Efficient Visual-Inertial Navigation using a Rolling-Shutter Camera with Inaccurate Timestamps

Chao X. Guo, Dimitrios G. Kottas, Ryan C. DuToit
Ahmed Ahmed, Ruipeng Li and Stergios I. Roumeliotis

Multiple Autonomous Robotic Systems Laboratory

Technical Report
Number -2014-001
February 2014
Repository Revision:
Last Update:

Dept. of Computer Science & Engineering
University of Minnesota
4-192 EE/CS Building
200 Union St. S.E.
Minneapolis, MN 55455
Tel: (612) 625-2217
Fax: (612) 625-0572
URL: http://mars.cs.umn.edu
Efficient Visual-Inertial Navigation using a Rolling-Shutter Camera with Inaccurate Timestamps

Chao X. Guo, Dimitrios G. Kottas, Ryan C. DuToit
Ahmed Ahmed, Ruipeng Li and Stergios I. Roumeliotis

Multiple Autonomous Robotic Systems Laboratory, TR-2014-001
February 2014

This technical report is structured as follows: In the first section, we introduce the expression for generating the position and orientation of an interpolated coordinate frame. In the second section, we derive the Jacobian matrices for the measurement model dealing with both the time synchronization and rolling shutter problems. The third and fourth sections, present the unobservable directions of the linearized system, while its non-linear counterpart is employed for proving the uniqueness of those directions. Finally, we provide a strategy for selecting the poses that comprise the estimator’s optimization window, for two common motion profiles: generic motion, that provides sufficient baseline between consecutive camera poses and hovering scenarios (i.e., static case or pure rotational motions).

I. INTERPOLATION MODEL

A. Translation

For generating the position of the interpolated frame at time $\phi$, we use the following linear interpolation model.

$$ p_\psi = (1 - \lambda) p_k + \lambda p_{k+1} $$

where $\lambda$ is the interpolation parameter, $p_k$ and $p_{k+1}$ are the camera position in the global frame at times $k$ and $k+1$ (with $t_k \leq t_\phi \leq t_{k+1}$), and $p_\psi$ is the camera position when capturing the feature measurement.

B. Rotation

For the orientation, we introduce the following small angle-based interpolation model:

$$ ^{k+1}\mathbf{C}_k \approx \mathbf{I} - \lambda [\theta \times] $$

$$ ^k\mathbf{C}_\psi \approx \mathbf{I} - \lambda [\theta \times] = (1 - \lambda) \mathbf{I} + \lambda ^{k+1}\mathbf{C}_k \mathbf{C} $$

$$ _G^\psi \mathbf{C} = _G^{k+1}\mathbf{C} = (1 - \lambda)^k \mathbf{C} + \lambda ^{k+1}\mathbf{C} $$

where $^j\mathbf{C}$ denotes the orientation of $j$ in frame $i$. When there is no superscript or subscript is used, this corresponds to the global frame $G$ (i.e., $^{k+1}\mathbf{C}_G \triangleq ^{k+1}\mathbf{C}$).

The interpolation parameter $\lambda$ can be defined as the time delay caused by the time synchronization and the rolling shutter.

$$ \lambda = \frac{t_{ts} + t_{rs}m}{t_n} $$

II. MEASUREMENT MODEL AND JACOBIANS

The measurement model can be written as $z = h'(x) + n$, where $n$ is the measurement noise, and

$$ h'(x) = \begin{bmatrix} h(x)_1 \\ h(x)_2 \\ h(x)_3 \\ h(x)_4 \end{bmatrix} $$
where
\[ h(x) = \psi_f = \psi C(p_f - p_w) \]
\[ = ((1 - \lambda)^k C + \lambda^{k+1} C) (p_f - (1 - \lambda)p_k - \lambda p_{k+1}) \]  
(7)

