Java Implementation based Heterogeneous Video Sequence Automated Surveillance Monitoring

Sankari Muthukarupan sankarim2@gmail.com
Information Systems and Technology Department
Sur University College, Sur, Oman

Bremananth Ramachandran bremresearch@gmail.com
Information Systems and Technology Department
Sur University College, Sur, Oman

Ahmad Sharieh ahmadsharieh@suc.edu.om
Information Systems and Technology Department
Sur University College, Sur, Oman

Abstract

Automated video based surveillance monitoring is an essential and computationally challenging task to resolve issues in the secure access localities. This paper deals with some of the issues which are encountered in the integration surveillance monitoring in the real-life circumstances. We have employed video frames which are extorted from heterogeneous video formats. Each video frame is chosen to identify the anomalous events which are occurred in the sequence of time-driven process. Background subtraction is essentially required based on the optimal threshold and reference frame. Rest of the frames are ablated from reference image, hence all the foreground images paradigms are obtained. The co-ordinate existing in the deducted images is found by scanning the images horizontally until the occurrence of first black pixel. Obtained co-ordinate is twinned with existing co-ordinates in the primary images. The twinned co-ordinate in the primary image is considered as an active-region-of-interest. At the end, the starred images are converted to temporal video that scrutinizes the moving silhouettes of human behaviors in a static background. The proposed model is implemented in Java. Results and performance analysis are carried out in the real-life environments.

Keywords: Anomalous Events Detection, Background Subtraction, Frame Extraction, Foreground Detection, Surveillance.

1. INTRODUCTION

Video surveillance is actively employed to monitor locations and the behavior of people in private and public areas. Since events resembling the terrorist attacks in all over the world, there have been a further deploying the need for video surveillance systems to assure the safety of people in diverse locality. These kinds of system can be utilized in case of monitoring crucial events such as crisis or disasters. For that, several cameras, technologist and supervisors are required to supervise the video streams for many hours. In order to provide the solution for this problem, an artificial intelligent system based surveillance system was proposed. Such an automated system can observe events to realize a perspective susceptible analysis of the diverse circumstances.

Automatic Video Surveillance (AVS) is a computationally challenging process due to its sequence of tasks to accomplish the identification of diverse foreground objects. Surveillance is commonly employed for describing observation of human behaviors based on recorded video sequence. In narration of video surveillances, plenty of works had been completed [1][2][3][6], however, detecting human behavior is still an open problem in order to take on diverse circumstances and timely actions.

Analysis of Human behavior (AHB) has various steps such as background subtraction, foreground detection, tracking meticulous behavior and analyzing behavior of a subject [1]. This approach has various applications such as lick recognition; video surveillance in public or private locality, behavior based multimedia retrieval, security access and prohibition of unauthorized people and many others. Furthermore, algorithms of intelligent video surveillances can involve complex computation and relatively powerful technology that acts like a human intelligence agent. However, keeping track of multiple people, vehicles, and their interactions, within a multifaceted urban environment are a difficult task. The user obviously should not be looking at two dozen screens showing raw video output in order to interpret the selective events. Our approach is to provide an interactive, graphical user interface and tracking the human.

In this paper, we propose a methodology to integrate heterogeneous environment for video surveillance monitoring system to detect the behavior of human using Java real-time classes. This method assists optical flow computation method in order to localize subjects or objects in real-life scenario. A Java based platform independent tool is proposed to detect the background, localize the objects and locate the anomalous events such as walking with diverse action.

The remainder of this paper is organized as follows: Section 2 describes the literature review of the proposed problem. Proposed Java based automated video sequence methodology is depicted in Section 3. In Section 4, the Java implementation and result analysis of the proposed system is illustrated. The concluding remarks and future enhancement are revealed in Section 5.

2. LITERATURE APPRAISAL

In the literature appraisal of automated surveillance, certain, notable researches were done based on pattern recognition and Computer vision algorithms. However, integration of surveillance system in the real-life scenario and platform independent is highly inadequate. In the Literature review [4], human movement analysis is an important task in the Computer vision perspective. It includes background subtraction, detection, tracking, and recognition of subjects in the real-life scenarios. Human movement analysis can be categorized into four categories [8], such as low level vision detection, intermediate level vision tracking, high level vision behavioral analysis and extreme level vision anomalous event analysis. Subject detection is a process of detecting subject from a video sequence by applying background subtraction, foreground extraction and locates active-region-of-interests. In the extreme level vision, a watch dog algorithm will be invoked to integrate and monitor several related video clips to localize and track the anomalous events, respectively.

A system was proposed by Collins et al. [1] to monitor multiple cooperative video sensors to provide continuous coverage of people and vehicles in a cluttered environment. In that, Video Surveillance and Monitoring (VSAM) detected the objects' or subjects' movements in a complex environment and also described to find the detection objects' categories such as truck or car or human or living thing. This is based on color analysis. Propositionally, time complexity is increased based on number of the subjects or objects which are available to classify in the color space.

There are three conventional approaches [2] to detect objects such as temporal differencing, Gaussian mixture model and Optical flow. Temporal differencing is a process of adaptive to dynamic environments, but generally does a poor job of extracting all relevant feature pixels. Gaussian mixture model provides the most complete feature data, but is extremely sensitive to dynamic scene changes due to lighting and extraneous events. Optical flow can be used to detect independently moving objects in the presence of camera motion. However, most optical flow computation methods are computationally complex, and cannot be applied to full-frame video streams in real-time without specialized hardware and software integration [3] [5] [10][11]. In [9], an approach of detecting objects in dynamic background was described for noisy environment.

One additional benefit of using our proposed approach is its lower cost. Other surveillance systems do not eliminate human participation in solving investigation task fully. The proposed approach significantly simplifies the job and assists an analyst or investigator, who is not a professional in statistics and programming to manage the process of extracting knowledge from a video scene [4][6][7][12] in heterogeneous environment.

