

Asynchronous Adaptive Conditioning for Visual-Inertial SLAM

Nima Keivan, Alonso Patron-Perez, Gabe Sibley

{nimski, apatron, gsibley}@gwu.edu

Abstract. This paper is concerned with real-time monocular visual inertial simultaneous localization and mapping (VI-SLAM). In particular a tightly coupled nonlinear-optimization based solution that can match the global optimal result in real time is proposed. The methodology is motivated by the requirement to produce a scale-correct visual map, in an optimization framework that is able to incorporate relocalization and loop closure constraints. Special attention is paid to achieve robustness to many real world difficulties, including degenerate motions and unobservability. A variety of helpful techniques are used, including: a relative manifold representation, a minimal-state inverse depth parameterization, and robust non-metric initialization and tracking. Importantly, to enable real-time operation and robustness, a novel numerical dog leg solver [17] is presented that employs multithreaded, asynchronous, adaptive conditioning. In this approach, the conditioning edges of the SLAM graph are adaptively identified and solved for both synchronously and asynchronously. In this way some threads focus on a small number of temporally immediate parameters and hence constitute a natural “front-end”; other threads adaptively focus on larger portions of the SLAM problem, and hence are able to capture functional constraints that are only observable over long periods of time – an ability which is useful for self-calibration, during degenerate motions, or when bias and gravity are poorly observed. Experiments with real and simulated data for both indoor and outdoor robots demonstrate that asynchronous adaptive conditioning is able to closely track the full-SLAM maximum likelihood solution in real-time, even during challenging non-observable and degenerate cases.

1 Introduction

It is well known that the batch bundle-adjustment solution to monocular SLAM is the gold standard, in that it’s form defines the Cramer-Rao lower bound and that it takes advantage of all measurements over all time to compute the maximum likelihood parameter estimate [2,21]. Visual-inertial bundle adjustment is significantly more challenging than vision-only BA [8]. Vision-only monocular systems suffer from a well-studied scale ambiguity. Adding an IMU can makes scale observable, however inertial measurements complicate matters when it comes to computing the global MLE solution incrementally in real-time.

For bundle adjustment to be real-time for use on robots, a *local* approach is typically employed [13]. With an IMU this is difficult since the local adjustment region may need to be very large in order to ensure observability of certain parameters. Indeed, under certain degenerate motions such as constant velocity forward motion, some parameters may never be observable (though this rarely if ever happens in practice) [6,5].

An alternative to local-bundle adjustment is to only keep a sliding window of the most recent poses and landmarks active, and marginalize the rest into a prior distribution [19,14,18]. This is equivalent to a fixed-lag Kalman smoother [11,3] and recently such systems have shown remarkable results [4,9,10].

Marginalization into a prior distribution like this is predominantly employed for computational efficiency – if it were possible to compute the full MLE solution in real-time it would be preferable. Marginalization is also costly because it introduces conditional dependencies between the remaining parameters causing “fill-in”. Fill-in can be addressed by cutting feature tracks and carefully marginalizing poses and landmarks simultaneously [15]. Marginalization is also potentially dangerous because it bakes in linearization errors which can lead to over-confident estimates or divergence unless one is careful to maintain consistency [4]. Carrying prior distributions induced from marginalization also necessitates an expensive global optimization at loop-closure to obtain the correct marginal. This paper attempts to remedy these issues by avoiding marginalization altogether.

Instead of relying on marginalization we take advantage of conditioning, which has shown surprisingly robust and accurate results in the computer vision community [7,2] and avoids locking in incorrect parameter estimates when used adaptively [20]. Using a relative manifold is also important because optimal relative transformation estimates in SE3 are by definition near zero. This fact allows multiple threads to asynchronously optimize and update different overlapping subsets of the full problem without detriment.

Adaptive asynchronous conditioning has other benefits: it can **a)** perform robust initialization even under degenerate motions, **b)** allow constant-time loop closure without expensive loop-long re-linearization, **c)** operate even during poor observability conditions, **e)** avoid inconsistency associated with early marginalization and re-linearization, **f)** track the relative-space maximum likelihood solution in constant time, **d)** enable power-on-and-go self-calibration. We find that adaptive asynchronous conditioning closely tracks the global batch optimal solution, at a fraction of the computational cost, which enables real-time operation.

