Periodic SLAM: Using Cyclic Constraints to Improve the Performance of Visual-Inertial SLAM on Legged Robots

Hans Kumar[1], J. Joe Payne[2], Matthew Travers[1], Aaron M. Johnson[2], and Howie Choset[1]

Abstract— **Methods for state estimation that rely on visual information are challenging on legged robots due to rapid changes in the viewing angle of onboard cameras. In this work, we show that by leveraging structure in the way that the robot locomotes, the accuracy of visual-inertial SLAM in these challenging scenarios can be increased. We present a method that takes advantage of the underlying periodic predictability often present in the motion of legged robots to improve the performance of the feature tracking module within a visual-inertial SLAM system. Our method performs multi-session SLAM on a single robot, where each session is responsible for mapping during a distinct portion of the robot's gait cycle. Our method produces lower absolute trajectory error than several state-of-the-art methods for visual-inertial SLAM in both a simulated environment and on data collected on a quadrupedal robot executing dynamic gaits. On real-world bounding gaits, our median trajectory error was less than 35% of the error of the next best estimate provided by state-of-the-art methods.**

I. Introduction

While there has been tremendous progress in the development of state estimation and simultaneous localization and mapping (SLAM) algorithms in recent years, dynamic motion can still induce failure on even the most robust systems [1]. More specifically, methods for state estimation and SLAM that rely on visual information experience a significant decrease in the performance of visual feature tracking when there are rapid changes in the viewing angle of cameras onboard a robot.

Legged robots are of particular interest to us in this work because they are examples of dynamical systems that maintain periodic structure in their motion when executing gait-like behaviors [2]. When locomoting with dynamically stable gaits, such as walking or running on flat ground, legged robots exhibit patterns in their footfall and resulting body orientation. Rapid orientation changes, such as those caused by contact events, have typically been thought of as hindrances to performing SLAM on legged systems [1]. However, our method uses the predictability of periodic motion resulting from these events to improve the performance of estimation compared to other approaches.

In this work, we present a novel factor graph [3] design for visual-inertial SLAM that exploits the periodic predictability in the visual information obtained by a legged robot. Our approach explicitly distinguishes visual features detected during each unique section of the robot's gait cycle, when visual information is more likely to be similar. By performing visual SLAM separately on each portion of the gait cycle, we show improved performance of the feature tracking module that is critical to the success of visual SLAM. Fig. 1 shows an example of how this method could introduce three different visual SLAM sessions to track different portions of the gait cycle of a pitching legged robot. To obtain a unified SLAM estimate, our approach connects each individual visual SLAM session to one another by incorporating inertial measurement unit (IMU) measurements from the robot. Our method tightly couples visual and inertial measurements in a factor graph optimization framework to achieve greater combined performance than any individual SLAM session.

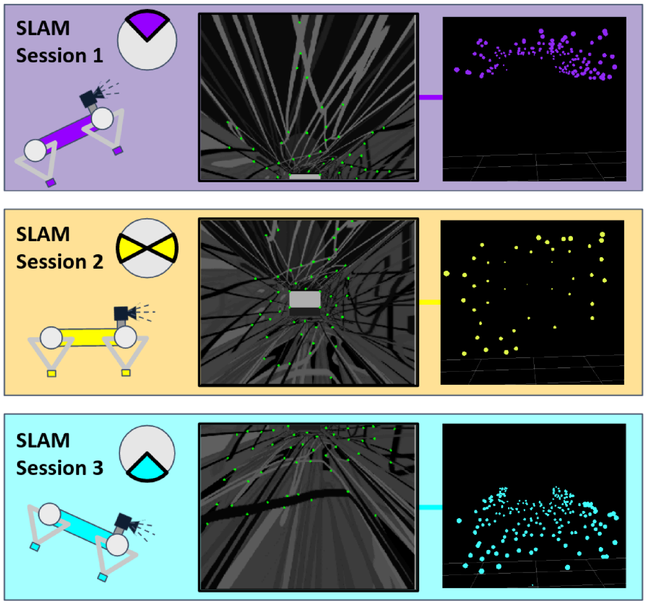

Fig. 1. Three SLAM sessions performing estimation periodically when a robot is looking upwards, forwards, and downwards. The dial represents which part of the robot's gait cycle images are taken from. The left column of images shows the result of periodic feature tracking in simulation. The right column of images shows the corresponding three-dimensional map of landmarks for each SLAM session.

