Robust Legged Robot State Estimation
Using Factor Graph Optimization

David Wisth, Marco Camurri, Maurice Fallon

Abstract—Legged robots, specifically quadrupeds, are becoming increasingly attractive for industrial applications such as inspection. However, to leave the laboratory and to become useful to an end user requires reliability in harsh conditions. From the perspective of perception, it is essential to be able to accurately estimate the robot’s state despite challenges such as uneven or slippery terrain, textureless and reflective scenes, as well as dynamic camera occlusions. We are motivated to reduce the dependency on foot contact classifications, which fail when slipping, and to reduce position drift during dynamic motions such as trotting. To this end, we present a factor graph optimization method for state estimation which tightly fuses and smooths inertial navigation, leg odometry and visual odometry. The effectiveness of the approach is demonstrated using the ANYmal quadruped robot navigating in a realistic outdoor industrial environment. This experiment included trotting, walking, crossing obstacles and ascending a staircase. The proposed approach decreased the relative position error by up to 55% and absolute position error by 76% compared to kinematic-inertial odometry.

I. INTRODUCTION

For legged robots to become truly autonomous and useful they must have a consistent and accurate understanding of their location in the world. This is essential for almost every aspect of robot navigation, including control, motion generation, path planning, and local mapping.

Legged robots pose unique challenges to state estimation. First, the dynamic motions generated by the robot footsteps can induce motion blur on camera images as well as slippage or flexibility in the kinematics. Second, the strict real-time requirements of legged locomotion require low latency, high frequency estimates which are robust. Third, the sensor messages are heterogeneous, with different frequencies and latencies. Finally, the conditions where legged robots are expected to operate are far from ideal: poorly lit or textureless areas, self-similar structures, muddy or slippery grounds are some examples.

For these reasons, legged robotics have traditionally relied on filter-based state estimation, using proprioceptive inputs (IMUs, force/torque sensors and joint encoders) [1], [2], [3]. While these approaches give reliable and high frequency estimates, they are limited in their ability to reject linear and angular position drift.

For statically stable walking, leg odometry drift is low enough that terrain mapping can be used for continuous footstep planning and execution [4]. However for dynamic locomotion, position drift is much higher which makes such mapping ineffective, as illustrated in Fig. 2. The result of this is that we do not know what the shape of the terrain is under the robot due to drift and cannot properly plan motions.

To overcome this limitation, some previous works have incorporated exteroceptive inputs (cameras and LIDAR) into filtering estimators in a loosely coupled fashion. This has been successfully demonstrated on legged machines operating in field experiments [6], [7]. However, because these filters marginalize all previous states of the robot, it is not possible to fully exploit a (recent) history of measurements, as smoothing methods can.

Research into smoothing approaches applied to Visual-Inertial Navigation Systems (VINS) is now well established in the Micro-Aerial-Vehicle (MAV) community. On a MAV this approach has been successful due to careful time synchronization of the IMU and cameras, smooth vehicle motion and also the absence of the challenges of articulated legged machines.

A recent work by Hartley et al. [8] demonstrated that a VINS approach could be adapted to inertial-legged legged robots — in their case, a biped. Their initial results were promising, but only tested in a controlled scenario. Additionally, vision was integrated as relative pose constraints, rather than incorporating the feature residuals directly into the optimization.

Contribution

This paper aims to progress the deployment of state estimation smoothing methods in realistic application scenarios.
A modular sensor head performs the fusion of very high 
primarily driven by inertial predictions with VO updates: 
Kalman Filter (EKF) with an error state formulation de-
veloped for the Boston Dynamics LS3. The system was 
primarily driven by inertial predictions with VO updates: 
a modular sensor head performs the fusion of very high 
quality tactical grade IMU with two hardware synchronized 
sterro cameras. The leg odometry (fused with an additional 
navigation grade IMU in the body) was used only in case 
of VO failure. Their extensive evaluation (over several km) 
achieved 1% error per distance traveled.

