Chapter 3 : IMPROVED APPROXIMATE MEDIAN FILTER BASED METHOD FOR MOVING OBJECT SEGMENTATION

In recent past, many moving object segmentation methods under varying lighting changes have been proposed in literature and each of them has their own benefits and limitations. The various methods available in literature for moving object segmentation may be broadly classified into four categories i.e. moving object segmentation methods based on (i) motion information (ii) motion and spatial information (iii) learning, and (iv) change detection. The objective of this chapter is two-fold i.e. firstly, this chapter presents a comprehensive comparative study of various classical as well as state-of-the art methods for moving object segmentation under varying illumination conditions under each of the above mentioned four categories and secondly, this chapter presents an improved approximation filter based method in complex wavelet domain and its comparison with other methods under four categories mentioned as above. The proposed approach consist of seven steps applied on given video frames which include: wavelet decomposition of frames using Daubechies complex wavelet transform; use of improved approximate median filter on detail co-efficient (LH, HL, HH); use of background modeling on approximate co-efficient (LL sub-band); soft thresholding for noise removal; strong edge detection; inverse wavelet transformation for reconstruction; and finally using closing morphology operator. The qualitative and quantitative comparative study of the various methods under four categories as well as the proposed method is presented for six different datasets. The merits, demerits, and efficacy of each of the methods under consideration have been examined. The extensive experimental comparative analysis on six different challenging benchmark data sets demonstrate that proposed method is performing better to other state-of-the-art moving object segmentation methods and is well capable of dealing with various limitations of existing methods.

3.1. Introduction

Moving object detection is a crucial part of automatic video surveillance systems and it is useful in robotics, object detection and recognition, indoor/outdoor object classification and many other applications [23, 24]. To design the moving object segmentation algorithm for intelligent video surveillance systems, several major challenges have to be concerned. Toyama *et al.* [25] have identified the following challenges in moving object segmentation such as (i) lighting changes, shadows and reflections (ii) dynamic backgrounds such as waterfalls or waving trees (iii) Motionless foreground (iv) small movements of non-static objects such as tree branches and bushes blowing in the wind (v) noise image, due to a poor quality image source (vi) movements of objects in the background that leave parts of it different from the background model (ghost regions in the image) (vii) multiple objects moving in the scene both for long and short periods (viii) shadow regions that are projected by foreground objects and are detected as moving objects. Out of all these issues, changing illumination conditions remain a major problem for moving object segmentation in real-life problems. To take into account these problems, many approaches for automatically adapting background model to dynamic scene variations are proposed [90, 91] and these approaches can be classified into two categories [92] such as non-recursive and recursive. A non-recursive approach uses a sliding-window for background estimation. It stores a buffer of the previous L video frames, and estimates the background image based on the temporal variation of each pixel within the buffer. This causes non-recursive approach to have higher memory requirements than recursive techniques. Recursive approach maintains a single background model that is updated with each new video frame. These approaches are generally computationally efficient and have minimal memory requirements.

The major contributions of this chapter include: (1) comparative study of various standard moving object segmentation methods which is classified into four categories i.e. moving object segmentation methods based on (i) motion information (ii) motion and spatial information (iii) learning (iv) and change detection (2) proposed an improved approximation filter based approach for moving object segmentation in complex wavelet domain (3) and presented the comparative study of the proposed method with other state-of -the-art algorithms on a set of challenging video sequences (4) analysis of the sensitivity of the most influencing parameters [84-87], and a discussion of their effects. (5) and analysis of the computational complexity and memory consumption of the proposed algorithm.

Rest of the chapter is organized as follows: Section 3.2 presents the Review of moving object segmentation methods. Section 3.3 presents the proposed method. Experimental results are given in Section 3.4. Finally, conclusion of the work is given in Section 3.5.

3.2. Review of Moving Object Segmentation Methods

Different kinds of methods exist to solve the problem of moving object segmentation. Good but incomplete reviews on moving object segmentation methods can be found in [93, 94]. As per available literatures moving object segmentation techniques can be broadly classified into four categories [95, 96, 97] namely (i) segmentation of moving object based on motion-information [33-34, 96, 98-103], (ii) segmentation of moving object based on motion and spatial information [27-28, 31-32, 104-109], (iii) segmentation of moving object based on learning [26-30, 110-115], and (iv) segmentation of moving object based on change detection [16, 35, 36-37, 116-125]. A review of some of the classical and state-of-the-art methods under each of the categories is presented in following subsections.

3.2.1. Moving Object Segmentation Methods Based on Motion-Information

The first category of moving object segmentation methods are based on motion-information which depends on motion estimation of moving objects. Some of the prominent methods available in literature are the works due to Bradski [98], Kim *et al*. [33], Liu *et al.* [34], Xiaoyan *et al.* [96], Mahmoodi [99] and Meier and Ngan [100]. Bradski [98] proposed a motion segmentation method using time motion history image (TMHI) for representing motion which is used to segment and measure the motions induced by the object in a video scene. The limitation of the method is that, it can only extract the moving objects but not the static one. A more refined application of this algorithm was proposed by Kim *et al.* [33] which is based on codebook approach where a codebook is formed to represent significant states in the background using quantization and clustering [33]. It solves some of the above mentioned problems existing in [98], such as sudden changes in illumination, but does not consider the problems of ghost regions or shadow detection. To deal with the issues mentioned in [33], Liu *et al.* [34] have proposed a moving object segmentation method which is based on cumulated difference, object motion and adaptive thresholding. Xiaoyan *et al.* [96] have proposed a video object segmentation technique on the basis of adaptive change. This method is not able to remove noise from the video frames. Mahmoodi [99] has proposed a shape based active contour method for video segmentation which is based on a piecewise constant approximation of the Mumford shah functional model. This method is slow beacause it is based on level set framework. Due to lack of spatial information of objects, these algorithms suffer from unwarranted ghost objects, shadows, changing background, clutter, occlusion, and varying lighting conditions. Meier and Ngan [100] have proposed a moving object segmentation which is based on Hausdorff distance. In this method, a background model is created which automatically adapts slowly and rapidly changing parts and matched against subsequent frames using the Hausdorff distance. The limitation of this method is that the boundaries of the extracted objects are not always accurate. In addition to above mentioned methods, in literature some other approaches [101-103] in the same domain have been proposed but they also suffer from most of the same problems mentioned as above.

Therefore the important features of the methods under the category moving object segmentation methods based on motion-information can be summarized as follows:

- The motion information based moving object segmentation methods [33-34, 96, 98-103] are fast and usually easy to implement.
- Motion information based moving object segmentation methods handle well the background changes but are not robust to sudden illumination changes.
- Furthermore, they are likely to fail if the contrast between the moving objects and the background is low.

3.2.2. Moving Object Segmentation Methods Based on Motion and Spatial Information

The second category of moving object segmentation methods are based on both motion and spatial information. The segmentation of moving objects based on motion and spatial information provide more stable object boundary extraction. Some of the prominent works under this domain are the works due to Mei *et al.* [28], Mcfarlane and Schofield [27], Remagnino *et al.* [104], Wren *et al.* [31], Zivkovic [32], Reza *et al.* [105], and Ivanov *et al.* [106]. In paper [28], Mei *et al.* proposed an automatic segmentation method for moving objects based on the spatial-temporal information of video. In this method, the author utilizes the spatial-temporal information. Spatial segmentation is applied to divide each image into connected areas to find precise object boundaries of moving objects. The limitation of this method is that the boundaries of the extracted objects are not always accurate enough to locate them in different scenes. Mcfarlane and Schofield [27] have proposed an approximation median filter method for segmentation of multiple video objects. This technique has also been used in background modeling for urban traffic monitoring [104]. The major disadvantage of this method is that it needs many frames to learn the new background region revealed by an object that moves away after being stationary for a long time [92] but this method is computationally efficient. Wren *et al.* [31] have proposed Running Gaussian Average model for moving object segmentation. This model is based on Gaussian probability density function (pdf) where a running average and standard deviation are maintained for each color channel. The drawback of this method lies in its complex nature which makes its processing slow because of the computational overhead involved in updating the mixture models. To deal with the issues mentioned in [31], Zivkovic [32] have proposed a moving object segmentation technique which is combination of temporal and spatial features. This approach automatically adapts the number of Gaussians being used to model for a given pixel. Reza *et al.* [105] have proposed a moving object segmentation technique, combining temporal and spatial features. This approach takes into account a current frame, ten preceding frames and ten next consecutive frames to segment the moving object. The method detects moving objects independent of their size and speed but there is no provision for reduction of blur and noise from frames, which may lead to inaccurate object segmentation. Ivanov *et al.* [106] have proposed an improvement over background subtraction method, which is faster than that proposed by [105] and is invariant to runtime change illuminations. In addition to above mentioned methods there are many other works reported in literature [107-109] under this second category but most of them suffers from the similar types of limitations associated with above mentioned methods.

Therefore the important features of the methods under the category moving object segmentation methods based on motion- and spatial information can be summarized as follows:

- The motion and spatial information based moving object segmentation methods [27-28, 31-32, 104-109] needs many frames to learn the new background region revealed by an object that moves away after being stationary for a long time [92].
- Motion and spatial information based moving object segmentation methods is adaptive to only the small and gradual changes in the background and in case of sudden changes it distorts.
- Computational complexity of spatial information based moving object segmentation methods is also very low.