Based on the state-of-art review in the automated surveillance monitoring, in this paper, we contributed Java implementation based heterogeneous video monitoring to assist optical flow computation method to localize subjects or objects in real-life scenario without any specially designed hardware.

3. PROPOSED METHOD

In this section, the proposed method for Java implementation based heterogeneous video monitoring is presented. It has a sequence of operations that are required for the automated silhouettes behavior observation.

3.1 Classes Association and Preprocessing

The proposed approach is associated with object oriented design (OOD) in Java Media Frame work. Association relationship of classes' attributes, methods and exceptions are illustrated in Figure 1. It also reveals the association of classes' hierarchy of the proposed approach in a Unified Markup Language (UML) static structure.

Audio Video interleaved (AVI) format is invoked and its essential tracks are interpreted for observing progression of events in a scenario. An exception is thrown if any problem exists while loading. In this process, Java Media Framework media player is used to open and to load and run the AVI in a player. Similarly, the panel is loaded to select the video file by using Java swing and media framework. Here, a program is designed to load a panel. This panel is invoked to load AVI. Another panel is utilized for extracting frames. It is called the loading panel program. The following is an algorithm to load AVI in the heterogeneous environment.

Step 1: Choose surveillance file over internet or intranet.

Step 2: For media, Universal Resource Locator (URL) is initialized as null for initial processing.

Step 3: Try for locating the file over internet. If it exists, then media tester is loaded; otherwise malformed URL exception is invoked and control goes to Step 6.

Step 4: If mediaURL is not Null, then video sequence is showed on the heterogeneous environment and control goes to Step 5 else Null content is expected in the stream of loaded sequence and execution goes to Step 6.

Step 5: For Manual interaction, User Navigation properties are loaded.

Step 6: End the Loaded process and control goes to Frame Extraction.

Once video sequence is loaded, then end-users can give their options of human behavior or extreme level vision for anomaly events.

Frame extraction is a process to extort sequence of frames from the loaded video sequence over the internet or intranet. Java Media Frame package creates the Processor class for extracting frame from the sequence. After creating and configuring, the processor control obtains the track control for video track. Codec class is utilized to instantiate JMF package to extract the set of frames to access codec from the data flow path. Then, Set the track control to flow of the path and then extract the progression of frames. The following algorithm describes frame extraction process.

Step 1: Processor initialization.

Step 2: Let State Transition is true.

Step 3: Try to create a processor from the process Manager.

Step 4: If process exception causes due to unable to create process, then return false and control invoked exit else add controllers to the processor.

Step 5: Put the Processor into configured state. If Processor is configured successfully then control goes to Step 6 for obtaining track control in the temporal data; otherwise Processor configuration is failed and control is treated as false alarm.

Step 6: If track_control is null, then there is no track control exists for the processor and control invoked exit else track control is found and proceed with Step 7.

Step 7: Seeking the track control for the video track as in (1).

$$\phi_{i+1} = \prod \phi_i + \alpha\lambda \tag{1}$$

where ϕ_i denotes object tracking control based on ith frame in the sequence and ϕ_{i+1} represents object track control based on (i+1)th frame in the sequence. The track control will be monitored as described in (2).

$$\omega = \begin{cases} True & if \phi_i \in V \\ False & if \phi_i \notin V \end{cases} \tag{2}$$

Step 8: If video_Track is null after seeking entire temporal data, then the input temporal media doesn't have any video_track and control goes to Step 15.

Step 9: Instantiate and set the frame access codec to the data flow path and realize the processor.

Step 10: parseVideoSize (video Format) is initialized for parsing the size of the video from the string of video format.

Step 11: Wait for state Synchronization and perform a Java coding to make synchronization of state transition. Next, event synchronization is performed to link the events in a given temporal data. Step 12 is carried out to do the event synchronization based on the controller keep posted.

Step 12: Process Controller update is performed to incorporate entire synchronized events. A Java coding is invoked to carry out controller update.

Step 13: The process of converting temporal sequence to static frame by means of converting AVI-to-JPEG (Joint Photography Expert Group). The Java source code is employed to implement an interface called preAccessCodec from the Codec (Step 9).

Step 14: Store the static frame into JPEG format. The accessFrame is invoked to convert each frames of temporal data into RGB JPEG file for further processing.

Step 15: Stop the process of Frame extraction and Gray scale Conversion model is invoked.

Gray scale pixels are more convenient for processing intensity variation of signals due to widely varied shine and out-door scenarios. In this, the images are composed exclusively of shades of gray varying from black at the weakest intensity to white at the strongest. Each pixel from the original image is extracted. Each pixel with RGB value is separated and Gray pixel value is computed. The Gconvert Java class is the source code to convert the three-dimensional color space value into one-dimensional gray space. After the gray conversion, each frame of the temporal background pixels is subtracted, which is based on the reference frame, in order to localize the foreground silhouettes.

