Kalman Filter Based Tracking in an Video Surveillance System

Caius SULIMAN[1], Cristina CRUCERU[2], Florin MOLDOVEANU[3]
Transylvania University of Braşov
No. 29, Eroilor Blvd., RO-69121 Braşov
[1]*caius.suliman@unitbv.ro,* [2]*cri.cruceru@yahoo.com,* [3]*moldof@unitbv.ro*

Abstract — **In this paper we have developed a Matlab/Simulink based model for monitoring a contact in a video surveillance sequence. For the segmentation process and corect identification of a contact in a surveillance video, we have used the Horn-Schunk optical flow algorithm. The position and the behavior of the correctly detected contact were monitored with the help of the traditional Kalman filter. After that we have compared the results obtained from the optical flow method with the ones obtained from the Kalman filter, and we show the correct functionality of the Kalman filter based tracking. The tests were performed using video data taken with the help of a fix camera. The tested algorithm has shown promising results.**

Index Terms — **Video Surveillance System, Optical Flow, Kalman Filtering, Image Processing, Tracking**

I. INTRODUCTION

The problem of using vision to track and understand the behavior of humans is a very important one. The main applications that it has are in the areas concerning human-robot interaction [7], robot learning, and video surveillance.

Here we try to focus our attention on video surveillance systems. A high level of security in public places is an extremely complex challenge. A number of technologies can be applied to various aspects of security, including biometric systems, screening systems, and video surveillance systems. Nowadays video surveillance systems act as large-scale video recorders, analog or digital. These systems serve two main purposes: to provide a human operator with images to detect and react to potential threats and recording for future investigative purposes.

From the perspective of real-time detection, it is well known that the human's visual attention drops below acceptable levels even if that operator is a trained one in the task of visual monitoring. Video analysis technologies can be applied to develop smart surveillance systems that can aid the operator in the detection and in the investigatory tasks.

For surveillance applications, the tracking problem is a fundamental component. In video surveillance one of the most used method for tracking contacts is the particle filter [8][10][11][13]. Another well known method in the research community is the use of the traditional Kalman filter [9]. In many cases the use of this type of filter is sufficient. This is due to the controlled indoor and outdoor environments that are used in the studies.

Many papers in the literature detail methods that track single persons only [6][10], but there are also many authors that describe different methods for the detection and tracking of multiple persons [2][3][5][11]. Most of these methods involve as testing grounds indoor environments [1][3][8][13] as well as outdoor environments [2][5][8][9], where these methods are applied to track groups.

The objective of this paper is the development of a video surveillance system capable of tracking a person in an outdoor environment. In Section II we describe the structure of the proposed video surveillance system. In Section III we present the method used for contact detection and the method used for the extraction of useful data from the video feed. Section IV describes the Kalman filter algorithm applied in our case. In Section V and VI we present the results obtained from the Simulink model's simulation, the conclusions drawn from this study and the possible future developments.

II. THE SYSTEM'S STRUCTURE

In this paper we examine the feasibility of using the optical flow algorithm in conjunction with the Kalman filter algorithm [9][12] for tracking a contact in a surveillance scene. In order to create an algorithm that is able to track a contact in a scene, three different, large-scale task must be accomplished (see Fig.1). First the algorithm needs to take an incoming surveillance video signal and segment it into a stream of frames where contacts are distinguished from the background of the scene. The next step is the tracking of the contact throughout the video sequence. Finally, the resulting track must be processed in order to analyze the contact's behavior.

For the segmentation process of the incoming video signal, the optical flow algorithm developed by Horn and Schunk was used [4]. The optical flow algorithm approximates the movement of the contact in the current frame as referenced to the previous frame. By determining the motion of objects, one can distinguish between the contact and the background of the scene. After careful tuning and processing, the output of the segmentation process is passed to the Kalman filter algorithm for further processing.

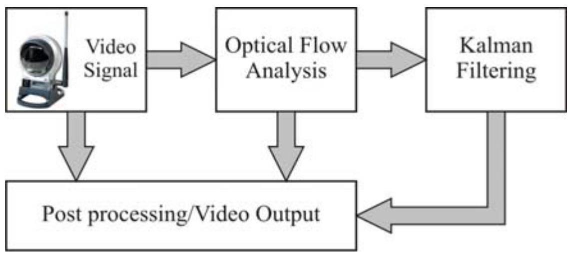

Figure 1. The structure of the surveillance system.

The Kalman filter is a recursive, adaptive filter that operates in the state space. It is well known for its ability to track objects in a timely and accurate manner. The tracking algorithm developed in this paper is able to process one contact at a time.