3.2.3. Moving Object Segmentation Methods Based on Learning

The Third category of moving object segmentation methods are based on learning which depends on some predefined learning patterns. Some of the prominent methods available in literature are the works due to Oliver *et al.* [110], Cucchiara *et al.* [111], Kushwaha *et al.* [26], Kato *et al.* [112], Ellis *et al.* [30], and Stauffer *et al.* [29]. Oliver *et al.* [110] proposed a moving object segmentation method which is based on spatial correlations. In this method, author constructs the background using principal component analysis. But it's suffered the problem of noise and blur. To deal the issue mention in [110], Cucchiara *et al.* [111] have proposed a moving object segmentation technique which is based on medoid filtering that can lead to color background estimation. The medoid filtering is capable of saving boundaries and existing edges in the frame without any blurring. But the computational complexity to construct the background is high. A more refined application of this algorithm proposed by Kushwaha *et al.* [26] which is based on construction of basic background model where in the variance and covariance of pixels are computed to construct the model for scene background which is adaptive to the dynamically changing background. The method described in [26] has the capability to relearn the background to adapt background changes. Kato *et al.* [112] have proposed a segmentation method for monitoring of traffic video based on Hidden Markov Model (HMM). In this method, each pixel or region is classified into three categories: shadow, foreground and background. This method comprises of two phases: learning phase and segmentation phase. Ellis *et al.* [30] have proposed online segmentation of moving objects in video using online learning. In this approach, motion segmentation is done using semi-supervised appearance learning task wherein supervising labels are autonomously generated by a motion segmentation algorithm but the computational complexity of this algorithm is very high. Stauffer *et al.* [29] have proposed a tracking method wherein motion segmentation was done using mixture of Gaussians and on-line approximation to update the model. This model has some disadvantages such as background having fast variations cannot be accurately modeled with just a few Gaussians (usually 3 to 5), causing problems for sensitive detection. In addition to above mentioned methods, in literature various other approaches [113-115] in the same domain have been proposed but they also suffer from most of the same problems mentioned as above.

Therefore the important features of the methods under the category moving object segmentation methods based on learning information can be summarized as follows:

- Learning based moving object segmentation methods [26, 29-30, 110-112, 113-115] are adaptive to the dynamically changing background.

- Computational complexity of learning based moving object segmentation methods is very high.
- Learning based moving object segmentation methods suffer the problem of shadow regions and the presence of ghosts like appearances.

3.2.4. Moving Object Segmentation Methods Based on Change Detection

The fourth category of moving object segmentation methods are based on change detection which depends on frame difference of two or more frames. Some of the prominent methods available in literature are the works due to Kim *et al.* [116], Chien *et al.* [117], Kim and Hwang [118], Shih *et al.* [119], Huang *et al.* [16, 35], Baradarani [36, 37, Hsia *et al.* [120], Khare *et al.* [121]. Kim *et al.* [116] proposed moving object segmentation and automatic object tracking approach for video sequences. In this approach, intra-frame and inter-frame segmentation modules are used for segmentation and tracking. The intra-frame segmentation incorporates the user interaction in defining a high level semantic object of interest to be segmented and detects precise object boundary. The inter-frame segmentation involves boundary and region tracking to capture temporal coherence of moving objects with accurate object boundary information. The drawback of this method is that user-interaction is required for separating moving objects from the background in video sequences. To deal with the issues mentioned in [116], Chien *et al.* [117] proposed moving object segmentation algorithm using background registration method. The background registration method is used to construct reliable background information from the video sequence. In this approach, a morphological gradient operation is used to filter out the shadow. The major disadvantage of this method is that it adapts only static background and suffers from the problem of ghost objects. Kim and Hwang [118] derive an edge map using change detection method and after removing edge points which belong to the previous frame, the remaining edge map is used to extract the video object plane. This method suffers from the problem of object distortion. To solve this problem, Shih *et al.* [119] used change detection method in three adjacent frames which easily handles the new appearance of the moving object. Huang *et al.* [16, 35] proposed an algorithm for moving object segmentation to solve the double-edge problem in the spatial domain using a change detection method with different thresholds in four wavelet sub-bands. Baradarani [36, 37] refined the work of Huang *et al.* [16, 35] using dual tree complex filter bank in wavelet domain. These methods [36, 37] suffer from the problem of noise disturbances and distortion of moving segmented objects due to change in speed of objects. To concern these issues, Hsia *et al.* [120] proposed a Modified Directional Lifting-based 9 /7 Discrete Wavelet Transform (MDLDWT) based approach, which is based on the coefficient of Lifting-based 9/7 Discrete Wavelet Transform (LDWT). Its advantages of low critical path, fast computational speed and the LL3-band of the MDLDWT is employed solely to reduce the image transform computing cost and remove noise but it cannot handle large dynamic background changes. Khare *et al.* [121] refine the work of Baradarani [36, 37] and Huang *et al.* [16, 35] using Daubechies complex wavelet. The method proposed by Khare *et al.* [121] reduces the noise disturbance and speed change, but it suffers from the problem of dynamic background changes and shadow detection and due to this segmenting coherence occurs [122]. In addition to above mentioned methods, in literature various other approaches [123-125] in the same domain have been proposed but they also suffer from most of the same problems mentioned as above.

Therefore the important features of the methods under the category moving object segmentation methods based on change detection [16, 35, 36-37, 116-125] can be summarized as follows:

- Change detection based moving object segmentation methods [16, 35, 36-37, 116-125] are adaptive to detect only "significant" changes while rejecting "unimportant" ones.
- Change detection based moving object segmentation methods [16, 35, 36-37, 116-125] handle noise disturbance and speed change very well.
- Change detection based moving object segmentation methods [16, 35, 36-37, 116-125] suffer from the problem of either slow speed of moving object or abrupt lighting variation changes.
- The other limitations include shadow regions, detection of only moving objects, and the presence of ghosts like appearances.

Table 3.1 presents the summary of various moving object segmentation methods under above mentioned four categories. The brief description of methods, their advantages, limitations, and conclusions of each category are highlighted. For comparative analysis purposes, only few prominent and latest methods in each category are considered which are performing better in their peer groups as reported in literature and demonstrated in results and analysis section.

Table 3.1: Summary of various moving object segmentation methods

Categories of Methods	Methods	Brief Description	Advantages	Limitations	Conclusions
Category I (Methods based on motion-information)	Bradski [98]	Use time motion history image (TMHI) for representing motion	Computationally Fast	Extracts only the moving objects but unable to extract the static one	Motion information based methods are fast and usually easy to implement. They handle well the background changes but are not robust to sudden illumination changes. Furthermore, they are likely to fail if the contrast between the moving objects and the
	Kim *et al*.[33]	Use quantization and clustering [13] for creating a codebook	Handle gradual illumination changes and computationally Fast	Have problems of ghost regions, shadow detection and noise	
	Liu *et al.* [34]	Use cumulated difference, object motion and adaptive thresholding	Handle ghost and noise problem	Problem to handle sudden illumination changes and shadow	
	Xiaoyan *et al.* [96]	Use adaptive change detection, Canny edge	Easily handle complex background and noise problem	Have problems of ghost regions and shadow	

		and improved active contour to obtain the segmented object			background is low.
	Mahmoodi [99]	Use shape based active contour and Mumford shah functional model	Easily handles the shadow problem	Computationally slow due to use of level set framework	
	Meier and Ngan [100]	Use Hausdorff distance for background modelling	Automatically adapts with the changing background	The boundaries of the extracted objects are not always accurate.	
Category II (Methods based on motion and spatial – information)	Mei *et al.* [28]	Use spatial-temporal information	Easily handle gradual illumination changes and shadow problem	Boundaries of the extracted objects are not always accurate enough to locate them in different scenes	Motion & spatial information based methods need many frames to learn the new background region revealed by an object
	Mcfarlane *et al.*[27]	Use frame differencing and	Computationally fast and handles noise	It needs many frames to learn the new	