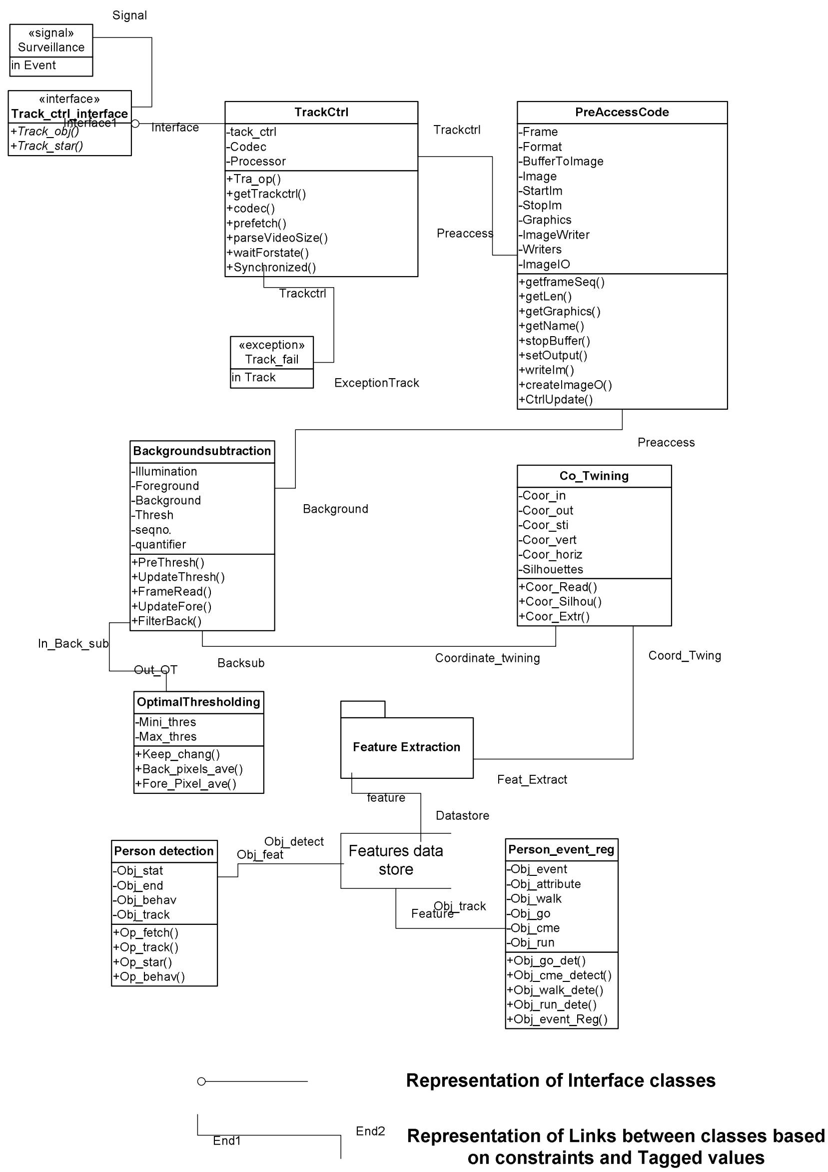

FIGURE 1: Class hierarchy using UML static structure for proposed video surveillance method.

3.2 Background Subtraction

In the background subtraction, initial frame has been treated as reference frame and do the deducting process on rest of the frames from the referred frame. A group of volume-of-pixels (voxels) of reference image is involved for the subtraction process with indexed image that is currently being considered for the background subtraction. By using the calculated threshold value, the newly obtained images with black and white pixels are stored for further preprocessing. The obtained result of the background subtraction is a possible of silhouettes which are obtainable from a scene. The background deduction process is performed using (3),

$$P(R_i) - P(B_i) = 0,$$
$$abs[(P(R_i) - (P(B_i) + P(F_i))] = P(F_i), \tag{3}$$

The $P(R_i)$ represents probability of the pixels which are in Reference frame R_i, $P(B_i)$ denotes probability of the pixel value in the current frame which is to be involved for background subtraction, and $P(F_i)$ is probability of absolute values which are obtained after the background subtraction.

3.3 Coordinates Twinning

In this module, silhouettes are taken as an input to scan the coordinates in order to obtain the twining process for localizing the Active-Region-of-Interest (AROI). This is the primary process of identifying the stimulus of diverse activities which are opted for the surveillances. In our algorithm, twining process starts from horizontal coordinate positions. Entire silhouettes are scanned until the black pixel occurs in a first row of scanning. If set of 'ON' pixel is encountered, then set of current positions is marked and then scan is continued. If set of volume-of-pixels is congregated, then the specified threshold is bounded. Next, scanning continue in vertical direction to mark the set of 'ON' pixels. The whole outline of the possible contour is treated as silhouettes. However, from the obtained silhouettes, we could not directly reveal that the contour belongs to stimulus of event. Once register the possible silhouettes then matching process is required for identifying the diverse activities of subjects from the given temporal sequence.

3.4 Optimal Threshold Estimation

The algorithm of the optimal threshold [7] is employed to estimate the threshold values. There is no knowledge about the exact location of objects in the extracted frame. Hence, consider as an approximation that the four corner of the image contains background pixels and remaining contains objects. Compute μ_B and μ_F as the mean background and foreground of the gray space, respectively. The thresholds are computed as given in (4),

$$\mu_B = \frac{1}{PQ}\sum_{i=1}^{P}\sum_{j=1}^{Q} B(i,j), \mu_F = \frac{1}{NM}\sum_{i=1}^{N}\sum_{j=1}^{M} F(i,j), \tag{4}$$

where P and Q are the number of background pixels, N and M are the number of foreground pixels, $B(i,j)$ and $F(i,j)$ are background and foreground information of the image, respectively. If background and foreground are equal in the distribution of pixels, then optimal threshold is the average of the means as described in (5) otherwise (6) has been computed in order to compensate the background and foreground pixel spaces.

$$\lambda_T^{t+1} = \frac{|\mu_B + \mu_F|}{2}, \tag{5}$$
$$\lambda_T^{t+1} = N_d(\mu_B, \sigma_B) + N_d(\mu_F, \sigma_F), \tag{6}$$

where λ_T^{t+1} is a threshold value at t+1 time, N_d denotes normal distributions of mean and standard deviation of pixels.

3.5 Feature Extraction and Object Tracking

In order to detect person, Haar features have been utilized. These Haar features are having low and high frequency components of multi-resolution signals. Based on the person features, connected component analysis have been involved to detect the coordinate of the subjects which are appearing in the scene. While training the manually extracted person silhouettes are implicated as optimistic exemplars, in that, a region of single person appearing in the template of surrounding substance. Furthermore, pessimistic exemplars such as no person on the scene are also trained. They are extracted randomly from the frame of image sequences based on empirical testing with no foreground occurrences. The followings are the sequence of steps of training process.

Step 1: Optimistic exemplars are trained with diverse backgrounds.