2 Sliding window optimization

2.1 Fixed window formulation

The state vector of the parameters inside the sliding window is defined as

$$\mathbf{x} = [\{ \mathbf{T}_{wv} \mathbf{v}_w \mathbf{b} \} \{ \rho_k \}]^T \quad (1)$$

where $\{ \mathbf{T}_{wv} \mathbf{v}_w \mathbf{b} \}$ is the set of pose parameters: $\mathbf{T}_{wv} \in \text{SE3}$ is the transformation from vehicle to world coordinates, $\mathbf{v}_w \in \mathbb{R}^3$ is the velocity vector in world

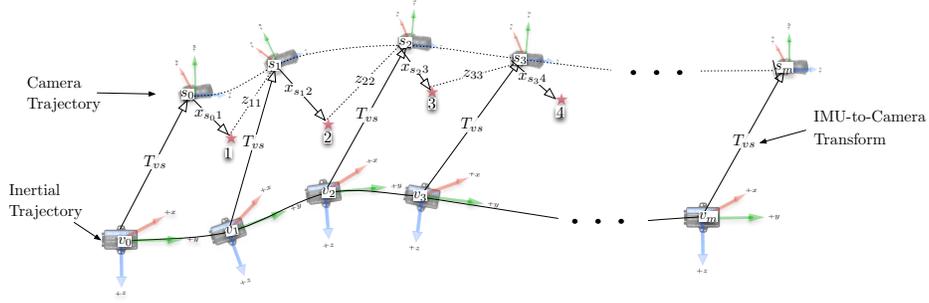


Fig. 1: Representation of the visual-inertial system.

coordinates and $\mathbf{b} \in \mathbb{R}^6$ is the IMU bias vector for the gyroscope and accelerometer. As mentioned previously, *world* coordinates here refers to the lifted local coordinates on which the optimization takes place. Similarly $\{\rho_k\}$ is the set of 1-d inverse-depth parameters for each landmark [16]. Since landmarks are parameterized in inverse depth, each must be back projected from its reference frame before being projected into the measurement frame. The reprojection error of the k th landmark with reference frame j into the i th frame is defined by:

$$\mathbf{r}_{\mathcal{P}_{ik}} = \mathbf{z}_{ik} - \pi(\mathbf{T}_{vs}^{-1} \mathbf{T}_{wv_i}^{-1} \mathbf{T}_{wv_j} \mathbf{T}_{vs} \mathbf{X}_{s_{jk}}) \quad (2)$$

where \mathbf{z}_{ik} is the measurement in image coordinates, $\mathbf{X}_{s_{jk}} = [u \ v \ 1 \ \rho]$ is the landmark inverse depth parameterization in the sensor frame obtained from back-projection, \mathbf{T}_{vs} is the transformation from the sensor to the vehicle frame, and π is the non-linear projection function. Inertial constraints are formed between subsequent states by integrating IMU measurements. The constraint between two subsequent states i and j with poses \mathbf{T}_{wv_i} and \mathbf{T}_{wv_j} and velocities \mathbf{v}_i and \mathbf{v}_j is defined as $\mathbf{r}_{\mathcal{I}_{ij}} \in \mathbb{R}^{15}$ and is formulated as

$$\mathbf{r}_{\mathcal{I}_{ij}} = \begin{bmatrix} \log \left((\mathbf{T}_{wv_i}) \mathbf{T}'_{v_i v_j} (\mathbf{T}_{wv_j})^{-1} \right) \\ \mathbf{v}_i + \mathbf{v}'_{ij} - \mathbf{v}_j \\ \mathbf{b}_j - \mathbf{b}_i \end{bmatrix} \quad (3)$$

where $\mathbf{T}'_{v_i v_j}$ and \mathbf{v}'_{ij} are the transformation and velocity deltas obtained by integrating the IMU measurements, and $\log(\cdot) \in \mathbb{R}^6$ is the SE3 logarithm function (with subsequent representation in minimal coordinates) as applied to an error state transformation. The residual on the gyroscope and accelerometer biases is derived from modeling them as random walk processes. The transformation delta due to the IMU integration is defined as

$$\mathbf{T}'_{v_i v_j} = \begin{bmatrix} \mathbf{R}'_{v_i v_j} & \mathbf{t}'_{ij} \\ 0 & 1 \end{bmatrix} \quad (4)$$

where $\mathbf{R}'_{v_i v_j} \in \mathbb{R}^{3 \times 3}$ is the rotation delta which is a function of the angular velocity measurements $\{\omega \in \mathbb{R}^3\}$ and the gyroscope biases $\mathbf{b}_g \in \mathbb{R}^3$. $\mathbf{t}'_{ij} \in \mathbb{R}^3$

is the translation delta which is a function of the acceleration measurements $\{\mathbf{a} \in \mathbb{R}^3\}$, the accelerometer biases $\mathbf{b}_a \in \mathbb{R}^3$, the gravity vector $\mathbf{g} \in \mathbb{R}^3$ and the initial velocity $\mathbf{v}_i \in \mathbb{R}^3$.

