RECOGNITION OF HUMAN ACTION UNDER VIEW CHANGE

by

MURUGAPPAN N 20084268
MAANICCKA SENTIL MA P 20084036
MOHAMED SHAMSUDEEN G 20084038

A project report submitted to the

FACULTY OF INFORMATION AND

COMMUNICATION ENGINEERING

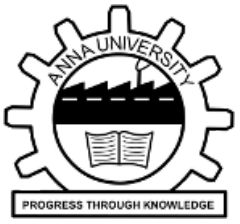
in partial fulfillment of the requirements

for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY CHENNAI

CHENNAI - 600025

April 2012

CERTIFICATE

Certified that this project report titled **"RECOGNITION OF HUMAN ACTION UNDER VIEW CHANGE"** is the *bonafide* work of **MURUGAPPAN N (20084268), MAANICCKA SENTIL MA P (20084036) and MOHAMED SHAMSUDEEN G (20084038)** who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these are any other candidates.

Place: Chennai
Date :

Dr.D.Shiloah Elizabeth
Project Guide,
Assistant Professor,
Department of Computer Science and Engineering,
Anna University Chennai,
Chennai - 600025

COUNTERSIGNED

Head of the Department,
Department of Computer Science and Engineering,
Anna University Chennai,
Chennai – 600025

ACKNOWLEDGEMENTS

We express our deep gratitude to our guide, **Dr.D.Shiloah Elizabeth** for guiding us through every phase of the project. We appreciate her thoroughness, tolerance and ability to share her knowledge with us. We thank her for being easily approachable and quite thoughtful. Apart from adding her own input, she has encouraged us to think on our own and give form to our thoughts. We owe her for harnessing our potential and bringing out the best in us. Without her immense support through every step of the way, we could never have it to this extent.

We are extremely grateful to **Dr.K.S.Easwarakumar**, Head of the Department of Computer Science and Engineering, Anna University, Chennai 600025, for extending the facilities of the Department towards our project and for his unstinting support.

We express our thanks to the panel of reviewers **Dr. Arul Siromoney, Dr. A.P. Shanthi and Dr.Madhan Karky** for their valuable suggestions and critical reviews throughout the course of our project.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

MURUGAPPAN N **MAANICCKA** **MOHAMED**
SENTIL MA P **SHAMSUDEEN G**

ABSTRACT

Visual recognition and understanding of human actions have attracted much attention over the past three decades and remain an active research area of computer vision. An automated visual surveillance system is used to detect abnormal behavior patterns and recognize the normal ones. When a person enters a room, video of him/her is captured and stored(both Front view and Top view). Then it is given to the training module where the video is split into frames and these split frames are used to extract the blob image. These blob images are stored for further comparison. In the detection phase, when a new person enters, anomaly is detected and alert system is generated to inform the administrator.

Contents

List of Figures

LIST OF ABBREVIATIONS

JMF	**J**ava **M**edia **F**ramework
AVI	**A**udio **V**ideo **I**nterleave
HMM	**H**idden **M**arkov **M**odel
SVM	**S**upport **V**ector **M**achine
RAM	**R**ead **A**ccess **M**emory
UML	**U**nified **M**odeling **L**anguage
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
FPS	**F**rames **P**er **S**econd

CHAPTER 1

INTRODUCTION

Visual recognition and understanding of human actions have attracted much attention over the past three decades and remain an active research area of computer vision. An automated visual surveillance system is used to detect abnormal behavior patterns and recognize the normal ones. When a person enters a room, video of him/her is captured and stored(both top view and front view). Then it is given to the training module where the video is split into frames and these split frames are used to extract the blob image.These blob images are stored for further comparison. In the detection phase, when a new person enters, anomaly is detected and alert system is generated to inform the administrator.

1.1 Methodology

Recent work has demonstrated the difficulty of the problem associated with the large variation of human action data due to the individual variations of people in expression, posture, motion, and clothing; perspective effects and camera motions; illumination variations; occlusions and dis occlusions; and distracting effects of scenes surroundings. Also, actions frequently involve and depend on manipulated objects, which add another layer of variability. As a consequence, current methods often resort to restricted and simplified scenarios with simple backgrounds, simpler kinematic action classes, static cameras, or limited view variations.

The abnormal behavior is detected by keeping track of the videos and blob frames and checking each frame values. The normality and abnormality of the captured motion is ascertained by checking the system for valid and invalid behavior that will be already trained. This involves training the module in the beginning for valid behavior by the controller and then video footage authenticated by the system.

One of the advantage of this system is that training phase and anomaly detection Phase can coexist together and it is adaptable to dynamic environment. (whenever needed video can be trained and detected)

When the video activity of a particular person is not matching with the trained dataset the system will generate error message to the administrator and if he/she wants to include this behavior as trusted and valid, the administrator can append this video for training anytime so that next time the video of same person is not detected as anomaly.

The user's gestures are taken as input and they are recorded in the system. Two view angles namely top view and front view are used to record the gestures. Whenever a user enters, the camera records his/her gestures and compares it with the prerecorded data in the system. If the gesture match is found,then the person is assumed to be authorized to enter. In case of mismatch, alert is initiated to the administrator that implies an unauthorized entry.

1.2 Definitions

User: A human being whose gestures are being monitored under video camera using JMF and given as input to the system for authentication.

Administrator: A person who is empowered to monitor the system and control it and who is responsible for addition of various trusted/valid action into the system and deletion of various invalid action from system. When a system monitors some invalid situation immediately the administrator is alerted.

Front View Video: Video capture of a person with the help of a web-camera integrated with Java Media Framework recording the front view of the person which will be used as input to the system.

Top View Video: Video capture of a person with the help of a web-camera integrated with Java Media Framework recording the top view of the person for enhanced accuracy and security which will be used as input to the system.

Training Phase: The process of training the system with new input behavior of various authenticated persons and storing them so that it could be used for comparison in detection phase.

Detection Phase: Final phase of the system which would distinguish between the authenticated and unauthenticated gesture of person and intimate the result to the administrator.

1.3 Problem Statement

To design and implement a dynamic and robust automated surveillance camera monitoring system that is capable of detecting invalid/unauthenticated user behavior with the help of pattern matching and trained data. It alerts the administrator and successfully performs various activities such as a video recording, segmentation, blob image extraction and data comparison without the use of any video database.

1.4 Project Scope

The following is the scope of the project system :

- Monotonous work of monitoring is automated and reduces manpower.
- Increase the accuracy and precision in monitoring with the help of top and front views.
- Provides more security.
- The system can be easily modified to suit dynamic requirements.

1.5 ORGANIZATION OF THIS REPORT

Chapter 1 deals with the brief introduction about the automation of surveillance camera system and the basic working of training phase, anomaly detection phase

and various definitions pertaining to the system.

Chapter 2 gives the details about previous publications dealing with recognition of human action,human action capture and analysis based on self-similarity

Chapter 3 deals with various system design features, detailed architecture diagrams and individual modules.

Chapter 4 deals with the complete implementation of the system along with class diagram, activity diagram, use-case diagram and brief description about every module.

Chapter 5 involves testing of the system under various conditions and parameters changed to obtain best output and detailed analysis.

Chapter 6 contains the results of the system along with performance analysis.

Chapter 7 provides conclusion for the project and scope for future enhancements.

CHAPTER 2

RELATED WORK

Visual recognition and understanding of human actions have attracted much attention over the past three decades and remain an active research area of computer vision. A good solution to the problem has yet an unexplored potential for many applications, such as the search and structuring of large video archives, video surveillance, human-computer interaction, gesture recognition and video editing.

Recent work has demonstrated the difficulty of the problem associated with the large variation of human action data due to individual variations of people in expression, posture, motion and clothing; perspective effects and camera motions; illumination variations and distracting effects of scenes surroundings. Also the actions frequently involved depend on manipulated objects, which add another layer of variability.

As a consequence, current methods often resort to restricted and simplified scenarios with simple backgrounds, simpler kinematic action classes, static cameras, or limited view variations.
Various approaches using different constructs have been proposed over the years for action recognition. These approaches can be roughly categorized on the basis of representation used by the authors. Time evolution of human silhouettes was frequently used as an action description.

For example, Bobick and Davis proposed capturing the history of shape changes using temporal templates and Weinland et al. extends these 2D templates to 3D

action templates. Similarly, the notions of action cylinders and space-time shapes have been introduced based on silhouettes. Recently, space-time approaches analyzing the structure of local 3D patches in the video have been shown to be promising in.

Using space-time or other types of local features, the modeling and recognition of human motion have been addressed with a variety of machine learning techniques, such as Support Vector Machines (SVMs) , Hidden Markov Models (HMMs). Most of the current methods for action recognition are designed for limited view variations.

A reliable and a generic action recognition system, however, has to be robust to camera parameters and different viewpoints while observing an action sequence. View variations originate from the changing and frequently unknown positions of the camera. Similar to the multi-view appearance of static objects, the appearance of actions may drastically vary from one viewpoint to another. Differently from the static case, however, the appearance of actions may also be affected by the dynamic view changes of the moving camera.

Bobick and Davis presented the view-based approach in their paper, ”Recognition of Human Movement Using Temporal Templates” [vol 23]. The basis of the representation is a temporal template which is a static vector-image where the vector value at each point is a function of the motion properties at the corresponding spatial location in an image sequence.Then a recognition method matching temporal templates against stored instances of views of known actions is developed. The method automatically performs temporal segmentation, is invariant to linear changes in speed, and runs in real-time on standard platforms.It

is in contrast to many recent efforts to recover the full three- dimensional reconstruction of the human form from image sequences, instead a view-based approach is developed to the representation and recognition of movement that is designed to support the direct recognition of the motion itself. ”Recognition of Human Action Under View Change” concentrates on the matching pattern of the human behavior and can be used as a real time application.

