Spline Fusion: A continuous-time representation for visual-inertial fusion with application to rolling shutter cameras

Steven Lovegrove
slovegrove@gwu.edu

Alonso Patron-Perez
apatron@gwu.edu

Gabe Sibley
gsibley@gwu.edu

Department of Computer Science
The George Washington University
Washington, DC, USA

In this paper, we describe a method for performing SLAM and visual-inertial calibration robustly using inexpensive sensors such as rolling shutter CMOS cameras and MEMS IMUs. We make use of a continuous-time model for the trajectory of the camera that naturally allows us to fuse information from many unsynchronized and potentially high-rate sensors whilst limiting state size. We model the rolling shutter of a camera explicitly and can form errors generatively on inertial measurements. This model is not limited to visual-inertial SLAM and may also simplify integration of other sensors such as spinning SICK Laser rangers.

At the heart of our approach lies a continuous trajectory representation similar to the one presented in [2]. We chose a formulation which offers:

- Local control, allowing the system to function online as well as in batch.
- C^2 continuity, to enable us to predict IMU measurements.
- A good approximation of minimal torque trajectories.

Cubic B-Splines are a well-known representation for trajectories in $\mathbb{R}^3$, but are not so easily applied when dealing with 3D rotations, such as interpolation in $\mathbb{SO}3$. For example, C^2 continuity is not necessarily preserved [3]. We choose to parameterize a continuous trajectory using cumulative basis functions formed using the Lie Algebra, equivalent to that proposed in [1]. Using cumulative B-Spline basis functions were first proposed for quaternion interpolation in [4] in the context of computer animation. This representation is not only C^2 continuous, but it also provides a very simple second derivative formulation.

The cumulative B-Spline parameterization enables the computation of analytical time derivatives at any point in the spline. This allows us to trivially synthesize accelerometer and gyroscope measurements, which we can in turn use to form direct errors on observed measurements.

Another advantage of a continuous-time framework is in dealing with rolling shutter cameras. Although the projective geometry of a rolling shutter camera remains the same as that of a global shutter camera, every line of the image is exposed for a different period, each one more delayed than the last. When the camera is in motion, this can cause the image to appear distorted and skewed (Figure 2a). Using a continuous-time model for the motion of the camera, we are free to treat every line of the image as its own exposure as shown in Figure 1.

Given generative models for visual and inertial data, we can solve for spline and camera parameters in batch or over a window by minimizing an objective function formed from the difference of measured to predicted observations. By using a continuous-time formulation, reprojection errors and inertial errors can be treated uniformly, weighted by their respective information matrices computed from device specifications or calibration.

We have conducted experiments in both simulation and with real data to evaluate our flexible continuous-time approach. We present sliding window visual odometry results on a simulated monocular rolling shutter dataset (Figure 2) and then continue by demonstrating our system on real data for joint visual-inertial SLAM and self-calibration (Figure 3).

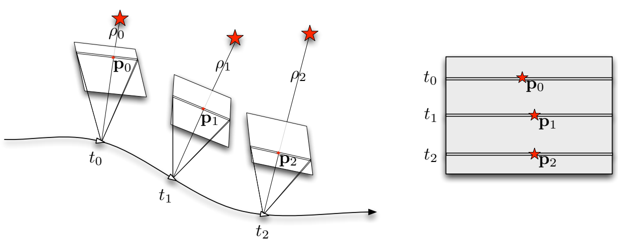
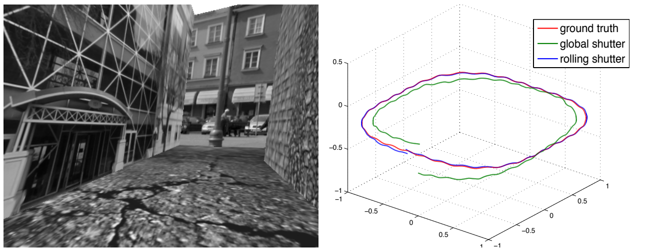


Figure 1: Rolling shutter cameras are easily modeled with continuous-time SLAM. Landmarks observed at pixel locations $\mathbf{p}_i$ are represented by their inverse depth, ρ_i and time of measurement, t_i. Each scanline is effectively a single push-broom camera (left); such scanline-camera measurements are captured over time, which defines the image returned by the actual sensor (right).



(a) (b)

Figure 2: Results from simulated monocular rolling shutter experiment. (a) Still from simulated sequence exhibiting large rolling shutter warp. (b) Comparison of ground truth trajectory (red) against estimated trajectories when modelling (blue) and not modelling (green) rolling shutter.

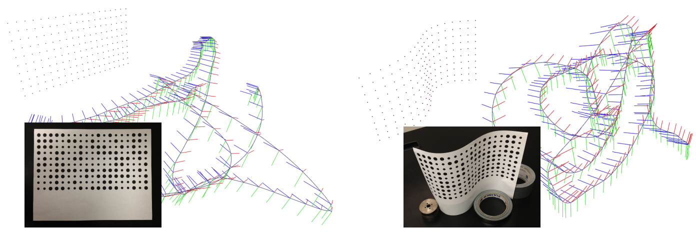


Figure 3: Example estimates computed for full-SLAM and self-calibration including camera-to-IMU, IMU biases, gravity vector, camera intrinsics (with distortion), 3D landmark locations and sensor trajectory. Again, the structure and scale in these figures are accurately estimated without any prior knowledge of the scene. This being our first experimental analysis, full and robust front-end feature tracking has not been completed, and the grid is only used to simplify data-association. This still allows evaluation and demonstration of full SLAM and visual-inertial self-calibration.

[1] P. Crouch, G. Kun, and F. Silva Leite. The de casteljau algorithm on lie groups and spheres. *Journal of Dynamical and Control Systems*, 5(3):397–429, July 1999.

[2] P. Furgale, T. D. Barfoot, and G. Sibley. Continuous-time batch estimation using temporal basis functions. In *International Conference on Robotics and Automation*, 2012.

[3] M-J. Kim, M.-S. Kim, and S. Shin. A c^2-continuous b-spline quaternion curve interpolating a given sequence of solid orientations. In *Computer Animation*, 1995.

[4] M-J. Kim, M-S. Kim, and S. Shin. A general construction scheme for unit quaternion curves with simple high order derivatives. In *SIGGRAPH*, 1995.