<table>
<tr><td rowspan="4"></td><td></td><td>Background modeling step</td><td></td><td>background region revealed by an object that moves away after being stationary for a long time</td><td rowspan="4">that moves away after being stationary for a long time [6] i.e. it is adaptive to only the small and gradual changes in the background in case of sudden changes it distorts and also the computational complexity of these algorithms are low.</td></tr>
<tr><td>Wren et al. [31]</td><td>Use running Gaussian average model and Gaussian probability density function</td><td>Handle gradual and sudden illumination changes</td><td>Complex nature which makes its processing slow because of the computational overhead involved in updating the mixture models.</td></tr>
<tr><td>Zivkovic [32]</td><td>Use number of Gaussians to create a model for a given pixel.</td><td>Easily handle gradual and sudden illumination changes</td><td>Computationally slow and also suffers from the problem of ghost and shadow</td></tr>
<tr><td>Reza et al. [105]</td><td>Use current frame, ten</td><td>Easily handle lighting</td><td>Computationally slow and have</td></tr>
</table>

		preceding frames and ten next consecutive frames to segment the moving object	changes, shadows and reflections	problem to handle sudden illumination changes	
	Ivanov *et al.* [106]	Use color intensity values at corresponding pixels	Computationally Fast and also handle noise, shadow problem	Problem to handle gradual and sudden illumination changes and ghost problem	
Category III (Methods based on learning patterns)	Oliver *et al.* [110]	Use spatial correlations and principal component analysis to construct the background	Easily handle gradual illumination changes and shadow problem	Computationally slow and Problem of noise and blur	Learning based methods are adaptive to the dynamically changing background but the computational complexity of these algorithm is very high
	Cucchiara *et al.* [111]	Use medoid filtering to construct the background	Easily handle gradual variations of the lighting	Computational complexity to construct the background is high	

			conditions in the scene		and they also suffer with the problem of shadow regions and the presence of ghosts like appearances.
	Kushwaha *et al.* [26]	Use variance and covariance of pixels to construct the background	Easily handle gradual and sudden variations of the lighting conditions in the scene, computationally fast and reduce the noise problem	Suffers from the problem of shadow and ghost object in the scene	
	Ellis *et al.* [30]	Use semi-supervised appearance learning task	Easily handle gradual variations of the lighting conditions in the scene	Computational complexity to construct the background is high	
	Stauffer *et al.* [29]	Use mixture of Gaussians and on-line approximation to update the model	Easily handle gradual and sudden variations of the lighting	Computational complexity to construct the background is high and also the suffer the	

			conditions in the scene	problem of noise and ghost object	
	Kato *et al.* [112]	Use Hidden Markov Model (HMM) to segment and learn the object	Easily handle gradual and sudden variations of the lighting conditions in the scene and also solve the problem of shadow, noise	Computational complexity to construct the background is high	
Category IV (Methods based on change detection)	Kim *et al.* [116]	intra-frame and inter-frame segmentation modules are used for objects segmentation and tracking	It detects accurate object boundaries	User-interaction is required for separating moving objects from the background in video sequences.	Change detection based methods can handle appearance of new objects in the scene. But they suffer from the problem of either slow speed of
	Chien *et al.* [117]	Background registration is used to construct	Easily handles object shadow and noise	1. it adapts only static background	

		reliable background information from the video sequence		2. it suffers with the problem of ghost objects	moving object or abrupt lighting variation changes. .
	Kim and Hwang [118]	Use single change detection	Easily handles the noise	Problem to handle new appearance of object in the scene	
	Shih *et al.* [119]	Use double change detection	Easily handles new appearance of object in the scene	Problem to handle gradual and sudden illumination changes due to this object is distorted and also suffers from noise and ghost object appearance	
	Huang *et al.* [16, 35]	Use single & double change detection in wavelet domain	Easily handle new appearance of object in the scene &	Problem to handle gradual and sudden illumination	

			computationally fast	changes due to this object is distorted also suffers noise and ghost object problem	
	Baradarani [36, 37]	Use change detection in dual tree complex wavelet domain	Easily handle new appearance of object in the scene and computationally fast	Suffer from the problem of noise disturbances and distortion of moving segmented objects due to change in speed of objects	
	Hsia *et al.* [120]	Use modified directional lifting-based 9 /7 discrete wavelet transform (MDLDWT)	Reduce the image transform computing cost, remove noise , and computationally fast	Problem to handle gradual and sudden illumination changes and due to this the object is distorted	
	Khare *et al.* [121]	Use single change	Reduces the noise	Suffers from the problem of	

		detection in Daubechies complex wavelet domain	disturbance and speed change	dynamic background changes and shadow detection and due to this segmenting coherence occurs	

After presenting the literature review of various moving object segmentation methods, discussed as above under each of the four categories, it is observed that the approximate median filter based method under second category i.e. a method based on motion and spatial information is better in comparison to methods presented in other categories also validated through experimental results and analysis presented in Section 3.4. The approximate median filter contains two steps to segment the object: (i) frame differencing of two consecutive frames and (ii) background modeling step. The brief working of approximation median filter based method for moving object segmentation is given as follows [27, 104]:-

Step I: Frame Differencing:

For background subtraction the frame difference $FD_n(i,j)$ is obtained by taken the absolute difference two consecutive frames (n-1) & n. This process can be written as follows:-

For every pixel location (i, j) ϵ the co-ordinate of frame

$$FD_n(i,j) = |f_n(i,j) - f_{n-1}(i,j)|$$
$$\text{If } FD_n(i,j) < V_{thr}$$
$$FD_n(i,j) = 0$$

Step II: Background Modeling:

In background modeling step, if the corresponding pixel in the current frame $f_n(i,j)$ is greater in value of previous frame $f_{n-1}(i,j)$ then previous frame is incremented by one otherwise previous frame is decreased by one. This process can be written as follows:-

$$\begin{aligned}&If\ (f_n(i,j) > f_{n-1}(i,j))\\&then \quad f_{n-1}(i,j) = f_{n-1}(i,j)+1\\&otherwise \ \ f_{n-1}(i,j) = f_{n-1}(i,j)-1\end{aligned}$$

Here, $f_n(i,j)$ is the value of $(i, j)^{th}$ pixel of n^{th} frame and $f_{n-1}(i,j)$ is the value of $(i, j)^{th}$ pixel of $(n-1)^{th}$ frame, V_{thr} is a threshold value and $FD_n(i,j)$ is the frames difference.

The main limitation of approximate median filter based method is that it does not adapt to the dynamic changes in background due to its weak background modeling steps. Due to this it suffers from the problems of (i) ghost like appearances in moving segmented object, (ii) slow adaptation toward a large change in background, and (iii) requirement of many frames to learn the new background region revealed by an object that moves away after being stationary for a long time.

Motivated by these facts, in this chapter, we have improved the background modeling step of traditional approximate median filter based method [27, 104] using different major changes such as background registration, background differencing, and background difference mask in complex wavelet domain. These major changes adapt the dynamic background changes and solve the above mentioned three problems in traditional approximate median filter. The effectiveness of the proposed method over traditional approximate median filter is validated through experimental result and analysis presented in section 3.4.

The main advantage of performing the above mentioned tasks in the complex wavelet domain is that the complex wavelet transform has better noise resilience nature as the lower frequency sub-band of the wavelet transform has the capability of a low-pass filter. The other advantage is that the high frequency sub-bands of complex wavelet transform represent the edge information that provide a strong cue to handle shadow. The proposed method is well capable of dealing with the problems of noise, ghost like appearances, distortion of objects due to the speed of moving objects, dynamic background scenes, varying illumination conditions, shadows, and computational complexity as demonstrated and reported in this chapter for several challenging test video sequences.

3.3. An Improved Approximation Median Filter Based Approach in Complex Wavelet Domain: The Proposed Method

In this chapter, an efficient approach for moving object segmentation under varying illumination conditions is proposed. The proposed method is the modified and extended version of traditional approximation median filter based method for moving object segmentation [27, 104] in complex wavelet domain as discussed in section 3.2. The proposed method consists of following seven steps as follows and also illustrated in Fig. 3.1:

(i) Complex wavelet decomposition of sequence of frames.

(ii) Application of approximate median filter on the wavelet coefficients.

(iii) Application of background modeling.

(iv) Application of soft thresholding for noise removal.

(v) Application of canny edge detector to detect strong edges.

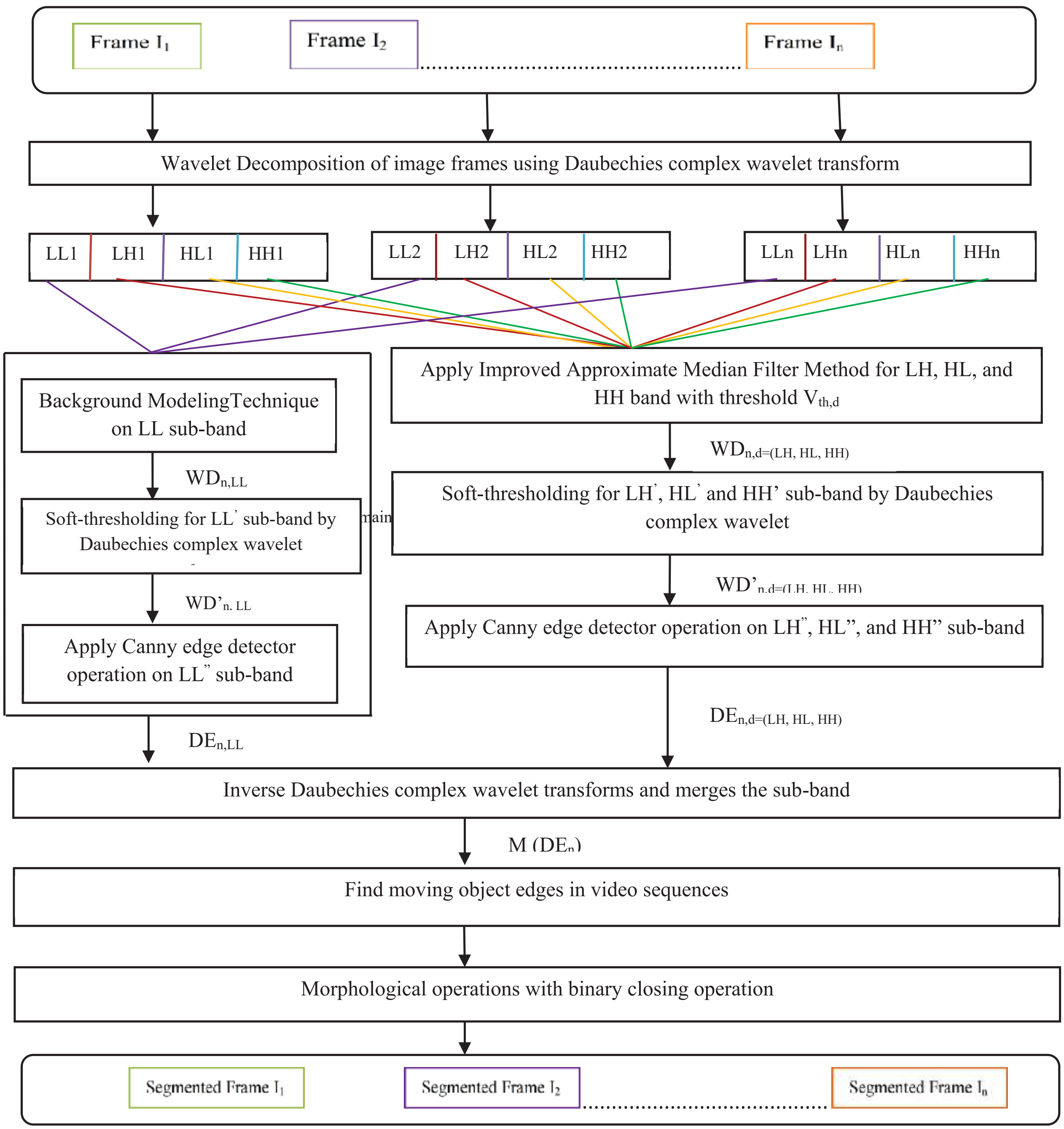
(vi) Application of inverse Daubechies complex wavelet transform.

