CHAPTER 4

EXPERIMENT I: FIRST LEVEL - DATA LEVEL OF THE TRI-LEVEL UNIFIED FRAMEWORK

Data Level is the first level of abstraction of the Tri-level Unified Framework and it describes the processing of the multimedia content (videos) for useful information to be extracted and reduce the size of the storage. The aim of this level is to reduce a large amount of data and retain useful information for the next higher levels of abstraction that require more processing tasks. The research work presented in this thesis deals with the analyzing videos of human gait. The Phase I of human gait analysis described in Chapter 3 deals with human detection and tracking. This chapter presents the details of techniques that were investigated for extraction of silhouettes of the moving humans detected in video sequences. There are several representations used for human body shape such as blobs, stick figures, 2D contours, volumetric models, kinematic and appearance based representations but many applications require the use of highly accurate silhouettes of objects. Silhouettes give a total outline of the human body shape providing information like location, shape, number of pixels present in the extracted shape. Huang and Wu [105] presented the following advantages for using human silhouettes for pose estimation:

- Human silhouettes provide rich information based on shape
- The sequences of human silhouettes are independent of the motion speed
- Human silhouettes in binary form are robust to variations in clothing or lighting.

In most applications relating to human gait analysis, the initial step is to detect moving humans. This provides a focus of attention for future tasks hence the objective of detecting
and segmenting moving human figures from video sequences is to obtain relevant information about the human figures located in all the frames of video sequences. Even small errors may result in a significant impact on performance. motion segmentation methods are commonly used to segment moving objects in video. The objective of segmentation is to obtain relevant information about the human figures located in the frames of human motion. The methods presented in this chapter have been focused on one of the segmentation methods known as background subtraction method. Existing motion segmentation algorithms were extended, and a simple, effective algorithm is proposed for obtaining clean silhouettes. The goal of this chapter is three fold:

- Evaluate the merits and demerits of the investigated techniques
- Propose and compare the observations
- Determine which method suits the best

4.1. BASELINE APPROACH FOR BACKGROUND SUBTRACTION

Performing background subtraction results in the segmentation of an image into background and foreground pixels. The baseline approach used for extraction of the moving human figure from a video sequence using background subtraction consists of the following steps:

1. Maintain an adaptive model of the reference frame, usually called background model
2. Compare the current frame with the reference frame
3. Locate the foreground and extract the moving human figure by choosing an appropriate representation such as shape, kinematic structure, appearance based or 3D-representation.

A background model can be generated by a variety of methods. A potentially robust approach would be successful in the accurate extraction of moving human figures by handling changes in illumination and be sensitive enough to identify moving figures. The various methods investigated in this work include:
1. Adaptive background modeling using OTSU threshold method.
2. Simple background modeling on ‘V’ layer in HSV color space followed by morphological operations.
3. GMM background modeling in HSV color space with tracking.
4. Blockwise GMM background modeling on RGB color space.
5. Recurrent blockwise GMM background modeling on RGB color space with tracking.

4.2. DATASETS USED

Survey regarding human action recognition using video sources is reported in chapter 2. However, in this section, a short review of the datasets used for the purpose of extracting moving silhouette by the methods investigated is described.

Dataset # 1: Weizmann Dataset [106]

The Weizmann dataset videos contain sample recordings with different lengths in term of the number of frames recorded. Fig. 4.1 shows samples of actions found in a dataset of Weizmann. The video format used for experiments is in ‘.avi.’ format. A dataset of Weizmann is recorded from a still camera with a static background. The dataset contains 93 low-resolution (180 x 144, with a frame rate of 25 fps) video samples of human actions.

Fig. 4.1 Sample frames of Weizmann dataset
There are 10 different actions in this dataset such as bending, jumping, walking and one hand waving, jumping in place, running, gallop sideways, skip jumping and two hands waving. These actions were performed by 9 actors. Each actor performed each action once, except one actor (‘Lena”), who performed three of the actions (running, skip jumping and walking) twice. One action, the person, is moving left to right, and other is vice versa.

**Dataset # 2: i3DPostMulti-view Human Action Dataset [107]**

The database has been created using eight convergent cameras setup to produce high definition multi-view videos. Various types of motions are recorded, where each video depicts one of eight performing one of the twelve different human motions. The database consists of 832 single view videos- each of 120 to 125 frames and each frame with a resolution (320 x 240, with a frame rate of 25 fps). The 12 different actions consist of the walk, run, jump forward, jump in place, bend, one hand wave, sit down - stand up, walk – sit down, run – fall, run – jump – walk, two persons handshaking, one person pulls another. Fig. 4.2 shows a sample of actions found in the i3DPostMulti-view Human action dataset.