The measurement Jacobians can be computed using the chain rule, as
\[ \frac{\partial h'(x)}{\partial x} = \frac{\partial h'(x)}{\partial h(x)} \frac{\partial h(x)}{\partial x} \]
(8)
where
\[ \frac{\partial h'(x)}{\partial h(x)} = \begin{bmatrix} \frac{1}{h(x)_3} & 0 & -\frac{h(x)_1}{(h(x)_3)^2} \\ 0 & \frac{1}{h(x)_3} & -\frac{h(x)_2}{(h(x)_3)^2} \end{bmatrix} \triangleq H_{pk} \]
(9)
and
\[ T_1 \triangleq (1 - \lambda)^k C + \lambda^{k+1} C \]
\[ t_2 \triangleq p_f - (1 - \lambda)p_k - \lambda p_{k+1} \]
\[ H_{\theta_k} = (1 - \lambda)[k^k C t_2 \times] \]
\[ H_{\theta_{k+1}} = \lambda [(k+1)^{k+1} C t_2 \times] \]
\[ H_k = (\lambda - 1) T_1 \]
\[ H_{k+1} = -\lambda T_1 \]
\[ H_{p_f} = T_1 \]
\[ H_{\lambda} = (k^{k+1} C - k^k C) t_2 + T_1 (p_k - p_{k+1}) \]
(10)
where \( H_{\theta_k} \) and \( H_{\theta_{k+1}} \) are the Jacobians of the orientation of the global in the camera frames at times \( k \) and \( k+1 \), \( H_k \) and \( H_{k+1} \) are the Jacobians of the position of the camera in global frame at times \( k \) and \( k+1 \), \( H_{p_f} \) is the Jacobian of the feature position, and \( H_{\lambda} \) is the Jacobian of the interpolation parameter \( \lambda \).

III. OBSERVABILITY ANALYSIS: LINEARIZED SYSTEM

In our observability analysis, we assume that the camera and IMU frames coincide so as to simplify notation. In practice, the extrinsic calibration parameters can be determined using the algorithm proposed in [1].

The system's state vector is: \([x_k \ x_{k+1} \ \xi]\), where \( x_i, i = k, k+1 \), is the IMU state:
\[ x_i = [q \ b_g \ v_i \ b_a \ p_i] \]
(11)
corresponding, \( q \) the global orientation in the IMU frame (in quaternion parametrization), \( b_g \), the gyroscope bias, \( v_i \), the IMU velocity in the global frame, \( b_a \), the accelerometer bias, and \( p_i \), the IMU position in the global frame respectively. \( \xi \) is defined as:
\[ \xi = [p_f \ \lambda] \]
(12)
where \( p_f \) is the feature position in the global frame, and \( \lambda \) is the interpolation parameter. Note that only one point feature is necessary.

A system’s observability matrix [2], \( M \), is defined as
\[ M = \begin{bmatrix} H^1 \\ H^2 \Phi^{2,1} \\ \vdots \\ H^k \Phi^{k,1} \end{bmatrix} \]
(13)
where \( \Phi^{k,1} \) is the state transition matrix from time step 1 to \( k \), and \( H^k \) is the measurement Jacobian at time step \( k \). As described in [2], a system’s unobservable directions, \( N \), span the observability matrix’s right nullspace.
\[ MN = 0 \]
(14)
From now on, we will use $\Phi$ and $H$ with superscripts denote the state transition and measurement Jacobian matrices of the whole system, while $\Phi$ and $H$ with subscripts denote the state transition and measurement Jacobian matrices corresponding to the IMU state $x_i$, $i = k, k+1$.

According to (10), the measurement Jacobian can be written as:

$$H^k = H_{p_k} \begin{bmatrix} H^i_{R} & H^{k+1}_{R} & H_{p_f} & H_{\lambda} \end{bmatrix}$$

where for $i = k, k+1$,

$$H^i_{R} \triangleq [H_{th} 0 0 0 H_{j}]$$

Similarly, the system state transition matrix can be defined as:

