor\tilde{\mathbf{u}}\times\rfloor\hat{\mathbf{u}}\\ &= \tilde{\mathbf{p}}_f + \frac{1}{\cos(\tilde{\phi})}\left(\cos(\tilde{\phi})\hat{\mathbf{p}}_f - \sin(\tilde{\phi})\lfloor\tilde{\mathbf{u}}\times\rfloor\hat{\mathbf{p}}_f\right)\end{aligned} \tag{12}$$

The first term (12) is a Cartesian addition to the original position estimate. The second term is a special case of a vector rotated by an angle $\tilde{\phi}$ about a vector $-\tilde{\mathbf{u}}$ using Rodrigues formula, and then scaled by a factor of $1/\cos(\tilde{\phi})$. The whole transformation is depicted in Fig. 1(right).

C. Linearization of Quaternions

It is clear from the special cases discussed above that a small $\tilde{\phi}$ may cause a large change in the Cartesian coordinates of the feature position. As in the perturbation analysis of quaternions used to represent rotation [24], $\tilde{\phi}$ is assumed to be small (i.e., $\cos(\tilde{\phi}) \approx 1$), and thus, only the vector part of the error quaternion $\tilde{\mathbf{v}}$ needs to be estimated. The dimension of the quaternion error state has now gone from 4 to 3, same as that of Cartesian error states. The error quaternion is hence approximated by $\tilde{\bar{f}} \approx \begin{bmatrix}\tilde{\mathbf{v}}^\top & 1\end{bmatrix}^\top$, and the quaternion update simply becomes [see (9)]:

$$\bar{f} \approx \begin{bmatrix}\mathbf{I}\hat{b} + \lfloor\hat{\mathbf{v}}\times\rfloor\\ -\hat{\mathbf{v}}^\top\end{bmatrix}\tilde{\mathbf{v}} + \begin{bmatrix}\hat{\mathbf{v}}\\ \hat{b}\end{bmatrix} =: \mathbf{F}\tilde{\mathbf{v}} + \hat{\bar{f}} \tag{13}$$

D. Transformation of Quaternions

Suppose $\bar{f}$ is the quaternion representation of a feature position in frame $\{J\}$, ${}^J\mathbf{p}_f$, and $\bar{g}$ is the quaternion representation of a feature position in frame $\{K\}$, ${}^K\mathbf{p}_f$.[2] The transformation between $\bar{f}$ and $\bar{g}$ is given by [see (4)]:

$${}^K\mathbf{p}_f = {}^K_J\mathbf{R}\,{}^J\mathbf{p}_f - {}^K_J\mathbf{R}\,{}^J\mathbf{p}_K \tag{14}$$

$$\Rightarrow \underbrace{\begin{bmatrix}\mathbf{w}\\ c\end{bmatrix}}_{\bar{g}} b = \underbrace{\begin{bmatrix}{}^K_J\mathbf{R} & -{}^K_J\mathbf{R}\,{}^J\mathbf{p}_K\\ \mathbf{0} & 1\end{bmatrix}}_{\mathbf{T}}\underbrace{\begin{bmatrix}\mathbf{v}\\ b\end{bmatrix}}_{\bar{f}} c \tag{15}$$

The estimate of $\bar{g}$ is easily obtained from the estimates of $\bar{f}$ and $\mathbf{T}$ by multiplying them and normalizing the result. The goal now is to determine the linear mapping from errors in $\bar{f}$ and $\mathbf{T}$ to that in $\bar{g}$, which can be determined by perturbation. Specifically, the quaternions are linearized as [see (13)]:

$$\mathbf{F} = \begin{bmatrix}\mathbf{I}\hat{b} + \lfloor\hat{\mathbf{v}}\times\rfloor\\ -\hat{\mathbf{v}}^\top\end{bmatrix} \qquad \mathbf{G} = \begin{bmatrix}\mathbf{I}\hat{c} + \lfloor\hat{\mathbf{w}}\times\rfloor\\ -\hat{\mathbf{w}}^\top\end{bmatrix} \tag{16}$$

$$\bar{f} \approx \mathbf{F}\tilde{\mathbf{v}} + \hat{\bar{f}} \qquad \bar{g} \approx \mathbf{G}\tilde{\mathbf{w}} + \hat{\bar{g}} \tag{17}$$

$$b \approx \hat{b} - \hat{\mathbf{v}}^\top\tilde{\mathbf{v}} \qquad c \approx \hat{c} - \hat{\mathbf{w}}^\top\tilde{\mathbf{w}} \tag{18}$$

[2]For example, J and K could be reference and camera frame, camera frame at $t = k$ and $t = k+1$, etc. depending on filter mechanization.