Nobili et al. [7] recently presented a state estimator for the 
HyQ quadruped robot which used an EKF to combine iner-
tial, kinematic, visual, and LIDAR measurements. In contrast 
to [6], the EKF was driven by an inertial process model with 
the primary corrections coming from leg odometry (synchron-
ized in EtherCAT) at the nominal control rate (1 kHz). The 
VO updates and an ICP-based matching algorithm were run 
on a separate computer at lower frequency and integrated 
into the estimator when available. This allowed the use of 
the estimator inside the control loop and has been recently 
demonstrated in dynamic motions with local mapping [11].

B. Smoothing Approaches

There has been a significant body of work on visual 
inertial navigation, especially for use with MAVs. A recent 
benchmark paper [12] evaluated a number of state-of-the-
art methods on the EuRoC dataset [13]. The maturity of 
the field was highlighted by the fact that many algorithms 
achieved an average Relative Position Error (RPE) of less 
than 20 cm per 35 m traveled. The authors concluded that 
the best performing algorithms were OKVIS [14], ROVIO 
[15], and VINS-Mono [16]. All of these algorithms perform 
windowed optimization to achieve the most accurate state 
estimate while bounding computation time. An exception 
was SVO+GTSAM [17] which loosely coupled the SVO 
visual odometry algorithm with IMU data using iSAM2 as 
the smoothing back-end [18].

The methods were, however, typically designed assuming 
that the IMU and cameras were synchronized in hardware, 
and that the vehicle motions were smooth. They are difficult 
to implement on legged platforms due to their hardware 
complexity and the high vibrations caused by locomotion.

As mentioned above, the first approach to use smooth-
ing/optimization on legged robots was the work of Hartley 
et al. [8] which presented a fusion of kinematic, inertial, 
and visual information, again using iSAM2 as the smoothing 
back-end. They presented a mathematical framework 
to model contact points as landmarks in the environment, 
similar to [11]. Their system incorporated contact information 
from only a single contact point at a time, and directly 
integrates (as pose constraints) the relative motion estimate 
of the SVO2 [19] algorithm. The approach was tested using 
inertial and visual input from a MultiSense S7 sensor (which 
was hardware synchronized) mounted on a Cassie biped 
(from Agility Robotics). A short indoor experiment of 60 s 
showed that adding vision into the optimization reduced the 
relative position error.

Our approach aims to combine best practice from the 
mature field of VINS – including windowed optimization and 
tight integration of visual features – with legged odometry to 
provide robust state estimation for legged robots. In Section
VI] we show our system outperforming both kinematic-inertial and visual-inertial approaches in large-scale outdoor urban and industrial experiments.

III. PROBLEM STATEMENT

We wish to track the linear position and velocity of a 12 Degrees of Freedom (DoF) legged robot equipped with an industrial grade MEMS IMU, an RGB/IR camera, joint encoders and torque sensors. The sensor specifications are detailed in Table I.

In Fig. 3 we provide a schematic of the reference frames involved. The pose of the robot’s base $B$ expressed in the fixed-world, inertial frame $\mathbf{w}$ is defined as:

$$ T_{\mathbf{w}B} = \begin{bmatrix} R_{\mathbf{w}B} & aP_{\mathbf{w}B} \\ 0 & 1 \end{bmatrix} \in SE(3) $$

The IMU and camera sensing frames are $I$ and $C$, respectively. The relative transformations between $B$, $I$ and $C$ are assumed to be known from CAD design. The location of a foot in base coordinates is expressed as $aP_{\mathbf{w}B}$.

A. State Definition

Borrowing the notation from [17], we define the state of the system at time $t_i$ as:

$$ x_i = [R_i, p_i, v_i, b_i] $$

(1)

where the couple $(R_i, p_i)$ expresses the robot pose and $v_i \in \mathbb{R}^3$ is the robot linear velocity. As is common in this field, the stack of gyro and accelerometer IMU biases $b_i = [b^g_\omega, b^a_\omega] \in \mathbb{R}^6$ replaces the angular velocity, which is directly measured by the IMU.