(vii) Finally the application of closing morphological operators.

All above steps are iteratively applied until the result does not surpass the set threshold value for object segmentation.

The workings of these steps are given as follows and illustrated in Fig. 3.1 & 3.2

.



Segmented Object

Figure 3.1: Block Diagram of the Proposed Method

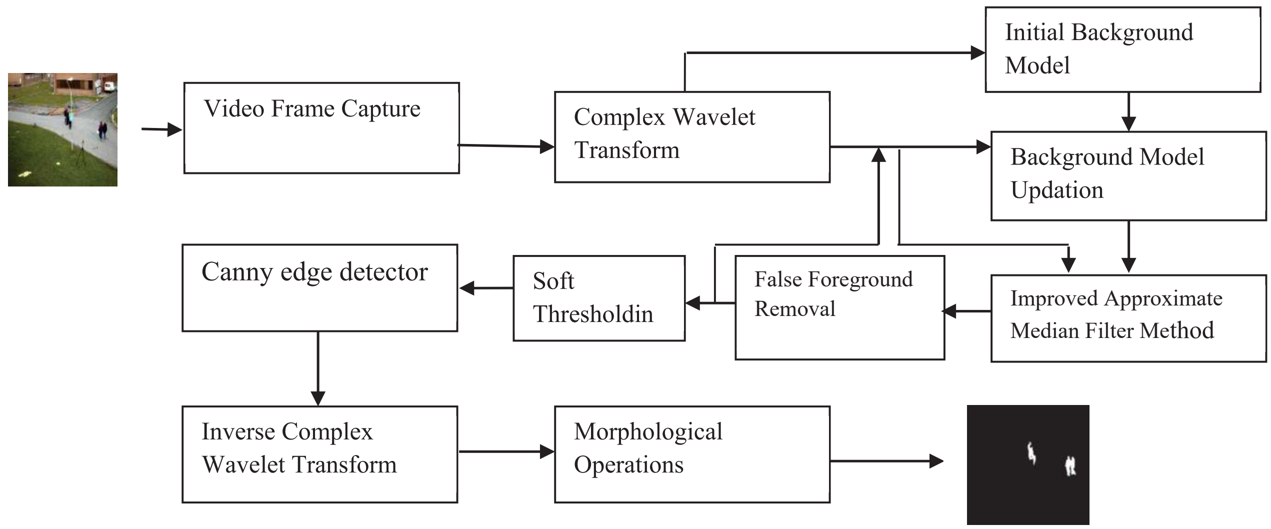


Figure 3.2: Sub-block Diagram of the proposed approach

Step 1: Wavelet Decomposition of frames

In the proposed approach, a 2-D Daubechies complex wavelet transform is applied on current frame and previous frame to get wavelet coefficients in four sub-bands: LL, LH, HL and HH. The generating Daubechies complex wavelet transform is described as follows:

The basic equation of multiresolution theory is the scaling equation [4]

$$\phi(u)=2\sum_{i} a_i \phi(2u-i) \tag{3.1}$$

where a_i's are coefficients, and $\phi(u)$ is the scaling function. The a_i's can be real as well as complex valued. Daubechies's wavelet bases $\{\psi_{j,k}(t)\}$ in one-dimension is defined using the above mentioned scaling function $\phi(u)$ and multi resolution analysis of $L_2(\Re)$ [4]. The generating wavelet $\psi(t)$ is defined as:

$$\psi(t)=2\sum(-1)^n \overline{a_{1-n}} \phi(2t-n) \tag{3.2}$$

Where $\phi(t)$ and $\psi(t)$ share same compact support [-L, L+1].

Any function f (t) can be decomposed into complex scaling function and mother wavelet as:

$$f(t) = \sum_k C_k^{j_o} \phi_{j_o,k}(t) + \sum_{j=j}^{j_{\max}-1} d_k^j \psi_{j,k}(t) \quad (3.3)$$

where, j_o is a given low resolution level, $\{C_k^{j_o}\}$ is called approximation coefficient and $\{d_k^j\}$ is known as detail coefficient.

Applying the approximate median filter based method [27, 104] in complex wavelet domain have following advantages (a) it is shift invariant and have a better directional selectivity as compared to real valued wavelet transforms [4] (b) it has perfect reconstruction property (c) it provides true phase information [4], while other complex wavelet transform does not provide true phase information (d) Daubechies complex wavelet transform has no redundancy [4].

Step 2: Application of improved approximate median filter method on wavelet co-efficient

In step 2, an approximate median filter based method is applied on detail wavelet coefficients i.e. on sub-bands: LH, HL, and HH. Let $Wf_{n,d}(i,j)(d=\{LH,HL,HH\})$ and $Wf_{n-1,d}(i,j)(d=\{LH,HL,HH\})$ are the wavelet coefficients at location (i, j) of the current frame and previous frame. Instead of assigning a fixed *a priori* threshold $V_{th,d}$ to each frame difference, this method uses the fast Euler number computation technique [126] to automatically determine $V_{th,d}$ from the video frame. The fast Euler numbers algorithm calculates the Euler number for every possible threshold with a single raster of the frame difference image using following equation:

$$E(i) = \frac{1}{4}[(q_1(i) - q_3(i) - 2q_d(i))] \quad (3.4)$$

where q_1 , q_3, and q_d is the quads (quad is a 2*2 masks of bit cells) contained in the given image.

The output of the algorithm is an array of Euler numbers: one of each threshold value. The Zero crossings find out the optimal threshold. Detailed algorithms for the fast Euler number computation method can be found in [126].

The wavelet domain frame difference $WD_{n,d}(i, j)$ for respective sub-bands are computed as:

for every pixel location (i, j) ϵ the co-ordinate of frame

$$WD_{n,d}(i, j) = \begin{cases} 1 & if \quad \left| Wf_{n,d}(i, j) - Wf_{n-1,d}(i, j) \right| > V_{th,d} \\ 0 & otherwise \end{cases} \tag{3.5}$$

Step 3: Application of background modeling using LL sub-band

This step of the proposed method deals with the problems of slow adaptiveness toward a large change in background and requirement of many frames to learn the new background region revealed by an object that moves away after being stationary for a long time as noted in traditional approximate median filter based method [27, 104]. To deal with these issues, here we propose to modify the background modeling approach which uses background registration mask, background difference mask and the frame difference mask to construct the background in LL sub band. The background modeling step is divided in to four major steps as shown in Fig 3.3.

The first step calculates the frame difference mask $WD_{n,LL}(i, j)$ of the LL image which is obtained by thresholding the difference between coefficients in two LL sub-bands as follows:

$$WD_{n,LL}(i, j) = \begin{cases} 1 & if \quad \left| Wf_{n,LL}(i, j) - Wf_{n-1,LL}(i, j) \right| < V_{th,WD} \\ 0 & otherwise \end{cases} \tag{3.6}$$

where $V_{th,FD}$ is a threshold of $WD_{n,LL}(i,j)$ determined automatically from the video frame by the fast Euler number computation method as explained in [126]. If $WD_{n,LL}(i,j)=0$, then the difference between two frames is almost the same.

The second step of background modeling maintains an up-to-date background buffer as well as background registration mask indicating whether the background information of a pixel is available or not. According to the frame difference mask of the past several frames, pixels that are not moving for a long time are considered as reliable background. The reliable background, $BR_{n,LL}(i,j)$ is defined as

$$BR_{n,LL}(i,j)=\begin{cases} BR_{n-1,LL}(i,j)+1 & if \quad WD_{n,LL}(i,j)=0 \\ 0 & otherwise \end{cases} \tag{3.7}$$

The $BR_{n,LL}(i,j)$ value is accumulated until $WD_{n,LL}(i,j)$ holds zero value. At any time that $WD_{n,LL}(i,j)$ is changed from 0 to 1, $BR_{n,LL}(i,j)$ becomes zero.