$\mathbf{R}'_{v_i v_j}$ is obtained by first integrating angular velocities in the world frame and then transforming the result to be relative to the starting orientation:

$$\mathbf{R}'_{v_i v_j} = \mathbf{R}'_{wv_i}{}^{-1} \mathbf{R}'_{wv_j}$$

where \mathbf{R}'_{wv_j} is the result of the discrete integration of angular velocities $\{\omega\}$ in the world frame, and also depends on the gyroscope biases \mathbf{b}_g . Each integration step is formulated as

$$\mathbf{R}'_{wt_{n+1}} = \exp(\mathbf{R}'_{wt_n} [\omega + \mathbf{b}_g] dt) \mathbf{R}'_{wt_n} \quad (5)$$

where \mathbf{R}'_{wt_n} is the rotation matrix from world coordinates to the coordinate frame resulting from the integration up to time t_n , $\omega \in \mathbb{R}^3$ is the angular velocity vector obtained by the gyroscope at time t_n and $\mathbf{b}_g \in \mathbb{R}^3$ is the gyroscope bias vector. The angular velocity measurement, taken in the body frame represents a rotation in the tangent space of \mathbf{R}'_{wt_n} , however since the rotation is integrated in world coordinates, the angular velocities must be transformed from the body frame to the world frame. This transformation is undertaken by the adjunct, which for SO3 is simply a multiplication by the rotation \mathbf{R}'_{wt_n} . Once in the world frame, the angular velocities are integrated and a rotation delta is obtained via the SO3 exponential exp. Note that the transformation from/to minimal coordinates in exp has been omitted for brevity.

The translation vector $\mathbf{t}'_{ij} \in \mathbb{R}^3$ is obtained by integrating the body accelerations in the world frame, and removing the translation of the initial frame \mathbf{t}_i as follows:

$$\mathbf{t}'_{ij} = \mathbf{t}'_{wj} - \mathbf{t}_{wi}$$

The discrete integration step for \mathbf{t}'_{wj} is formulated as

$$\mathbf{t}'_{wt_{n+1}} = \mathbf{t}'_{wt_n} + \int_{t_n}^{t_{n+1}} \mathbf{v}'_{t_n} dt \quad (6)$$

$$\mathbf{v}'_{t_n} = \mathbf{v}'_{t_{n-1}} + \int_{t_n}^{t_{n+1}} (\mathbf{R}'_{wt_n} [\mathbf{a} + \mathbf{b}_a] - \mathbf{g}) dt \quad (7)$$

where $\mathbf{v}'_{t_n} \in \mathbb{R}^3$ is the velocity integrated up to time t_n , $\mathbf{a} \in \mathbb{R}^3$ is the vector of accelerations measured in the body frame and $\mathbf{b}_a \in \mathbb{R}^3$ is the accelerometer bias vector. As the accelerometer measurements are integrated in the world frame, the measurements and biases in the body frame at time t_n must be transformed into the world frame, which is accomplished by multiplying by the body orientation at time t_n , \mathbf{R}'_{wt_n} .

Although the aforementioned derivation uses euler integration for simplification, all integrations, including the integration of angular velocities via the SO3 exponential exp are undertaken via fourth order Runge-Kutta. Considering (5)

through (7), it can be observed that the rotation delta $\mathbf{R}'_{v_i v_j}$ is independent of the translation delta \mathbf{t}'_{ij} , and also that the contribution of the starting velocity \mathbf{v}_i and the gravity vector \mathbf{g} can be factored out of the translation delta \mathbf{t}'_{ij} as follows

$$\mathbf{t}'_{ij} = \Delta t \mathbf{v}_i + \frac{1}{2} \Delta t^2 \mathbf{g} + \mathbf{t}^*_{ij}$$

where \mathbf{t}^*_{ij} is integrated as in equations 6 and 7, but with the starting velocity (\mathbf{v}_i) and gravity vector \mathbf{g} set to zero, and Δt is the entire duration over which \mathbf{t}^*_{ij} is integrated. Equation 4 can then be rewritten as

$$\mathbf{T}'_{v_i v_j} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \Delta t \mathbf{v}_i + \frac{1}{2} \Delta t^2 \mathbf{g} \\ 0_{1 \times 3} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}'_{v_i v_j} & \mathbf{t}^*_{ij} \\ 0_{1 \times 3} & 1 \end{bmatrix} \quad (8)$$