![Sample frames of i3DPost Multi-view Human action dataset](image)

**4.3. EXPERIMENTS AND RESULTS**

In this section, we present details of the investigated methods. The methods were implemented by slightly modifying the commonly implemented methods of running
average, single Gaussian and Gaussian Mixture Model. In the original approach, running average was implemented by user defined threshold value; the first method is demonstrated by using adaptive OTSU threshold for the running average method. The second approach converts the frames in the video sequence from RBG color space to HSV color space and exclusively from ‘V’ layer silhouette of the moving human figure is extracted and morphological operations – dilation and erosion are performed to refine the silhouettes. The original method by Stauffer and Grimson [108], GMM was implemented in RGB color space to handle complex moving backgrounds, however, the third method implements GMM in HSV color space along with tracking on human figures with bounding boxes. The fourth method consists of dividing the RGB frames into blocks with adaptive background modeling and consecutive frame differencing. The fifth method, Recurrent block wise GMM background modeling on RGB color space with tracking is a proposed method that is similar to the previous method but consists of intuitive frame differencing by selecting the frames.

The performances of all the investigated methods are evaluated on the processing time and the percentage of noise present in the extracted silhouettes of the frames. The investigated methods assumed that the silhouettes extracted using the proposed Recurrent blockwise GMM background modeling on RGB color space with tracking are 100% in performance.

4.3.1. Adaptive Background Modeling using OTSU Threshold Method

Using the baseline approach, a modification of running Gaussian average experiments were implemented. The threshold is automatically selected which helps in improving the computational efficiency. OTSU algorithm [109] is an automatic threshold selecting algorithm. The threshold is a value from the histogram of an image. It can be determined by the pixel's gray scale image and reduced a gray level image to a binary image. The method assumes that the image contains two classes of intensity value (e.g., foreground and background) then calculates the optimum threshold separating those two classes. The assumption is that all of the pixels can be classified into either class \( C_0 \) or \( C_1 \) by the intensity level of each pixel. This method is determined from the variance ratio of the class with total variance as equations below.
\[ \sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \]  
(4.1)
\[ \sigma_T^2 = \sum (i- \mu_T)^2 P_i, \ i = 1 \text{ to } N \]  
(4.2)
\[ \eta = \sigma_B^2 / \sigma_T^2 \]  
(4.3)

Where,
\( \sigma_B^2 \) is the between-class variance
\( \sigma_T^2 \) is the total variance of gray scale
\( P_i \) is the probability distributions
\( \omega_0 \) is the probability of accumulated occurrence
\( \omega_1 \)is the probability of accumulated mean level
\( \mu_1 \) is the first order cumulative moments of histograms
\( \mu_0 \) is the zeroth order cumulative moments of histograms
\( \mu_T \) is the total mean level of the original image
\( \eta \) is the criteria that are used to discriminate

**Major Steps in Adaptive Background Modeling using OTSU Threshold Method**

Step 1: Preprocessing: Normalize all frames
\[ r' = r/255, \ g' = g/255, \ b' = b/255 \]  
(4.4)
Where \( r, g, b \) represent the red, green, blue color layers and \( r', g', b' \) are the normalized

Step 2: Differencing of frames: Extract motion pixels by computing the difference between the incoming current frame and background model.
\[ \text{Diff}_{t+1} = |I_t - B_{t-1}| \]  
(4.5)
Where \( I_t \) is the pixel value of current frame and \( B_{t-1} \) is the pixel value of the background estimate. The background pixel is assumed to be a single Gaussian \((\mu, \sigma)\).

Step 3: Modeling: Initially obtain single Gaussian of all pixels of the first frame and use it as the reference frame and update the next consecutive frames as
\[ B_{t+1} = \alpha I_t + (1-\alpha)B_{t-1} \]  
(4.6)
Where, \( \alpha \) is the learning speed of the background. (The value of \( \alpha \) is chosen to be 0.5)
Step 4: The differenced image is threshold by OTSU [111], is an adaptive threshold that is obtained recursively by taking the variance of the Gaussian model.