Substitution of these identities into (15) yields:

$$\left(\mathbf{G}\tilde{\mathbf{w}}+\hat{\tilde{g}}\right)\left(\hat{b}-\hat{\mathbf{v}}^\top\tilde{\mathbf{v}}\right)\approx\left(\hat{\mathbf{T}}+\tilde{\mathbf{T}}\right)\left(\mathbf{F}\tilde{\mathbf{v}}+\hat{\tilde{f}}\right)\left(\hat{c}-\hat{\mathbf{w}}^\top\tilde{\mathbf{w}}\right)$$

where for simplicity (with a bit abuse of notation) we decomposed the true homogeneous transformation into its estimate and error, $\mathbf{T}\approx\hat{\mathbf{T}}+\tilde{\mathbf{T}}$. By neglecting high-order errors and using $\hat{\tilde{g}}\hat{b}=\hat{\mathbf{T}}\hat{\tilde{f}}\hat{c}$, this can be simplified to:

$$\left(\hat{\tilde{g}}\hat{\mathbf{w}}^\top\hat{b}\hat{c}^{-1}+\mathbf{G}\hat{b}\right)\tilde{\mathbf{w}}\approx\left(\hat{\tilde{g}}\hat{\mathbf{v}}^\top+\hat{\mathbf{T}}\mathbf{F}\hat{c}\right)\tilde{\mathbf{v}}+\tilde{\mathbf{T}}\hat{\tilde{f}}\hat{c}\quad(19)$$

It is not difficult to validate that $\mathbf{G}^\top\mathbf{G}=\mathbf{I}_{3\times3}$ and $\mathbf{G}^\top\hat{\tilde{g}}=\mathbf{0}_{3\times1}$. By multiplying both sides by $\mathbf{G}^\top\hat{b}^{-1}$, we obtain the desired linear relation of the quaternion error states:

$$\tilde{\mathbf{w}}\approx\frac{\hat{c}}{\hat{b}}\mathbf{G}^\top\hat{\mathbf{T}}\mathbf{F}\tilde{\mathbf{v}}+\frac{\hat{c}}{\hat{b}}\mathbf{G}^\top\tilde{\mathbf{T}}\hat{\tilde{f}}\quad(20)$$

Different parameterizations of $\tilde{\mathbf{T}}$ are possible in terms of the overarching navigation error states. For example, very often we consider the transformation error vector as $\tilde{\mathbf{t}}=\begin{bmatrix}\tilde{\theta}^\top & \tilde{\mathbf{p}}^\top\end{bmatrix}^\top$, where ${}^K_J\mathbf{R}\approx\left(\mathbf{I}-\lfloor\tilde{\theta}\times\rfloor\right){}^K_J\hat{\mathbf{R}}$, and ${}^J\mathbf{p}_K={}^J\hat{\mathbf{p}}_K+\tilde{\mathbf{p}}$. Then it becomes useful to re-derive (20) as follows:

$$\tilde{\mathbf{w}}\approx\mathbf{J}_{\tilde{\mathbf{w}}}(\tilde{\mathbf{v}})\tilde{\mathbf{v}}+\mathbf{J}_{\tilde{\mathbf{w}}}(\tilde{\mathbf{t}})\tilde{\mathbf{t}}\quad(21)$$

$$\mathbf{J}_{\tilde{\mathbf{w}}}(\tilde{\mathbf{v}})=\frac{\hat{c}}{\hat{b}}\mathbf{G}^\top\hat{\mathbf{T}}\mathbf{F}\quad(22)$$

$$\mathbf{J}_{\tilde{\mathbf{w}}}(\tilde{\mathbf{t}})=\frac{\hat{c}}{\hat{b}}\mathbf{G}^\top\begin{bmatrix}\lfloor{}^K_J\hat{\mathbf{R}}\left(\hat{\mathbf{v}}-\hat{b}{}^J\hat{\mathbf{p}}_K\right)\times\rfloor & -\hat{b}{}^K_J\hat{\mathbf{R}}\end{bmatrix}\quad(23)$$

We point out that care should be taken to account for cases when $\hat{b}=0$ (which can happen often as will become clear in the initialization discussed later). For instance, if $\hat{b}=0$, then $\hat{c}=0$ by definition and the ratio can be set to 1.