A similar factorization can be performed for \mathbf{v}'_{ij} to separate out terms that depend on the initial rotation and translation. The factorization of $\mathbf{T}'_{v_i v_j}$ and \mathbf{v}'_{ij} is undertaken in order to simplify the partial derivatives $\partial \mathbf{r}_{\mathcal{I}_{ij}} / \partial \mathbf{T}_{wv_i}$, $\partial \mathbf{r}_{\mathcal{I}_{ij}} / \partial \mathbf{T}_{wv_j}$, and $\partial \mathbf{r}_{\mathcal{I}_{ij}} / \partial v_i$ which are needed for the optimization. These would normally need to be propagated through the integration of inertial measurements via the chain rule. However, due to the factorization, these derivatives can be taken over the entire constraint by first integrating the inertial measurements as per (8) and then taking the aforementioned derivatives of (3). This both simplifies the process of calculating these derivatives, as well as avoiding the loss of accuracy due to the propagation of the derivatives through the Runge-Kutta integration. Unfortunately no such factorization can be made for $\partial \mathbf{r}_{\mathcal{I}_{ij}} / \partial \mathbf{b}$ which must be propagated through the Runge-Kutta integration via the chain rule.

2.2 Optimization formulation

The cost function minimized in the optimization consists of the aforementioned inertial and visual residuals and is formulated as

$$e = \sum_{i=1}^n \sum_{k=1}^m \|\mathbf{r}_{\mathcal{P}_{ik}}\|_{\Sigma_{\mathcal{P}_{ik}}}^2 + \sum_{k=1}^n \|\mathbf{r}_{\mathcal{I}_{ik}}\|_{\Sigma_{\mathcal{I}_{ik}}}^2 \quad (9)$$

where the notation $\|\mathbf{x}\|_{\Sigma}^2$ signifies the Mahalanobis distance given the measurement uncertainty Σ . In all cases residual uncertainties are calculated via Gaussian error-propagation from raw measurement uncertainties. In the case of visual measurements, a standard covariance of 1 pixel is used for both x and y image directions. For inertial measurements, the covariance of the final measurement must be propagated through the integration given the uncertainties in the accelerometer and gyroscope measurements supplied by the manufacturer. Since inertial measurements are integrated via the Runge-Kutta algorithm for accuracy, the uncertainties must also be propagated through each integration step as follows:

$$\mathbf{W}_{\mathcal{I}} = (\Sigma_{\mathcal{I}})^{-1} = \left(\frac{\partial \mathbf{r}_{\mathcal{I}}}{\partial \mathbf{x}_t} \left(\prod_{t=t_0}^{t_f} \frac{\partial \mathbf{x}_t}{\partial \mathbf{z}_t} \mathbf{C}_{\mathbf{z}_t} \frac{\partial \mathbf{x}_t^T}{\partial \mathbf{z}_t} \right) \frac{\partial \mathbf{r}_{\mathcal{I}}^T}{\partial \mathbf{x}_t} \right)^{-1} \quad (10)$$

where the weight for the particular residual $\mathbf{r}_{\mathcal{I}}$ is given by $\mathbf{W}_{\mathcal{I}} = (\Sigma_{\mathcal{I}})^{-1}$. To obtain the measurement covariance $\Sigma_{\mathcal{I}}$, the covariance of each inertial measurement at time t denoted by $\mathbf{C}_{\mathbf{z}_t}$ is propagated through the single-step state integration jacobian $\frac{\partial \mathbf{x}_t}{\partial \mathbf{z}_t}$. This step is then repeated for each inertial measurement and propagated via the chain rule to obtain the covariance for the final integration state. In order to obtain the covariance for the residual, the final integration state covariance is propagated through the residual jacobian $\frac{\partial \mathbf{r}_{\mathcal{I}}}{\partial \mathbf{x}_t}$, which is trivial with the exception of the derivatives for the SE3 logarithm.

Landmarks which are observed in the active window of frames but whose reference frame falls outside of it provide conditioning edges during the optimization. This is done by considering their reprojection error in poses outside of the active window where they were observed. A single IMU conditioning edge corresponding to the IMU residual between the last frame in the sliding window and its immediate parent frame is also used. The trajectory and map are represented in a relative graph [12]. This is shown in Figure 2.