**Results of Silhouette Extraction using Adaptive Background Modeling using OTSU Threshold Method**

Following Fig. 4.3 and Fig. 4.4 show the results obtained from the sample frames of the Weizmann video sequences of jumping and bending. All the samples demonstrate the adaptive algorithm backgroung modeling using OTSU threshold method.

![Frame 43 of 43](image1)

(a) Frame 43 of 43 video sequence of jumping type of gait,

![Adaptive Background](image2)

(b) With adaptive background,

![Binarized Difference Image](image3)

(c) Binarized difference image

Fig. 4.3 (a) A frame from Weizmann dataset video sequence of jumping type of gait,

(b) With adaptive background, (c) Binarized difference image
Fig. 4.4 (a) A frame from Weizmann video dataset sequence of bending type of gait, (b) With adaptive background, (c) Binarized difference image

It is observed that the silhouette extraction was successful, however, the algorithm, Adaptive background modeling using OTSU threshold method for the low resolution video sequences from Weizmann video was poor.

Fig. 4.5 and Fig. 4.6 show the results obtained from the sample frames of i3DPost Multi-view human action of walking and falling. All the samples demonstrate the adaptive modeling algorithm using OTSU threshold method.
Fig. 4.5 (a) A frame from i3DPost Multi-view Human action dataset video sequence of walking type of gait, (b) With adaptive background, (c) Binarized difference image

It is observed that the silhouette extraction was successful, however, the algorithm, Adaptive background modeling using OTSU threshold method for the high resolution video sequences from i3DPost Multi-view Human action dataset video sequence video was better than the video sequences of Weizmann. There was 40% noise present in the frames. Tracking was not implemented using this algorithm.

Thus, the depiction of the outcome of the silhouettes for minimal representation using Adaptive background modeling using OTSU threshold method is not recommended for next phase of work of human gaits analysis.
4.3.2. Simple Background Modeling on ‘V’ Layer in HSV Color Space followed by Morphological Operations

In this method, all the frames in a selected video sequence are initially converted from RGB color space to HSV color space and a background model is generated using the single Gaussian approach on the ‘v’ layer. The difference between the incoming frame and the reference frame is calculated by a Mahalanobis [110] distance and this distance is threshold by a pre-defined value (Usually 30). Then, on the extracted binary silhouettes morphological operations of dilation and erosion are performed to obtain refined silhouettes.
**Major Steps of Simple Background Modeling on ‘V’ Layer in HSV Color Space followed by Morphological Operations**

**Step 1:** Preprocessing: normalize the frames
\[ r' = r/255, \ g' = g/255, \ b' = b/255 \]
same as (4.4)

Where \( r, g, b \) represent the red, green, blue color layers and \( r', g', b' \) are the normalized

**Step 2:** Convert RGB to HSV color space.

**Step 3:** Background modeling: Extract the ‘v’ layer and construct background model using one Gaussian (1-G) [111].

Model each pixel with a Gaussian distribution \( \eta(\mu_{s,t}, \Sigma_{s,t}) \)

Where, \( \mu_{s,t} \) average background color and \( \Sigma_{s,t} \) Covariance matrix

**Step 4:** Silhouette extraction: calculate Mahalanobis distance between pixels of reference frame and the incoming frame

Mahalanobis distance: \( d_M = |I_{s,t} - \mu_{s,t}|\Sigma_{s,t}^{-1}|I_{s,t} - \mu_{s,t}|^T \) (4.7)

Where, \( I_{s,t} \), \( \mu_{s,t} \) are HSV vectors and \( \Sigma_{s,t} \) is the covariance matrix.

**Step 5:** Compare the distance to a threshold value 30.

**Step 6:** Apply morphological operations- erosion and dilation to obtain better silhouettes

**Step 7:** Update mean and covariance matrix of each pixel of the background model

\[ \mu_{s,t+1} = (1 - \alpha)\mu_{s,t} + \alpha I_{s,t} \]  
(4.8)

\[ \Sigma_{s,t+1} = (1 - \alpha)\Sigma_{s,t} + \alpha(I_{s,t} - \mu_{s,t})(I_{s,t} - \mu_{s,t})^T \]  
(4.9)

**Results of Extraction in Simple Background Modeling on ‘V’ Layer in HSV Color Space followed by Morphological Operations**