Our main contribution is a novel application of multi-session visual SLAM to improve feature tracking on a single robot with predictable, oscillatory motion. Additionally we perform an evaluation of our method against several state-of-the-art SLAM implementations on a robot in simulation and on a real-world quadrupedal robot, the Ghost Robotics Minitaur [4]. Experimental results show that this approach demonstrates clear improvement in the state estimation ac-

* This material is based upon work supported by the National Science Foundation under Grant No. IIS-1704256.

[1] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, {hansk,mtravers,choset}@andrew.cmu.edu

[2] Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA, {jjpayne,amj1}@andrew.cmu.edu

curacy of legged robots, especially when motion becomes highly dynamic. In the bounding gaits tested, this method's error is less than 35% of the median trajectory error of the best state-of-the-art method tested. We conclude our paper by noting future directions for using periodicity to improve the performance of SLAM algorithms.

II. Background/Related Work

Visual and visual-inertial SLAM are well studied fields with a wide variety of approaches, many of which are discussed in [1]. This section highlights the most relevant publications including systems used for benchmarking, alternative approaches, and inspiration for the methods presented in this paper.

A. State-of-the-Art General Use SLAM Systems

This section provides a brief description of the state-of-the-art SLAM systems used as comparison points for the methods presented within this paper. Only indirect methods are considered because the large frame to frame displacements present on legged robots may break the underlying assumptions of direct methods, which use local image intensity gradients to align subsequent images [5]. Although the systems mentioned in this section perform well in less dynamic scenarios, they fail to provide accurate estimates of robot pose when camera viewpoint changes become rapid.

1) ORB-SLAM2: ORB-SLAM2 is an optimization-based, visual-only SLAM system that uses data association over multiple time scales through local tracking and longer term loop closure and bundle adjustment through the use of keyframes [6]. This use of multiple timescales is one of the main features of ORB-SLAM2. However, when there are many difficult to estimate frames in a row, the system enters a lost state before being able to take advantage of multiple time scales of estimation.

2) VINS-Fusion: VINS-Fusion is an optimization-based visual-inertial SLAM system with a large focus placed on the integration of inertial measurements into the factor graph [7,8]. The addition of inertial information helps VINS-Fusion to outperform ORB-SLAM2 in moderately dynamic scenarios. However, at the highest levels of dynamic motion discussed in this work, the failure of VINS-Fusion's feature tracking module leads to inaccurate estimates of robot pose.

3) MSCKF VIO: Multi-State Constraint Kalman Filter (MSCKF VIO) is a filtering-based method that uses visual-inertial data in an extended Kalman filter rather than a factor graph optimization to estimate the state of the robot [9]. Unlike ORB-SLAM2 and VINS-Fusion, MSCKF does not maintain a long-term map of its surroundings as it is only performing odometry. However, because of its distinct filtering-based back-end, we include it as a comparison point.

B. Leg Odometry Based Methods

Recently, there has been a lot of interesting work on improving the quality of SLAM performance on legged robotic systems. In works such as [10–12], an additional factor is added to the factor graph optimization that represents the estimated motion from the forward kinematics of the system over the time period. While there have been many impressive results from these methods, they are presented on less dynamic gaits such as walks and slower trots with more careful leg placement. In gaits such as bounding, a large amount of camera pitch is expected, which would decrease the utility of visual odometry factors, which these systems are still reliant on. While in a mature system, leg odometry factors provide useful additional constraints, the visual constraints of the system must also be improved to achieve reliable performance in the most dynamic scenarios.

C. Multi-Agent SLAM

The main inspiration for this Periodic SLAM algorithm is cooperative mapping [13]. In that work, the constraints from visual SLAM sessions running on multiple different robots are simultaneously optimized. To address the challenge of obtaining a unified estimate from all of the sessions, "encounters" between robots are used to constrain the multiple SLAM sessions into one set of world coordinates. Each encounter, determined when more than one robot observes a similar set of visual features, is formulated as a relative pose constraint between the different robots.

In this work, we treat different portions of a legged robot's gait cycle as individual visual SLAM sessions. Unlike [13], our approach constrains each visual SLAM session to one another using IMU measurements since each session is running on a single robot.

III. Periodic SLAM

This section discusses details behind performing periodic data association and the relevant mathematics.