Step 2: Pessimistic exemplars are trained in order to determine the negative cases.

Step 3: Both Optimistic and Pessimistic exemplars are involved for learning process with diverse foreground for adopting dress variations from person-to-person.

Step 4: Initial approximation for both exemplars are obtained and they are classified manually.

Step 5: The system is retrained using diverse exemplars in order to adapt to different characteristics of scene.

Step 6: In order to make recalling process, a hetroassociator has been formed.

4. IMPLEMENTATION AND RESULTS ANALYSIS

Implementation has been done in Java in order to monitor the temporal space with static background video sequence. The following sections are described to illustrate result analysis of the proposed method.

4.1 Loading Input and Frame Extraction

This phase aims to focus on implementing heterogeneous video sequence to monitor the moving object using platform independent Java class. Figure 2 shows the media tester which is utilized to load AVI. Once the specified format is loaded into a default media player, then, the frame extracted module is invoked to extract the frames from the chosen video file and store the extracted frames in a knowledge base of silhouettes. The typical extracted frames are illustrated in Figure 3. It consists of a sequence of 145 frames for a real-life walking surveillance with data rate of 4500kbps, total bit rate was 4564kbps and frame rate was 30 frames / second.

FIGURE 2: Depiction of input Video File has length of five seconds, frame width and height is 320 and 240, respectively.

FIGURE 3: Representation of typical Extracted Frames from the given video.

4.2 Gray Scale Conversion

The extracted frames are involved to be read sequentially and then regenerated into gray image. From each extracted frames, pixel by pixel scanning is carried out for collecting RGB value of each pixel. This is done by using getRGB() method and r,g,b value of each pixels is multiplied with some specified predefined luster values in order to obtain a rearranged r, g, and b values for all pixels. Sum of the multiplied RGB value is assigned to each pixel by using setRGB() method to attain gray images. The result is shown in Figure 4.

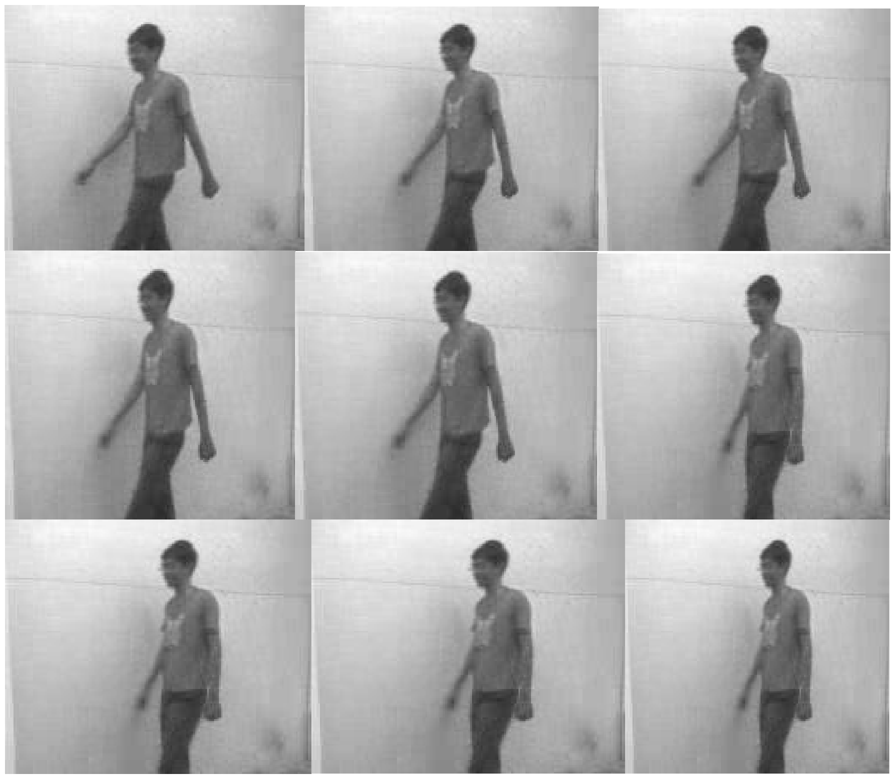

FIGURE 4: Sequence of Gray images convolved from the extracted frames.

4.3 Deduction Process

After gray conversion, the principal image is noted as a reference frame. This is shown in Figure 5. The remaining frames are deducted from reference frame (typically Figure 6) to obtain foreground images as shown in Figure 7.

Before Deduction

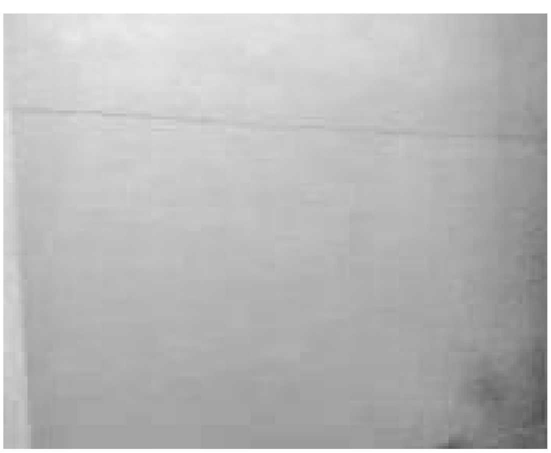

FIGURE 5: Reference image.

FIGURE 6: Foreground image.

After Deduction

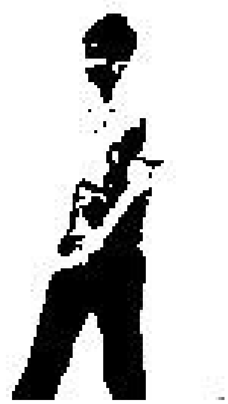

FIGURE 7: Deducted silhouette image that shows the walking chic.