To reduce complexity, the optimization is performed on a *lifted* window of the relative chain in which all poses and landmarks are transformed into a consistent local coordinate system, referred to as the *world* coordinate system in the aforementioned methodology. This allows reprojection errors to form without requiring the traversal of intermediate poses between the reference and measurement frames. Once the optimization has finalized, the results are transformed back into the relative representation for map storage.

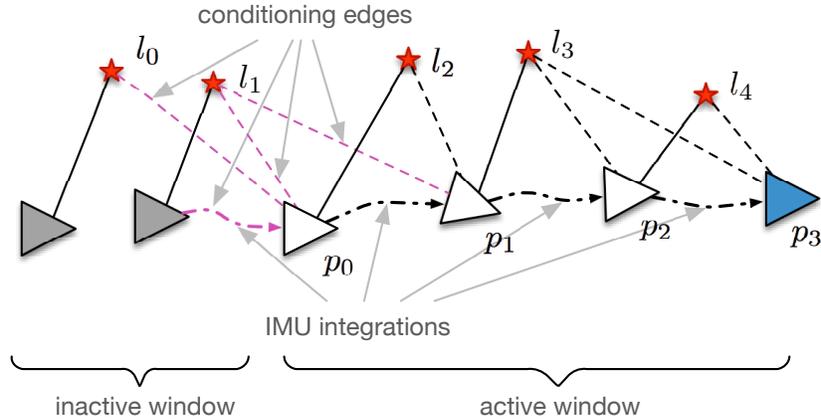


Fig. 2: Conditioning edges of the sliding window bundle adjustment.

2.3 Adaptive window implementation

Since marginalization is forgone, an adaptive local bundle adjustment is incorporated, which dynamically adjusts in size, in order to appropriately fold in parameters as needed. As is especially prominent when using an IMU, all parameters are not necessarily observable at any given time. Therefore, a fixed window cannot guarantee the optimization of a parameter at the time it becomes observable. Dynamically adjusting the window serves to allow the optimization to include parameters even if they are not immediately observable. Examples of these parameters are accelerometer and gyroscope biases and the direction of gravity, which is implicitly parameterized.

The condition used to assess whether the size of the window needs to be increased is based on the residuals observed in the conditioning edges shown in Fig. 2, after the optimization at iteration k has converged. The measurement covariances can then be used to assess whether the conditioning residuals are within expected bounds using a χ^2 test. The conditioning Mahalanobis distance is

$$e_c = \sum_{i \in C} \|\mathbf{r}_{P_i}\|_{\Sigma_{P_i}}^2 + \|\mathbf{r}_{I_c}\|_{\Sigma_{I_c}}^2$$

where the summation is over the set C comprising of all conditioning visual residuals, and \mathbf{r}_{I_c} is the single conditioning inertial residual connecting the active and inactive poses. Given e_c , an adaptive condition variable α_k can be defined as

$$\alpha_k = \frac{e_c}{\text{Inv } \chi^2(\beta, d)}, \quad d = 2|C| + 15$$

where $\text{Inv } \chi^2(\beta, d)$ is the inverse cumulative χ^2 distribution for d dimensions evaluated at probability β . The dimensionality d is derived from the 15 residuals of the single conditioning inertial residual plus 2 residuals for each visual conditioning residual. Initially if $\alpha_k > 1$, the conditioning residuals lie outside the β th percentile probability as expected from the residual covariance, so the window size is increased, and the optimization is run to convergence. While $\alpha_{k+1} > 1$ and $\alpha_{k+1} < \alpha_k$, the window size is continually increased and the optimization is run once again to convergence, without adding additional frames. Otherwise, the window is resized to its default minimum length and new frames are added to the window.

The intuition behind this adaptive criterion is that when new residuals render past parameters observable, and they are not present in the active state, tension will be introduced into the conditioning edges. The residuals defining these edges would then fall outside their expected distributions. Increasing the window size until the conditioning edges are within expected bounds ensures that unobserved dimensions become part of the active window. In the case that the conditioning error is not decreasing but is still outside expected bounds, the window size is returned to its default minimal value, as the error is more likely explained by outlier measurements.

The size of the dynamic window could stretch far if a parameter does not become observable for an extended period of time. In this case obtaining a real-time solution will become infeasible. In order to obtain a real time solution, an asynchronous adaptive window BA is used in conjunction with a small fixed size window BA which runs synchronously in real time. The use of a relative map representation ensures that updates to the map remain small, allowing multiple BAs to update it asynchronously without clashing.