Let $K_k$ be the set of camera keyframe indices up to time $t_k \in \mathbb{R}$. We assume that for each keyframe image $C_i$ (with $i \in K_k$) a number of landmark points $p_m$ are visible, where $j \in M_i \subseteq M$; $M_i$ indicates the set of landmark indices visible from keyframe $C_i$ out of the full set of landmarks, $M$. We then define the objective of our estimation problem $X_k$ as the history of robot states and landmarks detected up to $t_k$:

$$ X_k = \bigcup_{i \in K_k} \{x_i\} \cup \bigcup_{j \in M_i} \{m_j\} $$

(2)

C. Maximum-A-Posteriori Estimation

The aim of the factor graph framework is to maximize the posterior of the state $X_k$ given the inputs $Z_k$:

$$ X_k^* = \arg \max_{X_k} p(X_k | Z_k) \propto p(X_0)p(Z_k | X_k) $$

(4)

![Fig. 3. Reference frames conventions. The world frame $\mathbf{w}$ is fixed to earth, while the base frame $B$, the camera’s optical frame $C$, and the IMU frame, $I$ are rigidly attached to the robot’s chassis. When a foot touches the ground (e.g., the Right Front, RF), a contact frame $K$ (perpendicular to the ground and parallel to $\mathbf{w}$’s $y$-axis) is defined. The projection of a landmark point $m$ onto the image plane is $\pi(m)$.](image)

![Fig. 4. The sensors inputs are images, IMU measurements, and joint states, which are in general unsynchronized with each other.](image)
The first is the difference between the measured landmark location into image coordinates, as the error between the estimated state frame, typically during initialization. The residual is defined of the system (i.e., position and yaw) to a fixed reference frame. Prior Factors

In Fig. 5. In the next section we define each residual of (6).

where the residuals are from: landmark prior, IMU, state prior, camera and leg odometry factors, respectively. These factors will be used to create the factor graph structure shown in Fig. 5. In the next section we define each residual of (9).

IV. FACTOR DEFINITIONS

A. Prior Factors

Prior factors are used to anchor the unobservable modes of the system (i.e., position and yaw) to a fixed reference frame, typically during initialization. The residual is defined as the error between the estimated state $x_0$ and the prior $x_{p_0}$:

$$r_0(x_0, Z) = (\Phi(T_0^{-1}T_{p_0}), v_{p_0}, b_{p_0} - b_{p_0})$$

where $\Phi : SE(3) \rightarrow \mathbb{R}^6$ is the lifting operator [17].

B. Visual Odometry Factors

The visual odometry residual consists of two components. The first is the difference between the measured landmark pixel location, $\left(u_{i,j}, v_{i,j}\right)$, and the re-projection of the estimated landmark location into image coordinates, $\left(\pi_u, \pi_v\right)$ using the standard radial-tangential distortion model. The residual is defined as:

$$r_m = \left(\pi_u(R_i, p_i, m_j) - u_{i,j}, \pi_v(R_i, p_i, m_j) - v_{i,j}\right)$$

The second is the error between the prior on the landmark location $m_{j,0}$ and the estimated landmark location $m_j$:

$$r_{m_{j,0}} = m_j - m_{j,0}$$

The landmark prior helps to constrain the estimated landmark location in space if it is not well-constrained by the measurements, and is explained further in Section V.