The important process carried out here is that all the pixels are converted into binarization based on the predefined threshold. In this process, the first background image is considered as reference frame and rest of the frames are deducted from the reference frame. The crucial factor is how to estimate optimal threshold values for diversity of temporal sequences. The predefined threshold values are estimated using optimal threshold estimation method which includes luster and other artifacts. Since the result of all the three r,g,b picture elements are identical, red color elements is chosen for the deduction process. The value of each pixel in the reference image and the value of each pixel in the rest of the images are deducted one by one. If the deducted pixel value is less than the threshold value, then set the new image pixel as black or white and hoard those images. If the deducted pixel value is greater than threshold value, discard the obtaining images. Figure 8 depicts the frames with foreground object obtained using the proposed approach.

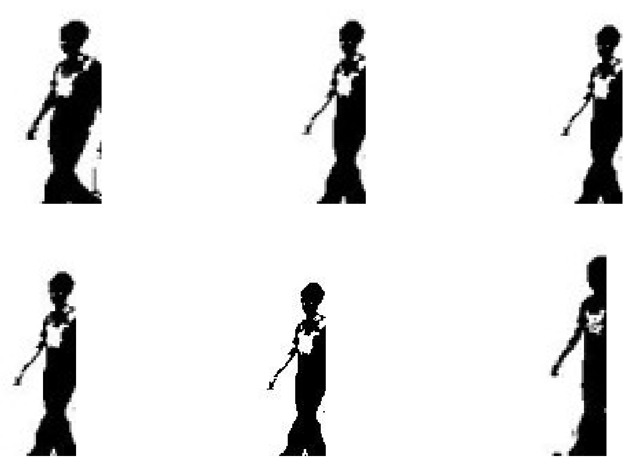

FIGURE 8: Deducted Images of extracted frames.

Once frames are deducted, histogram equalization was employed for enhancing the contour of the silhouettes in order to make the registration for silhouettes matching. The collection of silhouettes border is enriched and binary feature sets are indexed to store contour blob along with optimal thresholds of each temporal sequences. Figure 9 and Figure 10 show some possible silhouette of human body and walking chic, respectively.

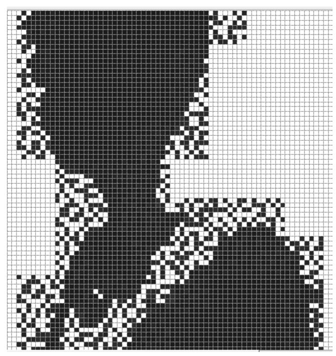

FIGURE 9: Representation of Silhouette of head portion and border of cloth.

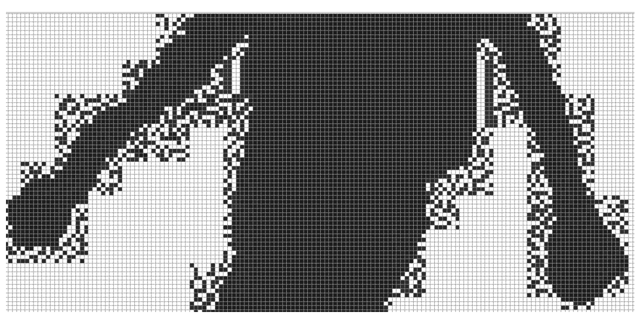

FIGURE 10: Illustration forward walking silhouettes.

4.4 Finding and Toning Coordinates

The deducted images have been used for the further process known as co-ordinates matching. In this process, at the most primitive x, y positions are fixed and scan the images horizontally until the set of black pixel is encountered. After finding the contour of black pixel, the encountered position value is perceived in x and y as the co-ordinate of original images equal by validating with ablated images. The acquired coordinate positions in the deducted image are matched with the original image. Silhouettes are stared. They are referred as Active-Region-of-Interest (AROI). This operation is employed using graphics class in Java as shown in Figure 11.

4.5 Performance Analysis

Intensity variations of silhouettes were observed with respect to dark blobs oscillation either in clock or anti-clockwise directions. Figure 12 illustrates intensity variations of black boundaries of silhouettes. It reveals that number of picture elements of silhouettes in AROI ranges from 0 to 450 pixels and peak intensities from 0 to 5 have more accounts of observations. Implicitly, it discloses that proposed approach of extraction of silhouettes in heterogeneous platform gives prominent improvement.

In next phase of experiment, rest of the intensities was observed. This is shown in Figure 13. It reveals that white region of the silhouettes ranges from 0 to 255 and number of picture elements in peak white region were conspicuously which is more suitable to matching the objects.

FIGURE 11: Illustration of Detection foreground images.

