International Journal of Membrane Science and Technology, 2023, Vol. 10, No. 1, pp 812-820

Developing Motion Detection Approach through Image Processing using Kalman Filtering and Gaussian Mixture Model

Patricia Arianne L. Barnuevo[1], Paul Andrew S. Orani[2], Flordeliza L. Valiente[3]

[1] *School of EECE, Mapua University, Barangay Santa Cruz, Makati City, Philippines.* *palbarnuevo@mymail.mapua.edu.ph*

[2] *School of EECE, Mapua University, Barangay Santa Cruz, Makati City, Philippines.* *pasorani@mymail.mapua.edu.ph*

[3] *School of EECE, Mapua University, Barangay Santa Cruz, Makati City, Philippines.* *FLValiente@mapua.edu.ph*

Abstract: Motion tracking through image processing requires validating the subject's presence which can use for automated systems. However, detecting a moving subject in a real-time video is a complex approach in terms of visual scenes. The Gaussian Mixture Model (GMM) is a flexible tool for image processing modeling. The concept of Kalman Filtering can be applied for noise filtering of a real-time video and tracking the motion in the image. The algorithms used in this study are to develop a motion-tracking system that uses a bounding box to assign tracks to the detected motion. The algorithm developed in MATLAB software includes the parameters for the Kalman Filtering and GMM. The testing involved an IP camera, Arduino Uno as the indicator if motion is detected, and different values of the parameters for the training frame. The confusion matrix is used for the statistical analysis of the five trials to detect the True Positive and measure the system's accuracy. The study showed promising results as it obtained a true positive rate of 0.6000 from trial 4, having the applied parameters of 3 for the gaussian number, 110 for the training frame, and 0.4 for the minimum background ratio.

Keywords: image processing, motion tracking, kalman filtering, gaussian mixture model, arduino uno.

1. INTRODUCTION

Motion tracking is useful in different aspects of automation systems that requires person or object's validation, it can be applied in areas such as rooms, traffic, and attendance monitoring. It is crucial for video compression, human detection, and behavior analysis in video surveillance systems. For real-time video surveillance systems, there is a need in motion detection that can provide accurate detection even in non-static backgrounds regardless of surroundings (outdoor or indoor), object speed and size, camera noisy pixels, or sudden change in light intensity. Various approaches have been used to detect motion in a continuous video stream. In relation to that, one of the notable sensors used by many when it comes to motion detection is the Passive Infrared sensors (PIR Sensor). PIR sensors only predict the motion and has several risks when in contact with sunlight. In detecting moving objects, it is usually applied through background subtraction, specifically the use specifically the Gaussian Mixture Model and the Kalman Filtering algorithm.

The concept of Kalman Filtering in image processing is to reduce the noise and use to track the motion in a real-time video [1][2] , while the Gaussian Mixture Model (GMM) is applied for data clustering and that is considered as a powerful tool for modeling in image processing. The GMM considers data points that will be derived from a mixture of the finite Gaussian distributions [3].

Interaction such as gestures were implemented for automation systems and it can be connected to certain sensors and may control different equipment, the previous study is considered accurate for hand gestures and recommends using a camera that has high quality for better detection [4]. Another previous study implemented an object detection and used a camera module to control the lights, it is concluded to have a supporting algorithm to enhance the quality of the system [5]. When it comes to real-time system predictive models and sensors, they may not work as expected, resulting in uncertainty; however, by using the Kalman filter, the uncertainty can be reduced with the help of a gaussian distribution [6].

This study aims to develop a motion tracking system through image processing using the Kalman Filtering and Gaussian Mixture Model specifically, (1) to develop a program including the two algorithms using MATLAB software, (2) to apply the program to a real-time video and measure the accuracy, and (3) to use an Arduino Uno as the prompt if there is motion detected from the program developed.

This paper will be significant to future researchers that will use image processing with their systems specifically applying the GMM and Kalman Filtering. The results of this paper may be used as a reference for the algorithm parameters and may further improve through innovations.

This paper focuses on developing an algorithm that can detect motion using GMM and Kalman filtering in a real-time video. The MATLAB software will be the main controller for this process and will be connected to an Arduino Uno to initiate a message if there is a detected motion. The result of the video must correspond to the motion detected through bounding boxes.