E. Perspective Projection

Suppose the transformed quaternion $\bar{g}=\begin{bmatrix}\mathbf{w}^\top & c\end{bmatrix}^\top$ represents the feature position in the camera frame, i.e., ${}^C\mathbf{p}_f=\mathbf{w}/c$. A monocular camera ideally measures the perspective projection of this position into the camera frame (5), and thus this measurement is: $\mathbf{y}=f\begin{bmatrix}w_x/w_z & w_y/w_z\end{bmatrix}^\top+\mathbf{c}+\boldsymbol{\eta}$, where f is the focal-length in pixels, $\mathbf{c}$ is the camera center offset, $\mathbf{w}=\begin{bmatrix}w_x & w_y & w_z\end{bmatrix}^\top$, and the noise $\boldsymbol{\eta}\sim\mathcal{N}(\mathbf{0},\mathbf{I}\sigma^2_{\text{pix}})$. With the error state, $\mathbf{w}=\hat{\mathbf{w}}+(\mathbf{I}\hat{c}+\lfloor\hat{\mathbf{w}}\times\rfloor)\tilde{\mathbf{w}}$, we compute the measurement Jacobian as:

$$\mathbf{J}_{\mathbf{y}}(\tilde{\mathbf{w}})=f\begin{bmatrix}\frac{(\hat{c}\hat{w}_z+\hat{w}_x\hat{w}_y)}{\hat{w}_z^2} & -\frac{\hat{w}_x^2}{\hat{w}_z^2}-1 & -\frac{(\hat{c}\hat{w}_x-\hat{w}_y\hat{w}_z)}{\hat{w}_z^2}\\ \frac{\hat{w}_y^2}{\hat{w}_z^2}+1 & \frac{(\hat{c}\hat{w}_z-\hat{w}_x\hat{w}_y)}{\hat{w}_z^2} & -\frac{(\hat{c}\hat{w}_y+\hat{w}_x\hat{w}_z)}{\hat{w}_z^2}\end{bmatrix}\quad(24)$$

F. Unit Vector Measurements

The most general parameterization of a (pinhole or omnidirectional) camera measurement is a unit vector pointed to the feature in the camera frame ${}^C\mathbf{u}$. To preserve the unit-norm constraint, the error in this vector can parameterized by a small rotation angle $\tilde{\boldsymbol{\alpha}}$, so that ${}^C\mathbf{u}=\mathbf{R}(\tilde{\boldsymbol{\alpha}}){}^C\hat{\mathbf{u}}\approx(\mathbf{I}-\lfloor\tilde{\boldsymbol{\alpha}}\times\rfloor){}^C\hat{\mathbf{u}}$, where ${}^C\hat{\mathbf{u}}$ is the measured unit vector. The corresponding quaternion formed by this unit vector becomes: $\bar{g}=\begin{bmatrix}{}^C\mathbf{u}^\top\sin(\phi) & \cos(\phi)\end{bmatrix}^\top$, where the angle component ϕ is unknown. As the cross product of the vector components is invariant to ϕ, we use this as the measurement: $\mathbf{y}=\mathbf{u}\times\mathbf{w}=\mathbf{0}$. Based on $\mathbf{w}=\hat{\mathbf{w}}+(\mathbf{I}\hat{c}+\lfloor\hat{\mathbf{w}}\times\rfloor)\tilde{\mathbf{w}}$, we have the following linearized measurement model:

$$\mathbf{y}\approx\hat{\mathbf{y}}+\mathbf{J}_{\mathbf{y}}(\tilde{\mathbf{w}})\tilde{\mathbf{w}}+\mathbf{J}_{\mathbf{y}}(\tilde{\boldsymbol{\alpha}})\tilde{\boldsymbol{\alpha}}\quad(25)$$

$$\mathbf{J}_{\mathbf{y}}(\tilde{\mathbf{w}})=\lfloor{}^C\hat{\mathbf{u}}\times\rfloor(\mathbf{I}\hat{c}+\lfloor\hat{\mathbf{w}}\times\rfloor)\quad(26)$$

$$\mathbf{J}_{\mathbf{y}}(\tilde{\boldsymbol{\alpha}})=-\lfloor{}^C\hat{\mathbf{u}}\times\rfloor\lfloor\hat{\mathbf{w}}\times\rfloor\quad(27)$$

The cross product measurements however only have 2 degrees of freedom. Performing the QR decomposition yields $\mathbf{J}_{\mathbf{y}}(\tilde{\mathbf{w}})=\mathbf{MN}$, where $\mathbf{M}^\top\mathbf{M}=\mathbf{I}$ and $\mathbf{N}$ is upper right triangular with the bottom row equal to $\mathbf{0}$. Using $\mathbf{M}^\top_{1:2}$ to denote the first two rows of $\mathbf{M}^\top$, it is possible to yield the transformed measurement $\mathbf{y}'$:

$$\mathbf{y}'\approx\underbrace{\mathbf{M}^\top_{1:2}\hat{\mathbf{y}}}_{\hat{\mathbf{y}}'}+\underbrace{\mathbf{M}^\top_{1:2}\mathbf{J}_{\mathbf{y}}(\tilde{\mathbf{w}})}_{\mathbf{J}_{\mathbf{y}}(\tilde{\mathbf{w}})'}\tilde{\mathbf{w}}+\underbrace{\mathbf{M}^\top_{1:2}\mathbf{J}_{\mathbf{y}}(\tilde{\boldsymbol{\alpha}})}_{\mathbf{J}_{\mathbf{y}}(\tilde{\boldsymbol{\alpha}})'}\tilde{\boldsymbol{\alpha}}\quad(28)$$