In third step of background modeling, if the value in $BR_{n,LL}(i,j)$ exceeds a predefined value, denoted by L, then the background difference masks $BD_{n,LL}(i,j)$ is calculated. It is obtained by taking the difference between the current frame and the background information stored. This background difference mask is the primary information for object shape generation i.e.

$$BD_{n,LL}(i,j)=\begin{cases} 1 & if \quad \left|Bf_{n-1,LL}(i,j)-Wf_{n,LL}(i,j)\right|>V_{th,BD} \\ 0 & otherwise \end{cases} \tag{3.8}$$

where $Bf_{n-1,LL}(i,j)$ is the pixel value in the current frame that is copied to the corresponding pixel in the $BR_{n,LL}(i,j)$, and $V_{th,BD}$ is a threshold value determined automatically from the video frame by the fast Euler number computation method as explained in [126]. In the case of $BR_{n,LL}(i,j)<L$, it is assumed that the background is

not constructed, so frame differences mask $WD_{n,LL}(i, j)$ is used which is calculated in the first step.

In the fourth step of background modeling, a background model is constructed using the background difference mask, background registration mask, and the frame difference mask. The background model generated has some noise regions because of irregular object motion and noise. Also, the boundary region may not be very smooth. The workings of these steps are given as follows and illustrated in Fig 3.3.

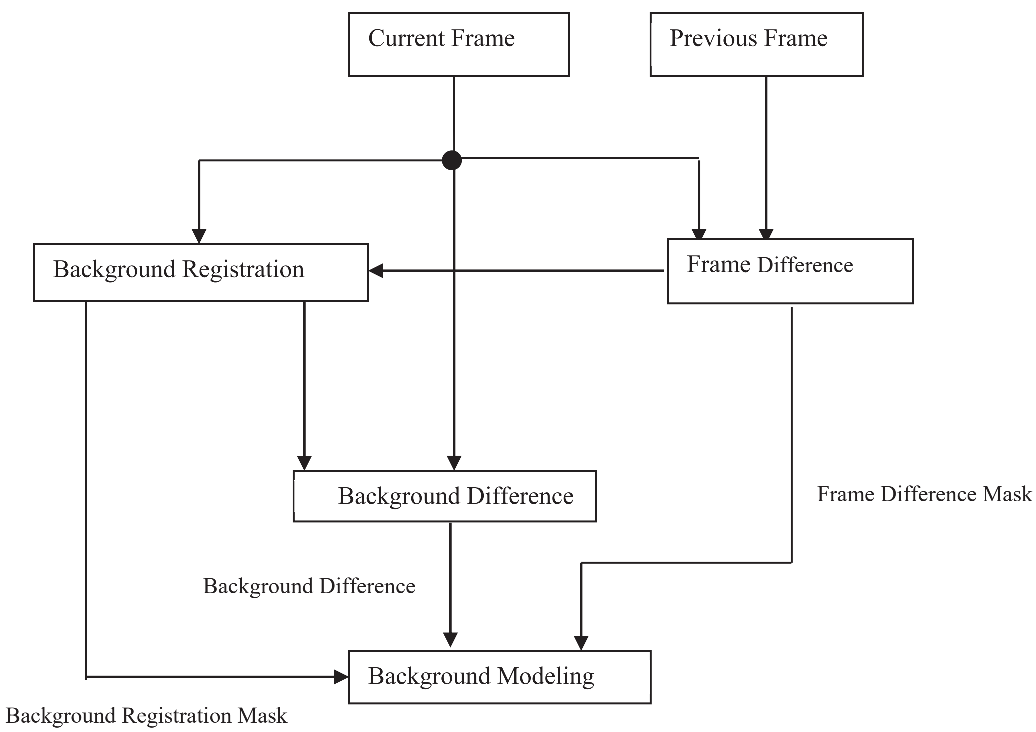


Figure 3.3: Block Diagram of the Background Modeling in LL sub-band

Step 4: Application of soft thresholding method for noise removal

After applying approximate median filter based method and background modeling, the obtained result may have noise. This step deals with the noise reduction from the data obtained in step 2 and step 3. In presence of noise, the equation is expressed as:

$$WD_{n,d=(LL,LH,HL,HH)}(i,j) = WD^{*}_{n,d=(LL,LH,HL,HH)}(i,j) + \eta \tag{3.9}$$

where $WD^*_{n,d=(LL,LH,HL,HH)}(i,j)$ is frame difference without noise, $WD_{n,d=(LL,LH,HL,HH)}(i,j)$ is the original frame difference with noise, and η is the additive noise. The wavelet domain soft thresholding T is applied on wavelet coefficients for noise reduction. The value of soft thresholding parameter T for de-noising is computed as [127]

$$T = \frac{1}{2^{j-1}}\left(\frac{\psi}{\xi}\right)\omega \tag{3.10}$$

where j is wavelet decomposition level and ψ , ξ and ω are standard deviation, absolute mean and absolute median of wavelet coefficients of a sub-band.

Step 5: Application of canny edge detector to detect strong edges in wavelet domain

Canny edge detection method is one of the most useful and popular edge detection methods, because of its low error rate well localized edge points and single edge detection response [128]. In next step, the canny edge detection operator is applied on $WD^*_{n,d=(LL,LH,HL,HH)}(i,j)$ to detect the edges of significant difference pixels in all sub-bands as follows:

$$DE_{n,d=(LL,LH,HL,HH)}(i,j) = canny(WD^*_{n,d=(LL,LH,HL,HH)}(i,j)) \tag{3.11}$$

where $DE_{n,d=(LL,LH,HL,HH)}(i,j)$ is an edge map of $WD^*_{n,d=(LL,LH,HL,HH)}(i,j)$.

Step 6: Application of inverse Daubechies complex wavelet transform

After finding edge map $DE_{n,d=(LL,LH,HL,HH)}(i,j)$ in wavelet domain, inverse wavelet transform is applied to get moving object edges in spatial domain i.e. E_n.

Step 7: Application of closing morphological operation to sub-band

As a result of step 6, the obtained segmented object may include a number of disconnected edges due to non-ideal segmentation of moving object edges. Extractions of object using these disconnected edges may lead to inaccurate object segmentation. Therefore, some morphological operation is needed for post-processing of object edge map to generate connected edges. Here, a binary closing morphological operation is used [128] which gives $M(E_n)$ i.e. the set of connected edge. In this step, the segmented output is obtained.

3.4. Experimental Results and Comparative Studies

3.4.1. Dataset Description

In this section, a brief overview of few datasets used for experimentation purpose in this chapter are presented.

Pets Dataset [129]

First video dataset used for experimentation in this chapter is the people video sequence which is part of Pets dataset available from [129]. This video data contains 2967 frames of frame size 480 x 272. The main characteristics of this video data are that they are record in outdoor environment wherein multiple objects (Human beings and cars) are present and cases of partial and full occlusions among human beings are also present.

Visor datasets [130-132]

The another video data considered for experimentation is the Visor dataset which is the largest publically available and most standard dataset widely used for benchmarking results for segmentation. In this chapter, three video data sets from this category are used for experimentation which are Intelligent Room video sequence [130] containing 299 frames each of size 320 x 240, Camera2_070605 video sequence [131] containing 2881 frames each of size 384 x 288 and HighwayI_raw dataset [132] containing 439 frames each of size 320 x 240. Camera2_070605 video sequence dataset is performed at particular angle and is of low-quality and low contrast. Intelligent Room video sequence is recorded in full noisy environment i.e. video quality is low with poor contrast and shadow of object is also present. In highwayI_raw video sequence is recorded in full noisy environment and full and partial occlusion occurs between fast moving cars.

Caviar Dataset [133]

The next video data considered for experimentation is the one step video sequence dataset which is the part of Caviar video dataset available from [133]. This video data contains number of video clips, having 1995 frames each of size 480 x 272, which were recorded acting out the different scenarios of interest. This video is recorded in stationary background situation and multiple human beings are present in the video.

CVCR Dataset (Crowdie Environment Dataset) [134]

The final data set used for experimentation contains videos of crowd's density environment. 4917-5_70 is one of the video sequences of CVCR dataset [134] which contain 1789 frames each of size 480 x 320. This video was shooted on much more height and in very crowdie environment which contains full occlusions, shadow and noise.

3.4.2. Performance Measures

It is very difficult to compare the segmentation results visually because human visual system can identify and understand scenes with different connected objects effortlessly. Therefore, quantitative performance metrics together with visual results are more appropriate. The performance measures are categorized into various categories for determining the performance of the chosen method or comparing the proposed method with other methods for moving object segmentation. The various categories of performance measures calculate the accuracy of moving object segmentation; measures for noise removal in moving object segmentation; and computational time and memory required in moving object segmentation. The performance measures listed under various categories are defined as follows:

3.4.2.1. Accuracy of Moving Object Segmentation

The accuracy of moving object segmentation is calculated in terms of Relative Foreground Area Measure (RFAM) [84], Misclassification Penalty (MP) [84], Pixel Classification Based Measure (PCM) [84], and Relative Position Based Measure (RPM) [84] as discussed in chapter 2 (in section 2.5).