A. Periodic Feature Tracking

Typically in visual odometry or visual SLAM systems, short-term data association happens between sequential camera frames. While tracking features across sequential camera frames works well on slow-moving robots, on dynamically locomoting legged robots this kind of data association often fails. When there is known periodicity in the viewpoint of a robot, it is beneficial to track features periodically at similar phases of the robot's gait cycle.

To perform periodic feature tracking, this method relies on being able to consistently extract and track features from images collected during an interval in which the phase of the robot's gait cycle is similar. Thus, this method makes two key assumptions about the robotic platform and its environment:

1) Periodic tracking has a global clock indicating the phase of the robot's gait.
2) Images taken at similar gait phases contain mutually visible features.

These assumptions have a few consequences. The first assumption limits the present application of this algorithm to scenarios in which the periodic motion of the robot is consistent with a structure that is known ahead of time. The second assumption requires that the robot maintains periodic motion that is fast enough, such that the scene does not change too much between oscillations.

Given a set of images with similar gait phases and mutual visual features, feature tracking begins with an initialization step and then a tracking step. After initializing visual features in the first image, the tracking step persists as long as there are enough features to track. If at any point the system “loses” too many features to track, the system re-initializes to add new visual features. While this approach is general to any feature detector, Harris corner detection [14] is used to initialize features and the Lucas-Kanade method [15] is used to track them.

Multiple periodic feature trackers are initialized to track different segments of the robot’s gait cycle. For each periodic feature tracker, a visual SLAM session is introduced, which is responsible for using the tracked features to build a sparse map of three-dimensional visual landmarks and to estimate the location of the robot during a certain phase segment. Initializing individual SLAM sessions leads to the potential for duplicated landmarks. For situations in which mapping accuracy is critical, post processing could de-duplicate landmarks between sessions at the expense of some added computation.

When performing multi-session SLAM, the problem of fusing different state estimates from each visual SLAM session must be addressed. While a naive approach to this problem might average the different SLAM sessions’ estimates, it would be advantageous if these sessions tightly shared information to achieve a more robust combined performance.

Since not enough visual features may be shared between different phases, IMU measurements are used to constrain separate SLAM sessions rather than visual constraints. IMU sensors can be used to provide measurements of the robot’s acceleration and angular velocity between different SLAM sessions when feature tracking is not reliable. By performing integration of the IMU measurements, relative pose constraints are introduced between each of the SLAM sessions.

B. *Periodic Factor Graph*

After obtaining periodically tracked features from the front-end of this SLAM system, the goal of the back-end is to obtain an estimate for the set of unknown robot poses (S) and landmark positions (L) by performing optimization-based probabilistic inference. Obtaining this estimate involves maximizing the conditional probability density of the set of unknown variables given the set of sensor measurements (Z) [16]. The values of S and L that maximize this probability density and the solution to SLAM are called the *maximum a posteriori* (MAP) estimate:

$$S_{MAP}, L_{MAP} = \arg\max_{S,L} P(S, L|Z) \tag{1}$$

Contemporary approaches for solving the problem of SLAM rely on sparse factor graph based optimization [17]. Factor graphs are a class of graphical models that are useful in representing sparsity within the distribution, $P(S, L|Z)$, to enable efficient MAP inference. More precisely, they are a bipartite graph that consists of two types of nodes: factors and variables [16]. In the context of SLAM, variable nodes are used to represent the unknown, latent robot and landmark states we wish to estimate, S and L, and factors are used to represent constraints (specified by sensor measurements) between states, $\phi(\cdot) : (S, L) \rightarrow \mathbb{R}$, such that,

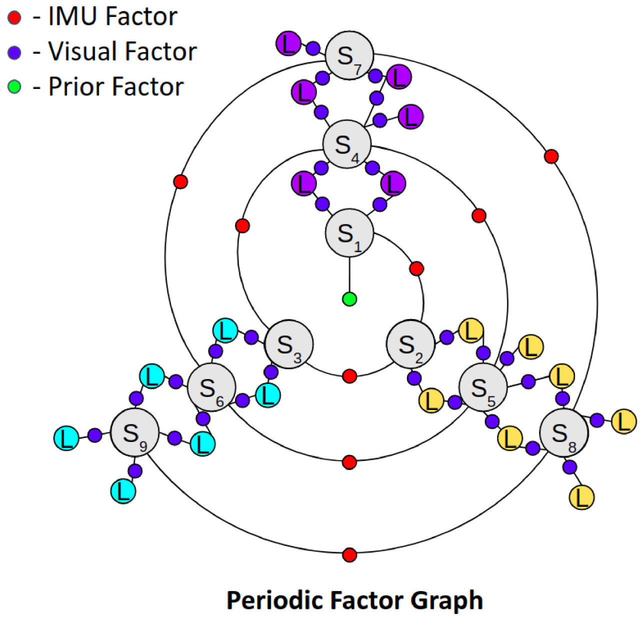

Fig. 2. Simplified Periodic SLAM factor graph. Each cycle around the circular graph represents one period of the robot’s gait cycle. Robot states positioned along a spoke of the graph have a similar gait cycle phase.

