# Pool-Predictor - a Kalman-Filter Implementation

Marlon Lückert B.Sc. - Julius Neudecker B.Sc.

Feburary 2020

In this article we're discussing an implementation of a Kalman-filter [1] to filter and predict the movement of balls on a pool table. We discuss the reasoning why we implemented the filter as a constant velocity model and its weak points in this situation. Since the performance of the filter deteriorates in cases of a rapid change in direction we derived two different implementations with adaptive behaviour. The implementations were tested in a simulator and with real world video footage of a pool table. In the end we compared the performaces of each filter to another.

## 1 Introduction

### 1.1 Game of Pool

At first glance the game of pool is very suitable to examine the behaviour 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. Also with only 2 DOF[1] a basic kalman filter implementation is fairly simple in this regard. But when two balls hit each other or a boundary, 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 behaviour is independent of the type of kalman implementation being CVM or CAM (see section 1.3). We call it the prediction problem in this regard. A kalman filter will assume the direction of movement on any given sample is the same as in the last sample. It will therefore create wrong estimations if the directon of movement changes drastically in a short period of time.

Because the filter allows to predict values for any given length into the future by feeding back its estimations as actual state, this will create a large estimation error if the direction is changed at some point. This however is why the game of pool is a good example to show the idea of a smart and adaptive kalman implementation.

### 1.2 Visual Recognition of moving elements

In order to create a state-vector input we to process we have to track the balls on the pool table. Since we needed reproductable results we used stock-footage as well as our own videos and processed them with the Python implementation of openCV. These videos 1 are put into an instance of simple video-processing steps to create a black and white mask of the ball contours as displayed in 2, where the center of the white pixel cluster is the ball we're looking for.

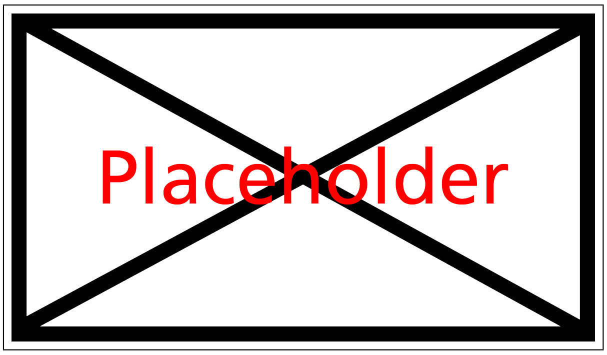

Figure 1: color picture of a pool table

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

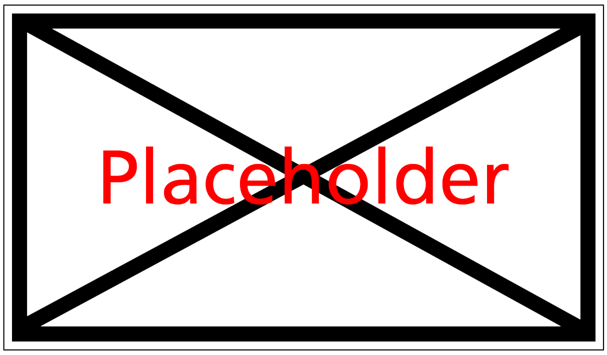

Figure 2: contours of balls after processing

### 1.3 Kalman-Filter

here comes some text

**Constant velocity model** here comes some text

**Constant acceleration model** here comes some text

## 2 Related Work

In the following, we discuss some previous work that is related to our work but none of these articles targets the problem in the same manner.

Jong-Yun Kim and Tae-Yong Kim [2] developed a method to provide robust tracking of a soccer ball. Their problem is that the soccer ball might be occluded by the player at any given time so an optical tracker could not deliver any results. 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.

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 threshhold in order to determine wheter 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 could 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] Systems. However their approach only targets the 'Q' and 'R' parameters of the filter.

Sarkka and Nunemmna 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 games.

## 3 Implementation

### 3.1 differences between implementations

### 3.2 CVM devel + Problems in Prediction

### 3.3 Dyn CVM devel

### 3.4 Smart Devel

## 4 Simulations

### 4.1 Structure of Simulation

optimal parameters

### 4.2 Sampling

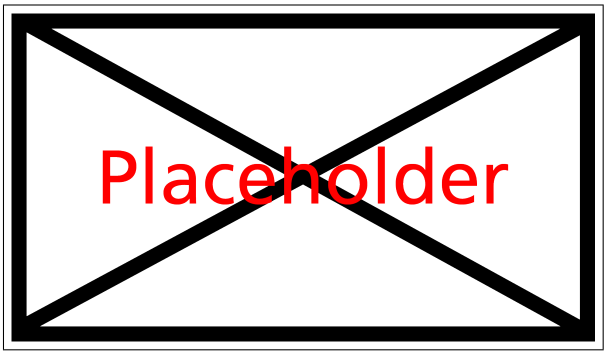

Figure 3: Caption: Text

[2] Inertial Navigation System / Global Positioning System

### 4.3 Noise

### 4.4 Velocity

## 5 Results

### 5.1 Real Video - Prediction vs. Filtered

#### No Ground Truth

### 5.2 CVM vs. CAM

## 6 Future Work

## 7 Improvements

Abstand von Bande durch diskrete abtastung ungenau, weil zeitliche unschärfe entsteht (abprall zwischen zwei frames). Prediction müsste unabhängig von frames gemacht werden.

### 7.1 AR App for Training

### 7.2 Broadcasting

## 8 Acknowledgement

T.Edeler wegen Vorlesung diesdas.

## 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.