OBJECT TRACKING USING KALMAN FILTER

Bodong Zhang, Sayali Tope, Soroosh Sabeti

University of Utah
Department of Electrical and Computer Engineering
Salt Lake City, Utah 84112

ABSTRACT

This report summarizes the key concepts and findings of our project on the topic of tracking objects by means of Kalman filter. In this project we tried implementing a combination of image processing methods to detect an object of interest within each frame of a video sequence and consequently use the acquired data to estimate and predict the location of the object in the following frames. The utilized algorithm for object detection is color based and can address challenges such as alterations in size of the object, and presence of multiple objects similar to the target. Our results also indicate, in the absence of abrupt variations in motion parameters, Kalman filter can successfully deal with occlusion and estimate the correct location of the object.

Index Terms— Object tracking, Kalman Filter, Occlusion, Multiple objects

1. INTRODUCTION

Object tracking has been an intriguing research topic for the past few decades due to its numerous, wide-ranging applications. Object tracking can be defined as the process of locating moving objects in a sequence of images. It is a key element in areas such as security and automated surveillance, robotics and computer vision, medical imaging, human-computer interaction, traffic monitoring, vehicle navigation and so on.

In the process of tracking objects there are some potential challenges. Brightness and size variations, unexpected changes in motion parameters (velocity, acceleration), as well as partial or full object occlusions are among these challenges. Tracking multiple objects with various complex shapes is another issue that needs to be dealt with.

There are different approaches to object tracking. These approaches have been categorized in the literature using various terminologies. Yilmaz et. al [1] classify object tracking methods into three main categories: point tracking, kernel tracking, and silhouette tracking. In the concept of point tracking there are some deterministic methods such as MGE tracker and GOA tracker. There are also some statistical methods such as JPDAF and Kalman filters.

In this project we focused on the use of Kalman filters for the purpose of tracking objects. Kalman filter provides a recursive solution and in the case of valid assumptions on distributions and noise this solution will be optimal. It applies to stationary as well as nonstationary environments. Each updated estimate of the state is computed from the previous estimate and the new input data, so only the previous estimate requires storage. We will delve into Kalman filtering algorithm in section 2.

The rest of this report is organized as follows. In section 2 we describe the general approach towards implementing the Kalman filtering algorithm. Section 3 is dedicated to an in-depth elucidation of the algorithm utilized in this project for tracking objects. In section 4 we present some experimental results. Finally, we conclude the report in section 5.

2. KALMAN FILTER

Kalman filter is in essence a sequential minimum mean square error (MMSE) estimator which can be employed to estimate some unknown parameters in a linearly modeled dynamic system typically in the discrete-time domain. It utilizes a series of potentially inaccurate or uncertain measurements or observations to estimate the parameters of interest.

These parameters along with other essential defining parameters can be organized into a vector called *state vector*. The state vector $\mathbf{s}_k$ at each time interval k is usually defined as the least amount of data required to describe the past behavior of a dynamic system in a unique manner in order to predict its future behavior [2]. The corresponding measured set of data can also be arranged into an observation vector $\mathbf{d}_k$.

There are two main equations involved in this recursive algorithm, namely the state transition equation and the measurement equation. The state transition equation can be written as

$$\mathbf{s}_{k+1} = \mathbf{F}_{k+1}\mathbf{s}_k + \mathbf{n}_k \tag{1}$$

where $\mathbf{n}_k$ is the system or *process noise* which is generally assumed to be zero-mean white Gaussian noise, and $\mathbf{F}_{k+1}$ is the *state transition* matrix from state at time k to time $k+1$.

The process noise samples at each time interval are assumed to be independent from other time intervals and the covariance matrix at time k is given by $\mathbf{Q}_k$.

The measurement equation can also be expressed as

$$\mathbf{d}_k = \mathbf{H}_k\mathbf{s}_k + \mathbf{v}_k \tag{2}$$

where $\mathbf{v}_k$ is the *measurement noise* which is also generally assumed to be zero-mean white Gaussian noise, and $\mathbf{H}_k$ is the *measurement* matrix. Similar to those of the process noise, samples of the measurement noise at different time intervals are assumed to be independent from one another and the covariance matrix at time k is given by $\mathbf{R}_k$. Furthermore, it is a typical assumption that the measurement noise and the process noise are independent from each other.