Following Fig 4.7, Fig. 4.8 and Fig. 4.9 are the results which are obtained from the sample frames of the Weizmann dataset video sequence walking. Initially, the RGB frames are converted to HSV color space and are separated into three layers H, S and V. In Fig. 4.9(c), the value layer consists of stable information. The algorithm, Simple Background Modeling on ‘V’ Layer in HSV Color Space followed by Morphological Operation was implemented with a simple background modeling on ‘V’ layer in HSV color space followed by morphological operations.
The following figure, Fig 4.10, shows the result of the extracted silhouettes using the. It is observed that pseudo silhouettes were present along with the original silhouettes. This method could not implement tracking. More than 40% of noise was present in the frames.
4.3.3. Adaptive GMM Background Modeling in HSV Space with Tracking

In this experiment, we combine the advantages of both detection and tracking in a single framework. The approximate articulation of each person is detected in every frame based on the model of the appearance of an individual person. We present experimental results that demonstrate how this method allows detection and tracking of a single person using the static camera in the HSV color space. The video sequences of i3DPost Multi-view human action dataset were used to implement, Adaptive GMM background modeling in HSV space with tracking method.
**Major Steps in Adaptive GMM Background Modeling in HSV Space with Tracking Method**

**Step 1:** Preprocessing: normalize the frames

\[ r' = r/255, \ g' = g/255, \ b' = b/255 \]

same as \( (4.4) \)

Where \( r, g, b \) represent the red, green, blue color layers and \( r', g', b' \) are the normalized

**Step 2:** Convert RGB to HSV color space.

**Step 3:** Background modeling: Construct background model for every pixel using Stauffer and Grimson [113] mixture of \( k \) Gaussians. Thus the probability of occurrence of a color at a given pixel is given by:

\[ P(I_{s,t}) = \sum \omega_{i,s,t}.\eta(I_{s,t}, \mu_{i,s,t}, \Sigma_{i,s,t}) \]  

(4.10)

Where \( \eta(I_{s,t}, \mu_{i,s,t}, \Sigma_{i,s,t}) \) is the \( i^{th} \) Gaussian model and \( \omega_{i,s,t} \) its weight. The covariance matrix \( \Sigma_{i,s,t} \) is diagonal. (Usually \( I_{s,t} \) is within 2.5 standard deviation of its mean)

**Step 4:** Parameters of matched component are updated as follows:

\[ \omega_{i,s,t} = (1 - \alpha) \omega_{i,s,t-1} + \alpha \]  

(4.11)

\[ \mu_{i,s,t} = (1 - \rho)\mu_{i,s,t-1} + \rho I_{s,t} \]  

(4.12)

\[ \sigma_{i,s,t}^2 = (1 - \rho)\sigma_{i,s,t-1}^2 + \rho(I_{i,s,t} - \mu_{i,s,t})^2 \]  

(4.13)

Where \( \alpha \) is the learning rate given by user usually taken as 0.0005 and \( \rho \) is the second learning rate given as \( \rho = \alpha \eta(I_{s,t}, \mu_{i,s,t}, \Sigma_{i,s,t}) \)

**Step 5:** Once the Gaussian has been updated, the \( K \) distributions are normalized so they sum up to 1. The \( K \) distributions are ordered based on fitness value \( \omega_{i,s,t}\sigma_{i,s,t} \) and only the \( H \) most reliable are chosen as part of the background

\[ H = \text{argmin}_h(\sum_\omega > T) \]  

(4.14)

Where, \( T \) is the threshold. Those pixels which are more than 2.5 standard deviations away from \( H \) distributions are labeled “in motion” and are tracked by bounding box.
Results of Extraction and Tracking using Adaptive GMM Background Modeling in HSV Space with Tracking

The results of extraction and tracking were based on different actions such as walking, falling, sitting and jumping from the i3DPost Multi-view Human action dataset video as seen in Fig. 4.11, Fig. 4.12, Fig. 4.13, Fig. 4.14 and Fig. 4.15.

Fig. 4.11 Original Frame from the video

Fig. 4.12 Extracted Silhouette from the video frame
Fig. 4.13 Extraction and tracking of human during falling

Fig. 4.14 Tracking of human during sitting

Fig. 4.15 Tracking of human during jumping
The silhouettes obtained by Adaptive GMM Background Modeling in HSV space with tracking were not binarised silhouettes. Though tracking was implemented successfully, the extracted silhouettes are not suitable for minimal representation that can be used for next level of analysis.

