Auto-Calibration with Stop Signs

Detector and Filter method

Yunhai Han
Department of Mechanical and Aerospace Engineering
y8han@eng.ucsd.edu
Yuhan Liu
Department of Computer Science and Engineering
yul139@eng.ucsd.edu

September 11, 2020

1 Introduction

The chapters in this report can be split into three parts:

1. System pipeline
2. Performance of mask RCNN(fast RCNN) algortihms
3. Calibration results after implementing Kalman Filter

2 System pipeline

The system pipeline can be illustrated as:

- Detection(bounding box for stop sign)
- Color Segmentation to find the edge of the sign
- Edge Detection to refine edges (early line detection)
- Line refinement with perpendicular search
- Estimation of intersection points for $\mathrm{N} > 4$ points
- Homography estimation for calibration
- Update of camera calibration

Currentlt, all of this can be done automatically. In other words, the input file is the raw **rosbag** file and the output files are the calibration results. But, the manual work are needed in two parts:

- Separate the whole video into several separate clips which capture stop signs.
- Manually remove some bad-quality images. For example, we obtain in total 21 sets of corner points(camera6), we have to manually remove one of them. The reason is that the estimated line does not match the real line well(due to some issues that Yuhan introduced before) and this would make the calibration result much worse.

3 Detector

4 Filter method

We implement Kalman Filter as professor mentioned in the report. Since both the state transition equation and measurement equation are linear and no control inputs are involved, it is very easy for implementation.

We think of three different modes on how to implement Kalman filter on the images we collected.

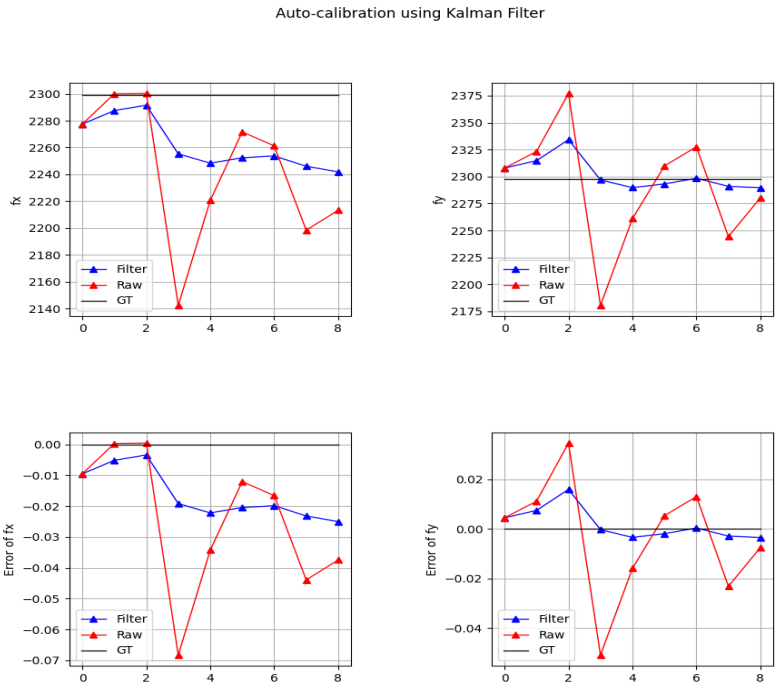
Suppose we have in total N images.

- Mode1: Each time we select $N-1$ images and calibration the camera. Run Kalman filter on N sets of results.
- Mode2: Set a moving window and its length is $K(K<N)$. We gradually select K images in the temporal order. Run Kalman filter on $N-K+1$ sets of results.
- Mode3: Gradually increse the number of images used for calibration. at the first iteration, only 2 images are used(minimum number of imgaes that are required) and at the last iteration, all N images are used.