In ”Advanced Vision-Based Human Motion Capture and Analysis”[vol 103], current scene is interpreted correctly from the front angle and tracking humans from the scene in one or more frames is done. It involves the analysis of the new patterns and makes a match with the predefined stored patterns. The work concentrates only on the front view which may be a limitation to the scope of the project as in some cases as top view video monitoring along with the front view video is required for enhanced security. ”Recognition of Human Action Under View Change” concentrates on both top and front angle for better precision and accuracy in tracking.

In IEEE paper ”Recognizing Human Action With Local SVM Approach”[vol 36], Schldt et al. presented a concept based on SVM and Local space-time features that capture local events in video and could be adapted to the size, the frequency and the velocity of moving patterns. In this work they demonstrate how such features can be used for recognizing complex motion patterns.According to the author they construct video representations in terms of local space-time features and integrated such representations with SVM classification schemes for effective human action recognition. For the purpose of evaluation they introduced a new dedicated video database containing 2391 sequences of six human actions

performed by 25 people in four different scenarios. The presented results of action recognition justified the proposed method and demonstrated its advantage.

The use of database for storing and retrieving videos made the system slow in operation and processing. ”Recognition of Human Action Under View Change” does not use any database hence fast in system working and detection.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 Functional Requirements

3.1.1 Video Recording

The input AVI video file that is captured using JMF used as input for this module and eventually splitted into sequence of JPEG images frames with an unique name comprising the time stamp of that particular image. Various parameters for recording video in JMF :

- RGB encoding format
- 320x240 video resolution.
- 15 fps (frames per second)

3.1.2 Training Phase

This module should initially train the system with new input behavior of various authenticated person and store the modeling into system as a new system doesn't contain any list of valid or trusted activities appended by the administrator. After this step the new activity should be recorded into the system and next time when the system detects the same new activity by the user there should be no mismatch with

the trained data value and it should ascertain as valid person. Main objective of this phase is to recognize and differentiate between the authenticated valid person and the unauthenticated invalid person.

3.1.3 Detection Phase

Once the initial training phase is completed the system should be ready to be deployed for checking the human action recognition under view change and when it has detected that a person enters a room, video of him is captured and stored (both side view and the top view) then it is given to the training module here the video is checked and if it is a normal behavior splited image is taken, whenever the action is recognized blob images are saved, and the frame counts are taken. In case an anomaly is detected a red alert should be displayed. The abnormal behavior should be achieved by tracking the videos and blob frames and checking each frame values.

3.1.4 Invalid Action Response

When the system encounters an abnormal behavior by tracking the video and blob frames, checking each frame values with that of system entry, then an alert should be sent to the administrator and he determines whether the particular person whose action has been absent in the system has to be updated into set of actions trusted or not.

3.2 Hardware Requirements

- Processor : 2 GHz and above
- RAM : 2 GB
- Hard Disk : 4 GB (free space for storing good resolution video)
- Web Camera : VGA resolution or above

3.3 Software Requirements

- Operating System : Windows XP, Vista or 7
- Platform : J2SE 1.5 Java 2 Platform, Standard Edition(J2SE)
- IDE : NetBeans 6.9 and above or Eclipse
- Special Tool : JMF 2.2.1e

More details about JMF are included in the Appendix.

3.4 Specific Requirements

3.4.1 Login Capability

The software shall provide the login capability only to the administrator who has the power to record other authorized users gestures. No other persons can view

their footage and do not have permission to modify it without the administrator's intervention.

3.4.2 Usability

The software is user friendly and self-explanatory with minimum knowledge about using computer is sufficient for effective usage of the system. Java provides many user friendly and interactive features for the project to be effectively used. Java language enabled with multi threaded interface helps in handling multiple tasks simultaneously. Java supports multi threading programs which implies that we need not wait for the application to finish one task before beginning another.

3.4.3 Reliability

The system is completely reliable due to the importance of data and the damages due to incorrect or incomplete data. Java is a robust language. It provides many safeguards to ensure reliable code. It has strict compiler time and runtime checking for data types. It is designed as a garbage-collected language relieving the programmers virtually from all memory management problems

3.5 List of Use Cases

3.5.1 Use case 1 : Initiate Training Phase

Goal :

Train the system with new input behavior of various authenticated person and store the modeling into system

Precondition:

1. Current activity has been successfully recognized by the system
2. Activity has no match with the database entry of trusted activities

Trigger:

The controller/administrator has authenticated the new activity and wants the new activity to be inserted into the system for future authenticated entry

Description:

New activity has been recorded into system and next time when the system detects the same new activity by the user there will be no mismatch with the system entry and it will ascertained as valid person

Post Condition:

New activity added to list of trusted activities in system

3.5.2 Use case 2 : Detection of Authenticated Activity

Goal :

Behavior of the user whose gestures are already trained should be identified as valid

Precondition:

Current activity has been successfully captured by the camera and input has been scanned and motion recognized by the system.

Trigger:

Activity has a match with the system entry of trusted activities

Description:

Activity that has been detected is a trusted activity which has been already entered into the system and the person entering into the room is an authenticated person

Post Condition:

Normal Behavior of the monitoring system.

3.5.3 Use case 3 : Detection of Unauthenticated Activity

Goal :

Invalid activity or behavior of the user is identified and alarm initiated

Precondition:

1. Current activity has been successfully captured by the camera, input has been scanned and motion recognized by the system.
2. Activity has no match with the database entry of trusted activities and the controller/administrator has been alerted

Trigger:

Mismatch occurs while comparison of current video with the previously trained video in the system

Description:

Activity that has been detected is not a trusted activity as there is no system entry and the person entering into the room is not an authenticated person

Post Condition:

Normal Behavior of the monitoring system.

CHAPTER 4

SYSTEM DESIGN

4.1 System Features

4.1.1 System Training Phase:

This module would initially train the system with new input behavior of various authenticated persons and stores it into the system as a new system doesn't contain any list of valid or trusted activities appended by the administrator. After this step the new activity has been recorded into the system and next time when the system detects the same new activity by the user there will be no mismatch with the trained data value and it will be ascertained as a valid person. Main objective of this phase is to recognize and differentiate between the authenticated valid person and the unauthenticated invalid person.

4.1.2 System Detection Phase:

Once the initial training phase is completed the system is ready to be deployed for checking the human action recognition under the view change and when it has detected that a person enters a room, video of him is captured and stored (both front view and the top view) then it is given to the training module. Here the video is checked, if it is a normal behavior split image is taken, whenever the action is recognized blob images are saved and the frame counts are taken. In case, anomaly

is detected a red color alert will be displayed. The abnormal behavior is achieved by tracking the videos and blob frames and checking each frame values.

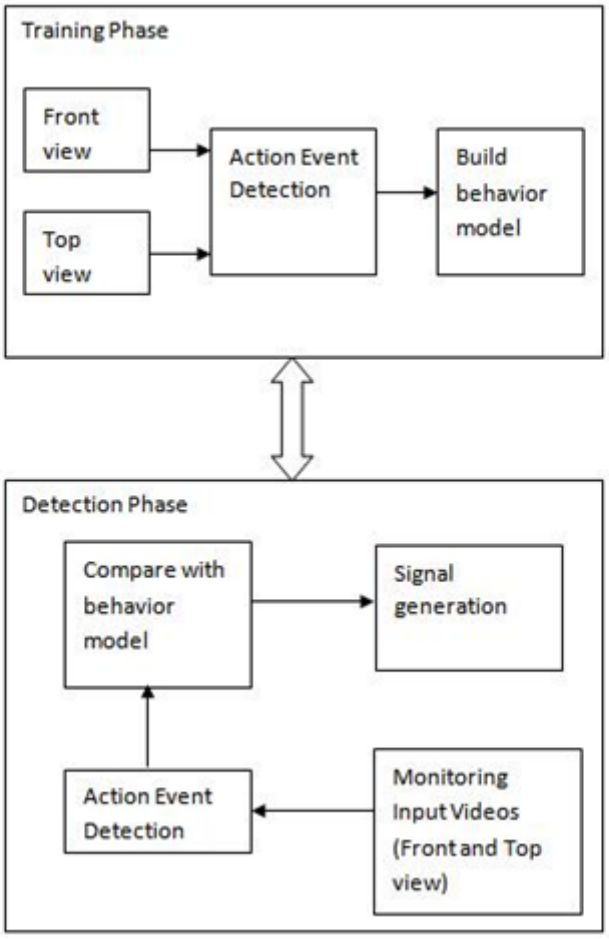


FIGURE 4.1: Two Phase Architecture Diagram

4.2 Detailed Design

4.2.1 Class Diagram

Class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematics of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling comprising the attributes of the class and methods of operations as shown in Fig 4.2