4.3.4. Blockwise GMM Background Modeling on RGB Color Space (BGMM)

In this method, each frame is partitioned into blocks each of 8 x 8 sizes and 1-G Background modeling is used, and the difference between frames is performed using the Mahalanobis’ distance. The threshold of 50% tolerance is selected, and boundaries are detected by setting the moving pixels to 1, non-moving to 0. Consecutive frames are differenced to obtain the moving humans. This process is repeated for all frames across the video sequences.

Major Steps of Blockwise GMM Background Modeling on RGB Color Space (BGMM)

Step 1: Preprocessing: normalize the frames
\[ r’ = r/255, \quad g’ = g/255, \quad b’ = b/255 \] same as (4.4)

Step 2: Background modeling: Initially, on the first frame, construct background model using one Gaussian (1-G) [113] and set it as the reference frame. Model each pixel with a Gaussian distribution \( \eta (\mu_{s,t}, \Sigma_{s,t}) \). Where, \( \mu_{s,t} \) average background color and \( \Sigma_{s,t} \) Covariance matrix

Step 3: Silhouette extraction: Divide the both the reference frame T1 and the current frame T2 into blocks (Block size = 8). Calculate Mahalanobis distance between pixels of reference frame and the incoming frame for all the blocks.

Mahalanobis distance: \( d_M = |I_{s,t} - \mu_{s,t}| \Sigma_{s,t}^{-1} | I_{s,t} - \mu_{s,t}|^T \) (4.15)

Where, \( I_{s,t}, \mu_{s,t} \)are RGB vectors and \( \Sigma_{s,t} \) is the covariance matrix.

Step 4: \( D = |T1 - T2| \) (4.16)

Step 5: Compare D to a threshold given by
\[ Th = (\max((\max(D))) / 1.3 \] (with 50% tolerance) (4.17)

Step 6: Select all the moving and non-moving pixels in the current frame and set non-moving to 0 and moving to 1
Step 7: Select the updated current for new mean and covariance matrix of each pixel of the background model

\[
\begin{align*}
\mu_{s,t+1} &= (1 - \alpha)\mu_{s,t} + \alpha I_{s,t} \\
\Sigma_{s,t+1} &= (1 - \alpha)\Sigma_{s,t} + \alpha(I_{s,t} - \mu_{s,t})(I_{s,t} - \mu_{s,t})^T
\end{align*}
\] (4.18) (4.19)

Results of Extraction using Blockwise GMM background modeling on RGB color space (BGMM)

Fig. 4.16, Fig. 4.17, Fig. 4.18, Fig. 4.19 show the results of extraction using Blockwise GMM background modeling on RGB color space (BGMM) and were demonstrated on different actions such as walking, jumping, bending and running from video sequences taken from the Weizmann dataset. The extracted silhouettes were desirable but noisy.
4.3.5. Recurrent Blockwise GMM Background Modeling on RGB Color Space with Tracking (RBGMM)

In this proposed method, background modeling is constructed on every pixel in RGB color space using blockwise partitions. Instead, of consecutive frames differencing, the only frame with less similarity is selected to obtain the moving humans. Finally, tracking using centroid points over silhouettes is obtained. This representation gives silhouette data invariant to noise.

**Major Steps in Recurrent Blockwise GMM Background Modeling on RGB Color Space with Tracking (RBGMM)**

Step 1: Preprocessing: normalize the frames
\[ r' = r/255, \quad g' = g/255, \quad b' = b/255 \]
same as (4.4)

Step 2: Background modeling: Initially, on the first frame, construct background model using one Gaussian (1-G) [113] and set it as the reference frame. Model each pixel with a Gaussian distribution \( \eta(\mu_{s,t}, \Sigma_{s,t}) \)

Where, \( \mu_{s,t} \) average background color and \( \Sigma_{s,t} \) Covariance matrix

Step 3: Silhouette extraction: Divide the both the reference frame \( T_1 \) and the current frame \( T_2 \) into blocks (Block size = 8).

Step 4: Calculate Mahalanobis distance between pixels of the reference frame and the incoming frame for all the blocks recurrently.

Mahalanobis distance:
\[
d_M = |I_{s,t} - \mu_{s,t}| \Sigma_{s,t}^{-1} |I_{s,t} - \mu_{s,t}|^T
\]

Where, \( I_{s,t}, \mu_{s,t} \) are RGB vectors and \( \Sigma_{s,t} \) is the covariance matrix.