III. THE OPTICAL FLOW ANALYSIS

One of the important blocks presented in the above scheme is the so called *optical flow analysis* block. The main purpose of this block is to determine the existence of possible contacts in the incoming video signal and process them in such manner that the Kalman filter will be able to track them with minimal error.

In Fig.2 we will present the main component of the Optical Flow Analysis block.

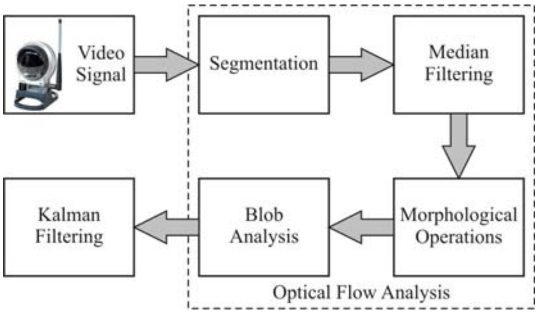

Figure 2. The optical flow analysis block.

In what follows we describe the functionality for each component block.

A. Segmentation

In our case, the term segmentation is used to describe the process through which a video signal passes to become a series of binary images. At the output of this sub-block, each of the resulting binary images will contain black and white areas. The black areas correspond to the portion of the frame where no motion was detected, and the white areas correspond to the portion of the frame where motion was detected.

The surveillance system was developed in Matlab's Simulink. At first the incoming video signal is coded in the RGB color space. Because we use the optical flow to detect motion, the video signal needs to be converted to the intensity color space (see Fig.3). To estimate the optical flow between two images we use the algorithm developed by Horn and Schunk. In our case this algorithm is used to compute the optical flow between the current frame and the previous one. This is one of the tunable parameters used in our experiments. Another important tunable parameter used in the optical flow estimation is the smoothness factor. This parameter is defined as a constraint which controls how smoothly the velocity field of the brightness pattern in images varies throughout the image. The Horn-Schunk algorithm quantifies the smoothness of the velocity filed using the magnitude of the gradient of the optical flow velocity defined as in:

$$\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2 \text{ and } \left(\frac{\partial v}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2 \qquad (1)$$

where u and v are the velocity vectors corresponding to the optical flow. A small value for this gradient indicates that the vector field is very smooth; a higher one indicates the contrary. A smooth vector field tends to zero-out regions where no motion is detected leaving only limited areas of non-zero vector fields. In the Simulink model of our surveillance system, this smoothness factor is inversely proportionally to the magnitude of the velocity gradients. Our experiments pointed out that the optimal value for the smoothness factor is 0.6.

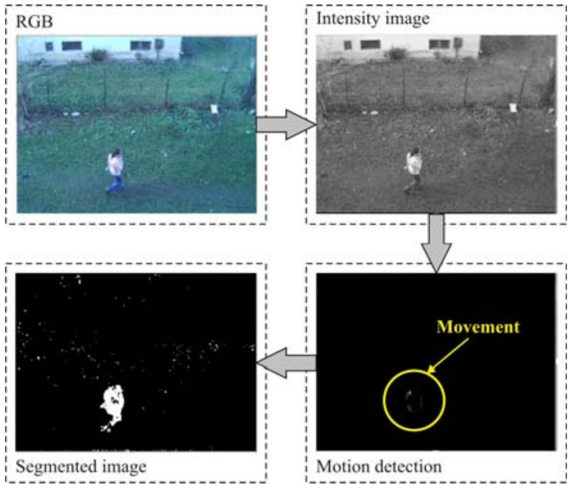

Figure 3. The segmentation process.

Before the processed video signal exits the segmentation sub-block it is compared with a certain threshold to keep only what interests us from the video feed.

B. Median Filtering

One of the biggest problems that optical flow has it's that it is very sensitive to changes in illumination or to the quality of the video feed. This sensitivity conduces in erroneous blobs appearing in individual frames. If these blobs are large, that means that they are approaching the average size of a real person, they can create problems for successful morphological operations.

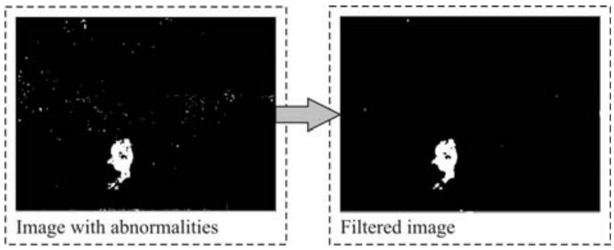

Figure 4. Result after the median filtering.