3 Experiments

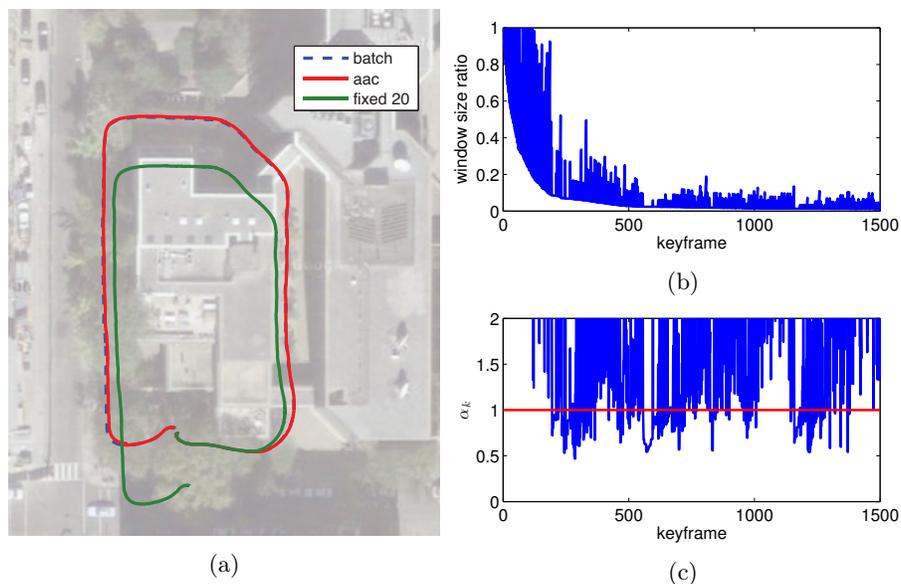


Fig 3: A loop consisting of a 200m dataset taken on foot around the GWU campus, superimposed over satellite imagery. The AAC optimization consists of a fixed window of 15 keyframes plus an asynchronous adaptive window with a minimum of 15 keyframes. The batch translation error as a percentage of traveled distance is 0.71% while the AAC error is 0.72% (a) The resulting poses obtained by running AAC, batch and a fixed-window optimization over the data. (b) The ratio of the AAC window to the total number of keyframes. A ratio of 1.0 indicates a batch solve. (c) The condition variable α_k for every keyframe. Values larger than one indicate an expansion is necessary.

To test the proposed method, experiments are run on two sensor platforms. Both platforms consist of a camera with wide-angle lens and a commercial grade MEMS accelerometer and gyroscope. The camera captures VGA images at 30fps

and the IMU sample rate is 120Hz. In order to evaluate how closely the AAC method matches the global MLE solution, a batch solution for each recorded trajectory is estimated. Comparisons are also made with trajectories that were estimated by keeping a fixed-size sliding window to demonstrate the effects of the window size on the quality of the solution. For all experiments the synchronous part of the AAC method is run with a fixed window of 15 poses.

The visual measurements are obtained by first extracting salient corners in the image where needed, to form landmarks. These landmarks are then tracked in subsequent images by minimizing the reprojective appearance error in a 9x9 pixel support area around the corner, between the two images similar to [1]. If the appearance error is between a predefined threshold and the reprojective constraints are not violated, the new position of the corner is added as a visual measurement of the landmark. All examples were run by attempting to track at most 128 landmarks. Keyframing was used as a means to increase the performance of the optimization by adding parameters only when sufficient motion was detected. This also alleviates problems arising from a stationary camera.