Now if we define $\hat{\mathbf{s}}_k$ to be the *a posteriori* estimate and $\hat{\mathbf{s}}_k^{\,-}$ to be the *a priori* estimate of the state we can have the *innovation* vector as

$$\tilde{\mathbf{s}}_k = \mathbf{s}_k - \hat{\mathbf{s}}_k \tag{3}$$

Assuming the innovation being uncorrelated with previous measurements, the a posteriori estimate of the state can be derived as

$$\hat{\mathbf{s}}_k = \hat{\mathbf{s}}_k^{\,-} + \mathbf{G}_k\left(\mathbf{d}_k - \mathbf{H}_k\hat{\mathbf{s}}_k^{\,-}\right) \tag{4}$$

where $\mathbf{G}_k$ is called the *Kalman Gain* matrix [2].

Using the above assumption and defining the *a priori covariance* matrix as

$$\mathbf{P}_k^{\,-} = E[(\mathbf{s}_k - \hat{\mathbf{s}}_k^{\,-})(\mathbf{s}_k - \hat{\mathbf{s}}_k^{\,-})^T] \tag{5}$$

where $E[.]$ denotes the expectation operator and $(.)^T$ the transpose operator, Kalman Gain matrix can be expressed as

$$\mathbf{G}_k = \frac{\mathbf{P}_k^-\mathbf{H}_k^T}{\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k} \tag{6}$$

The *a posteriori covariance* matrix can also be expressed in terms of its a priori counterpart and vice versa as

$$\mathbf{P}_k = \left(\mathbf{I} - \mathbf{G}_k\mathbf{H}_k\right)\mathbf{P}_k^{\,-} \tag{7}$$

$$\mathbf{P}_k^{\,-} = \mathbf{F}_k\mathbf{P}_{k-1}\mathbf{F}_k^{\,T} + \mathbf{Q}_k \tag{8}$$

Noting that state estimate at each time interval is updated using

$$\hat{\mathbf{s}}_k^{\,-} = \mathbf{F}_k\hat{\mathbf{s}}_{k-1}^{\,-} \tag{9}$$

the recursive Kalman filtering can be implemented.

Figure 1 summarizes the sequential operations performed in this algorithm given the particular definition of state vector utilized in this project. Further discussion on this definition is presented in section 3.

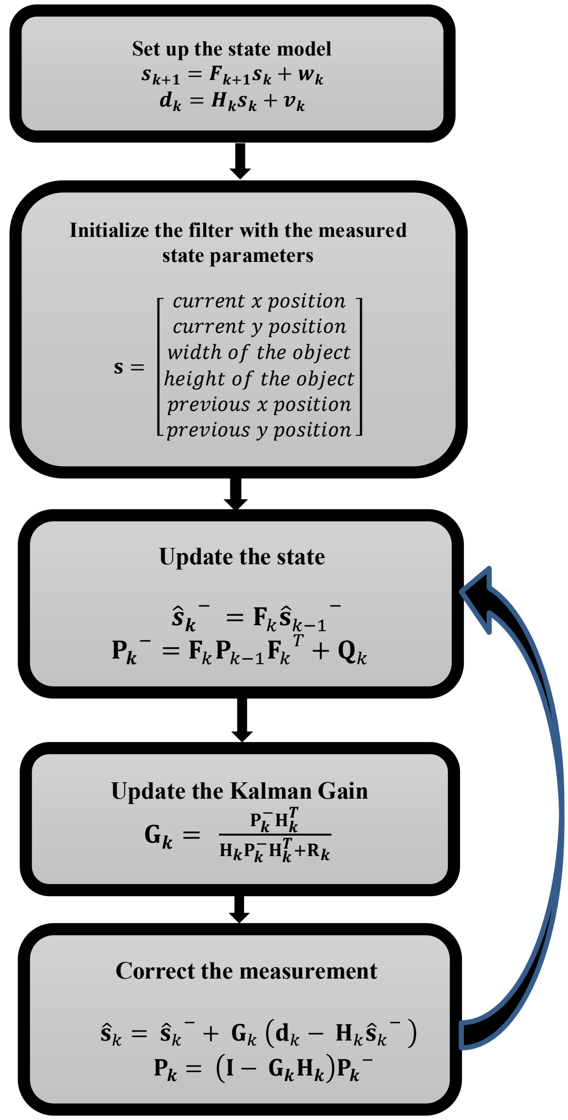

Fig.1. Summary of the Kalman filtering algorithm

3. OBJECT TRACKING ALGORITHM

In this section, we will explain the algorithm of how to perform video tracking by using image processing and Kalman filter.

First, one frame is selected as background frame, which means that there is no object of interest in this frame. Then for each frame, by performing background subtraction, we can get the possible areas that may have the target. Before processing the frames, one frame containing the target will also be selected to get the template of target object.