$$S_{MAP}, L_{MAP} = \arg\max_{S,L} \prod_i \phi_i(S, L) \tag{2}$$

Factor graphs are particularly useful in solving this multi-session SLAM problem because they are amenable to adding constraints between sequential and periodic robot states. Fig. 2 provides a simplified version of the factor graph that this method uses to solve the problem of SLAM with the incorporation of periodic feature tracking across three different portions of a robot’s gait cycle.

In Fig. 2 visual data association is performed across the spokes of the graph. Each spoke of the graph can be thought of as a SLAM session maintaining its own map of landmarks in the environment. The colors of the different landmark nodes are used to specify the corresponding portion of the robot’s gait cycle in which they are tracked. While this could be extended to any number of estimators, our experiments use a graph with visual SLAM being performed on three portions of the robot’s gait cycle: when the robot is looking upwards, forwards, and downwards. Moreover, although not pictured in Fig. 2, each visual SLAM session may contain multiple robot pose updates before moving to the next session. Additionally, SLAM sessions do not necessarily always have to update ”in order”. In practice the general cadence of the three session system would be: ”up”, ”forwards”, ”down”, ”forwards”.

C. Factor Graph Optimization

This section explains the mathematics behind each factor in the periodic factor graph shown in Fig. 2. To represent the state of the robot and each visual landmark, there are two types of variable nodes: s_i and l_j. Each robot state, s_i, represents the pose and velocity of the robot at a particular time. Each visual landmark state l_j represents the position of a unique point in the robot's map.

It is assumed that each of the factors is corrupted with zero-mean, additive Gaussian noise. Given this assumption, each Gaussian factor can be written in a form,

$$\phi \propto exp(-\frac{1}{2}\left\|F(s_i, l_j, z_k)\right\|^2_{\Sigma}) \tag{3}$$

where F can be thought of as a constraint or cost function which is dependent on the robot and landmark states as well as a sensor measurement z_k using the squared Mahalanobis distance $\|\cdot\|^2_{\Sigma}$ for weighting based on the measurement covariance Σ.

1) The Prior Factor: The prior factor is the simplest factor in the full periodic factor graph shown in Fig. 2. While all other factors are useful in estimating the robot's relative motion, the prior factor grounds the estimated state of the robot to a global reference frame. Given a prior measurement of the initial location of the robot, z^p, with covariance Σ^p, a Gaussian prior factor on the initial robot state is defined as:

$$\begin{gathered}\phi^{Prior} \propto exp(-\frac{1}{2}\left\|(h^{Prior}(s_1) - z^p)\right\|^2_{\Sigma^p}) \\ \text{where } h^{Prior} \text{ is trivially the identity function}\end{gathered} \tag{4}$$

2) The Visual Factor: Each visual factor in Fig. 2 represents a cost between its connected landmark and robot nodes that is dependent on a visual measurement, z^v_k. Visual measurements are periodically tracked stereo features from the front-end of SLAM with the form $z^v_k = [u^L_k, u^R_k, v_k]$. Here, u^L_k and u^R_k are the x coordinates of the tracked feature in the left and right stereo images and v_k is the y coordinate of the tracked feature in both images.

To calculate the cost for a particular visual measurement, the visual factor transforms a three-dimensional landmark into an estimated stereo feature, $\hat{z}^v_k$, at a corresponding robot state. The visual measurement function, h^{Visual}, performs this transformation in two steps: coordinate frame transformation (g) and projection (π):

$$h^{Visual}(s_i, l_j) = \pi(g(s_i, l_j)) = \hat{z}^v_k \tag{5}$$

After transforming the 3D landmark l_j into a stereo feature point, the re-projection error is calculated for a visual measurement with covariance Σ^v as follows:

$$\phi^{Visual} \propto exp(-\frac{1}{2}\left\|h^{Visual}(s_i, l_j) - z^v_k\right\|^2_{\Sigma^v}) \tag{6}$$