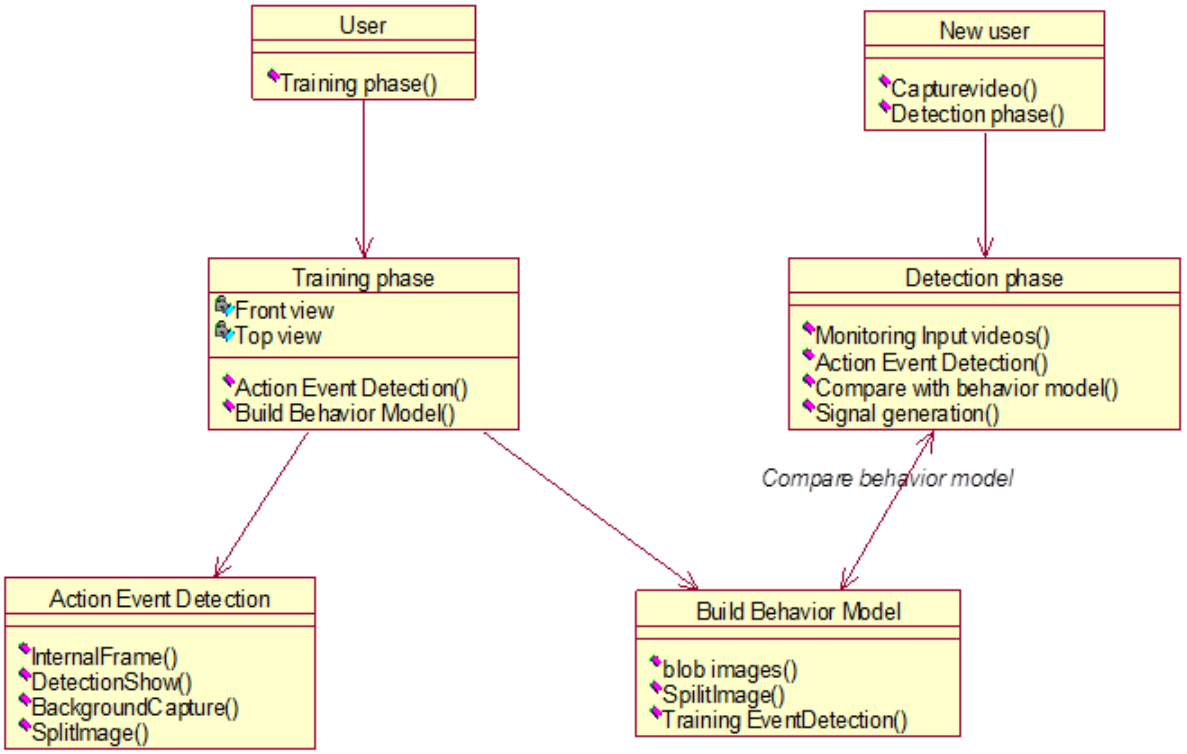


FIGURE 4.2: Detailed Class Diagram

4.2.2 Activity Diagram

Activity diagrams are graphical representations of work-flows of stepwise activities and actions with support for choice, iteration and concurrency. Activity diagram with all possible states of the system and corresponding inputs and outputs of each state along with control and information of entire system shown in Fig 4.3.

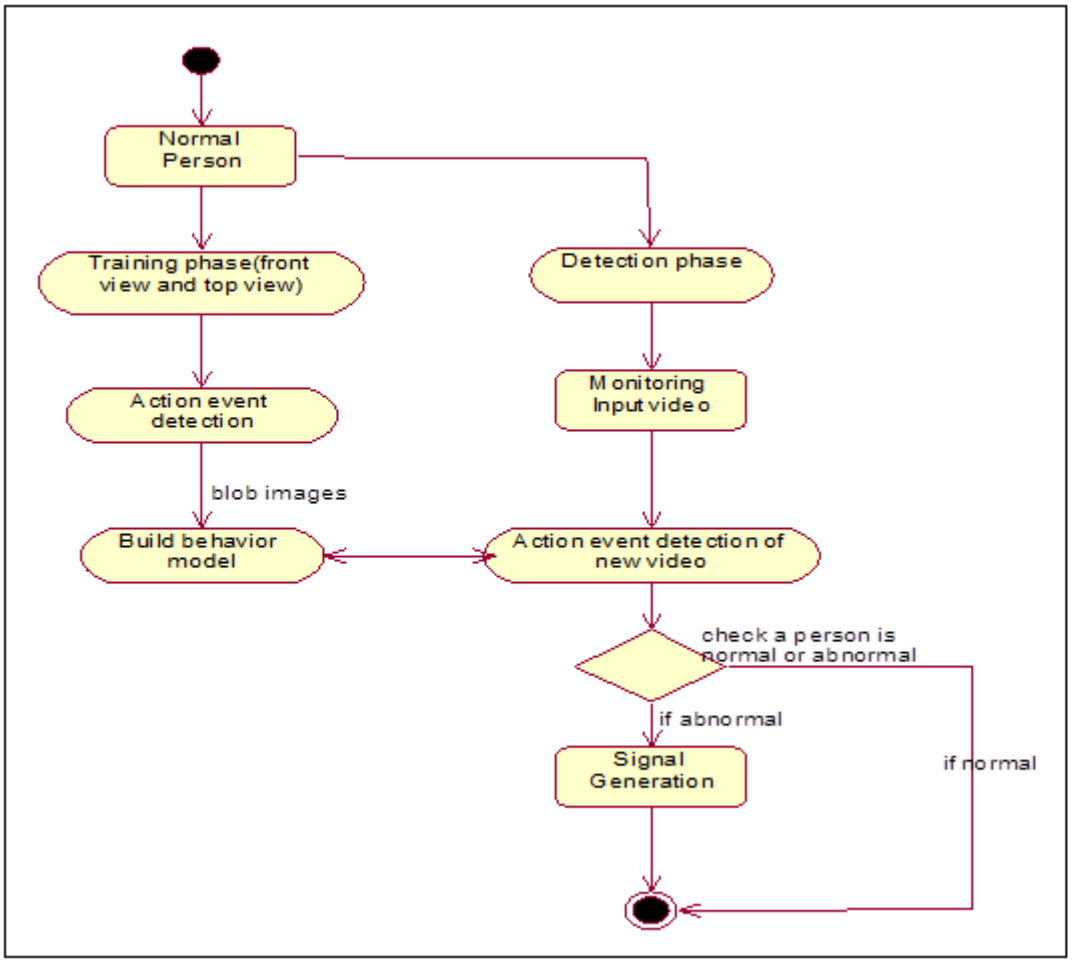


FIGURE 4.3: Activity Diagram

4.2.3 Sequence Diagram

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario as shown in Fig 4.4

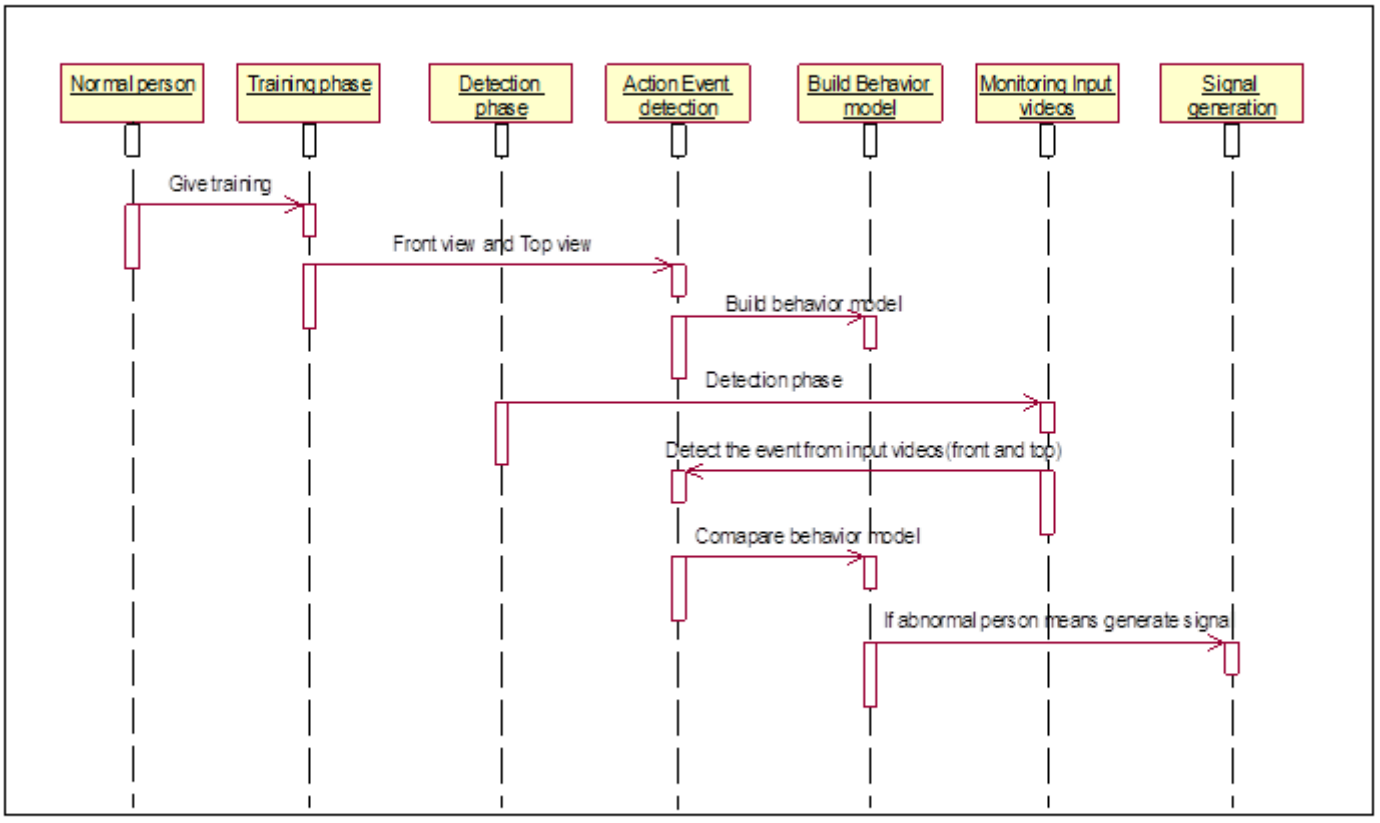


FIGURE 4.4: Sequence Diagram

4.2.4 Collaboration Diagram

A collaboration diagram is an illustration of the relationships and interactions among software objects as shown in Fig 4.5