In second step, for each frame, morphology is applied after background subtraction. Morphology is a very important topic in image processing and there are morphological operations which help to denoise and fill holes. After image processing step, only foreground objects would appear. Based on geographical connection, different objects would be separated and marked with different numbers.

After morphology process, the next step is detection. Each foreground object would be compared with the template to check the similarity. If one of the objects is very similar to template, then this object would be considered as target, or we assume that the target is occluded in this frame.

The last step is target state estimation by Kalman filter. Because observation is sometimes influenced by noise or not available due to occlusion, Kalman filter helps us to solve these problems very well. The experiment results show that our method can handle many challenging situations like change of size, occlusion, multiple objects, shadow interference and so on.

3.1. Background Subtraction

Background subtraction is an efficient method to find the foreground part of each frame and it is widely used in many object tracking algorithms, like the one used by Jin et al [3]. If in the video, there exists a frame with just the background, then we directly select it as background. If not, there are multiple ways to obtain the background. For example, the median value of each pixel could be calculated after reading the whole frames and the median value would be the background. This is because in most cases, the target would always move and would not cover any background pixel in more than half of the total number of frames. A quicker and more accurate method is to manually select several frames and combine obtained pieces to form background.

Background subtraction can also be used for extracting target model. Before processing detection, we need to set up a template of the target. To this end, one frame with target is selected. It is acceptable if in that frame, there are other objects, because human interaction is allowed to draw a rectangle surrounding the target. After doing background subtraction, only pixels inside the rectangle that are greatly different from background color would be considered as template of the target.

In background subtraction, RGB color is used rather than intensity to get accurate separation. In experiments, it is possible that intensity of background and objects are similar even though the RGB colors are totally different, so our method has better separation.

3.2. Morphology

Morphological operations play an important role in many areas. For example, Oscar et al.[4] use morphology in denoising for video tracking. In experiments, opening and closing are applied to process binary foreground images. Opening is performed through erosion followed by dilation, while closing is done using dilation followed by erosion. Opening first helps us to erase noisy points by erosion and then let other foreground parts keep their original size by dilation. After getting rid of noise problem, closing is implemented to fill the holes in the foreground. First, dilation helps us to fill the holes but it also can lead to a slight increase in size of the objects, so erosion is used after dilation to maintain the size. The results can be found in Figure 2.

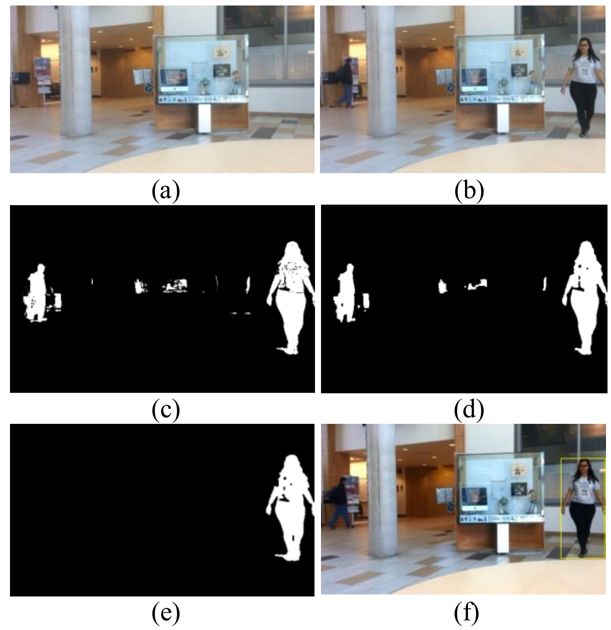

Fig.2. (a) Background (b) Frame with multiple objects (c) After background subtraction (d) After morphology (e) Detected object (f) Rectangle marking the object

After opening and closing, we are able to divide foreground into several different objects based on geographical connection. In the next step, different objects would be compared with target template separately.

3.3. Target detection

Target detection is based on morphology results that give clue about the number of foreground objects in the frame and their locations. Only the foreground objects would be the possible candidates of the target. The basic idea is that the target object should be similar to template. There are many ways to define similarity. A popular way is to compare intensity histogram, but Malik et al. [5] extract the dominant color for comparison. In our algorithm, RGB color is used for comparison and decision is made based on threshold. To determine the target, for each frame, each pixel of template is scanned and the similar RGB color is searched in each object. Objects with similar color have higher possibility of being the target. After scanning the whole template, the object with the highest number of matches with color of each pixel of the template would be the target. Also a threshold is added to reject detection in case of occlusion where even the object with the highest number of matches is not the actual target. If that happens, it means that the object is occluded and thus there is no detection in that frame.

