Project Report for CSCI – 5454
Design and Analysis of Algorithms

Kalman filter based dynamic tracking for visual odometry applications

Shiril Tichkule

Fall 2017

University of Colorado Boulder

Kalman filter based dynamic tracking for visual odometry applications

Shiril Tichkule

University of Colorado Boulder, 425 UCB, Boulder, Colorado 80309, USA
Shiril.Tichkule@colorado.edu

Abstract. This paper presents the formulation and implementation of a Kalman filter based dynamic object tracking algorithm. Object tracking is one of the most fundamental problems in the area of visual odometry, that deals with predicting and tracking the position, velocity, and attitude of a moving body. We have proposed a solution to this problem based on the principle of adaptive state estimation using Kalman filters. Specifically, we have analyzed the motion of an object across a static background, and used predictive filter outputs to track it's centroid coordinates across video frames. The Kalman filter was constructed using a physical model based on Newton's laws of motion, and was augmented with the relevant noise covariance matrices estimated from each image. The algorithm was found to yield sub-pixel tracking accuracy in the horizontal dimension of the object's motion. The vertical tracking accuracy was found to be on the order of ca. 5 pixels, owing to greater uncertainty (noise) in the object's vertical motion. Although the algorithm performs well for motion constrained to two dimensions, we predict that for more complex trajectories, similar accuracies can be obtained by use of additional data from on-board inertial sensors or multiple camera angles.

1 Background and Motivation

Dynamic object tracking deals with the problem of tracking motion of single/multiple objects, as they traverse through a static/dynamic background. With the increasing invasiveness and cost effectiveness of inertial sensors, such applications have gained significant importance in the field of guidance and navigation. Traditionally, and even in many current day applications, navigation is accomplished with the aid of GPS (Global Positioning System). However, with sensor fusion, it has become possible to augment GPS inputs with visual signals, for enhanced determinism in object tracking. This is especially useful when there is intermittent loss of the GPS tracking signal, in which case, visual localization keeps track of the object, and vice-versa [3].

Visual localization belongs to the general class of problems that occur in the area of odometry. From the perspective of navigation, and aerospace sciences, odometry refers to tracking the motion of a body with respect to its velocity, position, trajectory, or any combination of these parameters [2]. It can also be considered to be a state estimation problem from a control systems standpoint. In that case, any of the previously mentioned parameters, or a subset of those, can make up the state space which needs to be estimated during each time-slice of the prediction operation [6].

Using this adaptive state estimation concept, we have devised a mechanism to predict the position of an object of interest in two dimensions, moving across a static background. In our case, the state space is characterized by the two-dimensional coordinates corresponding to the centroid of the object being tracked. A Kalman filter (KF) implementation has been chosen since our goal is to estimate the current state of the object using and ensemble of knowledge regarding past states. This approach follows from the fundamental Kalman filtering principle which states that aggregating data from past states produces better estimates as opposed to a single measurement of the current state corrupted with noise.

As a top-level overview of our methodology, a sequence of images containing the object to be tracked is provided as input to the tracking algorithm. For each image in the video, the current state of the Kalman filter is determined using the previous (priori) state variable, and the error covariance matrix ('predict' phase). This is followed by the 'update' phase, where the Kalman filter state equations are updated to include the current observation corrupted with noise. This goes on recursively, and during each step, the 'predict' phase yields a prediction for the updated position of the object. In Sec. 2, we will mathematically formulate aspects of the Kalman filter setup, while the implementation and data processing aspects will be dealt with in Sec. 3. Secs. 4 and 5 will present the results and conclusions, respectively.

2 Kalman Filters – Adaptive State Estimation

As described in Sec. 1, a sequence of images containing to be tracked will be provided to the Kalman filtering algorithm, which will then yield position estimates on a frame-by-frame basis. More generally speaking, the entire problem can be succinctly represented by the following four steps [1]:

- Get prior knowledge of state space
- Predict current state based on a pre-defined physical model
- Update current state by comparing prediction to current input
- Aggregate this information into the prior knowledge, for application to the next time step