One of the main reason for choosing the median filter is that most of these abnormalities, the erroneous blobs, appear in singular frames and they do not appear again for several more frames. The median filter is used to decrease the effect of these abnormalities while still maintaining the information of the correctly detected contacts.

In Fig.3 it can be seen that the segmented image contains many abnormalities. After the median filtering many of these abnormalities are gone (see Fig.4).

C. Morphological Operations

The morphological operation will process the video signal coming from the output of the median filtering sub-block in such way that all erroneous blobs residing in the image are eliminated and all and only the correctly detected blob is maintained and classified as a real contact. The main morphological operations used by us in this study are the *erosion* and *dilatation*. Optimal erosion is achieved when the structuring element keeps at least the remnant of a blob for all correct contacts. If we use a sub-optimal structuring element for the erosion and dilatation operations, a valid contact could be lost completely, or an erroneous blob could be tracked. Both these errors can produce significant barriers to optimal dilatation and to Kalman filter tracking. Optimal dilation is obtained when the structuring element merges all remnants of a single blob into one contact. If a sub-optimal structuring element is used for dilation, one contact could be viewed as multiple contacts or multiple contacts could be viewed as one contact. After an optimal structuring element for erosion was determined, each frame was eroded using the chosen structuring element. Determination of the optimal structuring element for dilation was similar to that of erosion. Each frame was dilated with a square structuring element.

An infinite number of possibilities exist for size and shape of structuring elements. Depending on the data used, the size and shape of the optimal structuring element could vary significantly.

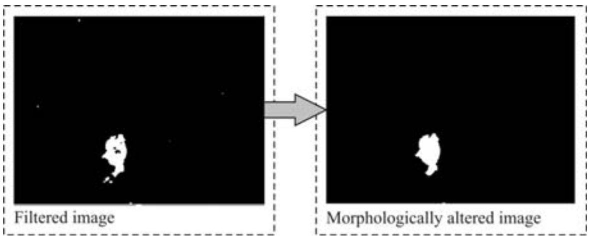

Figure 5. The result of morphological operations applied on the median filtered image.

Comparing the morphological altered image to the image resulting after the median filtering (see Fig.5) shows that the erosion operation removed the remaining erroneous blobs residing in the image, thus deciding that they were not contacts. Further dilation has created a solid blob out of the area where motion was detected, and this blob will be tracked as a contact.

D. Blob Analysis

The main functionality of the blob analysis sub-block is to determine the minimum size of a blob and the maximum number of blobs that will be used in the Kalman tracking process and in the visualization step. By setting the minmum blob size we obtain a new level of protection against abnormalities by specifying a minimum size that a blob must have in order to be correctly tracked. Thus, any blob that doesn't fulfill this condition will not be tracked.

The other tunable parameter of the blob analysis sub-block, the maximum number of blobs, is used to set the number of Kalman filters to be used in the tracking process. In our case this parameter was set to 1.

IV. KALMAN FILTERING

Filtering is a very used method in engineering and embedded systems. A good filtering algorithm can reduce the noise from signals while retaining the useful information. The Kalman filter is a mathematical tool that can estimate the variables of a wide range of processes. It estimates the states of linear systems. This type of filter works very well in practice and that is why it is often implemented in embedded control system and because we need an accurate estimate of the process variables. The discrete Kalman filter is characterized by both a process model and a measurement equation.

The process model is characterized by the assumption that the present state, x_k, can be related to the past state, x_{k-1}, as follows:

$$x_k = \Phi_k x_{k-1} + w_k \qquad (2)$$

where w_k is assumed to be a discrete, white, zero-mean process noise with known covariance matrix, Q_k; Φ_k represents the state transition matrix which determines the relationship between the present state and the previous one.

In our case we try to track the state of a contact based on its last known state. Here, the state vector consists of a two-dimensional position expressed in Cartesian coordinates, a two-dimensional velocity and a two-dimensional accelertion. By considering a constant acceleration, the state transtion matrix can be determined from the basic kinematic equations as follows:

$$s_k = s_{k-1} + v_{k-1}t + \frac{1}{2}a_{k-1}t^2 \qquad (3)$$

$$v_k = v_{k-1} + a_{k-1}t \qquad (4)$$

$$a_k = a_{k-1} \qquad (5)$$

where s is defined to be the contact's position, v is its velocity, a is the contact's acceleration and t is the sampling period. In a matrix form, the above equations can be written as:

$$\begin{bmatrix} s_{x,k} \\ s_{y,k} \\ v_{x,k} \\ v_{y,k} \\ a_{x,k} \\ a_{y,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0.5 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_{x,k-1} \\ s_{y,k-1} \\ v_{x,k-1} \\ v_{y,k-1} \\ a_{x,k-1} \\ a_{y,k-1} \end{bmatrix} \qquad (6)$$

Here, the subscripts x and y refer to the direction of the contacts position, velocity and acceleration in the two-dimensional plane. The value of the sampling period is set to 1.