G. Feature Initialization

To initialize the EKF-based visual SLAM system that includes feature quaternion $\bar{f}=\begin{bmatrix}\mathbf{v}^\top & b\end{bmatrix}^\top$ in the state vector, we need to compute both the feature state estimate $\hat{\bar{f}}$ and its error covariance. To this end, when the feature is measured for the first time, we set its quaternion as:

$$\hat{\bar{f}}_0=\begin{bmatrix}\hat{\mathbf{u}}_0^\top & 0\end{bmatrix}^\top\quad(29)$$

where $\hat{\mathbf{u}}_0$ is the first observed unit vector pointing from the camera to the feature, rotated into the desired coordinate system. In this way, it is indeed possible to obtain the initial measurement provided that the camera measurements do not have infinite variance. On the other hand, in order to compute an initial covariance, we first note that:

$$\tilde{\bar{f}}=\bar{f}\otimes\hat{\bar{f}}^{-1}=\begin{bmatrix}\hat{b}\mathbf{v}-b\hat{\mathbf{v}}+\lfloor\mathbf{v}\times\rfloor\hat{\mathbf{v}}\\ \hat{b}b+\mathbf{v}^\top\hat{\mathbf{v}}\end{bmatrix}\quad(30)$$

By evaluating it at the initial quaternion $\hat{\bar{f}}_0$, we have:

$$\tilde{\bar{f}}_0=\begin{bmatrix}-b\hat{\mathbf{u}}_0+\lfloor\mathbf{v}\times\rfloor\hat{\mathbf{u}}_0\\ \mathbf{v}^\top\hat{\mathbf{u}}_0\end{bmatrix}=\begin{bmatrix}-\cos(\phi)\hat{\mathbf{u}}_0+\sin(\phi)\lfloor\mathbf{u}\times\rfloor\hat{\mathbf{u}}_0\\ \sin(\phi)\mathbf{u}^\top\hat{\mathbf{u}}_0\end{bmatrix}\quad(31)$$

From above, the norm of the initial error quaternion can be computed as follows:

$$\begin{aligned}\|\tilde{\mathbf{v}}_0\|^2&=\left(-\cos(\phi)\hat{\mathbf{u}}_0^\top-\sin(\phi)\hat{\mathbf{u}}_0^\top\lfloor\mathbf{u}\times\rfloor\right)\\&\quad\left(-\cos(\phi)\hat{\mathbf{u}}_0+\sin(\phi)\lfloor\mathbf{u}\times\rfloor\hat{\mathbf{u}}_0\right)&(32)\\&=\cos^2(\phi)-\sin^2(\phi)\hat{\mathbf{u}}_0^\top\lfloor\mathbf{u}\times\rfloor^2\hat{\mathbf{u}}_0&(33)\\&=\cos^2(\phi)-\sin^2(\phi)\hat{\mathbf{u}}_0^\top\left(\mathbf{u}\mathbf{u}^\top-\mathbf{I}\right)\hat{\mathbf{u}}_0&(34)\\&=1-\sin^2(\phi)\left(\hat{\mathbf{u}}_0^T\mathbf{u}\right)\left(\mathbf{u}^T\hat{\mathbf{u}}_0\right)&(35)\\&=1-\sin^2(\phi)\cos^2(\theta)&(36)\end{aligned}$$

where θ is the angle between $\mathbf{u}$ and $\hat{\mathbf{u}}_0$.

This is an important result, because it means when a feature quaternion is initialized with $\hat{b}_0=0$ as suggested here, the error states are *bounded* by a sphere with radius 1. One condition where this bound is reached is when $\sin(\phi)=0$, which means that the feature is located at the origin of

the chosen coordinate system. The other condition is when $\cos(\phi) = 0$, which means that $\mathbf{u}$ and $\hat{\mathbf{u}}_0$ are perpendicular. Note that this bound applies for any $\mathbf{u}$ and $\hat{\mathbf{u}}_0$. As such, a finite covariance can be assigned to the quaternion error even when no information is available from observations. However, having a consistent distribution for $\tilde{\mathbf{v}}_0$ does not mean that the residual error distribution will be consistent if an arbitrary $\hat{\mathbf{u}}_0$ is chosen because the measurement equation is nonlinear. The initialization in (29) ensures that the residuals will be small at first which negates the nonlinearities in certain situations. In practice, the proposed initialization procedure works well in situations where the distance from the camera to the origin of the global coordinate frame is small relative to the distance from the landmark to the camera (see Fig. 2), but in the opposite case the non-linearities seem to prevent consistent recursive estimation.