$$\Phi^{k,1} = \begin{bmatrix} \Phi_{k,1} & 0_{15 \times 15} & 0_{15 \times 3} & 0_{15 \times 1} \\ 0_{15 \times 15} & \Phi_{k+1,2} & 0_{15 \times 3} & 0_{15 \times 1} \\ 0_{3 \times 15} & 0_{3 \times 15} & I_{3 \times 3} & 0_{3 \times 1} \\ 0_{1 \times 15} & 0_{1 \times 15} & 0_{1 \times 3} & 1 \end{bmatrix}$$

where $\Phi_{i,j}$, $(i, j) = (k, 1)$, $(k+1, 2)$ is the state transition matrix between IMU states $x_i$ and $x_j$. As shown in [3], the structure of $\Phi_{i,1}$ is written as:

$$\Phi_{i,j} = \begin{bmatrix} \gamma^i_{j} & C & \phi^{i}_{12} & 0 & 0 \\ 0 & I & 0 & 0 \\ \phi^{i}_{31} & \phi^{i}_{32} & I & \phi^{i}_{34} \\ 0 & 0 & 0 & 1 \\ \phi^{i}_{51} & \phi^{i}_{52} & (i-j)\delta t I & \phi^{i}_{54} \end{bmatrix}$$

Then, substituting $H^k$ from (10), the block columns of the $k$-th block row of the system’s observability matrix, $M$, is written as:

$$M^k = H_{p_k} \left( (1 - \lambda)^{k} C T_2 \times j^{k} C_{1} + (\lambda - 1)T_1 \Phi^k_{51} \right)$$

$$M^k_2 = H_{p_k} \left( (1 - \lambda)^{k} C T_2 \times j^{k} \Phi^{k}_{12} + (\lambda - 1)T_1 \Phi^{k}_{52} \right)$$

$$M^k_3 = H_{p_k} ((k-1)\delta t(\lambda - 1)T_1)$$

$$M^k_4 = H_{p_k} ((\lambda - 1) T_1 \Phi^{k}_{54})$$

$$M^k_5 = H_{p_k} ((\lambda - 1) T_1)$$

$$M^k_6 = H_{p_k} (T_1)$$

$$M^k_7 = H_{p_k} \left( \lambda^{k+1} C T_2 \times j^{k+1} C_{1} - \lambda T_1 \Phi^{k+1}_{51} \right)$$

$$M^k_8 = H_{p_k} \left( \lambda^{k+1} C T_2 \times j^{k+1} \Phi^{k+1}_{12} - \lambda T_1 \Phi^{k+1}_{52} \right)$$

$$M^k_9 = H_{p_k} (-k\delta t \lambda T_1)$$

$$M^k_{10} = H_{p_k} (-\lambda T_1 \Phi^{k+1}_{54})$$

$$M^k_{11} = H_{p_k} (-\lambda T_1)$$

$$M^k_{12} = H_{p_k} \left( (k+1)^{k} C - k^{k} C_{1} T_2 + T_1(p_k - p_{k+1}) \right)$$

In [3], it has been shown that with a time synchronized global shutter camera, the vision-aided inertial navigation
system’s unobservable directions are:

\[
N_{\text{global}} \triangleq \begin{bmatrix}
\mathbf{i} \mathbf{C} & \mathbf{0}_{3 \times 3} \\
\mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\
-([\mathbf{v}_1 \times] \mathbf{g}) & \mathbf{0}_{3 \times 3} \\
\mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\
-([\mathbf{p}_1 \times] \mathbf{g}) & \mathbf{I}_{3 \times 3} \\
\mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\
-([\mathbf{p}_f \times] \mathbf{g}) & \mathbf{I}_{3 \times 3} \\
\mathbf{0} & \mathbf{I}_{3 \times 3}
\end{bmatrix}
\]  

(20)

In our case, the system receives the same amount of information from the IMU and camera measurements, except there exists a time misalignment between the two sensors. Thus, we expect our system to have at least the same unobservable directions.