Based on this, it is apparent that we need a physical model which governs state transitions of the object from one frame to another. This can be most easily realized by making use of Newton's laws of motion. The position of the object p at time t can be represented by Newton's second law:

$$p_t = p_{t-1} + u\Delta t + \frac{1}{2}a(\Delta t)^2. \tag{1}$$

The velocity of the object u at time t can be represented by Newton's first law:

$$u_t = u_{t-1} + a\Delta t, \tag{2}$$

where a and Δt represent the acceleration and time increment, respectively. In the two-dimensional (x, y) plane, the instantaneous state variables of position and velocity can be represented by:

$$P_t = \begin{bmatrix} x_t \\ x'_t \\ y_t \\ y'_t \end{bmatrix}, \tag{3}$$

where the x'_t represents the time derivative of the x-position, equivalent to the x-component of the velocity. Eqs. 1, 2, and 3 can be concisely represented as follows:

$$P_t = T * P_{t-1} + aU, \tag{4}$$

where

$$U = \begin{bmatrix} \Delta t^2/2 \\ \Delta t \\ \Delta t^2/2 \\ \Delta t \end{bmatrix}, \tag{5}$$

$$T = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6}$$

Since we are interested only in tracking the position of the object, the T matrix may be modified to:

$$T_R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}. \tag{7}$$

We now need to augment the state dynamics equations with the following noise covariance matrices: Noise covariance of states, which will be a diagonal matrix, since the states can be assumed to be independent.

$$C = diag[\sigma_x^2, \sigma_{x'}^2, \sigma_y^2, \sigma_{y'}^2], \tag{8}$$

and a state transition covariance, which accounts for dynamism in the system, and can be computed using:

$$D = U^T * T_R^T * T_R * U, \tag{9}$$

which yields

$$D = \begin{bmatrix} \Delta t^2 & 0 & \Delta t^3/2 & 0 \\ 0 & \Delta t^4/4 + \Delta t^3/2 & 0 & \Delta t^4/2 \\ \Delta t^4/2 & 0 & \Delta t^2 & 0 \\ 0 & \Delta t^3/2 & 0 & \Delta t^4/2 \end{bmatrix}. \tag{10}$$

The measurement noise can also be assumed to be independent from state to state, and hence be given by the following diagonal matrix:

$$N = diag[\sigma_N^2]. \tag{11}$$

Since each state is characterized by the centroid coordinate of the object, the state covariance matrix can be represented using the centroid uncertainty that is introduced on account of partial sub-pixel illumination. In this case the RMS position error is on the order of 1 pixel, i.e. $\sigma_x^2 = \sigma_y^2 = 1$. Further, $\sigma_N^2 = 10^{-2}$ was calculated as the variance of the pixel blurring across all frames.

Assuming that the current input to the system can be given by $I_t = [x_t, y_t]'$, Eqs. 4 to 11 can be used to yield the following equations for the Kalman filter.

PREDICT:

$$\begin{aligned} P_t &= T * P_{t-1} + aU, \\ C_t &= T * C_{t-1} T^T + D. \end{aligned} \tag{12}$$

CORRECT:

$$\begin{aligned} KG_{t-1} &= C_{t-1} * T_R^T * (T_R * C_{T-1} T_R^T + N)^{-1}, \\ P_{t+1} &= P_t + KG_{t-1} * (I_t - T_R * P_t), \\ C_{t+1} &= (I - KG_t * T_R) * C_t. \end{aligned} \tag{13}$$

The following algorithm encompasses all of the pre-processing and Kalman filter components of the entire data processing workflow., and was realized using Matlab.

ALGORITHM: KALMAN_TRACK(video V):

- frames = read_video(V);
- background = dark_frame(frames);
- remove_background(frames,background);
- kalman_filter_initialize();
- for (i = 1 to n_frames)
  - input(x,y) = estimate_centroid(frames[i]);
  - current(x,y) = predict(past(x,y));
  - [past(x,y), KG, C] = correct(input(x,y));
  - end;
- assemble_video(frames);
- end;

---

3 Implementation and Data Processing

The Kalman filter implementation and additional image processing was tested using a short video, in which a bright colored ball (the object to be tracked) exhibits translational motion in the $x-y$ plane. The background was kept static, and the ball was the only object that moved throughout the entire video. The following is a summary of the video on which the tracking algorithm was run.