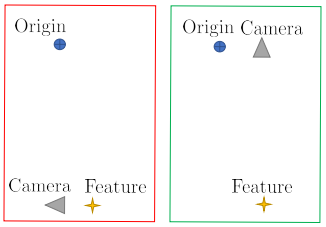

Fig. 2: Bad (left) and good (right) geometries for initialization.

It is important to note that in the Cartesian representation, the feature position errors have infinite variance prior to the initialization procedure. With inverse depth representations, the initial depth also has infinite variance, but a relatively small variance assigned to $\tilde{\rho}_0$ accounts for most depths that will ever be practically encountered. With the quaternion representation however, not only is the initial error state covariance finite, it is independent from the geometry of the problem. This is validated in the numerical experiment in Section IV, where an actual value for the covariance is calculated numerically, while analytically determining the covariance will be for future work. It should be also noted that while the distribution of quaternion error states has finite variance, it is not necessarily Gaussian. So, while the error states can be initialized with the correct variance in the EKF, there is still an approximation of the distribution type.

IV. Numerical Experiments

Three numerical experiments were performed to validate the results derived in this paper: (1) The first test is an extremely simple bundle adjustment problem in which a single feature point is estimated from known camera poses. While this is not a SLAM problem because the pose is not jointly estimated, it demonstrates the differences and similarities between two popular feature representations and the proposed quaternion parameterization when it comes to convergence and consistency. (2) The second test is designed to demonstrate that the error-states of an initial feature quaternion with no prior information is bounded as derived in the preceding section. (3) In the third test, simulated inertial measurement unit (IMU) and feature points are used to evaluate three robocentric visual-inertial monocular SLAM filters that use different point feature representations. An inverse depth parameterization is included as benchmark and two quaternion representations are presented to demonstrate the modeling advocated in this work.

A. Test 1: Batch Estimation

It is insightful to consider a simple example of triangulating a single feature from two camera poses that are separated by a baseline distance b as illustrated in Fig. 3. The cameras are aligned with the global coordinate system, i.e., ${}^{C}_{G}\mathbf{R} = \mathbf{I}$. In each trial of the experiment, each camera produces noisy 2D measurements of the feature projection [see (5)]:

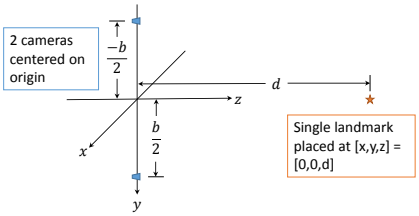

Fig. 3: Triangulation layout.

$$\mathbf{y}_C = \left(\mathbf{p}_{f_{x,y}} - \mathbf{p}_{C_{x,y}}\right) / \left(\mathbf{p}_{f_z} - \mathbf{p}_{C_z}\right) + \boldsymbol{\eta} \tag{37}$$

where $\boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}\sigma^2)$. The observations from both cameras are stacked to form the vector $\mathbf{y}$. For each trial of the experiment, the measurements from the two cameras are used to perform maximum likelihood estimation (MLE) of the feature point position using the Cartesian, inverse depth, and quaternion representations. To show convergence of the estimators, they are each initialized with a depth of $\hat{d}_0 = 10d$, and the MLE is performed using a Levenberg-Marquardt (LM) algorithm. In each iteration of the algorithm, a correction to the feature state $\mathbf{x}$ is computed from:

$$\tilde{\mathbf{x}} = \left(\mathbf{J}_{\tilde{\mathbf{x}}}(\mathbf{y})^\top \mathbf{J}_{\tilde{\mathbf{x}}}(\mathbf{y}) + \mathbf{I}\lambda\right)^{-1} \mathbf{J}_{\tilde{\mathbf{x}}}(\mathbf{y})^\top \left(\mathbf{y} - \hat{\mathbf{y}}(\hat{\mathbf{x}})\right) \tag{38}$$

where the feature for each parameterization is given by:

	Cartesian	Inverse Depth	Quaternion
$\mathbf{x}$:	$\begin{bmatrix} x & y & z \end{bmatrix}^\top$	$\begin{bmatrix} \alpha & \beta & \rho \end{bmatrix}^\top$	$\begin{bmatrix} v_x & v_y & v_z & c \end{bmatrix}^\top$
$\tilde{\mathbf{x}}$:	$\begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{z} \end{bmatrix}^\top$	$\begin{bmatrix} \tilde{\alpha} & \tilde{\beta} & \tilde{\rho} \end{bmatrix}^\top$	$\begin{bmatrix} \tilde{v}_x & \tilde{v}_y & \tilde{v}_z \end{bmatrix}^\top$