2. METHODOLOGY

A. Conceptual Framework

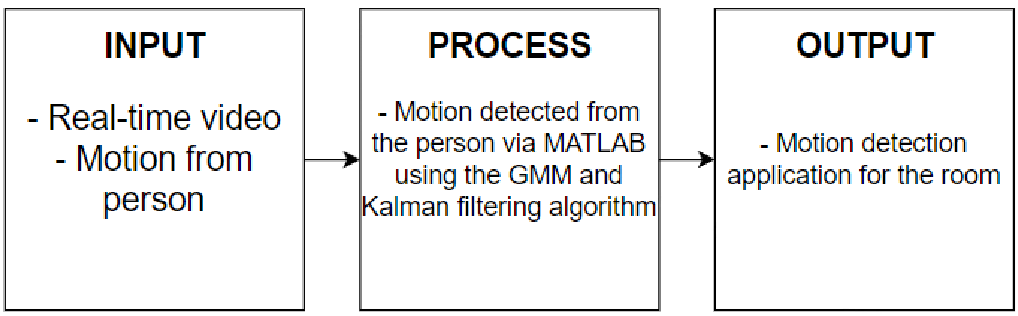

Fig. 1. Conceptual framework of the motion detection

The conceptual framework includes the input, process, output of the system. The input includes the real-time video that and motion from the person to be fed to the MATLAB software implementing the Gaussian Mixture Model (GMM) and Kalman Filtering algorithm and the output it can detect a motion by using the applied image algorithm.

B. Data Gathering

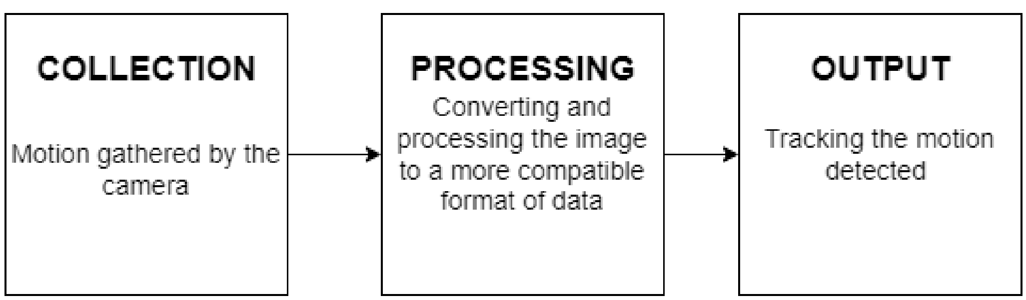

Fig. 2. Image collection process

The objective of the image collection process is to observe the accuracy of the algorithm's motion detection approach. In the collection phase, the IP camera will be set into a specific area where motion is present. Once it is collected, the MATLAB program will convert it to a more compatible format of data, and the output will be the motion tracks in a real-time video.

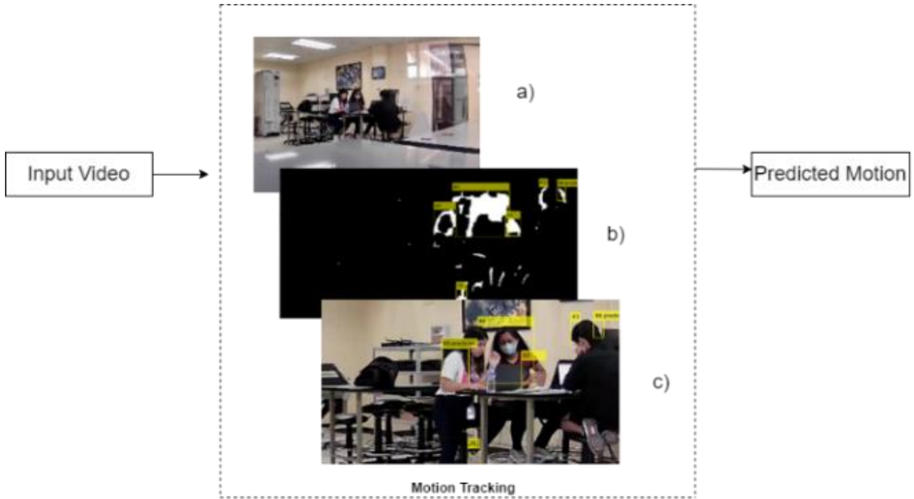

Fig. 3. Prediction model of the algorithm

The prediction model for the image processing system includes, (1) the input video wherein it is from the IP camera, (2) motion tracking that specify the algorithm and filtering: for a) it represents the real-time frame, b) motion detection by subtracting the background using GMM and Blob Analysis, c) Kalman filtering which bounding boxes is applied, and (3) the predicted motion in the frame through bounding boxes.