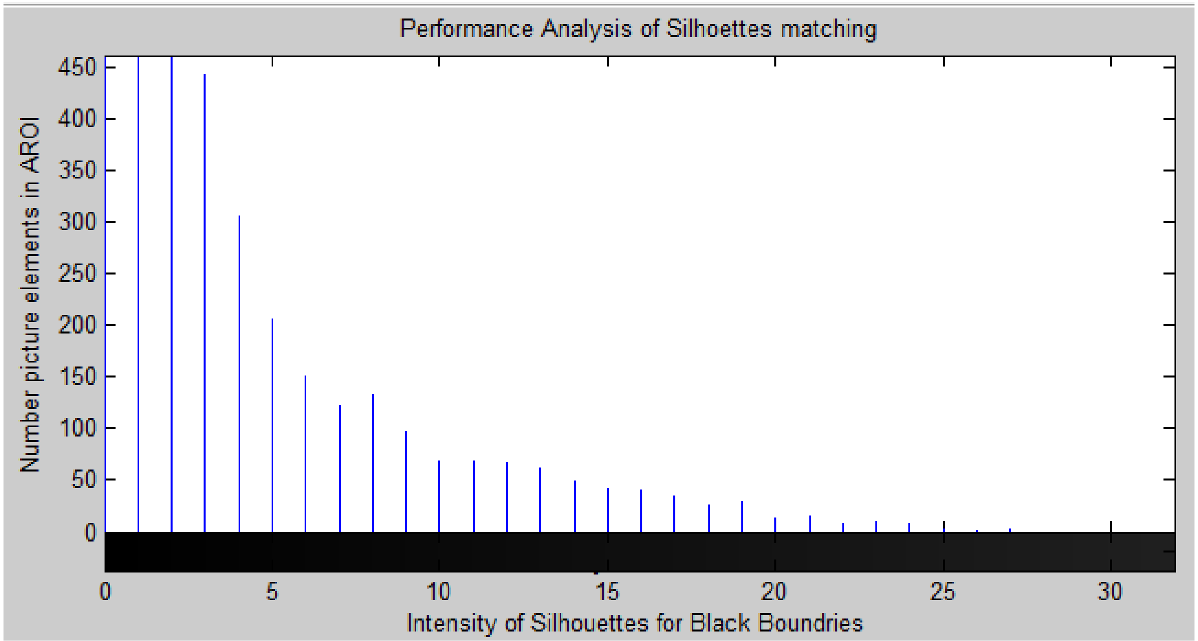

FIGURE 12: Intensity variations of black boundaries of silhouettes.

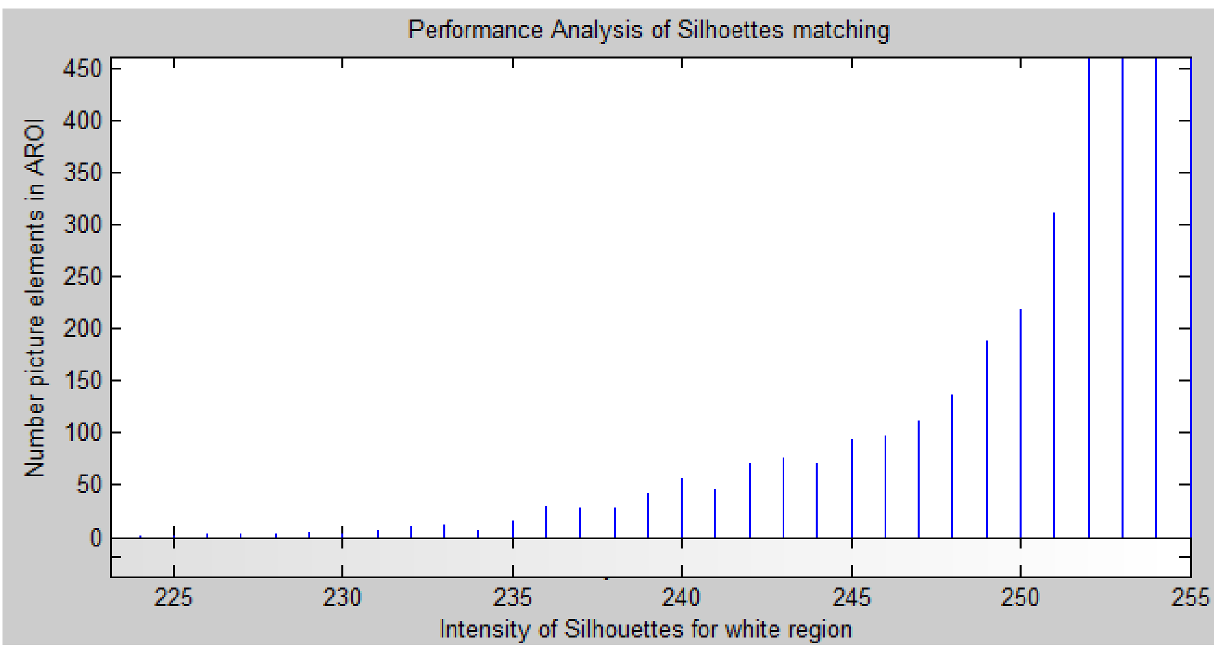

FIGURE 13: Intensity variations of white region of silhouettes.

The Mean Square Error (MSE) and the Peak-Signal-to-Noise Ratio (PSNR) are the two error metrics utilized to compare image quality. In our experiment, PSNR computes the ratio of decibels of noise in extracted silhouettes after invoking in a heterogeneous environment. It normally computes highest peak of the input signal of the extracted frame and noise embedded due to external sources in the resultant frame. This ratio is often used to evaluate the quality measurement of the performance of algorithm between a referenced frame and an output frame of sequence of images. The higher ratio of the PSNR is the better quality of the silhouettes. This is shown in Figure 14. It reveals that resultant frames in the initial states of silhouettes were approximately 27.8dB of PSNR and drops to approximately 19.8dB of focal point of silhouettes. This reveals that proposed algorithm produces better quality silhouettes within first four frames of computation.