3.4.2.2. Noise Removal Capacity in Moving Object Segmentation

Here three performance measurement metrics namely Peak Signal-to-Noise Ratio [87], Normalized Absolute Error (NAE) [86], and Normalized Cross Correlation [85] are used for noise.

3.4.2.3. Computational Time and Memory

Here two performance measurement metrics namely computational time and memory consumption are used for Computational time and memory.

3.4.3. Results & Comparative Studies

In this section, comparative studies of some prominent methods as reported in literature and as discussed in Section 3.2, under the four categories, is presented both qualitatively and quantitatively on six video datasets discussed as above [129-134] in Section 3.4.1. Further, the comparative study of the proposed method is also presented with various methods under each category. The object intended for segmentation in the test video clips are appearing after approximately 100 frames in the test cases under consideration. The performance measures were calculated for whole video clips at the frame interval of 25 after 100th frame. In this chapter, the result for only four frames viz. 125, 150, 175, and 200 are shown. However, the performance trend remained the same for all video frames. In Tables 3.2 through 3.8, results of various moving object segmentation methods under each of the four categories as discussed in Section 3.2 in terms of seven different performance metrics divided under two categories viz. segmentation accuracy and noise removal, as discussed in sub-section 3.4.2, are listed. In Table 3.9, average computation time (frames/second) and memory consumption for different methods for a video of frame size 320 x 240 for first 100 frames [133] are shown. The comparative study has been done on a computer with Intel 2.53GHz core i3 processor with 4 GB RAM using OpenCV 2.9 and MATLAB 2013a software.

3.4.3.1. Qualitative Analysis

In this section, we report the experimental analysis and results of methods under categories I to IV and that of the proposed method. In category-I, we report the experimental analysis and results of four latest methods proposed by Kim *et al.* [33], Bradaski [98], Liu *et al.* [34], and Meier and Ngan [100] based on their advantages and limitations. In category-II, three latest methods for experimentation and comparative analysis are considered which are due to Mcfarlane *et al*. [27], Wren *et al.* [31] and Zivkovic *et al.* [32]. In category-III, we consider three latest methods for experimentation and comparative analysis which are due to Kushwaha *et al.* [26], Cucchiara [111], and Oliver [110]. Similar way, in category-IV, we consider four latest methods for experimentation and comparative analysis which are due to Kim *et al.* [116],Chien *et al.* [117], Khare *et al.* [121] and Hsia *et al.* [120].

Some observations about the results obtained by methods in categories I to IV and proposed method are as follows for six different video data sets [129-134]. From Fig. 3.4-3.9, it can be observed that:

(a) The segmentation results obtained by the method proposed by Kim *et al.* [33] perform better to other methods such as by Bradaski [98], Liu *et al.* [34] and Meier and Ngan [100] in category-I because the results of methods reported in [98, 34, 100] depends on the motion of the object (see frame no. 125-200 (ix, x, xiv)). If object is static then methods reported in [98, 34, 100] are not able to segment the object but Kim *et al.* [33] method works well for different data sets [129-134] (see frame no. 125-200 (iv)).

(b) The segmentation results obtained by the method proposed by Mcfarlane *et al.* [27] perform better to other methods in category II (see frame no. 125-200 (iii)). From Fig. 3.4-3.5, one can conclude that Mcfarlane *et al*. [27] give better shape of moving object with least noise in segmented frames by the methods in category-II (see frame no. 125-200 (iii)). From Fig. 3.6-3.9, it is clear that Wren *et al.* [31] and Zivkovic *et al.* [32] both suffers from the ghost object, noise, and shadow (see frame no. 125-200 (v & xi)) but Mcfarlane *et al*. [27] give better result with respect to least noise in segmented frame in category-II.

(c) For the methods under category-III:

- The segmentation results obtained by the method proposed by Kushwaha *et al.* [26] perform better to other methods in category III (see frame no. 125-200 (xii)).
- Results obtained by Cucchiara [111] suffer from the problem of ghosts, noise and shadows and also some portion of the object is distorted (see frame no. 125-200 (vi)).
- Results obtained by the method proposed by Oliver [110] have the problem of disappearance of the object in the frame during segmentation process after some time and the object is also distorted (see frame 125-150 (xiii)).

(d) The segmentation results obtained by the method proposed by Khare *et al.* [121] under category–IV perform better to other methods such as Kim *et al.* [116], Chien *et al.* [117] and Hsia *et al.* [120]. From Fig. 3.4, it is clear that, results of methods reported in [116, 117, 120] is not accurate (i.e. objects are collapsed) due to occlusions between multiple objects in the frame (see frame no. 125-200 (vii, xv, xvi)). In this situation Khare *et al.* [121] method works well but it suffers from the problem of ghosts (see frame no. 125-200 (viii)). From Fig. 3.5, one can conclude that, methods reported in [116, 117, 120] is not able to give comparable shape structure as compared to the Khare *et al.* [121] (see frame no. 125-200 (vii, viii, xvi)). From Fig. 3.6, it is also seen that the method proposed by Khare *et al.* [121] suffered the problem of ghost as compared to the Chien *et al.* [117] and Hsia *et al.* [120] (see frame no. 125-200 (vii, viii, xvi)). From Figs. 3.8 and 3.9, it is clear that, result obtained by Hsia *et al.* [120] method is distorted (see frame 125-200 (vi)) due to speed change of cars but in this condition Khare *et al.* [121] method work properly (see frame no. 125-200 (vii & viii)).

(e) The segmentation results obtained by proposed method perform well to other methods in the category I to IV having fast moving objects, crowdie and shadow environment in the video dataset. The proposed method does not suffer from the problem of ghost, object distortion, shadow, and disappearance of object in video scene (see frame no.125-200 (ii)) in comparison to other method in the category I to IV for different datasets [129-134].

3.4.3.2. Quantitative Analysis

In this section, the performances of the proposed method have been compared quantitatively under categories I to IV and proposed method in terms of seven different performance metrics divided under two categories viz. segmentation accuracy and noise removal as discussed in section 3.4.2.

From Tables 3.2-3.9 and Figs. 3.10-3.16(a-f) it can be observed that the following methods are performing better under each of their respective categories. These methods are associated with high value of RFAM, RPM, PCM and low value of MP in comparison to other methods under each category which indicate better segmentation accuracy. The high values of PSNR and NCC and low value of NAE indicate better noise removal capacity in comparison to other methods under respective categories for different datasets [129-134]. Also, the following methods under each of the respective category are associated with less computational time and memory consumption in comparison to other methods in their respective categories. These observations are summarized as:

- Kim *et al.* [33] method is performing better in terms of segmentation accuracy, noise removal capacity and computational complexity in category-I for each of the datasets.
- Macflrane *et al.* [27] method is performing better in terms of segmentation accuracy, noise removal capacity and computational complexity in category-II for each of the datasets.
- Kushwaha *et al.* [26] method is performing better in terms of segmentation accuracy, noise removal and computational complexity in category-III for each of the datasets.
- Khare *et al.* [121] method is performed better in terms of segmentation accuracy, noise removal and computational complexity in category-IV for each of the datasets.

Further, the proposed method is associated with high value of RFAM, RPM, PCM, PSNR, NCC; and low value of MP and NAE in most of the frames in comparison to other methods under each category for different datasets [129-134]. From Table 3.9, one can also observe that the proposed method had taken less computational time and consumed only 3.90 megabytes of RAM which was the least in comparison with the other methods in category-I to IV. Hence, proposed method is performing better in terms of segmentation accuracy, noise removal and computational complexity in comparison to other methods in categories I to IV for each of the datasets.

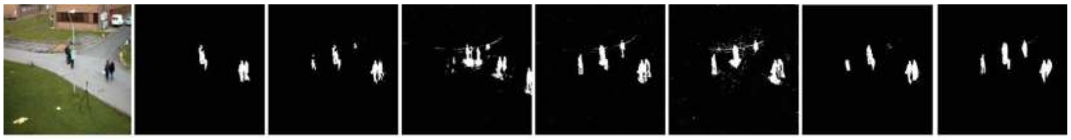
Overall observation of performance of methods under Categories I to IV and the proposed method:

From qualitative and quantitative observations of the comparative analysis and results of methods in Categories I to IV and the proposed method, we conclude that proposed method is performing better in comparison to all methods under consideration for different datasets [129-134]. For experimentation, we have taken different complex datasets i.e. multiple objects with partial and full occlusion, crowded object, and fast moving object with shadow. After overall observation, we conclude that the proposed method perform better to other methods from category I to IV. The other methods which perform better after the proposed method in decreasing order of their performances are Kushwaha *et al*. [26], Mcflarne *et al.* [27], Khare *et al.* [121], and Kim *et al.* [33].

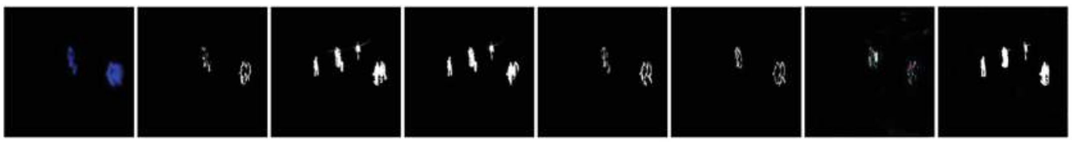
3.5. Conclusions

This chapter presented a review and experimental study of various recent moving object segmentation methods available in literature and these methods were classified into four categories i.e. moving object segmentation methods based on (i) motion information (ii) motion and spatial information (iii) learning, and (iv) change detection. The objective of this chapter was two-fold i.e. firstly, this chapter presented a comprehensive literature review and comparative study of various classical as well as state-of-the art methods for moving object segmentation under varying illumination conditions under each of the above mentioned four categories. Further, in this chapter, an efficient approach for moving object segmentation under varying illumination conditions was proposed and its comparative study with other methods under consideration was presented. The qualitative and quantitative comparative study of the various methods under four categories as well as the proposed method was presented for six different datasets [129-134]. The advantage, limitations, and efficacy of each of the methods under consideration have been examined.