- Static background, object of interest moves across
- 240 x 320 pixels, color images
- 6 s long, recorded at ca. 30 fps
- $\Delta t = 1/30$ s
- $a = 0.005$ m/s^2

The first step in preprocessing the video is performing a background estimation. Since we had only one object moving across a static background, a simple frame-averaging based method was used to calculate the background frame, which is shown in the left panel of Fig. 1. The background image appears as expected, although upon keen observation, a faded image of the ball can be seen at the right edge of the frame. This occurs because during the latter half of the video, the ball is near stationary along the right edge, as opposed to moving actively during the first few frames. The object stationarity along with frame-averaging causes a faded image of the ball to appear in the background frame.

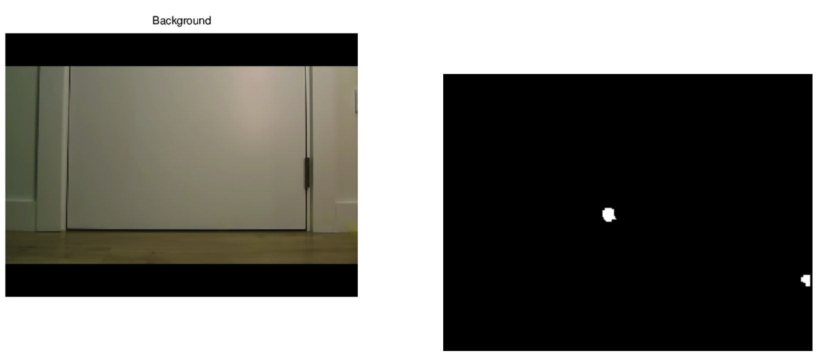

Fig. 1. Left: background frame obtained by averaging. Right: Binary image of frame no. 5 obtained by removing the background frame

This background frame is then removed from each frame of the video to yield a binary image representation of the object, as shown in the right panel of Fig. 1. This image corresponds to the fifth video frame, and the ball (along with the faded image along the right edge) are clearly visible as white spots. Centroid estimation is then performed using weighted averaging of pixel intensities of the object in each binary frame. To complete the Kalman filter setup, noise parameters referred to in Sec. 2 are computed. This entire corpus is then fed into the 'predict' and 'correct' equations of the Kalman filter to yield the tracked (filter-estimated) centroid coordinates of the object.

4 Results and Discussions

Presented in Fig. 2, are the centroid coordinates of the object as a function of time. In accordance with how the object moves, we see that the x-coordinate (blue curve) increases quasi-linearly, until the point where the ball motion greatly subsides. In parallel, the y-coordinate (red curve) exhibits oscillatory behavior, as the ball bounces across. Fig 3. shows a set of four frames in which the detected object is marked up on the basis of it's centroid coordinates. The overlap of the ball and the marked up areas show that the Kalman filter will have valid inputs for each frame.

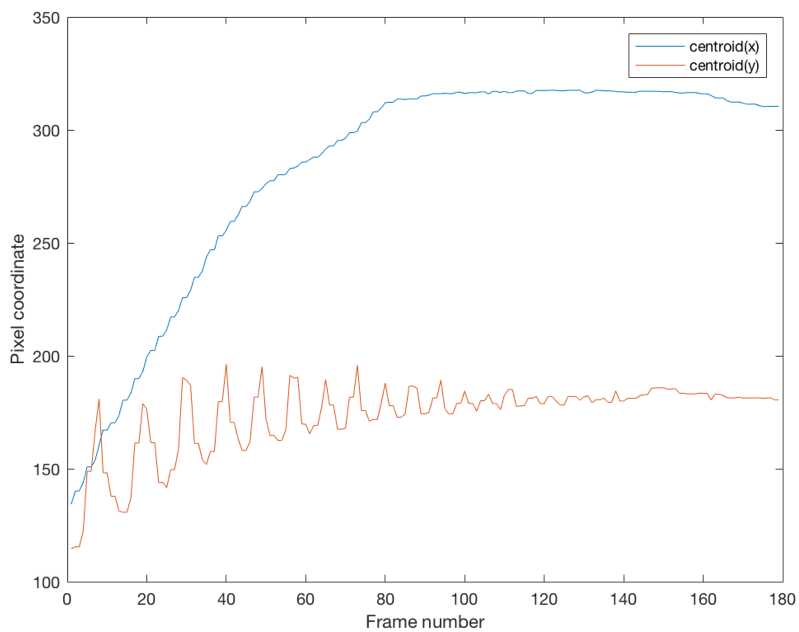

Fig. 2. Centroid coordinates of the tracked object as a function of the frame number. Blue and red curves correspond to the x and y coordinates, respectively.