3) The IMU Factor: Each IMU factor uses measurements from the two sensors that make up the IMU: the gyroscope and the accelerometer. Using these measurements, it is possible to describe the dynamics of the robot's state s_i between two sequential time instances. This process can be summarized with a dynamics function h^{IMU} which predicts the next robot state given a sequence of IMU measurements.

$$\hat{s}_{i+1} = h^{IMU}(s_i, z^{IMU}_k) \tag{7}$$

Using this IMU process function, each IMU factor can be written as an error between the predicted and estimated next state of the robot:

$$\phi^{IMU} \propto exp(-\frac{1}{2}\left\|s_{i+1} - h^{IMU}(s_i, z^{IMU}_k)\right\|^2_{\Sigma^{IMU}}) \tag{8}$$

To avoid adding states to the graph at a high rate, IMU preintegration [18,19] is used.

4) MAP Estimation and Robust Cost Function: After defining the form of each of the factors in the periodic factor graph, nonlinear optimization is performed to obtain a unified SLAM solution. Moreover, each of the factors can be plugged into (2) to arrive at the following minimization:

$$\begin{aligned} S_{MAP}, L_{MAP} &= \underset{s,l}{\operatorname{argmax}}\, \phi^{Prior} \prod_{n=0}^{N} \phi^{Visual}_n \prod_{m=0}^{M} \phi^{IMU}_m \\ &= \underset{s,l}{\operatorname{argmin}} \left\|(h^{Prior}(s_1) - z^p)\right\|^2_{\Sigma^p} \\ &\quad + \sum_{n=0}^{N} \left\|h^{Visual}(s_i, l_j) - z^v_k\right\|^2_{\Sigma_v} \\ &\quad + \sum_{m=0}^{M} \left\|(s_{i+1} - h^{IMU}(s_i, z^{IMU}_k))\right\|^2_{\Sigma^{IMU}} \end{aligned} \tag{9}$$

The general approach to solve this is to first use a Taylor series expansion to linearize the optimization objective, and then iteratively solve the linearized equation using Gauss-Newton or Levenberg-Marquardt [20] methods. Our implementation relies on the iSAM2 algorithm to perform incremental SLAM more efficiently [21].

In the standard L2 cost objective, all measurements of a specific type are modeled with the same uncertainty. In practice, we empirically tuned the covariance value of each factor. However, without explicitly pruning erroneous measurements, this method is particularly sensitive to these hand-tuned values. To combat this issue, a robust error model is incorporated into the optimization. While many different robust error models exist [22], the Geman-McClure cost function (ρ) is chosen because of its particularly high bias against large outliers:

$$\rho(r) = \frac{\frac{r^2}{2}}{1 + r^2} \tag{10}$$

IV. Testing Algorithmic Performance

A. Simulation

To evaluate the performance of the Periodic SLAM system, we first conducted experiments in a simple hallway environment made in Gazebo. Lines of different colors are drawn onto the walls of the simulated hallway to ensure an abundance of visual features is available. Rather than using a legged robot to collect data in the simulated environment,

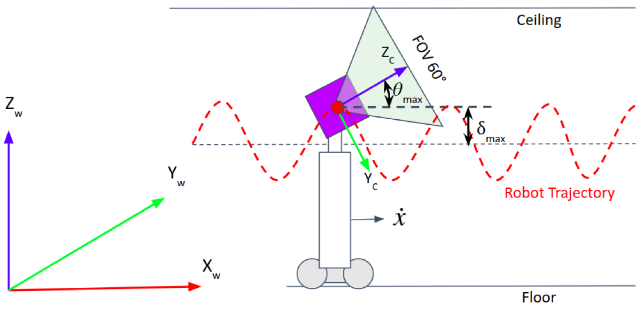

Fig. 3. The simulated robot setup which translates forward, raises and lowers the camera height, and pitches the camera viewing angle.