C. Preintegrated IMU Factors

We use the IMU preintegration algorithm described by Forster [17]. This approach preintegrates the IMU measurements between nodes in the factor graph to provide high frequency state updates between optimization steps. The preintegrated IMU measurements are then used to create a new IMU factor between two consecutive keyframes. This will use an error term of the form:

$$r_{\Delta t} = [r_{\Delta R_{\Delta t}}, r_{\Delta p_{\Delta t}}, r_{\Delta v_{\Delta t}}]$$

where $\Delta t$ are the IMU measurements between times $i-1$ and $i$. The individual elements of the residual are defined as:

$$r_{\Delta R_{\Delta t}} = \log \left(\Delta R_{\Delta t} \cdot \Delta R_{\Delta t}^T\right)$$

$$r_{\Delta p_{\Delta t}} = R_{t-1} \cdot \left(p_i - p_{i-1} - v_{i-1} \Delta t - \frac{1}{2} g \Delta t^2\right)$$

$$r_{\Delta v_{\Delta t}} = R_{t-1} \cdot \left(v_i - v_{i-1} - g \Delta t\right)$$

$$r_{\Delta b_{\Delta t}} = b_i - b_{i-1}$$

where $\Delta R_{\Delta t}$, $\Delta p_{\Delta t}$, $\Delta v_{\Delta t}$ are the preintegrated IMU measurements defined in [17].

D. Leg Odometry Factors

Leg Odometry (LO) is the process of estimating the incremental motion of a walking robot given its kinematic sensing as well as information about the contact the robot’s legs make with the ground. The main assumption behind LO measurements is that the absolute velocity of a contact point is zero. This is assumed to occur when the Ground Reaction Force (GRF) at a contact point is inside a hypothetical friction cone. Torques at contact points are neglected as quadruped robots’ feet are idealized as points.

Since the GRF, the terrain inclination, and the coefficient of static friction are typically unknown, an appropriate fusion of the dynamic model, kinematics, and IMU is required. In this work, we rely on the state estimator provided by ANYbotics, the Two-State Implicit Filter (TSIF) [3]. We resolve the estimate from the TSIF into estimates of relative base motion and use these to formulate additional factors to constrain the robot’s motion.

So as to synchronize kinematic measurements to the camera timestamps, when given a new keyframe at time
The proposed state estimation architecture consists of measurement handlers which process the sensor input and the current state from the optimizer to create the factors. These factors are then smoothed using iSAM2 for incremental factor graph optimization.

$t_i$ (with $i \in K_k$), we extract from the filter the base pose estimate immediately before and after $t_{i-1}$ and also $t_i$. The corresponding estimated poses $T_{i-1}$, $T_i$ are then computed via linear/slerp interpolation and used to produce the relative pose constraint:

$$r_{Q_{\Delta t}} = \Phi \left( (T_{i-1}^{-1} T_i)^{-1} T_{i-1}^{-1} T_i \right)$$

where the covariance $\Sigma_{Q_{\Delta t}}$ is provided by the filter, and $\Phi$ is the lifting operator defined in [17].

E. Zero Velocity Update Factors

To limit drift and factor graph growth when the robot is stationary, we detect zero velocity motion from camera frames by calculating the average feature motion:

$$\overline{\Delta x} = \frac{\sum_{i=k}^N \sqrt{(u_i - u_{i-1})^2 + (v_i - v_{i-1})^2}}{N + 1}$$

If $\overline{\Delta x}$ is below a certain threshold $\beta$ over $N$ successive frames, we stop adding image landmark measurements to the graph and simply add a zero relative pose factor to the graph of the same form as (15).

V. IMPLEMENTATION

The factor graph optimization was implemented using the iSAM2 incremental optimization library [18]. The structure of the system is shown in Fig. 6. The algorithm consists of a series of measurement handlers running in separate threads that process the different sensor inputs. When a new node is created (e.g., when an image measurement is received) each of the measurement handlers adds a new factor to the graph. The output of the factor graph optimization is then fed back to the measurement handlers.

A. Synchronization

Given two consecutive keyframe indices $i-1, i \in K_k$, preintegrating all IMU measurements between times $t_{i-1}$ and $t_i$ would result in an incorrect motion estimation as the IMU measurement timestamps may not be aligned with the image frames (see Fig. 7). To avoid this, we correct the $\Delta t$ of the IMU measurements directly before and after the camera image timestamp $t_i$:

$$\Delta t_{i-1} = t_{IMU} + \Delta t_{IMU} - t_i$$
$$\Delta t_i = t_i - t_{IMU}$$

where we assume constant acceleration and angular velocity between IMU measurements.