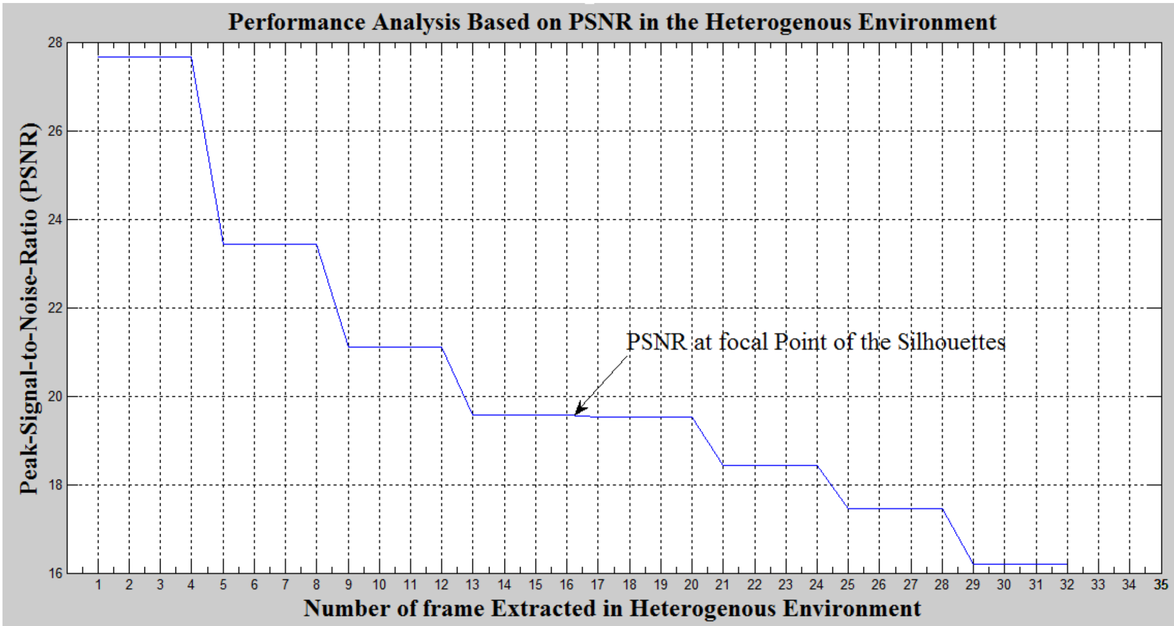

FIGURE 14: PSNR of silhouettes after extracting from the heterogeneous environment.

The MSE represents the cumulative squared error between the referenced and resultant frame, whereas PSNR represents a measure of the peak error. The lower the value of MSE is the lower the error of the silhouettes extraction algorithm. Figure 15 reveals that the first four frames have approximately 150 MSE on initially extracted silhouettes.

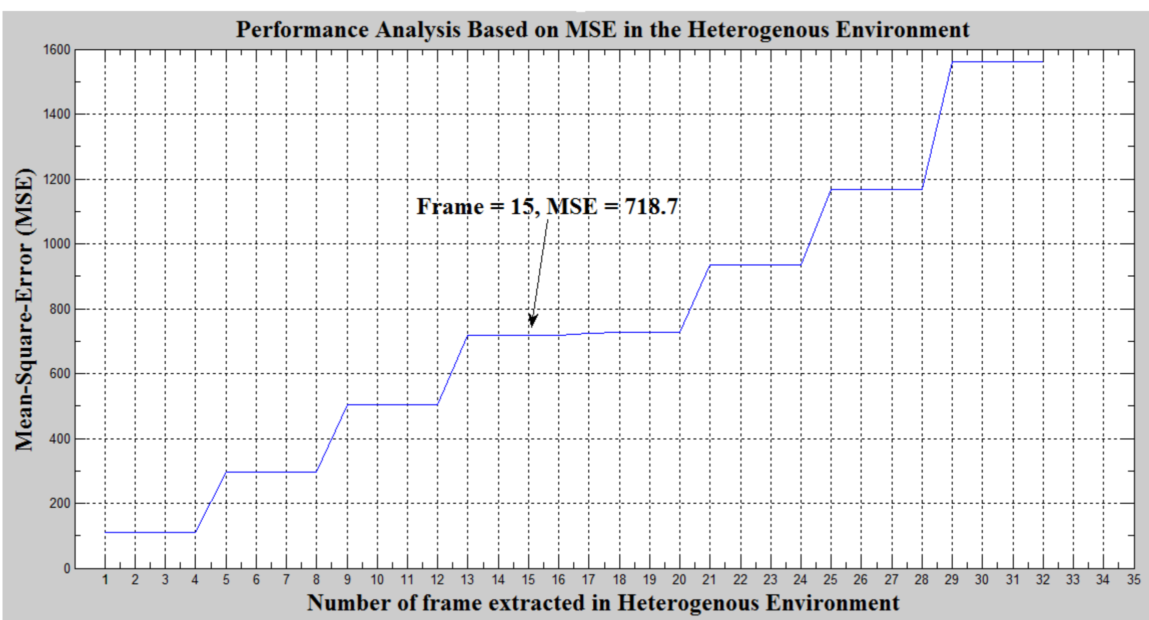

FIGURE 15: MSE of silhouettes after extracting from the heterogeneous environment.

This model reveals that the proposed heterogeneous environment algorithm will easily be deployed with existing traditional automated surveillance system in order to improve the performance over the internet oriented automated surveillance system. In addition, our proposed algorithm will incorporate in platform independent networks.

4.6 Comparative Study for F-Measure Computation

The proposed approach error rate computation was compared with wavelet based object starred approach. For that F-Measure error rate computation was taken. In this measure, the erroneous association of lost true positive pixels and the erroneous association of superfluous positive pixel were computed. In order to evaluate the correctness of both the foreground and background detection together, the overall performance of the object starred algorithms was compared with already existing method of object detection in moving frames. For this comparative study, 14 diverse video segments were chosen namely from A to N with different background and foreground objects. F-Measure was computed based on the error in pixel differences. The average error rate of proposed and existing method of F-Measure was observed and their respective errors were depicted in Figure 16. In Figure 16, X-label is depicted A to N video segments and Y-label has a various ranges of F-measures. It is observed that the proposed Haar starred algorithm detects the foreground objects which are chosen for tacking with less F-measure error rate. This is due to enhancement of the proposed approach in object detection for heterogeneous video formats adaptations in diverse circumstances and scenarios. The proposed approach achieved a minimum of 1.01 and maximum of 2.97 F-measure error rates whereas existing wavelet based starred approach minimum and maximum F-measure were 2.77 and 5.47, respectively. It is found that the proposed approach outperforms the existing wavelet based starred approach in terms of F-measure error rates. Furthermore, it is observed that existing wavelet approach could fail in heterogeneous video platform in terms of starred foreground objects.