The measurement equation is defined as:

$$z_k = H_k x_k + v_k \qquad (8)$$

where z_k represents the measurement vector, v_k is assumed to be a discrete, white, zero-mean process noise with known covariance matrix, R_k. The matrix H_k describes the relationship between the measurement vector, z_k, and the state vector, x_k. Given the fact that the state vector is of length six and the measurement vector is of length two, the matrix H_k must be of length six by two:

$$H_k = \begin{bmatrix} 1 & 0 & 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0.5 \end{bmatrix} \qquad (9)$$

From the process model and measurement equation it results that the Kalman filter attempts to improve the prior state estimate using the incoming measurement which has been corrupted by noise. This improvement can be achieved by linearly blending the prior state estimate, $\hat{x}_{k-1}$, with the noisy measurement, z_k, in:

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H_k \hat{x}_k^-) \quad (10)$$

Here $\hat{x}_k^-$ means the a-priori estimate; K_k is known as the blending factor. The minimum mean squared error of the estimate is obtained when the blending factor assumes the value of the Kalman gain:

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (11)$$

where P_k is known as the state covariance matrix. Generally the state covariance matrix is a diagonal matrix. The state covariance matrix is determined from the a-priori state covariance matrix as follows:

$$P_k = (I - K_k H_k) P_k^- \quad (12)$$

After the Kalman gain has been computed, and the state and state error covariance matrices have been updated, the Kalman filter makes projections for the next value of k. These projections will be used as the a-priori estimates during processing of the next frame of data.

$$\hat{x}_{k+1}^- = \Phi_k \hat{x}_k \quad (13)$$

$$P_{k+1}^- = \Phi_k P_k \Phi_k^T + Q_k \quad (14)$$

The above equations are the projection equations for the state estimate and for the state covariance matrix.

The main role of the Kalman filtering block is to assign a tracking filter to each of the measurements entering the system from the optical flow analysis block. For an easy implementation of the Kalman filter in Simulink, we wrote an embedded Matlab function. This method is often used when the function that needs to be implemented is more easily to express in Matlab's symbolic language than in Simulink's graphical language.

V. POST PROCESSING

The last block that we will discuss is the post processing/video output block. This block was used to process the output from the optical flow analysis block and the output of the Kalman filtering block.

The post processing block is composed of four video output sub-blocks. The first sub-block is used only to view the original video signal.

The second sub-block shows the optical flow lines superimposed on the original video signal (see Fig.6a). It is used only for scientific purposes, and that is to visualize the direction of the optical flow lines and from there to deduce in which direction the contact moves.

The third sub-block is used to visualize the resulting signal from the optical flow analysis block and to allow the user to be sure of the correct functionality of the optical flow analysis block. This sub-block is in direct connection with the blob analysis sub-block, block that produces the coordinates for a bounding box. This bounding box is a rectangle drawn around each correctly detected blob. The user is able to watch in real-time which contact in the video feed is being sent to the Kalman filtering block. If rectangles are not surrounding the correctly detected contacts in an image, this thing means that the optical flow analysis bloc is not working properly. Fig.6b presents the resulting output of the optical flow video viewer sub-block. We present only five frames taken at 17.5 FPS of each other. It can be clearly seen that a contact was detected and a bounding box was correctly superimposed on the contact.

The last sub-block discussed in this section is the Kalman filtering video viewer. This sub-block is in direct connection with the Kalman filtering block. This block produces at its output a matrix containing the position of the detected contact. The output is used by the Kalman filtering video viewer to draw markers in the video. These markers are represented here by red circles. The center of the circle represents the coordinates of the estimated position of the detected contact. The user is able to see in real-time if the detected contact is correctly tracked by the Kalman filter. If a marker doesn't follow consistently a contact we can say that the Kalman filtering block isn't working properly. Fig.7a presents five frames resulting from the Kalman filtering video viewer, taken like in the previous case, at 17.5 FPS of each other. We can clearly see that the marker is correctly tracking the detected contact thus confirming that the Kalman filter is working properly. The Kalman filter is even capable to track the contact that is leaving at some time the visual field of the camera and then correctly reassign the marker to the contact that reenters in the visual field (see Fig.7b).

(a)

(b)

Figure 6. Output sample from the optical flow video viewer. (a) the optical flow lines; (b) bounding box superimposed on the correctly detected contact.