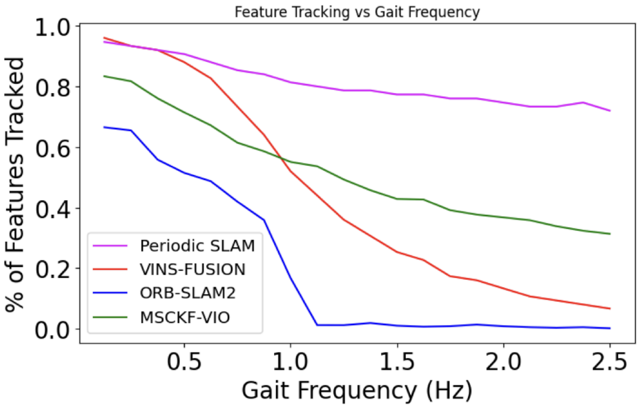

Fig. 4. Feature tracking performance of each SLAM system on the simulated dataset. The lines show the median performance for each SLAM system over 50 trials at each gait frequency. At higher gait frequencies, periodic feature tracking outperforms sequential feature tracking.

we use an actuated stereo-inertial camera on wheels, Fig. 3. Using two well controlled degrees of freedom in pitch, $\theta(t)$, and height, $z(t)$, the system approximates the motion of a camera attached to a hopping and pitching legged robot. Images from the simulated camera's perspective can be seen in Fig. 1.

We describe the motion of the simulated robot with a set of simple periodic functions,

$$\begin{aligned} x(t) &= \dot{x}t \\ \theta(t) &= \theta_{\text{max}} \sin(\omega 2\pi t) \\ z(t) &= \delta_{\text{max}} \sin(\omega 2\pi t) + z_0 \end{aligned} \tag{11}$$

that take four parameters as input: maximum pitch angle $\theta_{max} = 25°$, maximum heave distance $\delta_{max} = 0.05m$, the forward velocity $\dot{x} = 0.2ms^{-1}$, and gait frequency ω in Hz, which is varied in our experiments. To observe the effect of increasingly dynamic camera motion, we ran each SLAM algorithm on the robot as it moved forward 10 meters in the simulated hallway environment, and we ran 50 trials at different gait frequencies ranging from .125 Hz to 2.5 Hz.

Fig. 4 shows the feature tracking performance of our approach and each of the baseline methods as a function of gait frequency. The tracking metric is an average of the number of tracked features relative to the number of candidate features from the prior frame for all frames in the robot's trajectory.

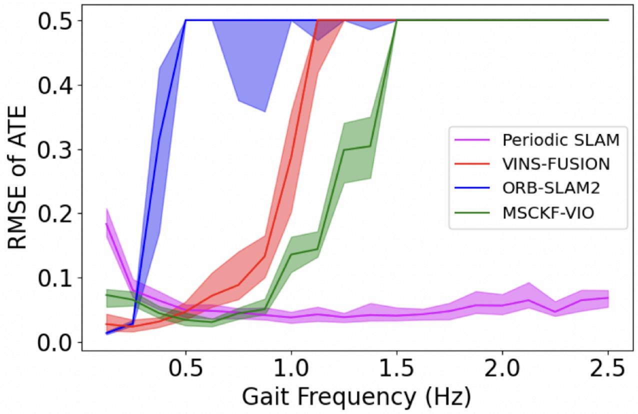

Fig. 5. Absolute Trajectory Error (ATE) of each SLAM system on the simulated dataset. The shaded regions represent the first to third quartiles across 50 trials at each gait frequency.

By tracking features periodically rather than sequentially, Periodic SLAM tracks a higher percentage of prior features than all 3 other methods at frequency values above 0.5Hz. Furthermore, Periodic SLAM consistently tracks over 75% of candidate features even on the most dynamic motion tested.

In Fig. 5, the root mean squared error (RMSE) of absolute trajectory error (ATE) of the four different methods are compared as the frequency of the motion varies. At lower gait frequencies, Periodic SLAM under-performs traditional methods because it ignores similarities between sequential visual information. However, at frequency values above approximately 0.75Hz, Periodic SLAM is more accurate than all of the other methods.

B. Minitaur

In addition to simulated experiments, we evaluated the performance of Periodic SLAM on data collected from an Intel RealSense D435i mounted on a Ghost Robotics Minitaur. Stereo images from the camera are collected at 30 Hz, and IMU data are collected at 300 Hz. As we were unable to access phase information from the robot's onboard controller, we hand labeled phase data for all sensor measurements. All experiments used an Optitrack motion capture system to provide ground truth pose information.

In our experiments, the Minitaur robot moved forward in a room while it executed two main types of gaits: a slow walking gait with relatively low camera pitch (easy gait) and a rapid bounding gait with large camera pitching (hard gait 1). To vary the visual information captured by the camera, we also collected data from the robot as it executed the rapid bounding gait while facing the opposite direction (hard gait 2). We ran each SLAM algorithm on data collected from 7 trials for each of these three situations. A video showing the robot's motion during our experiments can be found here: **https://www.youtube.com/watch?v=QygyDjVy5nY**.

