# Event-based Visual Inertial Odometry Supplemental

Alex Zihao Zhu, Nikolay Atanasov, Kostas Daniilidis
University of Pennsyvlania
{alexzhu, atanasov, kostas}@seas.upenn.edu

## 1. Event-based Camera Primer

Event-based cameras follow the same lens projection model as traditional cameras, and differ in how they process the light that hits each pixel. Instead of integrating light intensity over fixed time windows, event-based cameras trigger an event at a pixel when the log intensity over that pixel changes over a tunable threshold $C$.

$$\Delta \log(I(x, y)) \geq C$$

Events are triggered asynchronously as these intensity changes occur, and each event is comprised of a vector:

$$e_i = \{x_i, t_i, p_i\}$$

where $x_i$ is the image location of the event, $t_i$ is the exact time that the event occurred, with microsecond accuracy and $p_i$ is the direction (polarity) of the intensity change ($-1$ or $+1$). In this paper, we do not use the polarity information.

The high temporal accuracy of the timestamp, combined with the asynchronous nature of the sensor, allows for extremely high rate data acquisition, which forms the basis for this paper. In addition, the logarithm in the event generation equation allows for the sensor to work over a very wide range of light intensities, giving the camera very high dynamic range. There are also many other benefits such as low power consumption and low data rates compared to traditional cameras.

## 2. Full Results

In Figure 1, we present the full error plots of from our evaluation on the Event-Camera Dataset. We include both the raw position and rotation errors (left), as well as the percentage position error against distance traveled and rotation error against distance traveled (right). Solid lines represent EVIO while dashed lines represent KLTVIO.

## 3. Implementation Details

For initialization, the OptiTrack frame (ground truth) is set as the global frame for the filter, and the pose of the IMU is initialized according to the first OptiTrack pose. The initial velocity is determined by averaging the numerical derivative between the first eight OptiTrack position readings, and the biases are initialized at zero. We assume that gravity acts against the z axis of the OptiTrack frame.

100 features with spatial window size 31x31 pixels are tracked at all times, with a maximum feature track length of 100 frames. The temporal window length is initialized at 0.2 seconds, which worked across all sequences. Thereafter, the temporal window size is set at 3 times the median lifetime of features within the previous window.

For the MSCKF, the feature covariance is set to a diagonal matrix with $\begin{bmatrix} \frac{1}{1}^2}{f_u^2} & \frac{1}{1}^2}{f_v^2} \end{bmatrix}$ along the diagonal, where $f_u$ and $f_v$ are the focal length of the camera in the $x$ and $y$ directions, respectively. For further outlier rejection, the final error from the Gauss Newton estimate of each feature vector's 3D position is thresholded, and any error above $\frac{10^2}{f_u^2}$ is rejected. Our C++ filter code is based off of the MATLAB code from [1], and, like the original authors, we found the filter to be extremely sensitive to the tuning parameters, although the ones we note here worked for all of the sequences.

During testing, we also ran our algorithm on the outdoor sequences presented in the dataset. However, two main issues arose that caused tracking and state estimation to fail. First, as there is no ground truth data, we were unable to initialize the sensor state rotation, velocity and biases. This resulted in substantial error from assuming zero initial velocity. Second, the majority of the objects that generate events in these sequences are window frames, which tend to be horizontal and vertical lines. Due to the aperture problem, we were not able to maintain good feature tracking on these objects, and so the majority of tracked features are clustered in the horizon, where there is very little apparent motion, causing significant drift of the state estimate.