In frames with successful detection, a rectangle is introduced to mark the size and location of target. This is used as measurement data in Kalman filter for further process.

3.4. Kalman filter implementation

When the measurements are available, the final step would be state estimation by using Kalman filter. In some cases, the measurement is not accurate and occlusion is unavoidable. For example, when target is partly occluded, then the size of measured rectangle would shrink instantly, but actually the size of object does not change too much. Moreover, when target is occluded, there would be no detection at all. To estimate the location, Kalman filter iteration is modified when occlusion occurs because standard Kalman filter has measurement at every time interval.

First we need to set up state model with the equations given in (1) and (2). To determine vector of state for describing rectangle, we need at least 4 values [2] – x coordinate of rectangle, y coordinate of rectangle, width of rectangle, and height of rectangle, denoted in (10) by x_k, y_k, w_k, and h_k respectively. To estimate the location of next frame, the velocity information should also be saved. As a result, two more values are put into state vector - x coordinate of rectangle in the previous frame, y coordinate of rectangle in the previous frame. With these two values, the velocity could be calculated. To determine the variance of measurement and variance of state, the variance of detected rectangle is calculated and used. Thus, the state transition equation and measurement equation would be

$$\mathbf{s}_{k+1} = \mathbf{F}_{k+1}\mathbf{s}_k + \mathbf{n}_k = \begin{bmatrix} 2 & 0 & 0 & 0 & -1 & 0 \\ 0 & 2 & 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ w_k \\ h_k \\ x_{k-1} \\ y_{k-1} \end{bmatrix} + \begin{bmatrix} n_k^x \\ n_k^y \\ n_k^w \\ n_k^h \\ 0 \\ 0 \end{bmatrix} \quad (10)$$

$$\mathbf{d}_k = \mathbf{H}_k\mathbf{s}_k + \mathbf{v}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \mathbf{s}_k + \begin{bmatrix} v_k^x \\ v_k^y \\ v_k^w \\ v_k^h \end{bmatrix} \quad (11)$$

If detection result exists, the estimation of state would be updated following Kalman filter iteration. However, it is also necessary to consider occlusion. When it happens, there is no measurement available to update estimation of state. Instead, we have to assume the target is following uniform linear motion and update the state using the equations in (9) and (8).

$$\hat{\mathbf{s}}_k^{-} = \mathbf{F}_k\hat{\mathbf{s}}_{k-1}^{-} \quad (9)$$

$$\mathbf{P}_k^{-} = \mathbf{F}_k\mathbf{P}_{k-1}\mathbf{F}_k^{T} + \mathbf{Q}_k \quad (8)$$

Since measurement correction is not available, instead we have $\hat{\mathbf{s}}_k = \hat{\mathbf{s}}_k^{-}$ and $\mathbf{P}_k = \mathbf{P}_k^{-}$. So the variance of estimation would increase as the occlusion continues. When target shows up again, the Kalman filter can quickly put more weights on current detection because of high variance of estimation, which makes sure that Kalman filter can immediately turn back to using measurement data and have better estimation results.

4. EXPERIMENTAL RESULTS

As seen in previous sections, there were many challenges in implementation of Kalman filter. Surprisingly it was tougher to detect the object and get the data than actual implementation of Kalman filter. We started with the simpler objects like balls and then moved on to adding some complexities such as size variations, occlusion, multiple objects, interference by shadow, and complex object structures like humans.

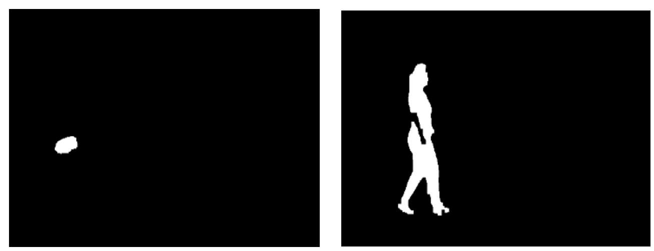

Fig.3. Detected target objects

Images in Figure 3 show the separated target objects for ball and human. These are the detection result after background subtraction and noise reduction using morphology.