C. Program

The objective for the program is to detect the motion by assigning tracks to the predicted motion and the present motion. The researchers used a blob analysis which is included in the program wherein it can track and compute the statistics for connected regions in a binary image. The blob analysis helps the system to cross-check the motion available if it is consistent or not. The Kalman filtering is applied to predict the centroid of each motion track detected in the current frame.

```
motiontracker.m

function motiontracker()

ARDUINO_AVAIL = true;
IP_CAM_AVAILABLE = true;
CAM_IP = "rtsp://192.168.1.21/stream1";
CAM_USERNAME = "camera12345";
CAM_PASSWORD = "cam12345";
MCU_PORT = "COM3";
MCU_MODEL = "Uno";
SHOW_BLOB_VIEW = true;
```

Fig. 4. Initial code of the program

The program requires the following information of the given parameters in fig.4 such as: (1) IP camera's status, it can be true or false, (2) IP camera's URL, if the camera is set to true it must include the IP address with the format of rtsp://ip address/stream 1, otherwise it must be set to 'webcam', (3) IP camera's username and password, this can be seen in the camera's built in application depending on the features, and (4) Arduino Uno's availability.

```
function obj = setupSystemObjects()
    DETECTOR_NUM_GAUSSIAN =  ; % 5 default
    DETECTOR_NUM_TRAINING_FRAME =  ; % 150 default
    DETECTOR_MINIMUM_BG_RATIO =  ; % 0.7 default

    if SHOW_BLOB_VIEW
        obj.maskPlayer = vision.VideoPlayer('Position', [740, 400, 700, 400]);
    end
```

Fig. 5. Creating system objects

The concept of GMM is applied in creating the system objects for the real-time video frame. The *NumGaussians* has a default rate of 5, this is for applying multiple background modes. The NumTrainingFrames has a default rate of 150, wherein it indicates the initial video frames for training the background model. The MinimumBackgroundRatio has a default rate of 0.7 wherein it can set the minimum possibility for pixels to be considered as background values.

```
function predictNewLocationsOfTracks()

    for i = 1:length(tracks)
        bbox = tracks(i).bbox;

        predictedCentroid = predict(tracks(i).kalmanFilter);

        predictedCentroid = int32(predictedCentroid) - bbox(3:4) / 2;
        tracks(i).bbox = [predictedCentroid, bbox(3:4)];
    end
```

Fig. 6. Predicting new motion

```
function [assignments, unassignedTracks, unassignedDetections] = ...
        detectionToTrackAssignment()

    nTracks = length(tracks);
    nDetections = size(centroids, 1);

    cost = zeros(nTracks, nDetections);
    for i = 1:nTracks
        cost(i, :) = distance(tracks(i).kalmanFilter, centroids);
    end

    costOfNonAssignment = 20;
    [assignments, unassignedTracks, unassignedDetections] = ...
        assignDetectionsToTracks(cost, costOfNonAssignment);
end
```

Fig. 7. Assigning tracks for motion

The concept of Kalman filter is applied to predict the motion tracks. The algorithm consists of two steps, which is to predict and update. In fig. 6, the prediction of new motion is set through a bounding box and length, it also includes the predicted centroid so that the shift of the bounding box is at the center of the predicted location. In fig.7 it includes the computing the cost of assigning each detection to each track which is the first step of the filtering, and to solve the assignment, the researchers applied the cost of non-assignment to 20 which pertains to unassigned tracks.

```
lcd.setCursor(0,0);
lcd.print("AUTO SYSTEM   ");
  bool twentyFour;
  bool amPm;
  byte hour=realTimeClock.getHour(twentyFour, amPm);
  if(hour<10){
    lcd.print("0");
    lcd.print(hour);
  } else lcd.print(hour);

  lcd.print(":");

  byte minute=realTimeClock.getMinute();
  if(minute<10){
    lcd.print("0");
    lcd.print(minute);
  } else lcd.print(minute);


if( hasPeopleMotion){
  lcd.setCursor(0,3);
  lcd.print(" motion detected");
 }
```

Fig. 8. Arduino Uno program

The Arduino Uno in this study includes a connection to a real-time clock module and connected to the MATLAB program. The researchers applied a message if it has motion detected.