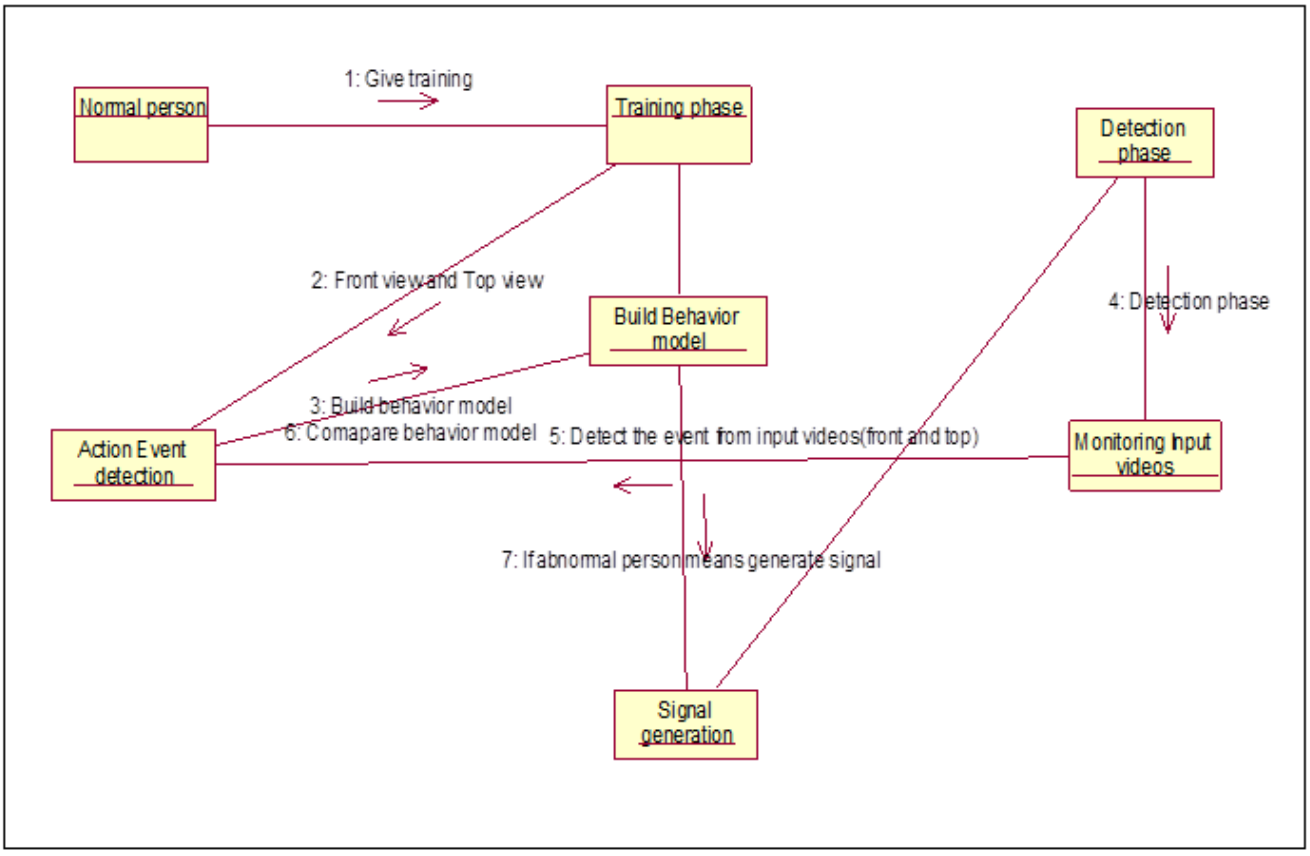


FIGURE 4.5: Collaboration Diagram

CHAPTER 5

SYSTEM DEVELOPMENT

This chapter details on various implementation testing procedures done over the completed modules in order to ensure the correctness of the implementation of the two phases and proper working of recording videos using JMF, segmentation of the video during training phase and accurate anomaly detection.

5.1 Implementation Overview

This project involves recording videos through JMF(Java Media Framework). The videos are recorded in AVI(Audio Video Interleave) format, RGB encoding, 15 frames per second, 320x240 resolution and training them into the system and storing the data for future comparisons involved in the testing phase and detecting authenticated and unauthenticated videos in the system detection phase.

5.1.1 Video Recording

Video input given to the system is recorded with JMF tool with specific parameters namely 320 X 240 video resolution, 15 fps, RGB encoding technique. JMF software can be easily used to record videos in .avi format with all required parameters. Go to File - Export the video, parameters can be entered and File - Capture records the video as shown in Fig 5.1 and 5.2 respectively.

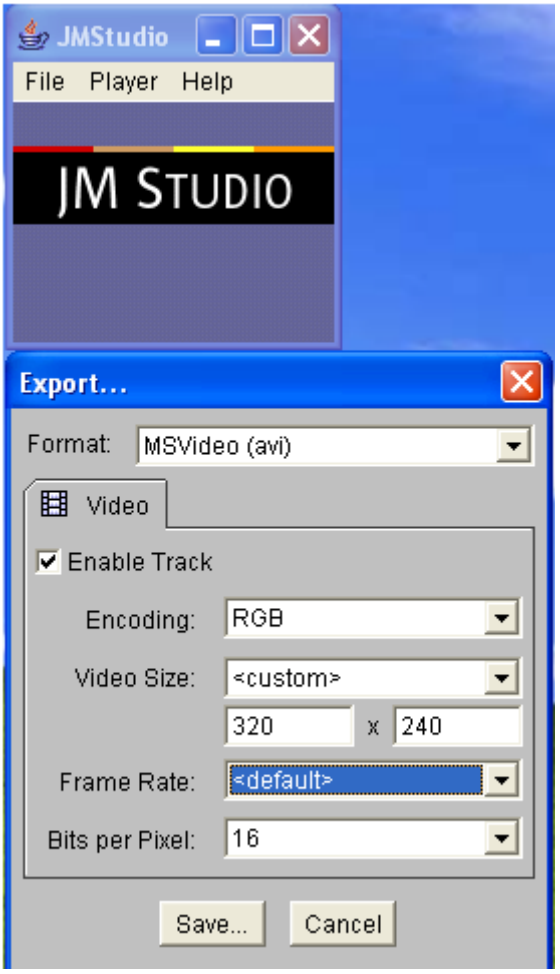


FIGURE 5.1: Setting required parameters for video

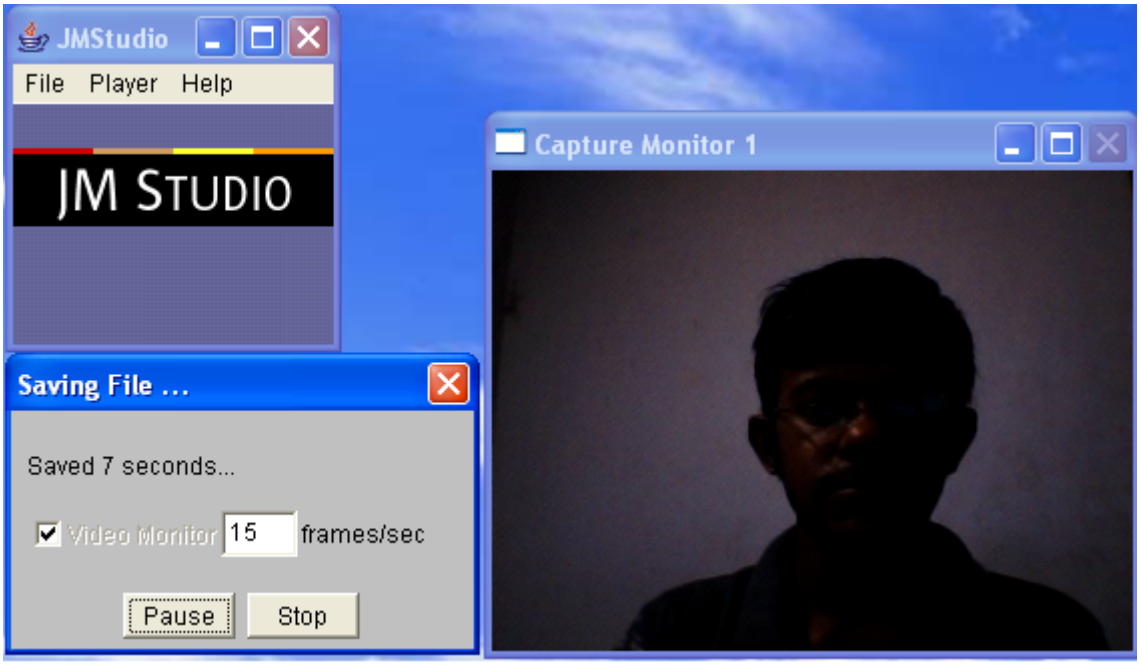


FIGURE 5.2: Capturing Video

5.1.2 Blob Image Extraction

The newly recorded video using JMF is trained in the training phase of the project. Training phase involves splitting the video into a sequence of frames called the splitted frames and from these frames the BLOB images are extracted and stored as system files, this reduces the load on the system as database is not involved which implies easier storage and retrieval of files in the system. The BLOB images and splitted frames are stored in separate folders.