## References

[1] L. Clement, V. Peretroukhin, J. Lambert, and J. Kelly. The battle for filter supremacy: A comparative study of the multi-state constraint kalman filter and the sliding window filter. In *Computer and Robot Vision (CRV), 2015 12th Conference on*, pages 23–30, 2015. 1

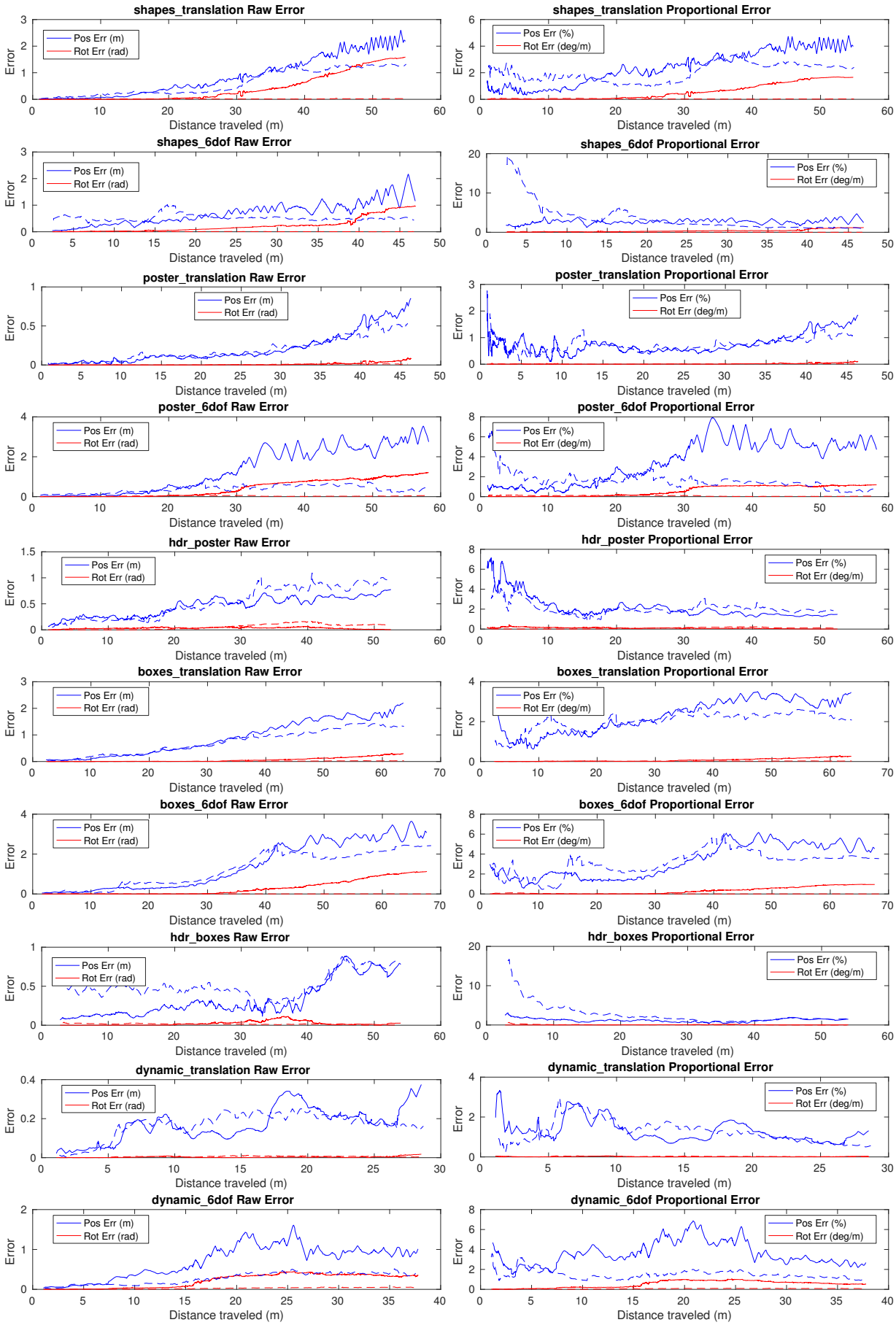


Figure 1: Raw and relative position and rotation errors for EVIO and KLTVIO compared to ground truth. Solid lines represent EVIO while dashed lines represent KLTVIO.