The extensive experimental results on six challenging data sets demonstrate that the proposed method is superior to other state -of-the-art background subtraction methods as well as this chapter also provided an insight about other methods available in literature.



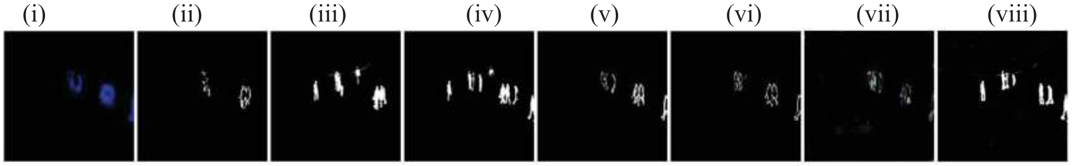
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

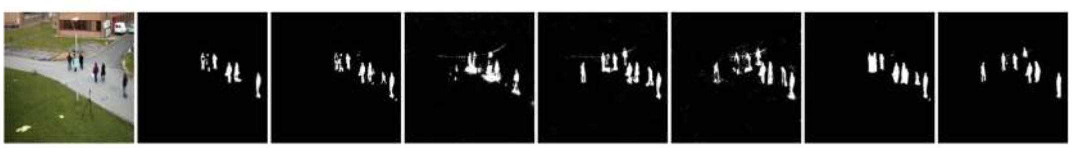
(a)Frame 125

(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

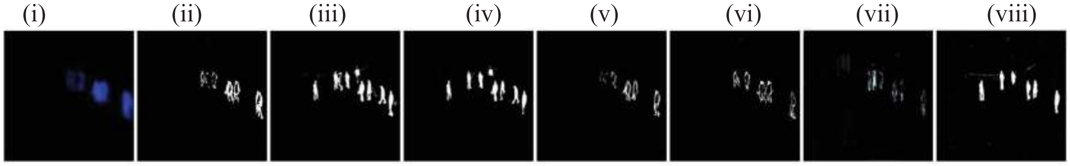


(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(b)Frame 150

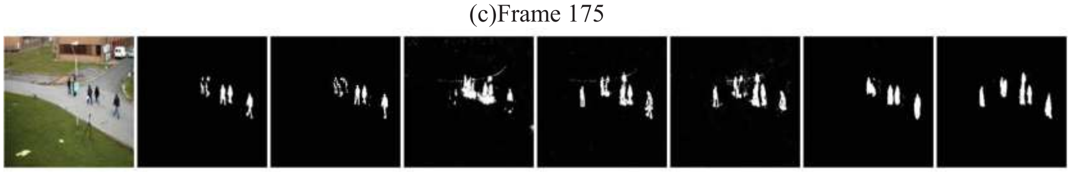


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(c)Frame 175

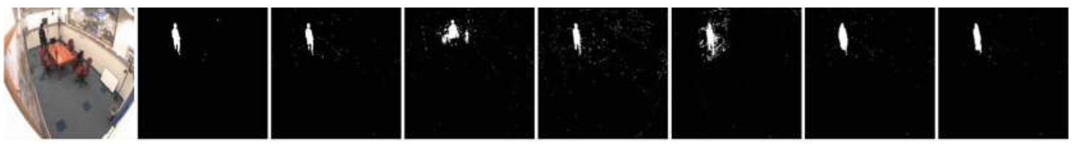


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

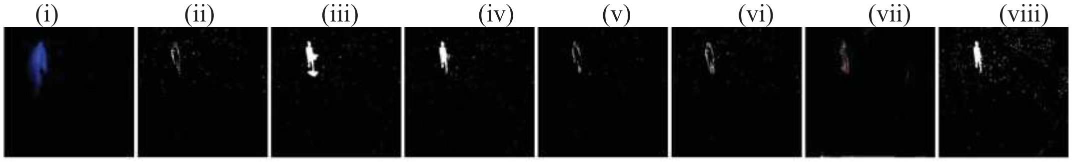
(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(d)Frame 200

Figure 3.4: Segmentation results for People video sequence [129] corresponding to (a) Frame 125,(b) frame 150, (c) frame 175, (d) frame 200 (i) original frame, and the segmented frame obtained by various methods such as: (ii) the proposed method, (iii)McFarlane and Schofield[27], (iv) Kim *et al.*[33], (v) Zivkovic[32] (vi) Cucchiara *et al.*[111], (vii)Hsia *et al.*[120], (viii) Khare *et al.*[121] (ix) Bradski[98], (x) Liu *et al.* [34], (xi) Wren *et al.*[31], (xii) Kushwaha *et al.* [26], (xiii) Oliver *et al.*[110], (xiv) Meier and Ngan [100], (xv) Kim *et al.* [116], and (xvi) Chien *et al.* [117].

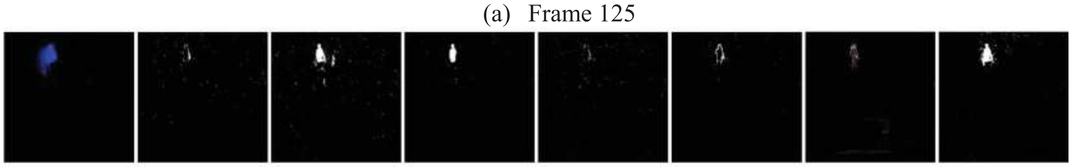


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

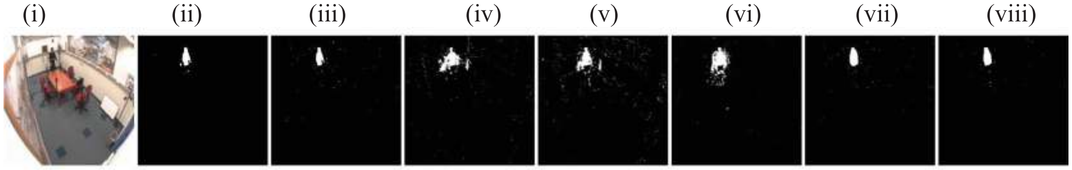


(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(a) Frame 125

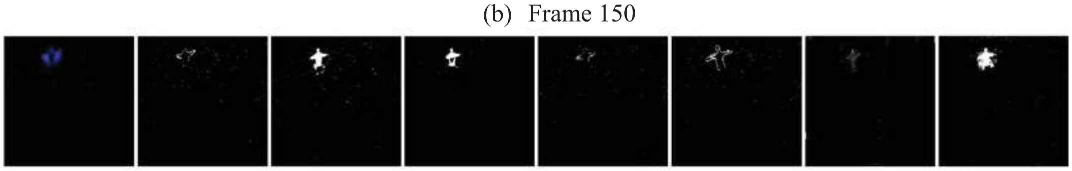


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

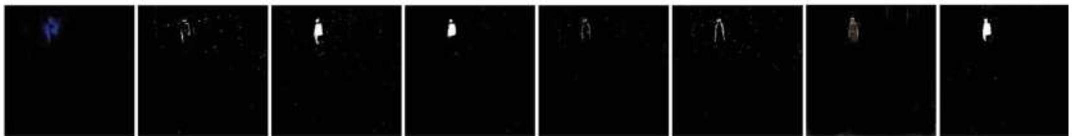
(b) Frame 150



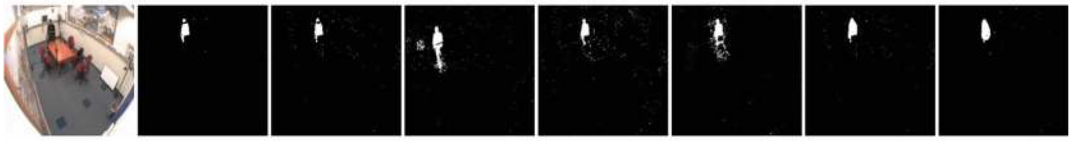
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(c) Frame 175



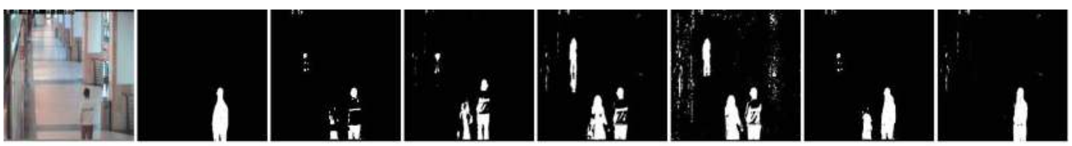
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



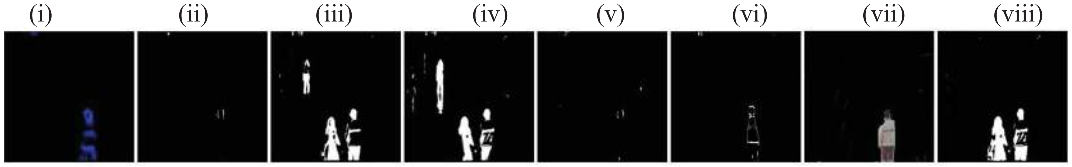
(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(d) Frame 200

Figure 3.5: Segmentation results for Intelligent Room video sequence [130] corresponding to (a) Frame 125,(b) frame 150, (c) frame 175, (d) frame 200 (i) original frame, and the segmented frame obtained by various methods such as: (ii) the proposed method, (iii)McFarlane and Schofield[27], (iv) Kim *et al.*[33], (v) Zivkovic[32] (vi) Cucchiara *et al.*[111], (vii)Hsia *et al.*[120], (viii) Khare *et al.*[121] (ix) Bradski[98], (x) Liu *et al.* [34], (xi) Wren *et al.*[31], (xii) Kushwaha *et al.* [26], (xiii) Oliver *et al.*[110], (xiv) Meier and Ngan [100], (xv) Kim *et al.* [116], and (xvi) Chien *et al.* [117].