The first step involved in system implementation is recording an empty background which is stored for future comparisons. Once the background is stored, videos with humans showing various gestures in the pre-stored background are recorded. Now the video is splitted into frames and for each frame the BLOB image is extracted

by comparing all the points in the current frame with the corresponding points in the previously stored empty background.

The point where there is a huge(level of pixel variation) difference in the pixels is treated as the starting of the BLOB and once the pixels start matching for the two frames in comparison, that point is treated as the end of BLOB for that particular frame and so on BLOB images are extracted for all the frames in the input video and stored. The size of the BLOB can be varied to concentrate onto specific details but normally its kept constant.

5.1.3 Algorithm

5.1.4 Training phase

5.1.4.1 Action Event Detection

```
start blob image extraction
if video NOT avi format
Print 'Incorrect File Name'
endif
if next.frame for input_video
compare split frame image with background_image
set r,g,b to R,G,B values of split frame image
set r1,g1,b1 to R,G,B values of background image
compare the pixel of corresponding points
if pixel comparison varies
mark the start point of blob_image
endif
if pixel comparison NOT varies AND blob_image marking started
mark the end point of blob_image
endif
endifend blob image extraction
```

In this module, the recorded video using Java Media Framework(JMF) is checked to be in the correct AVI format. Then, in the proper recorded video, the frames are split from the video for training. The background image without any person

is recorded to extract the blob image from the split frame. The blob image is compared with the background image and the blob image is extracted.

5.1.4.2 Building Behavior Model

```
start data serialization
if blob_image is extracted
serialize the blob_image
else
start blob image extraction
endif
end data serialization
```

Data Serialization refers to the process of converting an object into a format that can be stored(eg. a file format). So, the extracted blob image are used to be stored for anomaly detection. Hence these image objects are serialized in this module.

5.1.5 Detection Phase

5.1.5.1 Action Event Detection

```
start blob image extraction
if video NOT avi format
Print 'Incorrect File Name'
endif
if next.frame for input_video
```

```
compare split frame image with background_image
set r,g,b to R,G,B values of split frame image
set r1,g1,b1 to R,G,B values of background image
compare the pixel of corresponding points
if pixel comparison varies
mark the start point of blob_image
endif
if pixel comparison NOT varies AND blob_image marking started
mark the end point of blob_image
endif
endif
end blob image extraction
```

This module is similar to Action Event Detection module of training phase. In this module, the recorded video using Java Media Framework(JMF) is checked to be in the correct avi format. Then, in the proper recorded video, the frames are split from the video for training. The background image without any person is recorded to extract the blob image from the split frame. The blob image is compared with the background image and the blob image is extracted. These blob images are compared with the pre-stored blob images in the next module.

5.1.6 Comparison With Behavior Model

```
start comparison with behavior model
if next.frame for input_video
extract blob_image for each split frame
```

```
initialize checksquarewidth to 10
compare each pixel of checksquarewidth with corresponding pixels
if different_pixels < (checksquarewidth * checksquarewidth) /2
return TRUE
else
  return FALSE
endif
repeat for all squares of the blob image
endif
end compare with behavior model
```

The extracted blob images are compared with the pre-stored blob images. The comparison is made by constructing an imaginary square of width 10. The pixels in this imaginary square are compared with the corresponding pixels of the pre-stored blob image. The pixels that differs are counted and if the count is lesser than the half of the square of the width of the imaginary square means, then true value is returned. Similar comparisons are made for consecutive squares and the above process is repeated for all frames of the video.

5.1.7 Signal Generation

```
start signal generation
if blob_image mismatch
return RED SIGNAL
else
return GREEN SIGNAL
```

```
endif
end signal generation
```

If an anomaly is detected, then the detection phase generates an RED signal to alert the administrator. If no anomaly is detected, then the system shows that only authorized person is entered and there will be no change in the GREEN signal.