B. Visual Feature Tracking

A core component of the VILENS system is the tight integration of visual features into the optimization. It can provide lower drift state estimates by tracking and optimizing the robot state and landmark positions over many observations. Our visual feature tracking method is based upon the robust pixel-based tracking approaches used in ROVIO [20] and VINS-Mono [19].

Visual features are first detected using the Harris Corner Detector and then tracked through successive images using the Kanade-Lucas-Tomasi feature tracker. This method provides sub-pixel accuracy and is well-suited to the constrained but jerky motions typical of a legged robot. After the tracking, outliers are rejected using RANSAC with a fundamental matrix model similar to [16]. New features are then detected in the image to maintain a minimum number of tracked features. These new features are constrained to be a minimum distance (in image space) from existing features, to ensure an even distribution.

To limit the graph growth, we estimate the location of a feature in the world frame only after it has been observed more than $N_{\text{obs}} = 30$ times. When depth is available, it is used for the initial landmark location estimate. When only monocular data are available, we triangulate the landmark location $\mathbf{m}_{i,0}$ using the last $N_{\text{obs}}$ frames with the Direct Linear Transformation (DLT) algorithm from [21]. If the landmark is successfully triangulated with a depth smaller than $d_{\text{max}} = 50\, \text{m}$ then this and successive measurements of the same landmark are added to the graph.

C. Marginalization

An important consideration in a legged robot system is latency, and without marginalization, the time taken by iSAM2 optimization increases over time [18]. This becomes an important consideration when operating the robot for extended periods of time.

We marginalize states older than a threshold (typically, 10 s) and landmarks which are no longer observed, whilst keeping a minimum number of nodes in the factor graph. These marginalized states and landmarks are replaced with a
simple linear Gaussian factor based on the current linearization point of the node. This Gaussian factor has the same form as the residual defined in Eq. (7).

VI. EXPERIMENTAL RESULTS

In this section, we present the experimental evaluation of the proposed algorithm on three different datasets: EuRoC, Keble College and Oil Rig. The first dataset is a purely VINS dataset collected on a MAV. The other two datasets were collected using our ANYmal robot in different outdoor environments: a college campus and an industrial oil rig firefighter training facility.

A. EuRoC Dataset

To demonstrate that our approach builds upon a stand-alone VINS system, we evaluated it on a portion of the EuRoC benchmark dataset [13] and compared it to several state-of-the-art VINS algorithms including OKVIS [14], ROVIO [20], and VINS-Mono [16]. The estimated trajectory from the EuRoC V201 dataset is shown in Fig. 8.

In brief, we found that our core system can achieve comparable performance to these VINS algorithms. This demonstrates that the estimator can function without leg odometry information, which is important when that modality becomes unreliable (e.g., on slippery or soft ground).

B. Outdoor Datasets Experimental Setup

The outdoor experiments were conducted using the 12-DoF ANYmal quadruped [5] (Fig. 1). The measurements from the motors (i.e., joint states) were synchronized via EtherCAT, while the other sensors were by Network Time Protocol. The sensor configurations used for the two datasets are specified in Table I.

To generate ground truth, we collected a dense gravity-aligned prior map of the site using the commercial Leica BLK-360 3D laser scanner. Afterwards, we performed ICP localization (using the AICP algorithm [22]) against the prior map using data from ANYmal’s Velodyne VLP-16 LIDAR. This provided a ground truth trajectory with approximately 5 cm accuracy, but only at 2 Hz.

C. Keble College Dataset

The first outdoor dataset was collected in a urban environment at Keble College, Oxford, UK. The dataset consists of the robot trotting on a concrete path around a 28 m × 60 m open lawn surrounded by a residential building (Fig. 9). The main challenges were vegetation moving in the wind, long distances to visual features (>10 m), and limited angular motion, which made feature triangulation difficult.