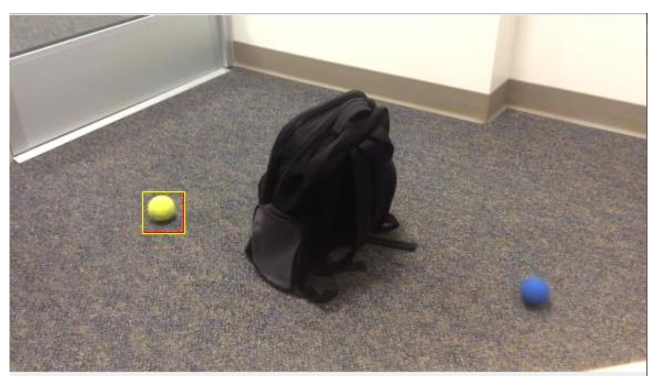

Fig.4. Tracking a simple object (ball in this case)

Figure 4 shows the result of Kalman filter for a simple object like ball dealing with occlusion and multiple objects. Red box indicates prediction by Kalman filter and yellow box is the detection of object. As we see, prediction and detection go hand in hand and show pretty good results. Also our

Kalman filter deals very well with multiple objects, it predicts and detects the yellow ball only and completely ignores blue ball. We have applied color based threshold detection technique, so our Kalman filter will have tough time identifying similar color objects.

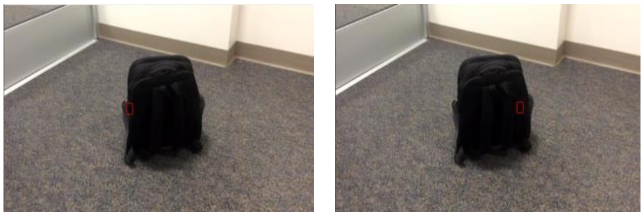

Fig.5. Kalman filter detects the location of the ball in the presence of full occlusion

Images in Figure 5 show how Kalman filter addresses occlusion. Even though ball is behind backpack, the red box shows prediction indicating general position of object. The red box moves along the occlusion based on the prediction following the previous motion of the ball.

Fig.6. Kalman filter detects the location of a complex object in the presence of partial occlusion

Figure 6 depicts an example of implementing Kalman filter on an object with complex structure like human body. Here also red box indicates Kalman filter and yellow is for detection. Threshold was created for blue object for color based detection. The image on the right shows results for occlusion. Our Kalman filter works very well for human as well.

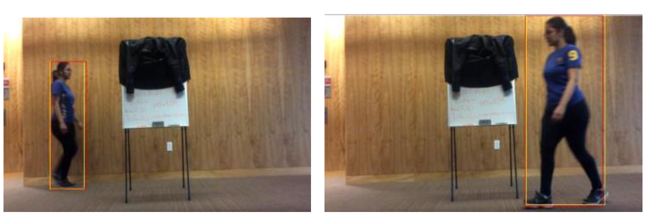

Fig.7. Variations in size

The pictures in Figure 7 demonstrate Kalman filter implementation for change in object size. The red and yellow boxes adjust their sizes according to the size of object. In human detection, shadow was a major issue as shadow moves along with object. This was solved by adjusting the thresholds to get more sophisticated results.

5. CONCLUSION

This project provides very good practical application for estimation theory concepts. It is based on Bayesian and MMSE estimator. Along with the basic implementation of Kalman filter, we have addressed a lot of challenges in this project described in section 4. Our detection uses many image processing techniques like color threshold, background subtraction and morphology. Our Kalman filter prediction agrees well with the detection. Hence the error is very small. Future work can address more complex challenges like variation in lighting conditions, changes in background, multiple objects of same color etc. Kalman filter has several critical applications like security surveillance, military applications, medical applications which need visual assistance, study of animal behaviors, robotics and computer vision etc. This project provided prefect amalgamation of estimation theory and its applications.

6. REFERENCES

[1] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” ACM Comput. Surv., vol. 38, no. 4, pp. 1–45, 2006.

[2] E. Cuevas, D. Zaldivar, and R. Rojas, “Kalman filter for vision tracking,” Freie Universität Berlin, Inst. Informatik, Berlin, Germany, Tech. Rep. B 05-12, 2005.

[3] Jin Wang, Fei He, Xuejie Zhang, Yun Gao, “Tracking Objects through Occlusions Using Improved Kalman Filter”

[4] Oscar Efrain Ramos Ponce, Mohammad Ali Mirzaei, Frederic Merienne, “Tracking in Presence of Total Occlusion and Size Variation using Mean Shift and Kalman Filter”

[5] Malik M. Khan, Tayyab W. Awan, Intaek Kim, and Youngsung Soh, “Tracking Occluded Objects Using Kalman Filter and Color Information” International Journal of Computer Theory and Engineering, Vol. 6, No. 5, October 2014