Step 5:
\[
D = |T_1 - T_2|
\]

Step 6: Select all the moving and non-moving pixels in the current frame and set non-moving to 0 and moving to 1
Step 7: To update T2 for next iteration, each iteration compares D to the similarity of at least 30 matches in the next five consecutive frames. Select only those frames with > 30 matches and set that frame as next current frame.

Step 8: Update mean and covariance matrix of each pixel of the background model

\[
\mu_{s,t+1} = (1 - \alpha)\mu_{s,t} + \alpha I_{s,t}
\]

(4.22)

\[
\Sigma_{s,t+1} = (1 - \alpha)\Sigma_{s,t} + \alpha(I_{s,t} - \mu_{s,t})(I_{s,t} - \mu_{s,t})^T
\]

(4.23)

Step 9: After extraction of silhouettes all of them are aligned and tracked by their centroid coordinates.

Results of extraction in Recurrent Blockwise GMM Background Modeling on RGB Color Space with Tracking (RBGMM)

The results of extraction are seen in Fig. 4.20, Fig. 4.21, Fig. 4.22, and Fig. 4.23 were demonstrated on different actions such as, walking, jumping, bending and running from video sequences taken from the Weizmann dataset. The extracted silhouettes were desirable. The advantage of using this method is that the 30 matches for each set of consecutive frames selected only silhouettes with necessary poses that are used for further processing.

Fig. 4.20 Sequence of walking using RBGMM

Fig. 4.21 Sequence of jumping using RBGMM
4.4. DISCUSSION AND ANALYSIS

4.4.1. Discussion on the Investigated Methods

The adaptive background modeling using OTSU threshold method yielded binarized difference images, but the results obtained were of vast variation. The binarized images of Weizmann consisting of low resolution images yielded poor results compared to high resolution images of i3DPost Multi-view Human action dataset. Also, there were many holes in the extracted human figures which will have to look for improvement.

The second method was implemented by converting the frames of video sequences from RGB to HSV, and the ‘V’ layer was extracted where the illumination variances were found to be stabilized. From the ‘V’ layer the silhouettes were extracted using GMM method and later refined with the morphological operations. Though the silhouettes were generated, it was observed that Pseudo silhouettes were generated which were difficult to remove.

The third method was implemented using GMM in HSV color space for human detection and tracking. This method was not useful for the low resolution video sequences of...
Weizmann however for the high resolution images of i3DPost Multi-view Human action dataset, humans were detected and were tracked with bounding boxes across the video sequence. The results obtained were not binary silhouettes but were of appearance based.

The fourth method was implemented by dividing the all the RGB frames in the video sequence into blocks. By applying single Gaussian background modeling along with threshold, moving silhouettes was extracted, and the resultant silhouettes were noisy.

The fifth method is a proposed method that improvised upon the previous method by intuitively selecting the next frame which is less similar to the current frame and recurrently all the selected frames were processed with noise threshold and finally, the silhouettes extracted were with almost nil noise. The proposed method was found to be the best on both low as well as high resolution video sequences.

4.4.2. Analysis on Performance of the Investigated Methods

As a comparative study of segmenting human gait from the video sequences using the methods discussed, the methods were evaluated and compared in terms of speed of execution and noise levels found in Weizmann and i3DPost multi-view human action datasets. Research in this thesis is limited to using static camera sources but included low resolution and high resolution as well as single and multi-view angles datasets. The figures, in Fig. 4.24, Fig. 4.25, Fig. 4.26 and Fig. 2.27 show the show the processing time taken for different gait related actions such as walking, jumping, running and bending using different methods that were used to extract the silhouettes.