We ran four trials each approximately 22 m in length and evaluated the mean and standard deviation of the Relative Position Error (RPE) over a 10 m distance (see Table II).

Compared to the kinematic-inertial estimator, our algorithm reduces the RPE by 15% to 55% and yaw error by 42% to 85%, as visual features tracked over many frames constrain pose drift.

D. Oil Rig Dataset

A second outdoor dataset was collected at an industrial firefighter training facility in Moreton-In-Marsh, UK (Fig. 1). The facility closely matches the locations where ANYmal is likely to be deployed in future.

This 110 m (22 min) long dataset involves the ANYmal robot trotting through the facility, climbing over a slab, and walking up a staircase into a smoke-blackened room. Challenging situations include: featureless areas, stationary
periods with intermittent motion, dynamic obstacles occluding large portions of the image, non-flat terrain traversal, and foot slip caused by a combination of mud, oil, and water on the ground (Fig. 11).

Figure 10 shows the estimated trajectory from VILENS, compared to TSIF [3] and ground truth. The Absolute Translation Error (ATE) for VILENS is 76% lower compared to the TSIF (1.65 m and 6.88 m, respectively).

Looking at the relative performance over 10 m, the error reduction is 7% for RPE and 25% for yaw (Table III). This suggests that the Oil Rig dataset is more challenging than Keble, since the accuracy at small scale is closer to the TSIF (see Section VII).

Note that performance evaluation against the VINS algorithms mentioned in Section VI-A was not possible because they either failed to initialize due to the lack of motion or diverged after a short period. This is also confirmed in Table III where we evaluated the performance of VILENS as standalone VINS system against TSIF and full VILENS. Without Leg Odometry factors, VILENS fails for all the datasets except the first 70 s of Keble 1 and Oil Rig, where it performs worse or same.

E. Timing

A summary of the key computation times in the proposed algorithm are shown in Table IV. Since the kinematic-inertial filter, the image processing, and the optimization are run in separate threads, the system is capable of outputting a high-frequency kinematic-inertial state estimate (for control purposes) at 400 Hz, and an optimized estimate incorporating visual features at approximately 10 Hz.

VII. DISCUSSION

In the previous section, we have demonstrated that VILENS outperforms kinematic-inertial and visual-inertial methods for all the datasets. However, from Table II we can see that the gap between VILENS and TSIF is not uniform across the datasets. Nonetheless, we see that the feature quality has an influence of the drift rate over small scales.

In Fig. 12 we compare the drift rate of VILENS and TSIF over different distance scales for the Keble College and the Oil Rig datasets. The latter experiment is more challenging from the perspective of exteroception and as a result the scale at which VILENS starts to outperform TSIF is larger. This is due to the tracking of poor quality features in the conditions highlighted in Fig. 11.

In future work, we are motivated to improve performance in these challenging scenarios by exploring the use of redundant or wider field-of-view cameras as well as different methods of incorporating the kinematics directly into the
factor graph. Additionally, as part of an ongoing project, we intend to test performance in soft and compliant surface materials such as sand, mud and gravel where we envisage the visual part of the estimator predominating during sinking, sliding and slipping.

VIII. CONCLUSION

In this paper, we have presented VILENS (Visual Inertial LEgged Navigation System), a robust state estimation method for legged robots based on factor graphs, which incorporates kinematic, inertial and visual information. This method outperforms the robot’s kinematic-inertial estimator and robustly estimates the robot trajectory in challenging scenarios, including textureless areas, moving occludants, reflections and slippery ground. Under the same conditions, current state-of-the-art visual-inertial algorithms diverge rapidly.

IX. ACKNOWLEDGEMENTS

This research has been conducted as part of the ANYbotics research community. It was part funded by the Innovate UK-funded ORCA Robotics Hub (EP/R026173/1) and the EU H2020 Project THING, a Royal Society University Research Fellowship (Fallon) and a Google DeepMind studentship (Wisth).

REFERENCES