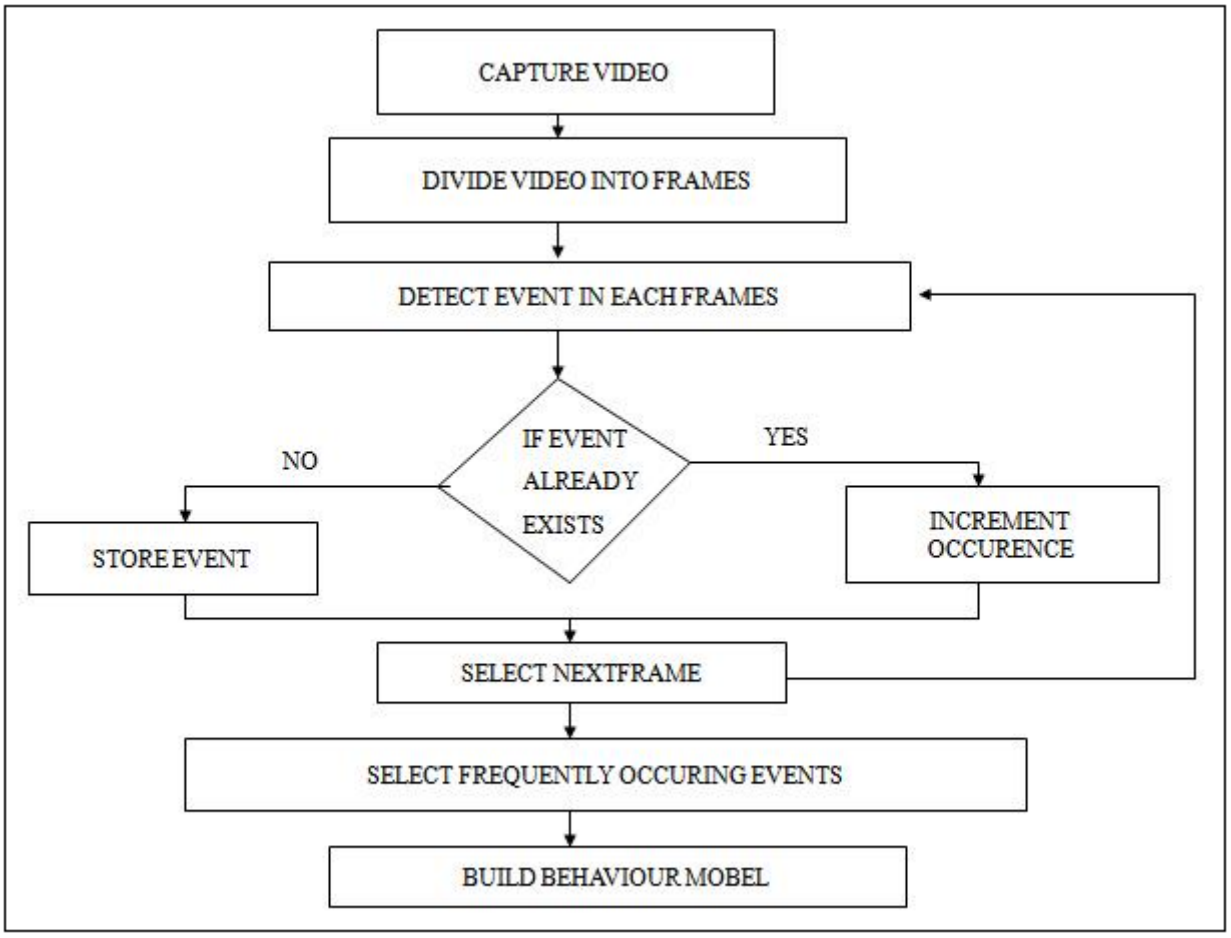


FIGURE 5.3: Flow Diagram-Training Phase

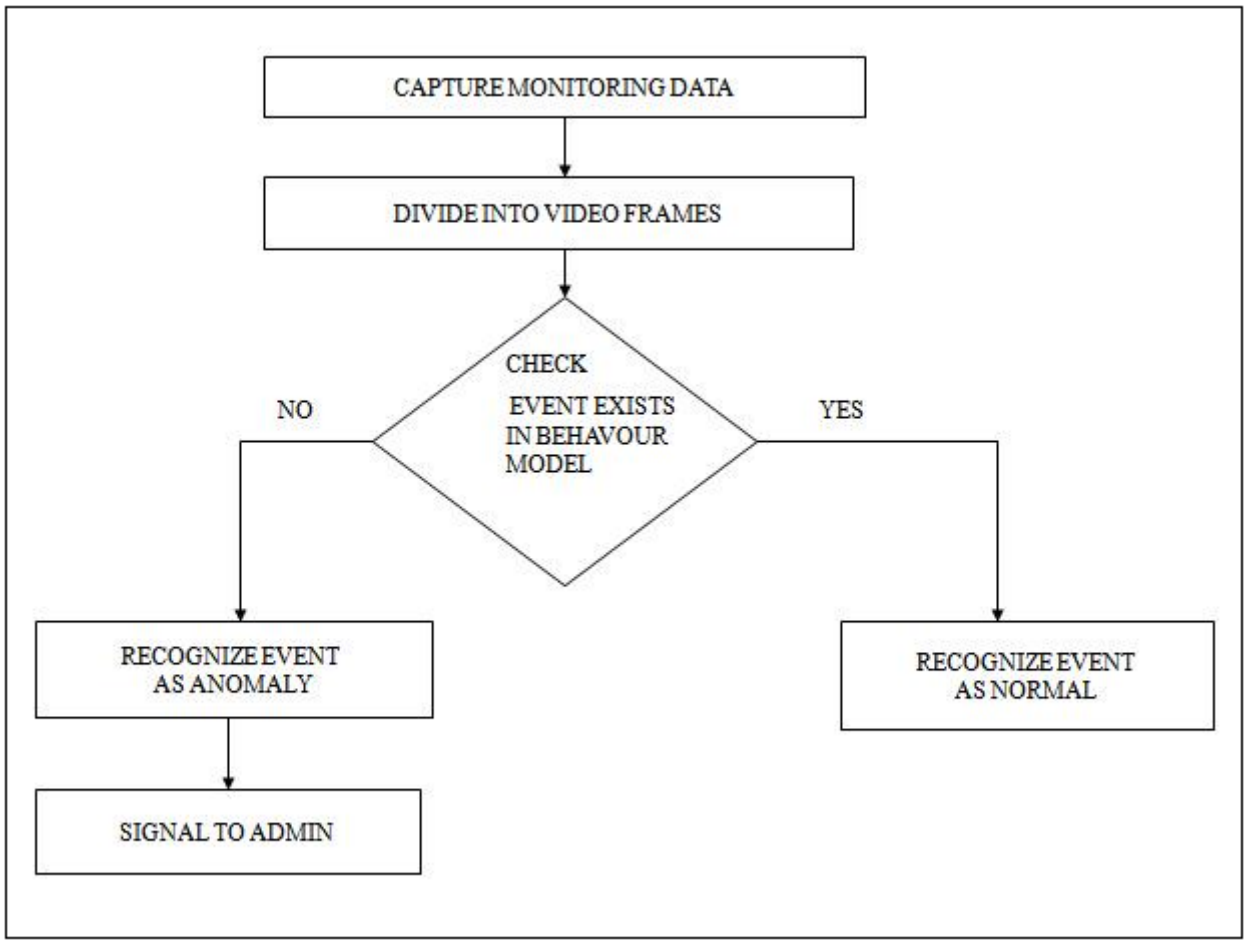


FIGURE 5.4: Flow Diagram-Detection Phase

There are two phases involved in the overall project, namely training phase and the detection phase. Each phase consists of certain number of modules whose pseudocode has been specified. The flow diagram also gives a detailed picture of what process is being done during the overall execution as show in Fig 5.3 and Fig 5.4.

CHAPTER 6

RESULTS AND DISCUSSION

This chapter provides various results and performance analysis of the system along with graphs that are used for critically evaluating the system based on varied input values and parameters.

6.1 Result

The proposed system gets an input video in AVI format at 15 frames per second, recorded in RGB encoding. The video is split into frames, the BLOB images are extracted from each frame and finally stored in separate folders for future comparisons. An empty background video is also needed to be stored at the start which will be used in the BLOB image extraction process but finding out the difference in pixels.

The next stage is the detection phase or anomaly detection phase in which a video is given as input to the system to check if it is authenticated or not. In this phase again the video is split into frames and BLOB images are extracted which will be compared with the pre- stored BLOB images using pixel comparison. Once an anomaly is detected via pattern matching, a Red signal indicates mismatch which can be handled by the administrator.

There is an option to save the background data used for comparison which can be used when the need arises. The data is stored in a separate file which can be edited if needed by the administrator. There may be certain cases where a trained

person may go unauthenticated due to wide changes in the gestures recorded during training phase.

Under such a situation the administrator can make the person authenticated and append the new gestures, this feature guarantees better precision and accuracy in the long run.

6.2 Data Analysis

6.2.1 Analysis Based on Check Square Width

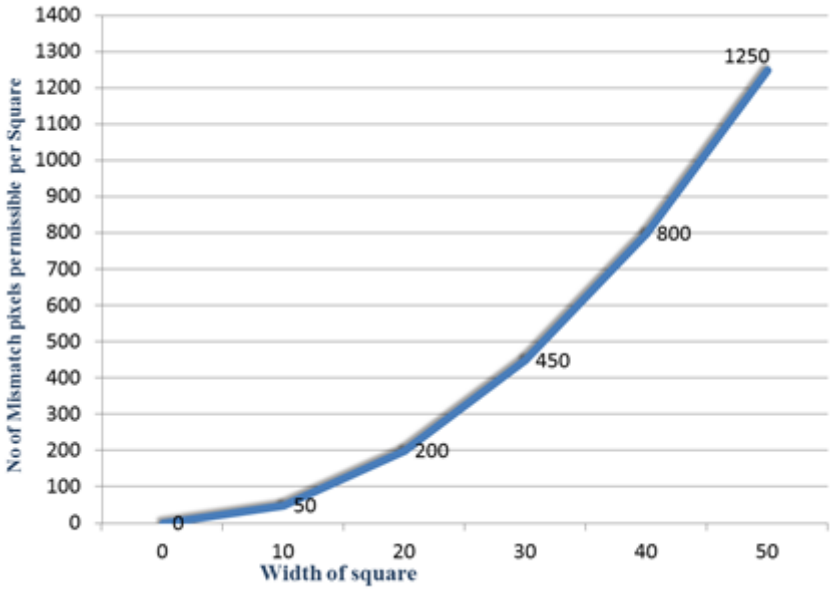


FIGURE 6.1: Data Analysis Diagram

6.2.2 Analysis Based on Pixel Approximation

In any given video frame of 320X240 resolution there will be a total of 76800 pixels available (320 pixels column wise and 240 pixels row wise matrix arrangement). The variation in number of same pixels detected with respect to the level of approximation of pixel variations is shown in the Fig 6.2.

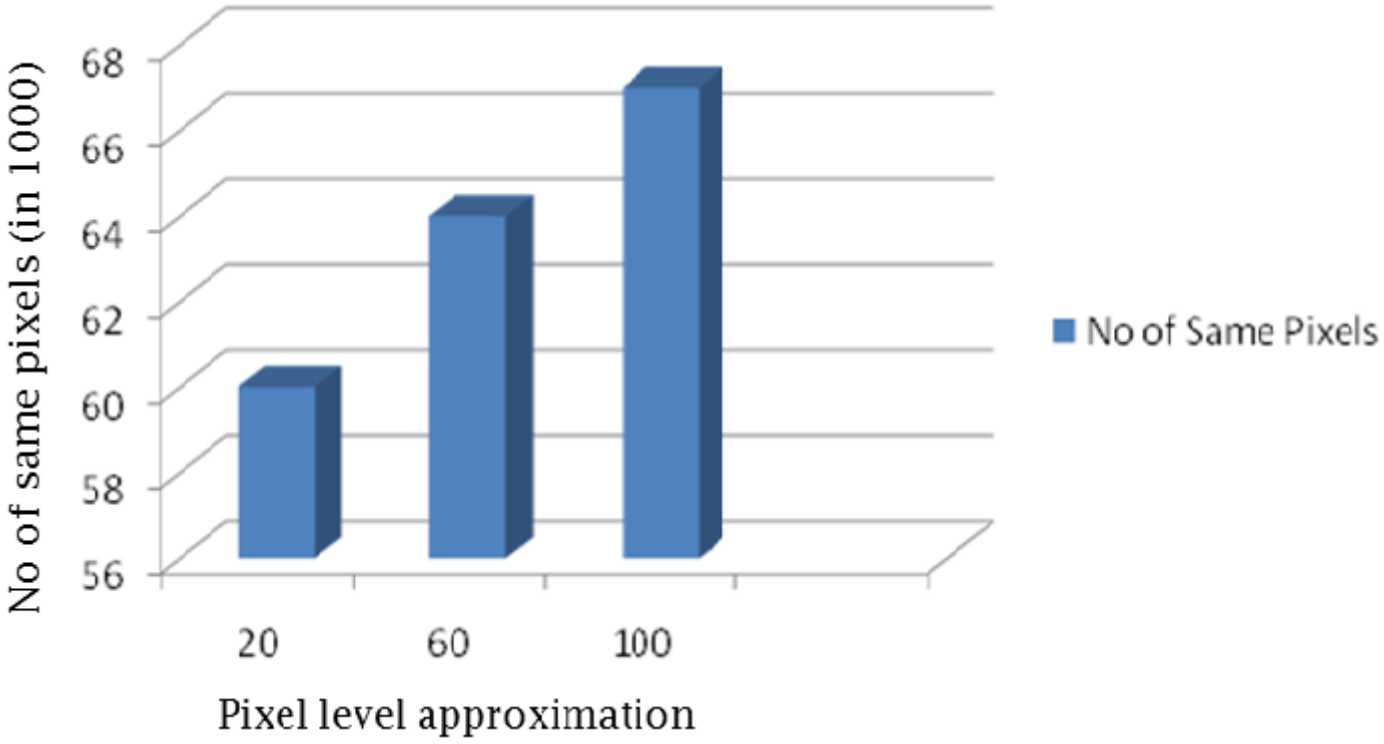


FIGURE 6.2: Data Analysis Diagram

When comparing the current frame with the previously trained frame during anomaly detection phase at pixel level, approximation of 20 pixel variation yields about 59000 same pixels (i.e. when one pixel value in current frame is 180 then plus or minus 20 values is accepted as same pixel for a trained data. In this case

pixel value range between 160 and 200 are accepted). When level of approximation slightly increased to 60 pixels then 63000 same pixels are detected. Similarly when the 100 pixel approximation is kept 100, this amount rises to 68000 same pixels. The most optimum results are achieved with about 20 to 25 pixel approximation.

6.2.3 Performance Analysis

To check the performance of the system 30 videos (10 videos each for durations 10 sec, 20 sec, 30 sec) were recorded on the whole, in addition 10 videos with invalid and contradictory gestures where also recorded to check the anomaly detection efficiency.