**Lemma 1**: The system with time misaligned IMU and camera measurements has at least the following four unobservable directions:

\[
\mathbf{N} \triangleq \begin{bmatrix}
\mathbf{i} \mathbf{C} & \mathbf{0}_{3 \times 3} \\
\mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\
-([\mathbf{v}_1 \times] \mathbf{g}) & \mathbf{0}_{3 \times 3} \\
\mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\
-([\mathbf{p}_1 \times] \mathbf{g}) & \mathbf{I}_{3 \times 3} \\
\mathbf{0}_{3 \times 1} & \mathbf{0}_{3 \times 3} \\
-([\mathbf{p}_2 \times] \mathbf{g}) & \mathbf{I}_{3 \times 3} \\
\mathbf{0} & \mathbf{I}_{3 \times 3}
\end{bmatrix} = [N_g \quad N_p]
\]  

(21)

**Proof**: First, it can be easily seen that \(M^kN_p = H_{p_k}(-T_1 + T_1) = \mathbf{0}\), for any \(k\). Thus, \(MN_p = \mathbf{0}\). To verify \(N_g\) is also the system’s unobservable direction, we multiply it to the system’s observability matrix as:

\[
\mathbf{M}^k \mathbf{N}_g = H_{p_k} \left((1 - \mathbf{C}^k \mathbf{C}[T_2 \times] \mathbf{g} + (\mathbf{v} - 1)T_1 [\mathbf{p}_1 - \mathbf{p}_k \times] - (\mathbf{v} - 1)T_1 [\mathbf{p}_1 \times] \mathbf{g} - T_1 [\mathbf{p}_f \times] \mathbf{g}\right)
+ H_{p_k} \left((1 - \mathbf{C}^k \mathbf{C}[T_2 \times] \mathbf{g} - \mathbf{v}T_1 [\mathbf{p}_2 - \mathbf{p}_{k-1} \times] \mathbf{g} + \mathbf{v}T_1 [\mathbf{p}_2 \times] \mathbf{g}\right)
\]

\[
= H_{p_k} \left((1 - \mathbf{C}^k \mathbf{C}[T_2 \times] \mathbf{g} + (\mathbf{v} - 1)T_1 [-\mathbf{p}_k \times] \mathbf{g} - T_1 [\mathbf{p}_f \times] + \mathbf{v}T_1 [\mathbf{p}_2 \times] \mathbf{g}\right)
\]  

(22)

In which \(\phi\) is substituted with the following expression, as proved in [3]:

\[
\phi = \left| \mathbf{p}_1 + (k - 1) \delta t \mathbf{v}_1 - \frac{1}{2} ((k - 1) \delta t)^2 \mathbf{g} - \mathbf{p}_k \right| \mathbf{C}
\]  

(23)

### IV. Observability Analysis: Non-linear System

In the last section, we have proved \(\mathbf{N}\) is the system’s nullspace. In this section, we will show that the time offset between the IMU and camera, corresponding to the interpolation ratio, is observable for the VINS. Thus, there exists no other unobservable directions other than \(\mathbf{N}\).

**Theorem 1**: Consider a nonlinear, continuous-time system:

\[
\begin{aligned}
\dot{\mathbf{x}} &= \sum_{i=0}^{l} \mathbf{f}(\mathbf{x})u_i \\
\mathbf{z} &= \mathbf{h}(\mathbf{x})
\end{aligned}
\]  

(24)

if a time delay \(\tau\) is unobservable, then

\[
\mathcal{O}\dot{\mathbf{x}} = \mathbf{0}
\]  

(25)

where \(\mathcal{O}\) is the system’s observability matrix, as defined in [4].