Fig. 4.24 shows the processing time taken for silhouette extraction of walking pose using the different methods investigated. The processing time required for Walking pose extraction using the proposed Recurrent Blockwise GMM background modeling on RGB Color Space with tracking method shows minimum processing time whereas, Blockwise GMM background modeling on RGB Color Space without tracking method shows the maximum time taken for processing.
Fig. 4.24 Processing time comparison for walking

Similarly, Fig. 4.25, shows the processing time taken for silhouette extraction of bending pose using the different methods investigated. The processing time required for bending pose extraction using the proposed Recurrent Blockwise GMM background modeling on RGB Color Space with tracking method shows minimum processing time whereas, Blockwise GMM background modeling on RGB Color Space without tracking method shows the maximum time taken for processing.
1. Adaptive background modeling using OTSU threshold method.
2. Simple background modeling on 'V' layer in HSV color space followed by morphological operations.
3. GMM background modeling in HSV color space with tracking.
4. Blockwise GMM background modeling on RGB color space without tracking.
5. Recurrent blockwise GMM background modeling on RGB color space with tracking.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive background</td>
<td>5.6</td>
</tr>
<tr>
<td>OTSU threshold method.</td>
<td></td>
</tr>
<tr>
<td>Simple background modeling on 'V' layer</td>
<td>6.8</td>
</tr>
<tr>
<td>in HSV color space followed by morphological</td>
<td></td>
</tr>
<tr>
<td>operations.</td>
<td></td>
</tr>
<tr>
<td>GMM background modeling in HSV</td>
<td>5.8</td>
</tr>
<tr>
<td>color space with tracking.</td>
<td></td>
</tr>
<tr>
<td>Blockwise GMM background modeling on RGB</td>
<td>7.2</td>
</tr>
<tr>
<td>color space without tracking.</td>
<td></td>
</tr>
<tr>
<td>Recurrent blockwise GMM background modeling</td>
<td>4.1</td>
</tr>
<tr>
<td>on RGB color space with tracking.</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 4.25 Processing time comparison for bending

Fig. 4.26 shows the processing time taken for silhouette extraction of running pose using the different methods investigated. The processing time required for running pose extraction using the proposed Recurrent Blockwise GMM background modeling on RGB Color Space with tracking method shows minimum processing time whereas, Blockwise GMM background modeling on RGB Color Space without tracking method shows the maximum time taken for processing.
Fig. 4.26 Processing time comparison for running

Fig. 4.27 shows the processing time taken for silhouette extraction of jumping pose using the different methods investigated. The processing time required for jumping pose extraction using the proposed Recurrent Blockwise GMM background modeling on RGB Color Space with tracking method shows minimum processing time whereas, Blockwise GMM background modeling on RGB Color Space without tracking method shows maximum time taken for processing.
1. Adaptive background modeling using OTSU threshold method.
2. Simple background modeling on ‘V’ layer in HSV color space followed by morphological operations.
3. GMM background modeling in HSV color space with tracking.
4. Block wise GMM background modeling on RGB color space without tracking.
5. Recurrent block wise GMM background modeling on RGB color space with tracking.

The accuracy of segmentation is measured by comparison of the results obtained by various methods to the probe silhouettes using the qualitative comparisons of best similarity and calculating the percentage of noise levels present according to the performance of methods. The probe silhouettes of various gait poses used for comparison are taken from the results generated by the proposed method, Recurrent Blockwise GMM Background Modeling with tracking (RBGMM).
<table>
<thead>
<tr>
<th>Methods/Actions</th>
<th>Walking (%)</th>
<th>Jumping (%)</th>
<th>Running (%)</th>
<th>Bending (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running average with OTSU</td>
<td>40</td>
<td>45</td>
<td>44</td>
<td>55</td>
</tr>
<tr>
<td>HSV segmentation with morphological operations</td>
<td>35</td>
<td>40</td>
<td>38</td>
<td>45</td>
</tr>
<tr>
<td>GMM in HSV color space with tracking</td>
<td>30</td>
<td>20</td>
<td>25</td>
<td>15</td>
</tr>
<tr>
<td>Blockwise GMM Background Modeling</td>
<td>25</td>
<td>32</td>
<td>25</td>
<td>50</td>
</tr>
<tr>
<td>Recurrent Blockwise GMM Background Modeling and tracking</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 4.1 Percentage of noise level present in the silhouettes extracted using various methods

4.5. CONCLUSION

This chapter demonstrated silhouette extraction of human gaits found in two datasets of Weizmann and i3DPost multi-view human action. The investigated methods were discussed in detail. Existing motion segmentation algorithms were modified, and the proposed Recurrent Gaussian Mixture Model method (RBGMM) is found to be performing well for both low as well as high resolution video sequences.

The task defined by the first level, Data level of the Tri-level Unified framework is a minimal representation, and the corresponding phase I specified for analysis of human gaits is human detection and tracking. The task was implemented by extracting silhouettes. Silhouettes require minimum storage.