Kalman-filter for partial linear systems for tracking pool balls

Marlon Lückert
Bachelor of Science
marlon.lueckert@haw-hamburg.de

Julius Neudecker
Bachelor of Science
julius.neudecker@haw-hamburg.de

March 2020

Contents

In this proposal we are going to discuss an advanced implementation of the Kalman-filter [1] to improve measurement quality and predict the movement of balls on a pool table. We are going to point out the necessity of an advanced filter design in this particular case. The Problem we are going to solve is the following: Since the performance of the filter deteriorates in cases of a rapid change in direction, the filter has to be able to adapt more quickly to these rapid changes. Our plan is to derive an implementations with adaptive behavior. The implementation will be tested in a simulator and with real world video footage of a pool table. The goal is to provide a filter design with a significant lower MSE than a vanilla implementation.

1 Introduction

At first glance the game of pool is very suitable to examine the behavior of a kalman-filter enhanced tracking system based on pure visual tracking. The surface of a pool table is made of a thin fabric which covers a hard surface i.e. slate or granite. The balls nowadays are usually made out of resin. This combination of materials creates very small rolling resistance and the balls behave almost fully elastic on collision. Since this is only a 2 DOF[1] problem, this can be solved with a simple linear kalman filter implementation. The problem is, when two balls hit each other or a cushion the velocity vector changes its orientation instantly. If this isn't taken into account, the filter needs some time to adapt to the new direction of movement and will produce wrong estimations during this time.

This behavior is independent of the type of kalman implementation being constant-velocity-model or constant-acceleration-model. A kalman filter will assume the direction of movement on any given sample is about the same as in the last sample. It will therefore create wrong estimations if the direction of movement changes drastically in a short period of time. The time the filter needs to recover depends on the filter gain.

We also use the filter to predict values for any given length into the future by feeding back its estimations as actual state. The quality of this prediction however depends on several factors, i.e. the applied process noise and framerate of the video.

2 Problem and Motivation

As introduced in the previous section is the lack of adaptability in a vanilla Kalman implementation. One could argue that this can be taken into account by using a higher overall process noise which would in turn lead to significantly reduced overall quality. This gets worse with increased speed and lower framerate. Essentially rendering the filter useless at a certain point.

We think this is a good point to develop this algorithm for future application in VR and AR based pool trainers and augmented broadcast experiences.

3 Hypothesis

The tracking and prediction of billiard balls by a Kalman-Filter improves by explicitly taking the physical conditions of a billiards table into account when implementing the filter.

4 State of research

Jong-Yun Kim and Tae-Yong Kim [2] developed a method to provide robust tracking of a soccer ball. They provide a solution for the problem for the case that the soccer ball might be occluded by the player at any given time, which results in a diminished tracking accuracy. In this case they used the velocity vector of the player to substitute for the ball presuming that the ball moves in the same direction as the player does.

Jia et.al. [3] conducted research in the trajectory of pool balls, which helped us to decide which kalman model is the most suitable.

Shiuh et.al. [4] provided a good starting point how to create a tracking algorithm for pool balls. They also developed an algorithm to track occluded objects using an adaptive kalman filter. In this case they used to threshold in order to determine whether the object can still be reliably tracked. If this isn't the case the filter will rely only on predicted values until the object can be tracked reliably again.

Salzmann and Urtasun [5] proposed a more general approach for tracking. They were able to recreate a highly accurate tracking from a noisy picture based on newtons 2nd law and markov models. Using different constraints and presumptions they were even able to extract physical parameters like friction and trajectories.

Mohamed and Schwarz [6] are using partly the same approach as we do to improve the results created by INS/GPS[2] Sys-

[1] Dimensions Of Freedom - determines the possible rotation or translation along each given axis

[2] Inertial Navigation System / Global Positioning System

tems. However their approach only targets the 'Q' and 'R' parameters of the filter.

Sarkka and Nummenmaa [7] created an adaptive kalman implementation which adapts itself to time-varying noise parameters. Since our input data is constant in this regard, we decided to simulate for the optimal filter parametrization instead of relying on the filter to adapt itself.

Gabdulkhakova and Kropatsch [8] use a kalman filter to create a video analysis tool for snooker game broadcasting.

5 Research design

5.1 Operationalization

In the following section we are going to operationalize the different terms of our hypothesis.

5.1.1 Tracking

The tracking describes the deviation from the filtered position of the billiard ball (result of Kalman-Filter) to the real position of the billiard ball. The position divides into three dimensions, the x-coordinate, the y-coordinate, the z-coordinate. Since we are investigating a billiard game we disregard the up-direction (z-coordinate), because the ball moves on a flat surface and will not leave the plane. We are measuring the coordinates in pixels. The range of the pixels is defined by the resolution of the input video on which the Kalman-Filter gets applied to. The origin, coordinate (0,0), is always the top-left corner. The final deviation from the two position is calculated with the Euclidean distance, which results in a length in pixels.