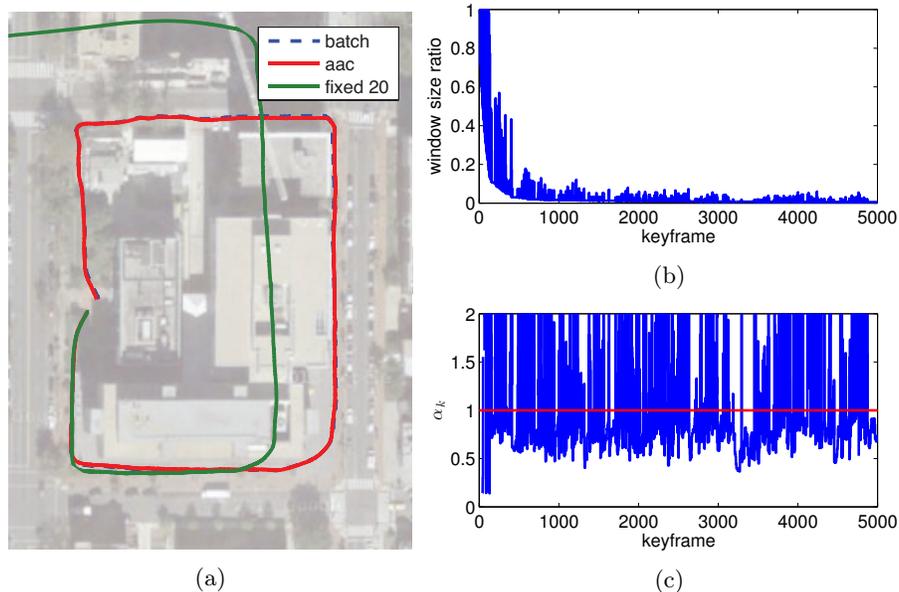


Fig. 4: A loop consisting of a 400m dataset taken on foot around the GWU campus, superimposed over satellite imagery. The AAC optimization consists of a fixed window of 15 keyframes plus an asynchronous adaptive window with a minimum of 15 keyframes. The batch error as a percentage of traveled distance is 1.33% while the AAC error is 1.42. For details on sub-figures (a), (b), and (c) refer to Figure 3

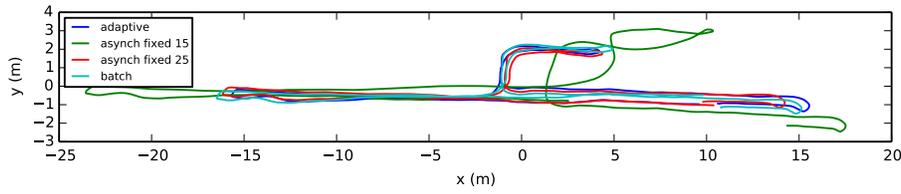


Fig. 5: Comparison of trajectories estimated by different bundle adjustment configurations. It can be seen that the 25 long asynchronous fixed window BA and the adaptive window BA both produce trajectories close to the batch solution, however the 15 long fixed asynchronous fixed window BA diverges substantially from the batch solution.

The first experimental setup consists of a person walking in different indoor and outdoor environments. The first experiment within this setup was undertaken indoors and consisted of a closed loop sequence along a corridor. The trajectory length is approximately 80m. The resulting images and IMU data were then processed with four different configurations of the solver: batch, two different fixed windows (15 and 25 poses) and AAC. The results from the four configurations run over the corridor sequence are shown in Figure 5. From these results it can be observed that as the window size increases, the trajectory converges to the batch solution, as expected. However, it can also be seen that the adaptive mode, with an average window size of 33, matches the batch solution closely without a predetermined window size.



Fig. 6: (a) Autonomous vehicle used for data collection in this paper. (b) The 900m trajectory estimated by the adaptive asynchronous system superimposed over aerial imagery.

Figures 3 and 4 show data obtained from running batch, AAC and fixed-window optimizations on two separate datasets taken on foot around the GWU campus. On both datasets, the AAC solution manages to adequately match the batch solution. Results using a fixed window size show that an adaptive

solution is necessary to adequately approximate the batch solution. The subplots depicting the ratio of the active window to the total number of keyframes denote the initialization phase, where the AAC system pushes the optimization to batch, if the initial parameter estimates such as velocity and orientation with respect to gravity are mis-estimated. However after the initialization phase, the AAC window size reaches a constant-time phase where only minor expansions are required to keep an optimal estimate.

The experiments with the second platform were run using the autonomous car depicted in Figure 6a. Images and IMU data were captured while driving around the GWU campus in Washington D.C. The trajectory generated by our adaptive method for a 900m segment of the data is shown in Figure 6b.

4 Experimental Insights

It was observed that in real-life situations, parameters such as velocity, gravity and bias are observable with adaptive conditioning. This is of course contingent upon sufficient excitation of the sensors. In the corridor dataset there is an ever present oscillatory acceleration which quickly renders the unknown parameters observable. Given this, we see a shorter required window size in order to closely estimate the MLE solution. As expected window growth is also seen in situations where scale and consequently velocity are ambiguous. An example of this is at the ends of the corridor where sharp turns introduce a slew of uninitialized new landmarks while simultaneously cutting tracks from established landmarks. The net result is a scale ambiguity that requires a larger window size to resolve, which is automatically discovered.