To minimize alignment errors between the camera frame and the motion capture frame, all error metrics are presented after processing with the Python evo package [23]. Fig. 6 shows the trajectories estimated by each of the implementa-

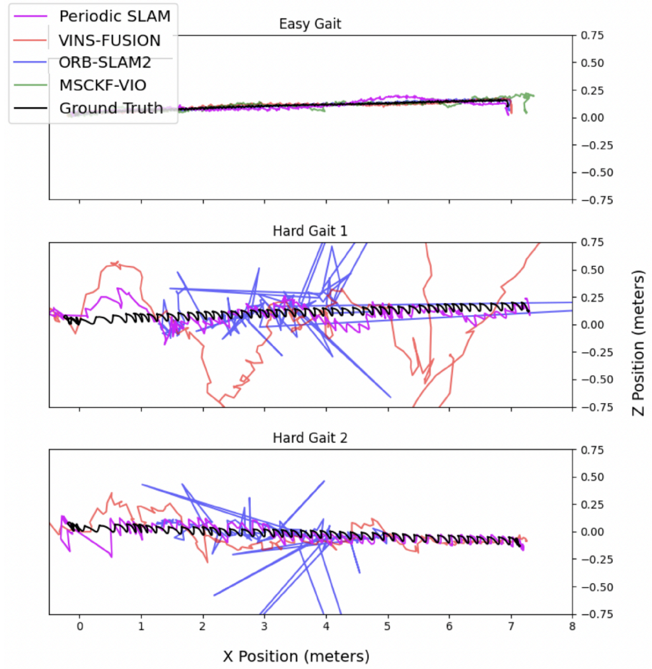

Fig. 6. Comparison of the estimated trajectories from each method on the three datasets. For the two difficult robot gaits, our method's estimated trajectory (pink) follows the ground truth trajectory (black) most closely. For Hard Gaits 1 and 2, MSCKF-VIO immediately diverges and its trajectory was unable to be reasonably plotted.

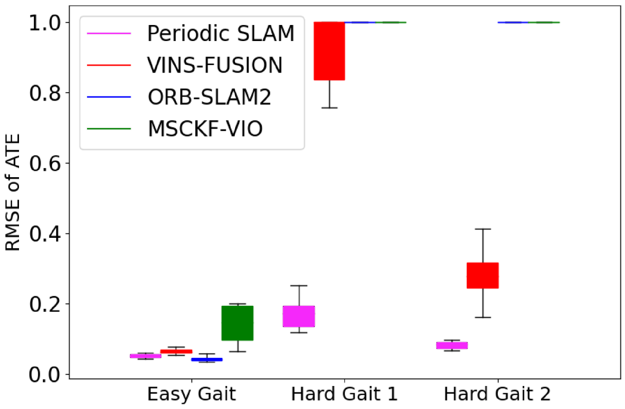

Fig. 7. Comparison of Absolute Trajectory Error (ATE) on the Minitaur data across SLAM implementations. Boxes represent the interquartile ranges and the whiskers represent the minimum and maximum values across 7 trials.

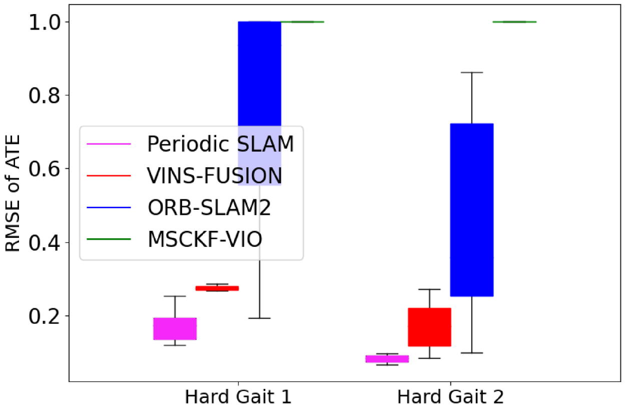

Fig. 8. Comparison of Absolute Trajectory Error (ATE) on the Minitaur data when the state-of-the-art methods are only given frames from when the robot is facing up. Only the two bounding gaits with significant pitching motion are shown. Boxes represent the interquartile ranges and the whiskers represent the minimum and maximum values across 7 trials.

tions for one trial in each of the three different conditions. Fig. 7 compares the RMSE ATE metrics of the systems under consideration across the 3 conditions.