D. Statistical Analysis

This paper will use the confusion matrix as the measurement of the accuracy results for each trial. The researchers will derive the values of certain measurement from the confusion matrix table of each trial.

1. True Positive Rate

$$TPR = \frac{TP}{TP + FN} \quad (1)$$

2. True Negative Rate

$$TNR = \frac{TN}{TN + FN} \quad (2)$$

3. False Positive Rate

$$FPR = \frac{FP}{FP + TN} \quad (3)$$

4. False Negative Rate

$$FNR = \frac{FN}{FN + TP} \quad (4)$$

3. RESULTS AND DISCUSSION

A. Algorithm accuracy

TABLE I. Applying GMM Parameters

Trial	Gaussian Number	Training Frame	Minimum BG Ratio
1	3	120	0.5
2	5	150	0.7
3	4	120	0.5
4	3	110	0.4
5	3	110	0.3

The researchers conducted five trials for the given parameters in Table I. The parameters are for observing the accuracy of the Gaussian Mixture Model concept. The Kalman filter settings shown in methodology is also applied. The testing aim to observe the accuracy, sensitivity, and update of the bounding box accordingly.

Fig. 9. Trial 1 of image accuracy testing

Fig. 10. Trial 2 of image accuracy testing

Fig. 11. Trial 3 of image accuracy testing

Fig. 12. Trial 4 of image accuracy testing

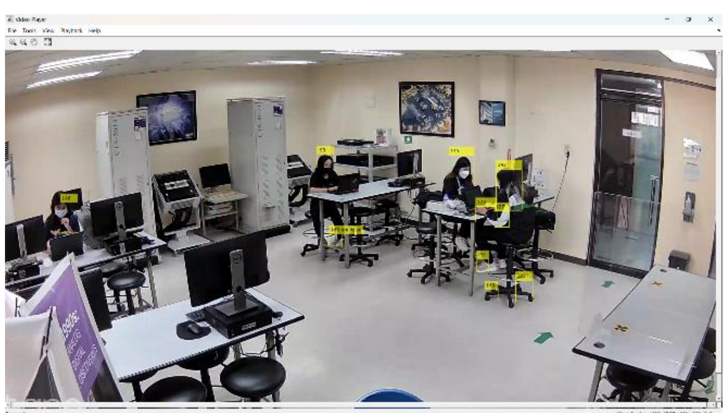

Fig. 13. Trial 5 of image accuracy testing

B. Confusion matrix results

Predicted value	Actual Value: Person Considered inside the bounding box	Actual Value: Person not considered inside the bounding box
Person Considered inside the bounding box	2	1
Person not considered inside the bounding box	4	0

Fig. 14 Confusion matrix table for trial 1

Predicted value	Actual Value: Person Considered inside the bounding box	Actual Value: Person not considered inside the bounding box
Person Considered inside the bounding box	3	0
Person not considered inside the bounding box	4	0

Fig. 15. Confusion matrix table for trial 2

Predicted value	Actual Value: Person Considered inside the bounding box	Actual Value: Person not considered inside the bounding box
Person Considered inside the bounding box	3	0
Person not considered inside the bounding box	3	0

Fig. 16. Confusion matrix table for trial 3

Predicted value	Actual Value: Person Considered inside the bounding box	Actual Value: Person not considered inside the bounding box
Person Considered inside the bounding box	3	0
Person not considered inside the bounding box	2	0

Fig. 17. Confusion matrix table for trial 4

Predicted value	Actual Value: Person Considered inside the bounding box	Actual Value: Person not considered inside the bounding box
Person Considered inside the bounding box	4	0
Person not considered inside the bounding box	6	0