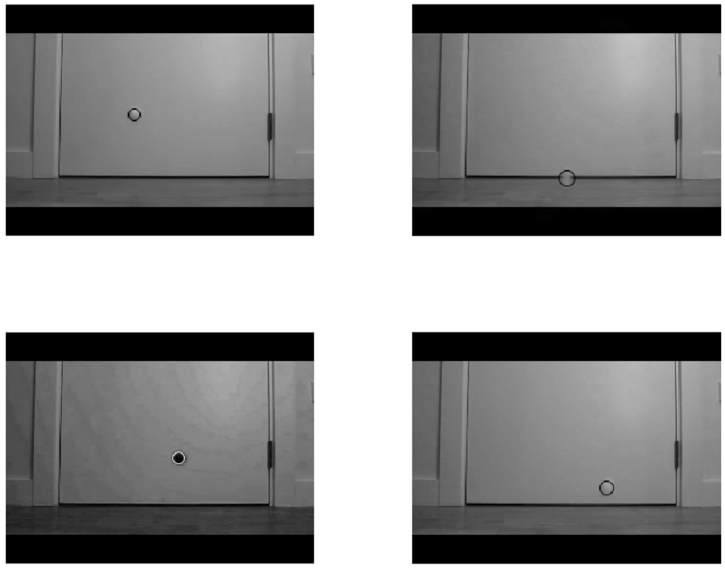

Fig. 3. Clockwise order from top-left – frame numbers: 5, 25, 45, 65. The circular ring around the tracked object has been derived on the basis of the estimated centroid coordinates in each frame.

In Fig. 4, the same four frames as Fig. 3 are presented, but now with the Kalman filter predicted output superimposed on the images (indicated by the bold circles). As expected, the Kalman filter predictions are not as accurate as the pure centroid based estimations. This is because the predictions are derived using the ensemble of centroid coordinates aggregated using all of the previous frames.

All of the Kalman filter outputs are presented in the form of a time-series in Fig. 5. Centroid time series from Fig. 2 are also included for comparison. It is observed that the blue and green curves (corresponding to the x-coordinate derived from centroids and Kalman tracking, respectively) track each other very well. This is to be expected since the object exhibits relatively smoother translational motion across in the horizontal direction. On the other hand, the red and black curves (corresponding to the y-coordinate derived from centroids and Kalman tracking, respectively) have a greater degree of disagreement. This can be attributed to the greater variability of the object's motion in the vertical direction, which introduces more 'noise' in the Kalman filter equations, and thereby relatively inaccurate predictions as compared to the horizontal direction.

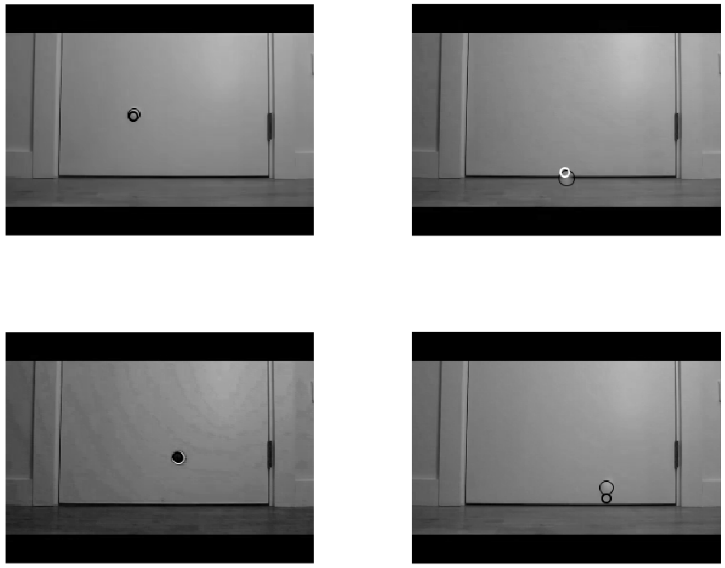

Fig. 4. Same as Fig. 3, but now the Kalman filter predicted centroids and regions of interest have been super-imposed onto the tracked object in each frame.