In the easy gait, the performance of all 4 systems are comparable. However the harder gaits demonstrate that the Periodic SLAM approach outperforms all of the other SLAM systems in dynamic regimes, experiencing median error values less than 35% of the other methods' errors.

A similar comparison is made in Fig. 8, however in this experiment, the other methods are provided only camera frames in which the robot is looking upwards. While simply discarding all frames that cannot have features matched improves the performance of the baseline methods, Periodic SLAM still has a lower ATE because it fuses information from multiple SLAM sessions.

V. Conclusion

In this paper, we present a method for performing visual-inertial SLAM during especially aggressive motion found on legged robots. We show that on dynamic systems with periodic structure, performing feature tracking periodically rather than sequentially increases feature tracking performance. We develop an algorithm that maintains multiple visual SLAM sessions that each track features periodically across different parts of the robot's gait cycle. By connecting each SLAM session with measurements from an IMU in a factor graph optimization, our approach produces a unified estimate.

While our approach addresses the issue of feature tracking in the presence of dynamic motion, it does not address some other phenomena that cause visual-inertial SLAM to fail on legged robots. In future work, periodicity can also be leveraged to tackle issues such as IMU saturation and motion blur. By increasing the covariance of measurements during predictable impact events, the effect of outlier measurements due to these phenomena can be lessened.

This work can also be extended by performing an optimization to determine the optimal number and phase of different visual SLAM sessions within a robot's gait cycle. This would make our approach more easily adaptable to different gait patterns and even different types of periodic motion, not necessarily on legged robots.

Acknowledgement

This work was inspired in part by earlier unpublished work by Mack White.

References

[1] C. Cadena, L. Carlone, H. Carrillo *et al.*, "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.

[2] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM review*, vol. 48, no. 2, pp. 207–304, 2006.

[3] F. Dellaert and M. Kaess, *Factor Graphs for Robot Perception*. Now Publishers Inc., August 2017.

[4] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.

[5] G. Younes, D. Asmar, E. Shammas, and J. Zelek, "Keyframe-based monocular SLAM: design, survey, and future directions," *Robotics and Autonomous Systems*, vol. 98, pp. 67–88, 2017.

[6] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[7] 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.

[8] T. Qin, S. Cao, J. Pan, and S. Shen, "A general optimization-based framework for global pose estimation with multiple sensors," *arXiv preprint arXiv:1901.03642*, 2019.

[9] K. Sun, K. Mohta, B. Pfrommer *et al.*, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.

[10] R. Hartley, J. Mangelson, L. Gan *et al.*, "Legged robot state-estimation through combined forward kinematic and preintegrated contact factors," in *IEEE International Conference on Robotics and Automation*, 2018, pp. 4422–4429.

[11] D. Wisth, M. Camurri, and M. Fallon, "Robust legged robot state estimation using factor graph optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4507–4514, 2019.

[12] M. Bloesch, M. Hutter, M. A. Hoepflinger *et al.*, "State estimation for legged robots – consistent fusion of leg kinematics and IMU," in *Robotics: Science and Systems*, 2012, pp. 17–24.

[13] B. Kim, M. Kaess, L. Fletcher *et al.*, "Multiple relative pose graphs for robust cooperative mapping," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 3185–3192.

[14] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover." Stanford Univ CA Dept of Computer Science, Tech. Rep., 1980.

[15] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *International Joint Conference on Artificial Intelligence*, 1981, pp. 674–679.

[16] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[17] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, "Factor graph based incremental smoothing in inertial navigation systems," in *International Conference on Information Fusion*. IEEE, 2012, pp. 2154–2161.

[18] L. Carlone, Z. Kira, C. Beall *et al.*, "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," in *IEEE International Conference on Robotics and Automation*, 2014, pp. 4290–4297.

[19] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "IMU preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Robotics: Science and Systems*, 2015.

[20] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis*. Springer, 1978, pp. 105–116.

[21] M. Kaess, H. Johannsson, R. Roberts *et al.*, "iSAM2: Incremental smoothing and mapping using the bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[22] M. J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *Computer vision and image understanding*, vol. 63, no. 1, pp. 75–104, 1996.

[23] M. Grupp, "evo: Python package for the evaluation of odometry and SLAM," https://github.com/MichaelGrupp/evo, 2017.