Fig. 18. Confusion matrix table for trial 5

TABLE II. Applying the statistical analysis formula and confusion matrix

Trial	TPR	FPR	TNR	FNR
1	0.3333	1	0	0.6667
2	0.4286	0	0	0.5714
3	0.5000	0	0	0.5714
4	0.6000	0	0	0.4000
5	0.4000	0	0	0.6000

In trial 1, the applied parameters are: Gaussian Number is 3, Training frame is 120, and Minimum BG ratio is 0.5. There are three people that must be considered in the frame and there are only two people in the bounding box however one person is not identified as moving. Applying the statistical analysis, the TPR of trial 1 is 0.3333. In trial 2, the Gaussian Number is 5, Training frame is 150, and Minimum BG ratio is 0.7. There are three people that must be considered in the frame, however the shadow detected in the doorway is considered, the trial 2 has the TPR of 0.4286. In trial 3, the Gaussian Number is 4, Training frame is 120, and Minimum BG ratio is 0.5. There are three people that must be considered in the frame, however the shadow detected behind the monitor is considered, having the TPR of 0.50000. In trial 4, the Gaussian Number is 3, Training frame is 110, and Minimum BG ratio is 0.4. There are three people that must be considered in the frame, and all the people inside are considered and not detecting the shadows as well, the trial 4 has the TPR of 0.6000. While in trial 5, the Gaussian Number is 3, Training frame is 110, and Minimum BG ratio is 0.3. There are three people that must be considered in the frame, and the chairs are considered in the bounding box, trial 5 has the TPR of 0.4000.

C. Arduino Uno connection

Fig. 19. LCD of the Arduino Uno for prompt

Fig. 20. Motion detection message

The fig. 19 and fig. 20 shows the LCD screen of an Arduino Uno connected to the MATLAB program. The researchers set a sample time to the screen for an actual demonstration of the possible connection of the algorithm to a microcontroller.

4. CONCLUSION

Using the MATLAB software, the researchers developed an image processing algorithm that includes the Gaussian Mixture Model and Kalman filtering to predict motion tracks which can also be applied for microcontroller. The frame per second is also considered and observed that it has a low value, but it does not affect the tracking of the predicted motion. The camera must be placed in a fixed area to prevent noise adjustments. The sensitivity of the detection depends on the parameters set to the algorithm, the accuracy of the algorithm was tested by applying different sets of parameters to the algorithm and comparing the trials to each other. It is determined that by using confusion matrix, trial 4 has a higher TPR of 0.6000.

5. RECOMMENDATIONS

This study recommends to further calibrate the measurement of parameters for testing the accuracy of Kalman filtering and GMM. It is also recommended to consider the frame rate of the real-time video to have a higher quality of image and detection of the area. Since the camera used is an IP camera, it is recommended that the camera and PC/laptop has a high bandwidth connection to prevent errors and secure the quality control in running the program.

6. REFERENCES

[1] MATLAB, "MATLAB: Kalman Filtering," *https://www.mathworks.com/help/control/ug/kalman-filtering.html*, 2021.

[2] A. S. Katerina Fronckova, "Kalman Filter Employment in Image Processing," *ICCSA 2020 Comput. Sci. Its Appl.*, pp. 833–844, 2020, [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-58799-4_60#citeas

[3] F. Najar, S. Bourouis, N. Bouguila, and S. Belghith, "A comparison between different Gaussian-based mixture models," *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl. AICCSA*, vol. 2017-Octob, pp. 704–708, 2018, doi: 10.1109/AICCSA.2017.108.

[4] M. S. Verdadero, C. O. Martinez-Ojeda, and J. C. D. Cruz, *Hand gesture recognition system as an alternative interface for remote controlled home appliances*. 2019. doi: 10.1109/HNICEM.2018.8666291.

[5] J. A. Galindo and M. V. Caya, *Development of street lighting system with object detection*. 2019. doi: 10.1109/HNICEM.2018.8666410.

[6] S. Sanyal, "An Intuition about Kalman filter for Computer Vision," 2021. https://www.analyticsvidhya.com/blog/2021/10/an-intuition-about-kalman-filter/#:~:text=Kalman filter has two stages. It is a,"prediction" and "update" are called "propagation" and "correlation."

DOI: https://doi.org/10.15379/ijmst.v10i1.2666