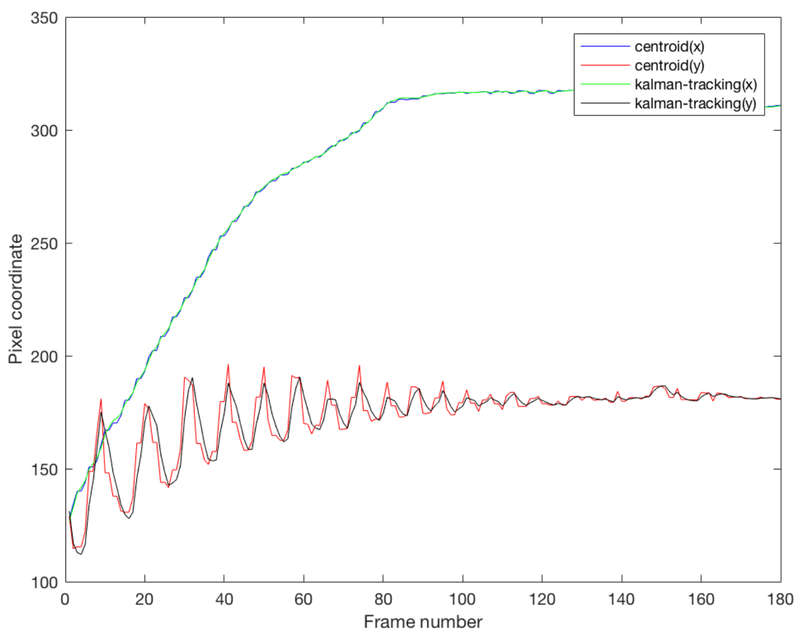

Fig. 5. Same as Fig. 2, but now with added centroid predictions derived from the Kalman filter. Green and black curves correspond to the Kalman-predicted x and y coordinates, respectively.

Observations based on these four figures can be summarized as follows:

- KF tracking in the horizontal direction has an accuracy of 0.62 pixels.
- KF tracking in the vertical direction has an accuracy of 5.27 pixels.
- Errors are greater when the object is oscillating vertically; they reduce when vertical motion subsides.
- This is as expected due to greater uncertainty in object's vertical motion.

5 Conclusions and Future Scope

Based on the findings presented in this paper, we can conclude that dynamic object tracking is achievable using a simple Kalman filter implementation. A Kalman filter was used in conjunction with Newton's laws of motion to develop a predictive object tracking algorithm. This algorithm was able to achieve sub-pixel tracking accuracy for motion constrained to the dimension (horizontal) with the least degree of uncertainty. Augmenting this algorithm to be applicable for use with a dynamic background and multiple objects is a natural choice for future exploratory work. Practical implementations of such an algorithm should be able to provide accurate predictions in cases of object occlusion, i.e. when there is intermittent loss of object imagery, due to a visual blockage [4]. For example, in the case of tracking a car along a road, there might be a tunnel which occludes the car from the observation camera. In such cases, the algorithm should be able to make accurate predictions of the car's motion based on it's previously aggregated visual data. The applicability and performance of this algorithm can be further enhanced by fusing data from multiple visual/inertial sensors [5], and will be the focus of future work.

References

1. Grewal, Mohinder S. and Andrews, Angus P.: Kalman Filtering: Theory and Practice with MATLAB. 4th ed., Wiley-IEEE Press (2014)
2. Civera, J., Grasa, Oscar G., Davison, Andrew J., Montiel, J. M. M.: 1-Point RANSAC for extended Kalman filtering: Application to real-time structure from motion and visual odometry. Journal of Field Robotics, 27 (5), 609–631 (2010)
3. Williams, B., Reid, I.: On combining visual SLAM and visual odometry. IEEE International Conference on Robotics and Automation, 3494-3500 (2010)
4. Nistr, D., Naroditsky, O., Bergen, J.,: Visual odometry for ground vehicle applications. Journal of Field Robotics, 23 (1), 3–20 (2006)
5. Sirtkaya, S., Seymen B., Alatan, A. A.: Loosely coupled Kalman filtering for fusion of Visual Odometry and inertial navigation. Proceedings of the 16th International Conference on Information Fusion, Istanbul, 219-226 (2013)
6. Simon, D.: Optimal state estimation: Kalman, H infinity, and nonlinear approaches. John Wiley & Sons (2006)