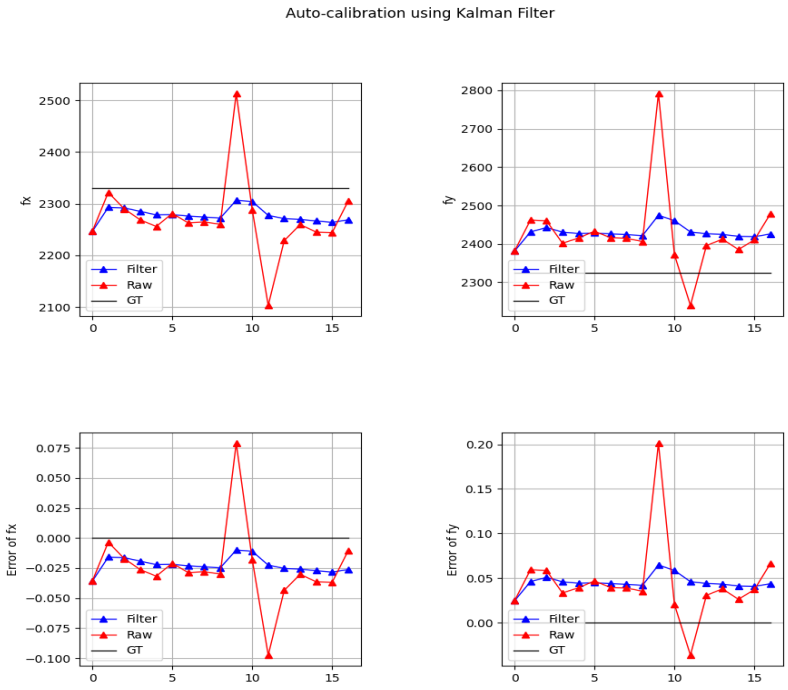
We do experiments for camera1 and camera6 and this are the results for the three modes(the noise parameters are the same):

- Mode1:

  camera1:



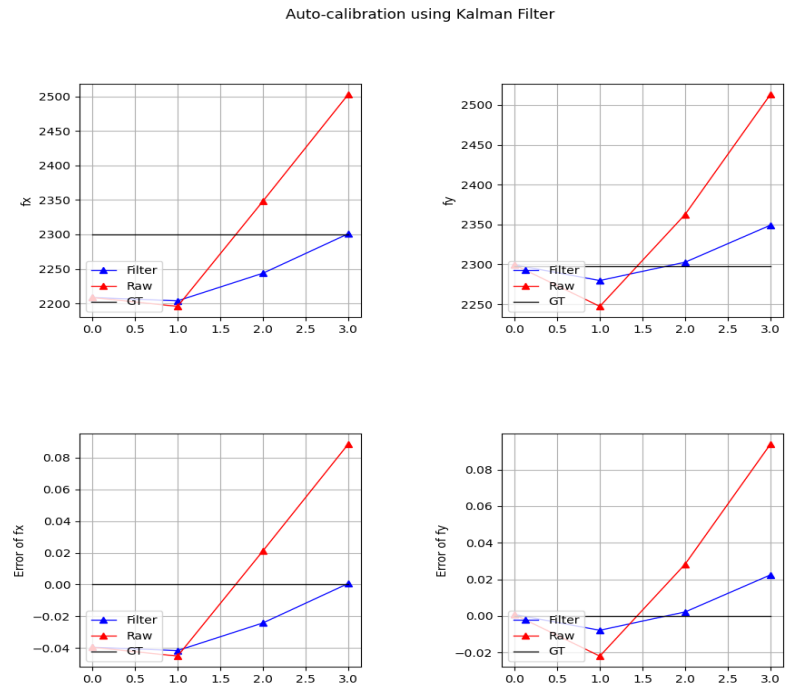
  camera6:



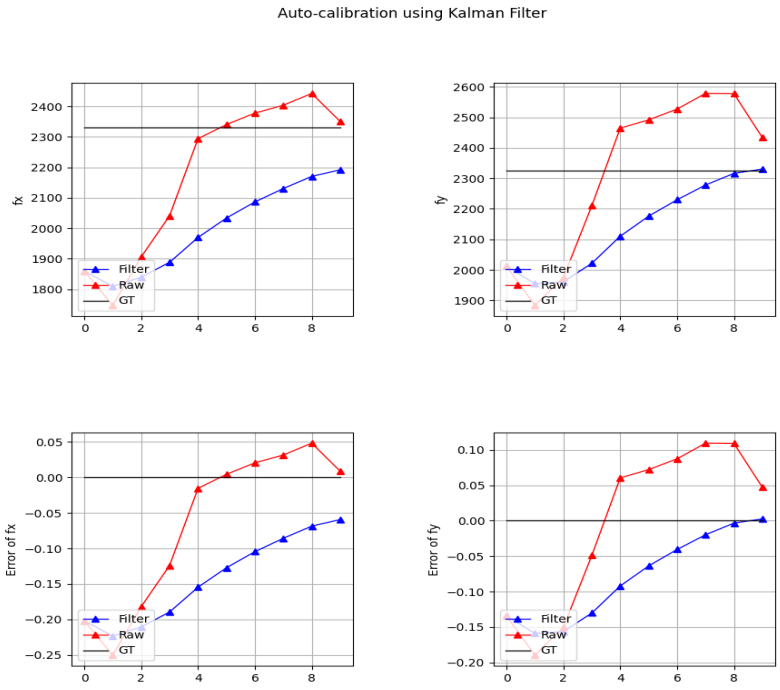
Because each time we use $N-1$ images, the calibration results themselves are the most accurate, there is no big difference after implementing Kalman Filter.