The relaxation factor λ is set to 1e-5, and each experiment runs the LM algorithm for 100 iterations. The experiment is run 500 times for depths of $d = \{1e1, 1e2, 1e3\}$. The error states are calculated from the true and estimated Cartesian coordinate, inverse depth parameters, and quaternion. The covariance of the estimated error states is computed from the information matrix evaluated at the converged estimates:

$$\hat{\mathbf{P}} = \left(\mathbf{J}_{\tilde{\mathbf{x}}}(\mathbf{y})^\top \mathbf{J}_{\tilde{\mathbf{x}}}(\mathbf{y})\right)^{-1} \sigma^2 \tag{39}$$

The final error values in the form native to each parameterization as well as the predicted 3σ error bounds computed from the covariance are shown in Fig. 4.[3] As expected, all parameterizations converge to a consistent estimate when the depth-to-baseline ratio d/b is only 10. As this ratio is increased to 100 and 1,000, the Cartesian representation fails to converge in the given number of iterations. Both the inverse-depth and quaternion parameterizations converge to consistent estimates.

[3]Note that although this was a 3D experiment, the results are shown as 2D cross sections to improve visibility.

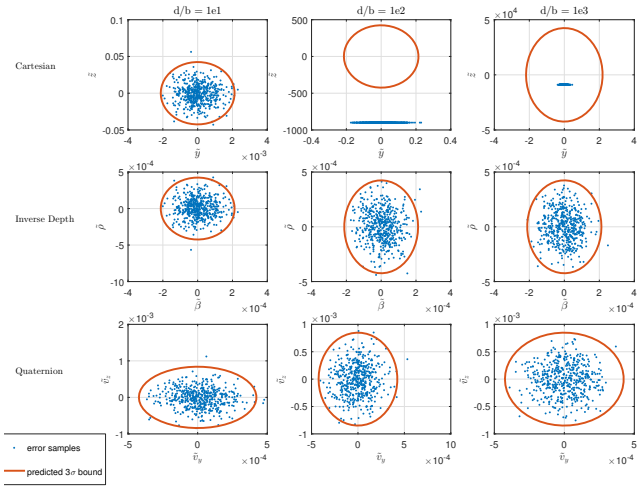

Fig. 4: Native parameter errors $b = 1$, σ =1e-4.

The inverse-depth and quaternion estimates were also transformed back into Cartesian coordinates, along with their associated covariances based on a linearized transformation from native-to-Cartesian coordinates. It was observed in our tests that if the linearization errors are not excessive, both inverse-depth and quaternion estimates can be used to consistently produce a Cartesian location if this is desired. When the noise is increased beyond a certain level relative to d/b, the linearization errors become too large and this transformation begins to break, while fully understanding this phenomenon is left as a subject for future work.

B. Test 2: Initialization Error Bound

A simple test is conducted to verify the error bound of the initial feature quaternion with no prior information as described in Section III-G. To do this, we perform the following procedure:

`Repeat N times:`

1) `Generate random feature quaternion` $\bar{f} = \begin{bmatrix}\mathbf{v}^\top & b\end{bmatrix}^\top = \begin{bmatrix}\mathbf{u}^\top \sin(\phi) & \cos(\phi)\end{bmatrix}^\top$.
2) `Initialize feature quaternion estimate with` $\hat{\mathbf{v}} = \hat{\mathbf{u}}_0$ `and` $\hat{b} = 0$.
3) `Compute vector component of error quaternion` $\tilde{\mathbf{v}}$.

The random quaternions are generated by creating $\mathbf{u}$ samples with a uniformly distributed direction, and ϕ samples uniformly distributed between -2π and 2π, which are then used to compute (6). The $\hat{\mathbf{u}}_0$ samples are also uniformly distributed in direction. The results of this test with $N = 100$ are shown in Fig. 5. As predicted, all of the error states lie within a unit sphere. The experiment is repeated with $N = 1e6$, and the resulting sample covariance of the error states is $\mathbb{E}\left[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^\top\right] \approxeq 0.277\mathbf{I}$.

C. Test 3: EKF-based VIO

To illustrate how the quaternion representation will be used in practice, a monocular EKF-based VIO system is implemented. Note that our goal of this experiment is to validate that (i) a full EKF can be built with quaternion features, (ii) the un-delayed initialization can work, and (iii)

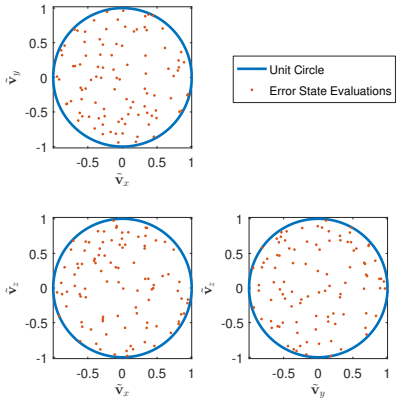

Fig. 5: The x, y and z components of the quaternion error states after being initialized without prior information.