For data collected on the vehicle, scale was observable only over a large period of time where significant accelerations were imparted on the vehicle for example during turning. As such, a longer window was required in situations where scale became ambiguous and especially for initialization where prior estimates of velocity, biases and the gravity direction were not available.

When using asynchronous BA, care must be taken so as to ensure sufficient update frequency of the asynchronous solution in order to ensure overlap with the synchronous BA. This is required to keep the synchronous BA in the overall solution basin as solved by the asynchronous BA. As expected from the relative framework, the updates to the edges and inverse depth parameters for landmarks are small and no interference was observed between the two threads.

Adaptive asynchronous conditioning (AAC) is a novel solution to real-time visual-inertial SLAM. AAC is interesting because it automatically scales and focuses computation to capture the full MLE solution, and avoids the downsides associated with marginalization, such as incorrect linearization and inconsistency. Further, AAC avoids the computational difficulties associated with carrying prior distributions, such as the need to compute global optimizations at loop closure.

The proposed method offers a natural “front-end” while simultaneously allowing larger portions of the problem to influence the solution. It is thus able

to produce estimates in real-time, and also capture functional-constraints that are only observable over long periods of time – an ability which is useful for self calibration, during degenerate motions, or when bias and gravity are poorly observed.

References

1. D. Scaramuzza C. Forster, M. Pizzoli. Svo: Fast semi-direct monocular visual odometry. In *IEEE Conference on Robotics and Automation*, 2014.
2. C. Engels, H. Stewenius, and D. Nister. Bundle adjustment rules. In *Photogrammetric Computer Vision*, 2006.
3. A. Gelb. *Applied Optimal Estimation*. MIT Press, Cambridge, MA, 1974.
4. J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-imu-based localization: Observability analysis and consistency improvement. *International Journal of Robotics Research*, 33:182–201, 2013.
5. E. Jones, A. Vedaldi, and S. Soatto. Inertial structure from motion with autocalibration. In *ICCV Workshop on Dynamical Vision*, 2007.
6. J. Kelly and G. S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration". *International Journal of Robotics Research*, 2010.
7. G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *International Symposium on Mixed and Augmented Reality*, 2007.
8. S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart. Keyframe-based visual-inertial slam using nonlinear optimization. In *Robotics: Science and Systems*, 2013.
9. M. Li, B. Kim, and A.I. Mourikis. Real-time cellphone localization using inertial sensing and a rolling-shutter camera. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2013.
10. M. Li and A. I. Mourikis. High-precision, consistent ekf-based visual-inertial odometry. *International Journal of Robotics Research*, 33:690–711, 2014.
11. P. S. Maybeck. *Stochastic Models, Estimation, and Control*, volume 141 of *Mathematics in Science and Engineering*. Academic Press, Inc, Boston, 1979.
12. C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, pages 1–17, 2010.
13. E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyse, and P. Sayd. Real time localization and 3d reconstruction. In *Proceedings of Computer Vision and Pattern Recognition*, New York, New York, June 2006.
14. A. Mourikis and S. Roumeliotis. A multi-state constraint kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3565–3572, Rome, April 2007.
15. E. D. Nerurkar, Kejian J. Wu, and S. I. Roumeliotis. C-klam: Constrained keyframe localization and mapping for long-term navigation. In *IEEE International Conference on Robotics and Automation Workshop on Long-term Autonomy*, 2013.
16. T. Pietzsch. Efficient feature parameterisation for visual SLAM using inverse depth bundles. In *British Machine Vision Conference*, 2008.
17. M. J. D. Powell. An efficient method for finding the minimum of a function of several variables without calculating derivatives. *Computer Journal*, 7(2):155–162, 1964.

18. G. Sibley. Sliding window filters for SLAM. Technical report, University of Southern California, Center for Robotics and Embedded Systems, CRES-06-004, 2006.
19. G. Sibley, L. Matthies, and G. Sukhatme. Sliding window filter with applications to planetary landing. *Journal of Field Robotics*, 27(5):587–608, September/October 2010.
20. G. Sibley, C. Mei, I. Ried, and P. Newman. Adaptive relative bundle adjustment. In *Robotics: Science and Systems*, 2009.
21. B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment - a modern synthesis. In *ICCV '99: Proceedings of the International Workshop on Vision Algorithms*, pages 298–372. Springer-Verlag, London, UK, 2000.