The system performance for 10 second video clip is presented and out of the 10 videos, 7 videos were accepted as valid and 3 videos were misinterpreted as invalid gestures and detected as anomaly video. Out of these 7 accepted videos, only 5 videos are non anomalous and other 2 are anomalous. Similarly, out of the 3 anomaly videos detected, only 2 are anomaly and 1 is a non anomalous video. Hence error is 3, detection is 7 and false alarm is 2 as shown in Fig 6.3.

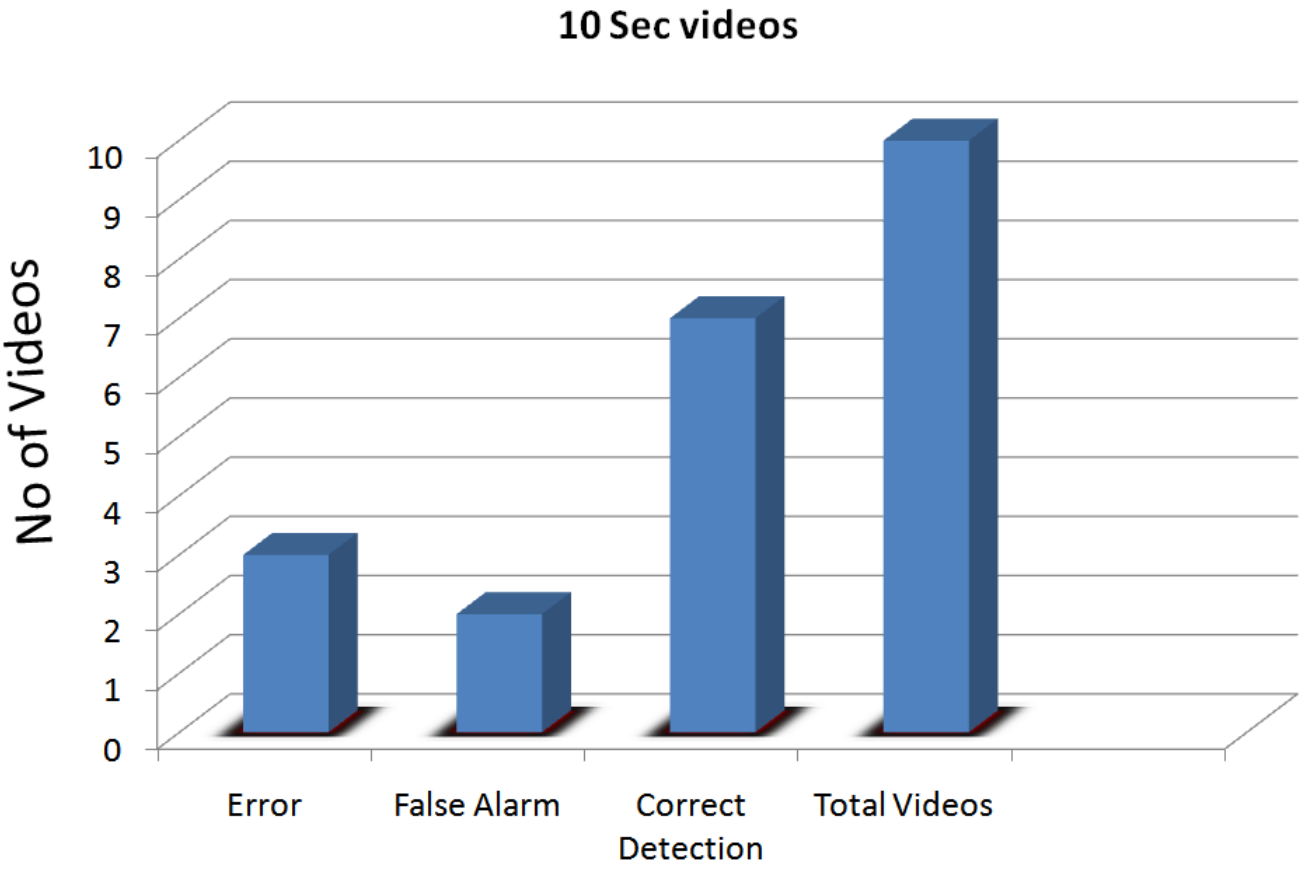


FIGURE 6.3: Performance Analysis Graph : 10 second Video

The system performance for 20 second video clip is presented and out of the 10 videos, 7 videos were accepted as valid and 3 videos were misinterpreted as invalid gestures and detected as anomaly video. Out of these 7 accepted videos, only 6 videos are non anomalous and other 1 are anomalous. Similarly, out of the 3 anomaly videos detected, only 2 are anomaly and 1 is a non anomalous video. Hence error is 2, detection is 8 and false alarm is 1 as shown in Fig 6.4.

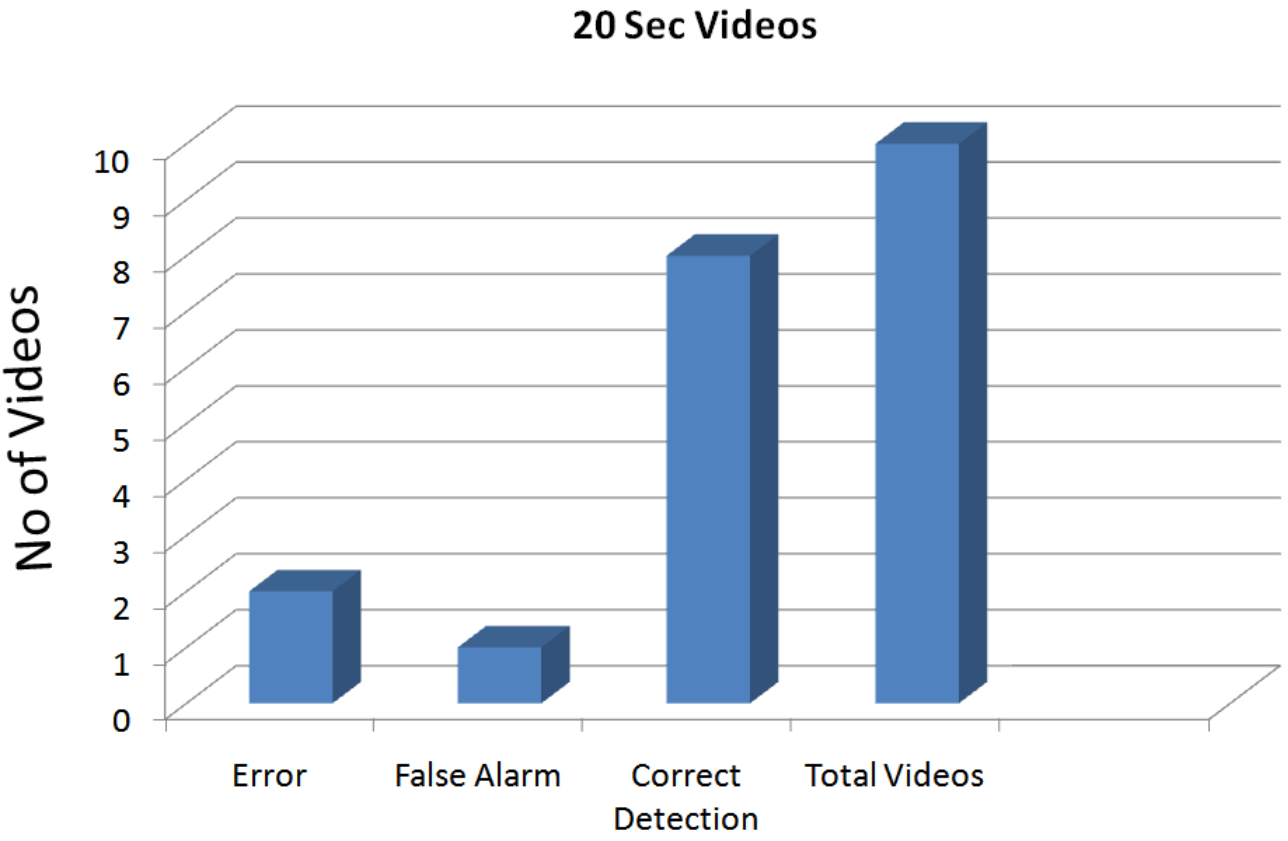


FIGURE 6.4: Performance Analysis Graph : 20 second Video

The system performance for 30 second video clip is presented and out of the 10 video, 6 videos were accepted as valid and 4 videos were misinterpreted as invalid gestures and detected as anomaly video. Out of these 6 accepted videos, only 4 videos are non anomalous and other 2 are anomalous. Similarly, out of the 4 anomaly videos detected, all the four videos are anomalous. Hence error is 2, detection is 8 and false alarm is 2 as shown in Fig 6.5.