the same results can be obtained regardless of the coordinate system used for the quaternion features. The IMU (body) state vector for the EKF consists of the rotation quaternion, the gyroscope bias, the velocity, the accelerometer bias, and the position: $\mathbf{x}_B = \begin{bmatrix}{}^B_G\bar{q}^\top & \mathbf{b}_\omega^\top & {}^G\mathbf{v}_B^\top & \mathbf{b}_a^\top & {}^G\mathbf{p}_B^\top\end{bmatrix}^\top$. The IMU state dynamics is well known and can be found in [7] and many others. The system state vector is augmented with features which are defined relative to the body-frame (which is assumed co-located with the camera frame) using three different parameterizations:

- Inverse-Depth VIO: The feature positions in the camera frame are parameterized by: ${}^C\mathbf{p}_f = \begin{bmatrix}\alpha & \beta & 1\end{bmatrix}^\top \rho^{-1}$, and the state vector is augmented with inverse depth parameters for each feature.
- Quaternions in Camera Frame: The features are parameterized by $\bar{f} = \begin{bmatrix}\mathbf{v}^\top & c\end{bmatrix}^\top$ so that ${}^C\mathbf{p}_f = \mathbf{v}c^{-1}$. Here, the feature points are initialized as $\bar{f}_0 = \begin{bmatrix}{}^C\mathbf{u}_0^\top & 0\end{bmatrix}^\top$, and assigned an initial covariance of $\mathbb{E}\left[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^\top\right] \approxeq 0.277\mathbf{I}$.
- Quaternions in Camera/Global Frame: The features are parameterized by $\bar{f} = \begin{bmatrix}\mathbf{v}^\top & c\end{bmatrix}^\top$ so that ${}^G\mathbf{p}_{f/B} = \mathbf{v}c^{-1}$ (to clarify, ${}^G\mathbf{p}_{f/B} = {}^G\mathbf{p}_{f/G} - {}^G\mathbf{p}_{B/G}$). Here, the feature points are initialized as $\bar{f}_0 = \begin{bmatrix}({}^C_G\mathbf{R}_0^\top {}^C\mathbf{u}_0)^\top & 0\end{bmatrix}^\top$, and assigned an initial covariance of $\mathbb{E}\left[\tilde{\mathbf{v}}\tilde{\mathbf{v}}^\top\right] \approxeq 0.277\mathbf{I}$.

1) Simulations: To generate synthetic data with which to evaluate the above algorithms, a low fidelity quadcoptor model is created and equipped with a simulated ADIS16448 IMU and a monocular camera providing feature point correspondences. The ADIS16448 uses the noise and bias repeatability specifications listed in [25], but the higher order errors are turned off since they are not modeled in the EKF. The camera model produces feature points on the ground so that there would be approximately 20 of them in view if the camera is pointed straight down. The noise on the feature measurements is small (approximately 0.04 deg 1σ). The 40° field-of-view camera is pointed slightly down from the horizon for the majority of the trajectories when the quadcoptor flying at altitudes between 50 to 140 m. Several trajectories are shown in Fig. 6.

2) Results: Ten trajectories were run, and then the data from each was passed through each VIO filters. The po-

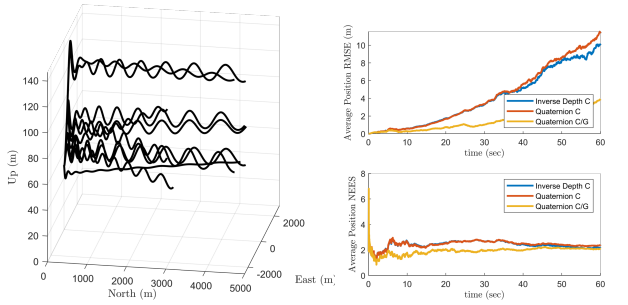

Fig. 6: VIO results of trajectories and position RMSE and NEES.

sition RMSE ($\sqrt{{}^G\tilde{\mathbf{p}}_B^\top {}^G\tilde{\mathbf{p}}_B}$) and NEES ($\sqrt{{}^G\tilde{\mathbf{p}}_B^\top \mathbf{P}^{-1} {}^G\tilde{\mathbf{p}}_B}$) are shown in Fig. 6. From this result, while there is no statistically significant difference in accuracy or consistency between the three feature parameterizations, it does verify the efficacy of the quaternion feature parameterization and the initialization procedure.

V. Conclusions and Future Work

Unit quaternions are ideally suited for representing the homogeneous coordinates of feature points at any scale. In this paper, we have shown that updating a feature quaternion by quaternion multiplication of an error quaternion can produce changes in the 3D feature position that scale with the magnitude of the 3D position. Therefore, the error quaternion only has minimum 3 degrees of freedom and can be linearized as in rotation estimation problems. A simple expression for initializing the feature quaternion was presented, and it was shown that when this initialization is used the covariance of the quaternion error states is bounded and the bound is the same regardless of the scene geometry. This means that unlike other feature representations, the feature quaternion error covariance can be set to a known and finite value for use in a covariance-form Kalman filter. Numerical experiments were performed that validate the analytical results, showing that quaternion representations are good alternatives of inverse depth, e.g., in VIO systems.