- Mode2:

  camera1($K=6$):

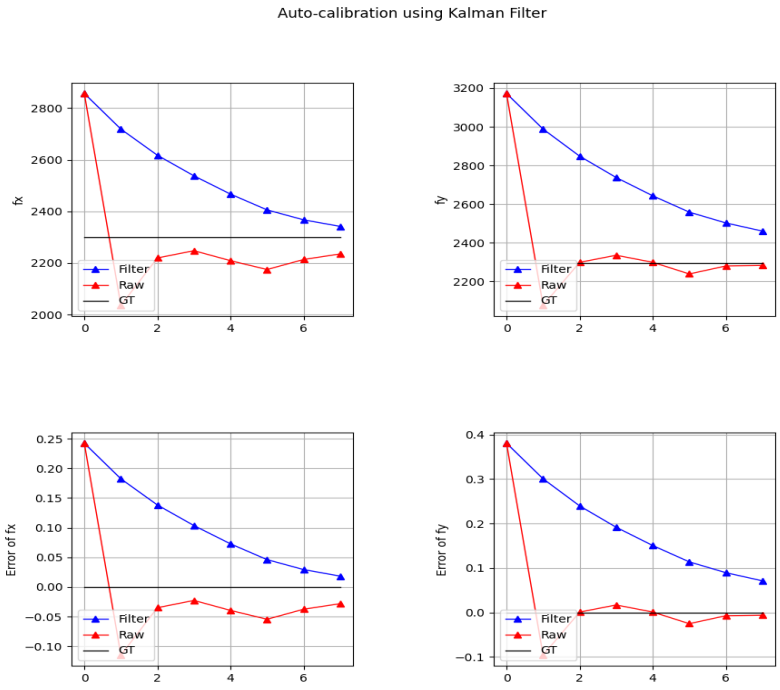


  camera6($K=8$):

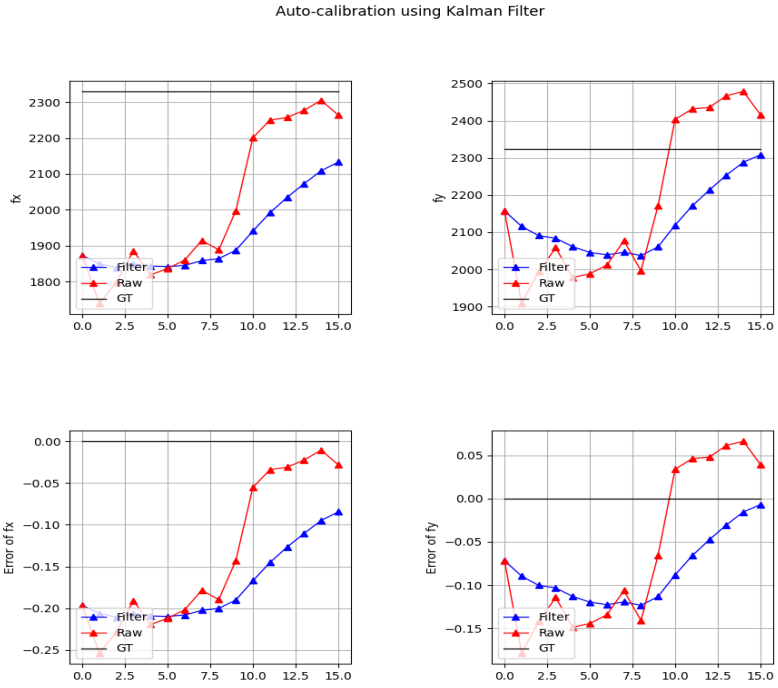


- Mode3:

  camera1:



  camera6:



The shapes of two curves are more desirable, since with the number of images increasing, the results are getting more accurate. We think if we can adjust the measurement noise:it decrease with the number of images increasing, the final results would be much better.