The position of the billiard ball changes over time, because we are investigating video sources. To measure the deviation of the whole video and to get a key figure, we are calculating the mean square error (mse) in which the error is the pixel distance between the two positions.

5.1.2 Prediction

The prediction describes the deviation from the calculated future position of the billiard ball to the real position at this specific time. The positions use the same measurements as described in the previous chapter. The time is measured in frames. The specific number of frames we will look in the future will vary in our experiments. To make the frame count comparable with different input video source we align them with the framerate (frames per second) of the video.

5.1.3 Improvement

The improvement compares a standard Kalman-Filter to our implementation of the Kalman-Filter. The improvement is measured as the difference between the different mean square errors of the tracking and the prediction. The smaller the mse of our Kalman-Filter compared to the standard implementation the greater the improvement.

5.2 Data Acquisition

We are going to acquire our data from billiard simulations and real videos.

5.2.1 Simulation

We construct a simulation to fully control every parameter and the state of the billiard game. This helps us to accomplish a better conclusion to the performance in real video. Our examination units are the framerate, the ball velocity, the sensor noise.

Framerate The framerate describes the framerate of the video, which is equal to the sampling rate of the Kalman-Filter. A standard Kalman-Filter performs much worse on low framerate because it gets less data to process. We test different framerate to inspect the performance of our implementation for different environments. For realtime applications on mobile devices our implementation has to perform well on low framerates. We are going to test the standard framerates 30 and 60 FPS and a very low framerate at 10 FPS.

Ball Velocity The ball velocity describes the speed of the ball, which depends on how hard the player hits the ball. The velocity is measured in pixel per frame. The cover a lot of cases, we choose a low velocity of 300 pixel per frame, an average velocity of 500 pixel per frame and a high velocity of 700 pixel per frame. We have to keep in mind that a high velocity also leads to more collisions because the ball will move a longer distance.

Sensor Noise When we analyze realtime videos we have detect the balls in the video frame. The detection is not always perfect and heavily relies on the light conditions and resolution of the camera. To simulate the quality of detection we added a noise value which changes the detected ball position by random value in a certain range. We test a deviation of 15 pixel, 30 pixel and 60 pixel to inspect how well the Kalman-Filter will perform.

Change of parameters To test the parameters individually we only change one parameter at the time. Three variations on each parameter lead to 9 simulation environments in total.

5.2.2 Real Videos

After testing the filter in the simulation we are also doing some test on random billiard videos. Since we are relying on a ball detection algorithm to identify the billiard balls, we cannot obtain the real position of the ball like we could in the simulation. That's why we cannot make statements about the tracking quality, but we can evaluate the performance of the tracking.

5.3 Data Evaluation

The key figure of evaluation is the mean square error. If the means square error of our implementation is lower compared to the standard implementation in every experiment we can prove that our implementation improved the tracking and prediction of billiard balls.

6 Proposed timetable

To conduct our research in a timely manner, we propose the following schedule to have the finished research paper by the end on June 2020.

- until 20th April: develop the mathematical framework
- until 15th May: develop the simulation and create footage for evaluation
- until 31st May: implement the filter according to optimized parameters
- until 30th June: conclude results and write research paper

We are planning to submit this paper to a conference which has relevance in this particular field of research. This has yet to be determined.

References

[1] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 1960.

[2] J.-Y. Kim and T.-Y. Kim, "Soccer ball tracking using dynamic kalman filter with velocity control," *IEEE, Tianjin, China*, 2009.

[3] Y.-B. J. et. al., "Trajectory of a billiard ball and recovery of its initial velocities," *Department of Computer Science Iowa State University*, 2011.

[4] S.-K. et. al., "Video object tracking using adaptive kalman filter," *Journal of Visual Communication and Image Representation*, 2006.

[5] M. Salzmann and R. Urtasun, "Physically-based motion models for 3d tracking: A convex formulation," *International Conference on Computer Vision*, 2011.

[6] A. H. Mohamed and K. P. Schwarz, "Adaptive kalman filtering for ins/gps," *Journal of Geodesy*, 1999.

[7] S. Sarkka and A. Nummenmaa, “Recursive noise adaptive kalman filtering by variational bayesian approximations,” *IEEE Transactions on Automatic Control*, 2009.

[8] A. Gabdulkhakova and W. G. Kropatsch, “Video analysis of a snooker footage based on a kinematic model,” *Vienna University of Technology*, 2012.