There is a large amount of future work that can be pursued from the results presented here. The unit-quaternion feature parameterization can be applied in all non-linear SLAM, bundle adjustment, and visual odometry approaches that make use of feature points observed with bearing-only measurements. In addition to the applications, theoretical results remain to be obtained. These include analysis of the linearity, derivation of an analytical expression for the initial feature quaternion error covariance, and observability analysis of systems using the feature quaternion parameterization.

References

[1] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[2] J. Civera, A. Davison, and J. Montiel, "Inverse Depth to Depth Conversion for Monocular SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Roma, Italy, April 2007.

[3] J. Fuentes-Pacheco, J. Ruiz-Ascencio, and J. Rendn-Mancha, "Visual simultaneous localization and mapping: a survey," *Artificial Intelligence Review*, pp. 1–27, 2012.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 15, no. 2, pp. 1147–1163, 2015.

[5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[6] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment – a modern synthesis," *Lecture Notes in Computer Science*, vol. 1883, pp. 298–375, Jan. 2000.

[7] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proceedings of the IEEE International Conference on Robotics and Automation*, Rome, Italy, Apr. 10–14, 2007, pp. 3565–3572.

[8] M. Li and A. I. Mourikis, "Improving the accuracy of EKF-based visual-inertial odometry," in *Proc. of the IEEE International Conference on Robotics and Automation*, St. Paul, MN, May 14–18, 2012, pp. 828–835.

[9] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis, "Consistency analysis and improvement of vision-aided inertial navigation," *IEEE Transactions on Robotics*, vol. PP, no. 99, pp. 1–19, 2013.

[10] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Madrid, Spain, Oct. 1-5, 2018, (to appear).

[11] T. Lemaire, S. Lacroix, and J. Sola, "A practical 3D bearing-only SLAM algorithm," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Alberta, Canada, Aug. 2–6, 2005, pp. 2449–2454.

[12] T. Vidal-Calleja, M. Bryson, S. Sukkarieh, A. Sanfeliu, and J. Andrade-Cetto, "On the observability of bearing only SLAM," in *Proc. of the IEEE International Conference on Robotics and Automation*, Roma, Italy, Apr. 10–14, 2007, pp. 4114–4118.

[13] S. I. Roumeliotis, G. S. Sukhatme, and G. A. Bekey, "Circumventing dynamic modeling: Evaluation of the error-state Kalman filter applied to mobile robot localization," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, Detroit, MI, May 10-15 1999, pp. 1656–1663.

[14] J. L. Crassidis, F. L. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of guidance control and dynamics*, vol. 30, no. 1, p. 12, 2007.

[15] J. B. Kuipers *et al.*, *Quaternions and rotation sequences*. Princeton university press Princeton, 1999, vol. 66.

[16] T. Lemaire, C. Berger, I. Jung, and S. Lacroix, "Vision-Based SLAM: Stereo and Monocular Approaches," *International Journal of Computer Vision*, vol. 74, no. 3, pp. 343–364, 2007.

[17] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, Oct. 2008.

[18] R. Mur-Artal and J. D. Tardos, "Orb-slam2: an open-source slam system for monocular, stereo and rgb-d cameras," *arXiv preprint arXiv:1610.06475*, 2016.

[19] J. Montiel, J. Civera, and A. Davison, "Unified inverse depth parametrization for monocular SLAM," in *Proc. of Robotics: Science and Systems*, Philadelphia, PA, Aug. 16-19 2006, pp. 81–88.

[20] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct EKF-based approach," in *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hamburg, Germany, Sep. 28 - Oct. 2, 2015, pp. 298–304.

[21] D. P. Koch, D. O. Wheeler, R. Beard, T. McLain, and K. M. Brink, "Relative multiplicative extended Kalman filter for observable gps-denied navigation," Brigham Young University, Tech. Rep., 2017. [Online]. Available: http://hdl.lib.byu.edu/1877/3102

[22] J. Sola, "Consistency of the monocular EKF-SLAM algorithm for three different landmark parametrizations," in *Proc. of the IEEE International Conference on Robotics and Automation*, Anchorage, AK, May 3-7, 2010, pp. 3513–3518.

[23] M. D. Shuster, "A survey of attitude representations," *Journal of the Astronautical Sciences*, vol. 41, no. 4, pp. 439–517, Oct.-Dec. 1993.

[24] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005.

[25] *Data Sheet ADIS16448*, Analog Devices, Inc., 2017. [Online]. Available: www.analog.com