(a)

(b)

Figure 7. Output sample from the Kalman filtering video viewer. (a) tracking a contact that passes through the camera's field of view; (b) contact exiting and reentering the camera's field of view.

VI. CONCLUSION

There are two main factors that affect the problem of tracking: the accuracy to distinguish between contacts passing through the scene and the speed to process the video feed in real-time. In this paper we have shown that with the help of the optical flow and Kalman filter algorithms it is possible to detect and track a person passing through a scene.

The video signal used in our experiments it is provided by a Linksys WVC200 PTZ IP video camera at a resolution of 240x320. The entire experiment was conducted using an Intel Core2Duo T9300 computer with 4 GB of RAM.

From the optical flow analysis used in our research we have deduced that there is an inevitable trade-off between the accuracy and speed of processing. To accurately distinguish a contact that passes through a scene, the computational time of the optical flow algorithm must be increased. If this increase of the processing time is too large, the algorithm will not operate in real-time.

The Kalman filter algorithm presented in this research was able to correctly process a contact and to correctly assign a filter to the processed contact. After reviewing the results we deduced that the algorithm performed quite well showing a moderate consistency in tracking. Due with the success with the data used in our experiments, any inconsistencies in the tracking process can be traced back to the fluctuations in performance of the optical flow algorithm.

Future research in the area of surveillance systems should be focused in two directions. First, research should be made to determine an objective measure of performance of the optical flow algorithm and to see if other existing algorithms are better suited to accurate, real-time processing of a video signal. The second research should be focused in the area of determining contact behavior, and in areas such as merging contacts into groups or dividing groups into separate contacts. For a future research, we will try to implement the presented Kalman filter algorithm into a system that is capable of tracking multiple persons.

ACKNOWLEDGMENTS

This paper is supported by the Sectoral Operational Programme Human Resources Development (SOP HRD), financed from the European Social Fund and by the Romanian Government under the contract number POSDRU/6/1.5/S/6.

REFERENCES

[1] M. A. Ali, S. Indupalli, B. Boufama, "Tracking Multiple People for Video Surveillance," First Intern. Workshop on Video Processing for Security, June 2006.

[2] B. Benfold, I. Reid, "Guiding Visual Surveillance by Tracking Human Attention," Proc. of the 20th British Machine Vision Conf., September 2009.

[3] L.M. Fuentes, S.A. Velastin, "From tracking to advanced surveillance," Proc. of the Intern. Conf. on Image Processing, vol. 3, pp. 121-124, September 2003.

[4] K.P. Horn, B.G. Schunck, "Determining optical flow," Artificial intelligence, vol. 17, pp. 185-203, 1981.

[5] C. C. Hsieh, S. S. Hsu, "A Simple and Fast Surveillance System for Human Tracking and Behavior Analysis," Proc. of the 3rd Intern. IEEE Conf. on Signal-Image Technologies and Internet-Based System, pp. 812-818, December 2007.

[6] F. Jean, R. Bergevin, A.B. Albu, "Body tracking in human walk from monocular video sequences," Proc. of the 2nd Canadian Conf. on Computer and Robot Vision, pp. 144-151, May 2005.

[7] N. Koenig, "Toward real-time human detection and tracking in diverse environments," Proc. of the 6th IEEE Intern. Conf. on Development and Learning, pp. 94-98, July 2007.

[8] S. Kong, M.K. Bhuyan, C. Sanderson, B.C. Lovell, "Tracking of Persons for Video Surveillance of Unattended Environments," Proc. of the 19th Intern. Conf. on Pattern Recognition, pp. 1-4, December 2008.

[9] W. Niu, L. Jiao, D. Han, Y. F. Wang, "Real-time multiperson tracking in video surveillance," Proc. of the 4th Pacific Rim Conf. on Multimedia, vol. 2, pp. 1144-1148, December 2003.

[10] A.W. Senior, G. Potamianos, S. Chu, Z. Zhang, and A. Hampapur, "A comparison of multicamera person-tracking algorithms," Proc. IEEE Int. Works. Visual Surveillance, May 2006.

[11] J. Wang, Y.g Yin, H. Man, "Multiple Human Tracking Using Particle Filter with Gaussian Process Dynamical Model," EURASIP Journal on Image and Video Processing, vol. 2008, Article ID 969456, 10 pages, 2008.

[12] G. Welch, G. Bishop, "An Introduction to the Kalman Filter,", Technical Report: TR95-041, University of North Carolina, 2006.

[13] J. Yao, J.M. Odobez, "Multi-Camera 3d Person Tracking With Particle Filter In A Surveillance Environment, "Proc. of the 16th European Signal Processing Conf., August 2008.