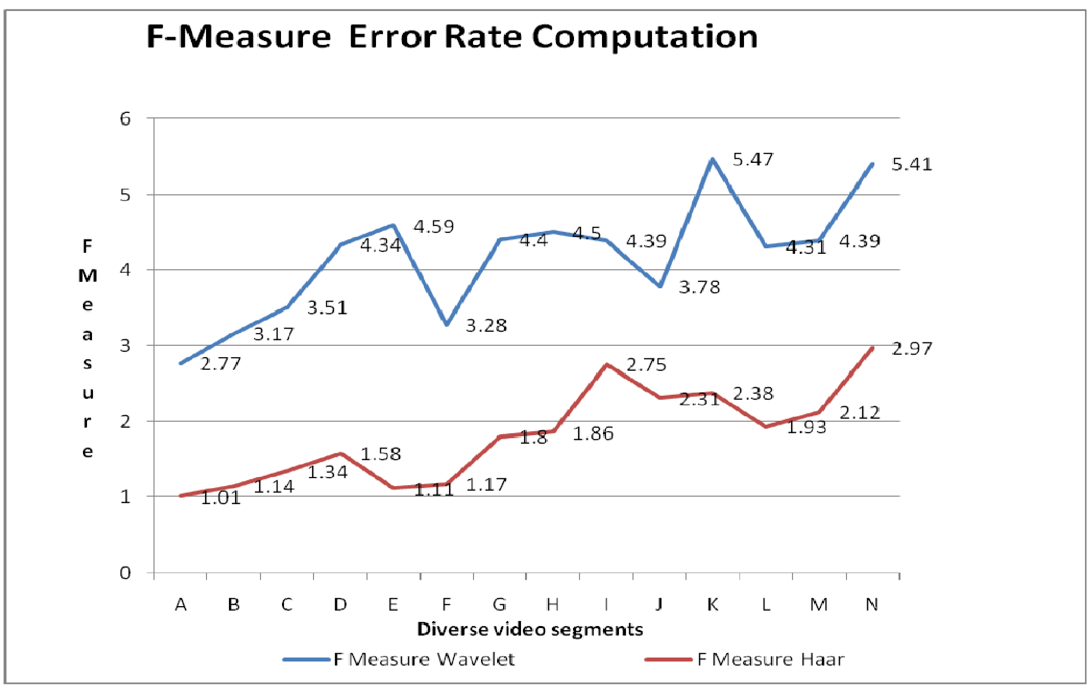

FIGURE 16: Comparative study of proposed approach of F measures, Haar with wavelet starred algorithms.

5. CONCLUSION AND FUTURE ENHANCEMENT

In this paper, we present a Java implementation based heterogeneous video surveillance monitoring in a static background. This is a generic smart video surveillance model in the platform independent environment. JMF is used to extract images from the video file and those images are then used for pre-processing. Binary silhouettes images are extracted and registered for matching process. Based on binary features, video is highlighted the moving object. Java Graphics classes are used for making starred on active region. The starred images are converted to video which observes the human movement.

In further research, a method will be suggested for synthesizing multiple threats and anomalous stimuli from several cameras to hypothesis the human characteristics from the pops up of silhouettes.

6. ACKNOWLEDGEMENT

Authors would like to acknowledge their student **Mr. Abdul Rahman B** for his effect to make front interface design and be grateful to monetary support provided by the Management, Sur University College, Sur, Oman.

7. REFERENCES

1. Collins R.T., A. J. Lipton, and T. Kanade, “A system for video surveillance and monitoring”, Carnegie Mellon University, Proceedings of DARPA Image Understanding Workshop, 2000, pp.1-68.

2. Comaniciu, D., Ramesh, V., Meer, P., “Kernel-Based object tracking”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.25, no. 5, pp. 564-577, 2003.

3. Gavrila, D. M., “The Visual Analysis of Human Movement: A Survey”, Computer Vision and Image Understanding, vol. 73, no. 1, pp 82-98, 1999.

4. Girisha R, and Murali S, “Tracking Humans using Novel Optical Flow algorithm for Surveillance Videos”, Proceedings of the Fourth Annual ACM Bangalore Conference Compute'11, Mar 25-26, Bangalore, India, no.7, 2011.

5. Haritaoglu, I.; Harwood, D.; Davis, L.S., “W4:Real-Time Surveillance of People and Their Activities”, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 22, no. 8, pp.809-830, August 2000.

6. Leon Rothkrantz, Zhenke Yang, Michael Jepson, Dragos Datcu, Pascal Wiggers, “A bottom-up approach of fusion of events in surveillance systems”, International Conference on Computer Systems and Technologies - CompSysTech’09, ACM, 2009, pp.III-A.3-1-3.6.

7. Milan Sonka, Vaclav Hlavac and Roger Boyle, “Image processing, analysis, and Machine Vision”, Second edition, ITP Press, USA, 1999, pp. 76-97.

8. Moeslund & Erik Granum, “A Survey of Computer Vision-Based Human Motion Captures”, Computer Vision and Image Understanding, vol. 81, pp. 231-268, 2001.

9. Sankari M, C. Meena, “Detecting Object From The Dynamic Background From The Noisy Image In Visual Surveillance”, Proc. of Int. Conf. on Advances in Recent Technologies in Communication and Computing 2011, Sep 14-15, 2011 © 2011 IET, UK, pp. 108-112.

10. Wang, L., W. Hu, T. Tan, “Recent developments in human motion analysis”, Pattern recognition, vol. 36, no.3, pp.585-601, 2003.

11. Chue-Poh Tan, Ka-Sing Lim and Weng-Kin Lai, “Multi-Dimensional Features Reduction of Consistency Subset Evaluator on Unsupervised Expectation Maximization Classifier for Imaging Surveillance Application”, International Journal of Image Processing, vol. 2, no. 1, pp.18-26, 2008.

12. Wai-Kit Wong, Zeh-Yang Chew, Hong-Liang Lim, Chu-Kiong Loo & Way-Soong Lim, "Omnidirectional Thermal Imaging Surveillance System Featuring Trespasser and Faint Detection", International Journal of Image Processing (IJIP), vol.4, no. 6, pp. 518-538, 2010.