**Proof**: For a system (24), given the control inputs \(u_i\) and the measurements \(\mathbf{z}\), if the time delay \(\tau\) is unobservable,
then:

\[ z = h(x(t)) = h(x(t + \tau)) \]  

(26)

By applying Taylor series expansion on (26), we have:

\[
\begin{align*}
    h(x(t)) &= h(x(t + \tau)) \\
           &= h(x(t)) + \tau h + \frac{1}{2} \tau^2 h + \frac{1}{6} \tau^3 h + \cdots
\end{align*}
\]

\[
= h(x(t)) + \left( \tau \nabla^0 h + \frac{1}{2} \tau^2 \sum_{i=0}^1 \nabla^1 h_i u_i + \frac{1}{6} \tau^3 \sum_{i=0}^l \nabla^2 h_{ij} u_i u_j + \cdots \right) x
\]

(27)

where \( \hat{h}, \hat{h}, \) and \( \ddot{h} \) are substituted with the following expressions:

\[
\begin{align*}
    \hat{h} &= \frac{\partial h}{\partial t} = \frac{\partial h}{\partial x} \frac{\partial x}{\partial t} = \nabla^0 h x \\
    \hat{h} &= \frac{\partial (\nabla^0 h x)}{\partial t} = \frac{\partial (\sum_{i=0}^1 \nabla^1 h_i u_i)}{\partial t} = (\sum_{i=0}^1 \nabla^1 h_i u_i) x \\
    \ddot{h} &= \frac{\partial (\sum_{i=0}^1 \nabla^1 h_i u_i x)}{\partial t} = \frac{\partial (\sum_{i=0}^l \sum_{j=0}^i \nabla^2 h_{ij} u_i u_j)}{\partial t} = (\sum_{i=0}^l \sum_{j=0}^i \nabla^2 h_{ij} u_i u_j) x
\end{align*}
\]

(28)

Since (27) needs to be satisfied by any set of control inputs \( u_i \), the following relation needs to hold:

\[
\begin{align*}
    \Theta \dot{x} &= 0, \quad \Theta = \begin{bmatrix}
    \nabla^0 h \\
    \nabla^1 h_i \\
    \nabla^2 h_{ij} \\
    \vdots
\end{bmatrix}
\end{align*}
\]

(29)

where \( i, j, k = 0, \ldots, l \), and the matrix \( \Theta \), is by definition [4], the observability matrix of system (1).

Lemma 2: For a Vision-aided Inertial Navigation System, \( \Theta \dot{x} \neq 0 \)

Proof: The VINS process model can be written as:

\[
\begin{align*}
    \dot{x} &= \begin{bmatrix}
        s \\
        \dot{v} \\
        \dot{p} \\
        \dot{p}_f \\
        \dot{b}_u \\
        \dot{b}_e
    \end{bmatrix} = \begin{bmatrix}
        \frac{1}{2} Db_c \\
        g - \lambda T \dot{b}_c \\
        v \\
        0 \\
        0 \\
        0
    \end{bmatrix} + \begin{bmatrix}
        \frac{1}{2} D' \\
        0 \\
        0 \\
        0 \\
        0 \\
        0
    \end{bmatrix} \omega + \begin{bmatrix}
        0 \\
        C^T \\
        0 \\
        0 \\
        0 \\
        0
    \end{bmatrix} a
\end{align*}
\]

(30)

where \( \frac{1}{2} D \triangleq \frac{\partial \hat{h}}{\partial \hat{h}} \). According to [5], the system’s observability matrix can be split into the product of two matrices:

\[
\Theta = \Xi B
\]

(31)

where \( \Xi \) is a full column rank matrix, and \( B \) is:

\[
B = \begin{bmatrix}
    \xi & 0_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & I_3 & 0_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & I_3 & 0_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & I_3 & 0_3 \\
    0_3 & 0_3 & 0_3 & 0_3 & I_3
\end{bmatrix}
\]

(32)
where $\xi$ is a $3 \times 3$ full rank matrix. Then:

$$\mathbf{B} = \mathbf{B}(f_0 + f_1 \varphi + f_2 a)$$

$$\mathbf{x} = \mathbf{B} \begin{bmatrix} \xi & 0_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & I_3 & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & I_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 \end{bmatrix} \begin{bmatrix} C(p_f - p) \partial \theta \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= \mathbf{B} \begin{bmatrix} 0_3 & -C & 0_3 & 0_3 & 0_3 \\ 0_3 & C & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & I_3 & 0_3 \\ 0_3 & 0_3 & 0_3 & 0_3 & I_3 \end{bmatrix} \begin{bmatrix} \frac{1}{2} D b_a \\ g - C T b_a \\ v \\ 0 \\ \omega \end{bmatrix}$$

$$= A \begin{bmatrix} -[C(p_f - p)] b_g - C v + [C(p_f - p)] \omega \\ -[Cv] b_g + Cg - b_a + [Cv] \omega + a \\ 0 \\ -[Cg] b_g + [Cg] \omega \\ 0 \end{bmatrix} \begin{bmatrix} \mathbf{1} \\ 0 \\ \mathbf{1} \end{bmatrix}$$