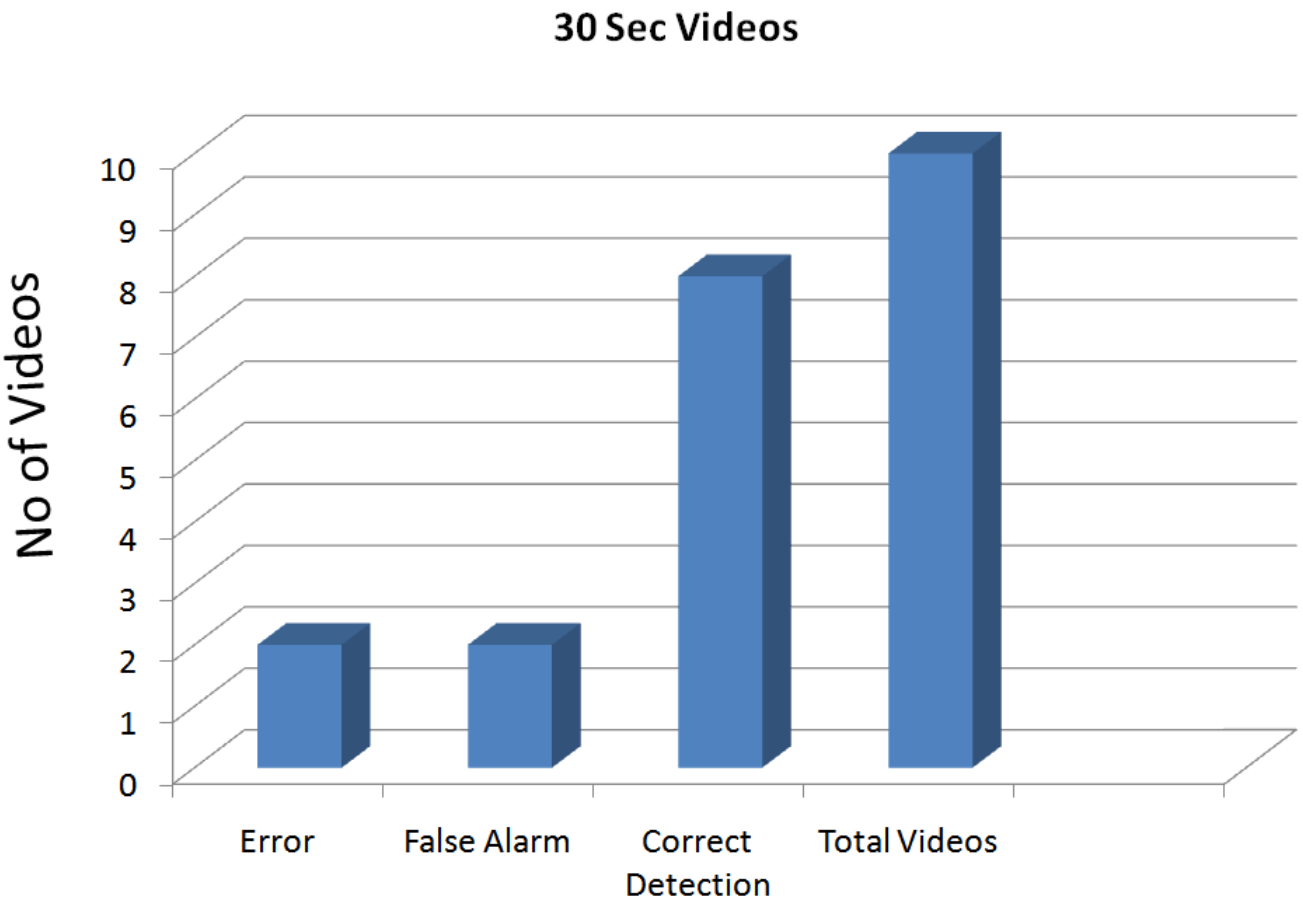


FIGURE 6.5: Performance Analysis Graph : 30 second Video

CHAPTER 7

CONCLUSIONS

7.1 Contributions

Previous works in the human action recognition domain were based on the use of video database which would be time consuming operations due to retrieval and storing heavy media content .In this system absence of database provides more optimized and quick results as all essential informations such as BLOB images,frames are stored in the file system directly.

Generally human action recognition in the past has concentrated only with the Front view camera position and the videos, but in this system for enhanced security and precision both front view and top view videos are simultaneously trained and detected for anomaly. Even if one view video is unclear or vague in deciding the validity problem of a particular video footage the other view could be used as a backup.Monitoring front and top view would help in finding synchronous behavior pattern of the human subject.

7.2 Future Work

In the future, the two view camera could be scaled up to multi-view camera positions and monitoring human action with 3 dimensional video camera is also

possible which would provide more accuracy.With the use of 3D technology slightest change could be monitored and detected which could be deployed in highly sensitive areas such as military base, airport, VVIP offices etc .

While extracting the subject in BLOB image extraction, the Height to Width ratio of the human subjects could be measured for further analysis about human gestures and actions in real time environment.

Instead of a static camera used in this system, advanced dynamically moving cameras could be used for effective and efficient recording of videos that could be used in applications such as an automated car driving system with multi-view cameras mounted on to car for sensing the traffic around and drive safely.

APPENDIX A

APPENDIX

A.1 Java Media Framework

A.1.1 Introduction

The Java Media Framework (JMF) is a Java library that enables audio, video and other time-based media to be added to Java applications and applets. This optional package, which can capture, play, stream, and transcode multiple media formats, extends the Java Platform, Standard Edition (Java SE) and allows development of cross-platform multimedia applications.JMF provides a unified architecture and messaging protocol for managing the acquisition, processing and delivery of time-based media. JMF enables Java programs to

1. Present (playback) multimedia contents,
2. Capture audio through microphone and video through camera,
3. Do real-time streaming of media over the Internet,
4. Process media (such as changing media format, adding special effects),
5. Store media into a file.

A.1.2 Capture device

A capture device represents the hardware you use to capture data, such as a microphone, a still camera, or a video camera. Captured media data can be fed into a player to be rendered, processed to convert the data into another format, or stored for future use. Capture devices can be categorized as either push or pull sources. With a pull source, the user controls when to capture an image. As an example, think of a still camera where a user clicks a button to take the shot. In contrast, a microphone acts as a push source because it continuously provides a stream of audio data.

A.1.3 Player

A player takes as input a stream of audio or video data and renders it to a speaker or a screen, much like a CD player reads a CD and outputs music to the speaker. A player can have states, which exist naturally because a player has to prepare itself and its data source before it can start playing the media. To understand this, insert a CD into your stereo and play the fourth song on the CD. What would happen? The CD player does not instantly play the song. It first has to search the track where the fourth song begins and do some other preparations. After about half a second (depending on your CD player), you start to hear the music. Likewise, the JMF Player must do some preparation before you can hear the audio or see the video. In normal operations, a player steps through each state until it reaches the final state. JMF defines six states in a player:

- Unrealized: In this state, the player object has been instantiated. Like a newborn baby who does not yet recognize its environment, a newly instantiated player does not yet know anything about its media.

- Realizing: A player moves from the unrealized state to the realizing state when you call the player's realize() method. In the realizing state, the player is in the process of determining its resource requirements. A realizing player often downloads assets over the network.

- Realized: Transitioning from the realizing state, the player comes into the realized state. In this state the player knows what resources it needs and has information about the type of media it is to present. It can also provide visual components and controls, and its connections to other objects in the system are in place.

- Prefetching: When the prefetch() method is called, a player moves from the realized state into the prefetching state. A prefetching player is preparing to present its media. During this phase, the player preloads its media data, obtains exclusive-use resources, and does whatever else is needed to play the media data.

- Prefetched: The state where the player has finished prefetching media data – it's ready to start.

- Started: This state is entered when you call the start() method. The player is now ready to present the media data.

A.1.4 Format

A format object represents an object's exact media format. The format itself carries no encoding-specific parameters or global-timing information; it describes the format's encoding name and the type of data the format requires. Format subclasses include audio format and video format. In turn, video format contains six direct subclasses:

- H261 format
- H263 format
- IndexedColor format
- JPEG format
- RGB format
- YUV format

A.1.5 Capture Media Data

Media capture is another important task in JMF programming. You can capture media data using a capture device such as a microphone or a video camera. It can then be processed and rendered, or stored in a media format. To capture media data, you need to:

1. Locate the capture device you want to use by querying the CaptureDeviceManager

2. Obtain a CaptureDeviceInfo object for the device
3. Get a MediaLocator from the CaptureDeviceInfo object and use it to create a DataSource
4. Create either a player or a processor using the DataSource
5. Start the player or processor to begin the capture process

You use the CaptureDeviceManager to access capture devices available on the system. This manager acts as the central registry for all capture devices available to JMF. You can obtain an available device list by calling the getDeviceList() method. A capture device is represented by a CaptureDeviceInfo object. You use the CaptureDeviceManager's getDevice() method to get the CaptureDeviceInfo for a particular capture device.

To use the capture device to capture media data, you then need to get the device's MediaLocator from its CaptureDeviceInfo object. You can either use this MediaLocator to construct a player or a processor directly, or use the MediaLocator to construct a DataSource that you can use as the input to a player or processor. Use the player's or processor's start() method to initiate the capture process.

References

[1] A. Bobick and J. Davis,“ The Recognition of Human Movement Using Temporal Templates”, *IEEE Transactions, Pattern Analysis and Machine Intelligence, vol. 23, no. 3, pp. 257-267*, 2001.

[2]] T. Syeda-Mahmood, M. Vasilescu, and S. Sethi, “Recognizing Action Events from Multiple Viewpoints”, *Proc. IEEE Workshop Detection and Recognition of Events in Video, pp. 64-72*, 2001.

[3] T. Moeslund, A. Hilton, and V. Kruger, “A Survey of Advances in Vision-Based Human Motion Capture and Analysis”, *Computer Vision and Image Understanding, vol. 103, nos. 2-3, pp. 90-126*, 2006.

[4] L. Wang, W. Hu, and T. Tan,“Recent Developments in Human Motion Analysis, Pattern Recognition”, *vol. 36, no. 3, pp. 585-601*, 2003.

[5] D. Weinland, R. Ronfard, and E. Boyer,“ Free Viewpoint Action Recognition Using Motion History Volumes”, *Computer Vision and Image Understanding, vol. 103, nos. 2-3, pp. 249-257*, 2006.