Since $A$ has full column rank [5], $\mathbf{B} \neq 0$.

Based on Theorem 1 and Lemma 2, the time delay $\tau$ is observable. Therefore, there exists no unobservable direction, that is not spanned by $N$.

V. CLONING DURING HOVERING

Given sufficient baseline, between consecutive camera poses, the proposed sliding window estimator manages cloned poses, using a first in, first out (FIFO) scheme. Specifically, before cloning the IMU pose corresponding to the new image, into the estimator’s sliding window, the oldest pose is marginalized, such as the length of the sliding window remains constant.

Assume that at time step $k - 1$, the sliding window comprises $N$ poses, $\{x_{k-N}, \ldots, x_{k-1}\}$. After processing feature tracks, firstly observed in image $k - N$, the oldest pose $x_{k-N}$ is marginalized. At the next time step $k$, the new pose $x_k$ is added to the estimator’s optimization window.

In short, the FIFO scheme slides the window of camera poses forward in time, which is the most commonly used image management scheme employed by sliding window filters. Under sufficient motion, a FIFO cloning strategy leads to poses uniformly distributed in space and time, which allows i) the triangulation of features, for linearization purposes, and ii) the representation of camera poses as an interpolation of consecutive IMU poses.

Furthermore, under motion with sufficient baseline, the system has four unobservable directions, equal in number to the system’s theoretical minimum.

However, if the sensor platform is "hovering" (i.e., the camera performs pure rotational motion, or remains static), an alternative cloning strategy is required such that no new unobservable directions are introduced [6].

A simple solution, is to use a last in, first out (LIFO) [6] cloning strategy, during which a new (hovering) IMU pose replaces the newest (hovering) clone in the sliding window. Such a scheme, maintains non-hovering camera frames, in the estimator’s theoretical minimum.

Consider a sliding window comprising the poses $\{x_{k-N}, \ldots x_{k-2}, x_{k-1}\}$. At the arrival of image $k$, the platform enters a hovering state, and the management of cloned poses switches to a LIFO scheme. After $M$ images, the sliding window will comprise the poses $\{x_{k-N}, \ldots x_{k-2}, x_{k-1+M}\}$.

Clearly, the increasing time interval between the newest poses of the window, $\{x_{k-2}, x_{k-1+M}\}$, does not allow us to employ the interpolation model, for processing measurements originating from image $k - 1 + M$. Instead, we propose an alternative cloning strategy (Algorithm V).

Specifically, instead of maintaining only the most recent hovering pose, in the estimator’s window, we keep the two most recent hovering poses (i.e., $x_{k-2+M}, x_{k-1+M}$). Such a strategy, allows us to use this pair of clones, for modeling measurements originating from the newest image $k - 1 + M$, while at the same time non-hovering poses remain in the estimator’s window such as baseline is maintained, between the camera frames comprising the estimator’s window.

REFERENCES

Algorithm 1 Proposed clone management at the arrival of image $k$, for a sliding window of size $N$

```
if Regular Motion then
    Drop oldest pose, $\{x_{k-N}\}$
    Add new pose, $\{x_k\}$
else if Switched from Regular Motion to Hovering then
    Drop the two most recent poses $\{x_{k-2}, x_{k-1}\}$
    Add new pose, $\{x_k\}$
else if Remained in Hovering then
    Drop second most recent pose, $\{x_{k-2}\}$
    Add new pose, $\{x_k\}$
end if
```