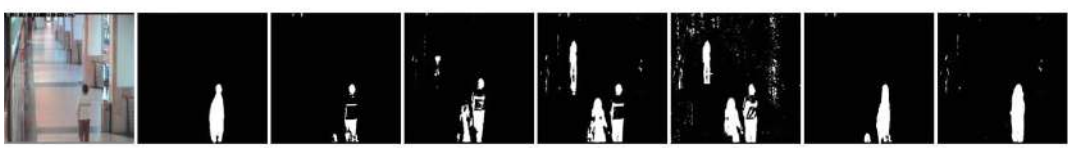


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

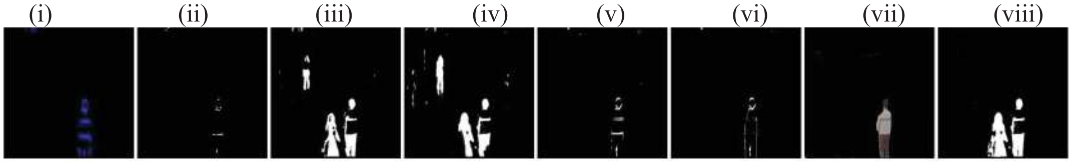


(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(a) Frame 125

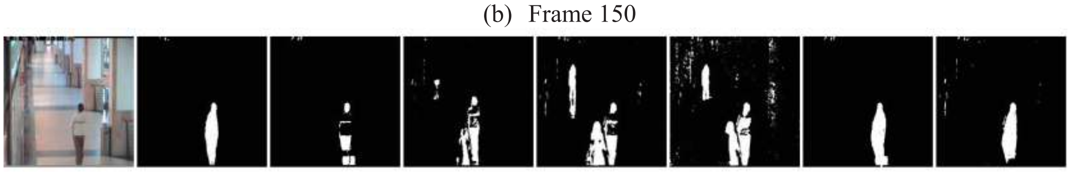


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

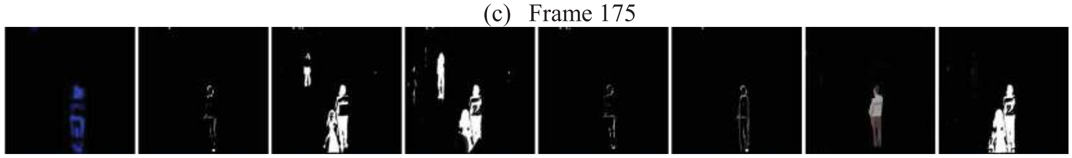
(b) Frame 150



(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(c) Frame 175



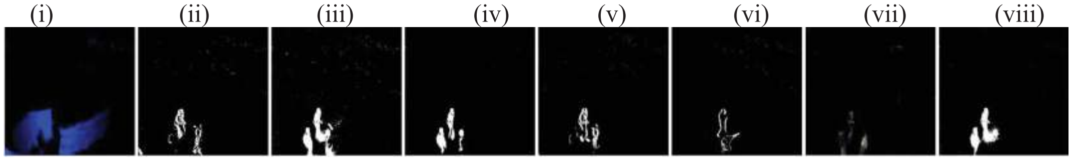
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(d) Frame 200

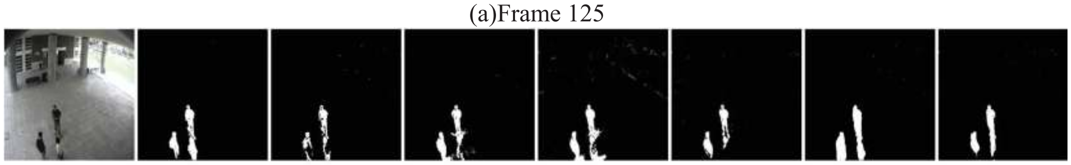
Figure 3.6: Segmentation results for One Step video sequence [133] corresponding to(a) Frame 125,(b) frame 150, (c) frame 175, (d) frame 200 (i) original frame, and the segmented frame obtained by various methods such as: (ii) the proposed method, (iii)McFarlane and Schofield[27], (iv) Kim *et al.*[33], (v) Zivkovic[32] (vi) Cucchiara *et al.*[111], (vii)Hsia *et al.*[120], (viii) Khare *et al.*[121] (ix) Bradski[98], (x) Liu *et al.* [34], (xi) Wren *et al.*[31], (xii) Kushwaha *et al.* [26], (xiii) Oliver *et al.*[110], (xiv) Meier and Ngan [100], (xv) Kim *et al.* [116], and (xvi) Chien *et al.* [117].

(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

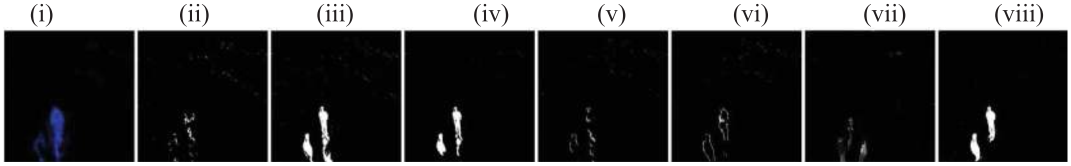


(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(a)Frame 125

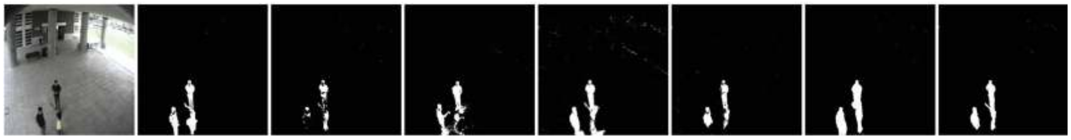


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(b)Frame 150

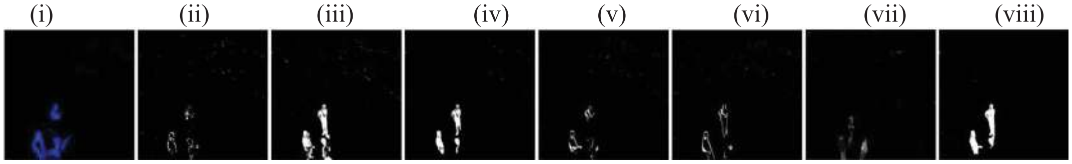


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(c)Frame 175

(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

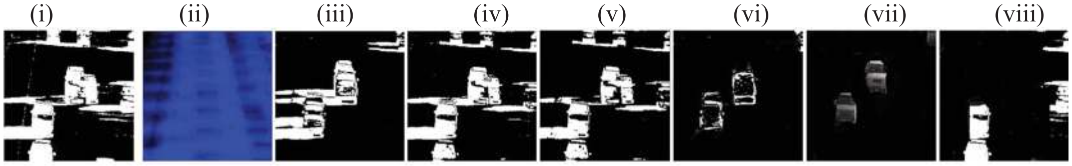


(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(d)Frame 200

Figure 3.7: Segmentation results for Camera2_070605 video sequence [131] corresponding (a) Frame 125,(b) frame 150, (c) frame 175, (d) frame 200 (i) original frame, and the segmented frame obtained by various methods such as: (ii) the proposed method, (iii)McFarlane and Schofield[27], (iv) Kim *et al.*[33], (v) Zivkovic[32] (vi) Cucchiara *et al.*[111], (vii)Hsia *et al.*[120], (viii) Khare *et al.*[121] (ix) Bradski[98], (x) Liu *et al.* [34], (xi) Wren *et al.*[31], (xii) Kushwaha *et al.* [26], (xiii) Oliver *et al.*[110], (xiv) Meier and Ngan [100], (xv) Kim *et al.* [116], and (xvi) Chien *et al.* [117].

(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



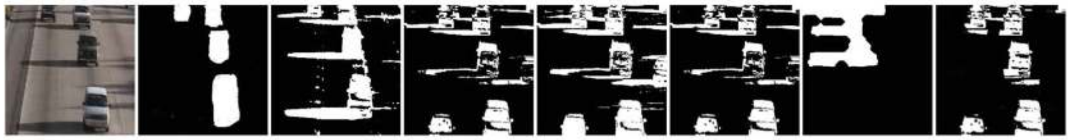
(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(a)Frame 125

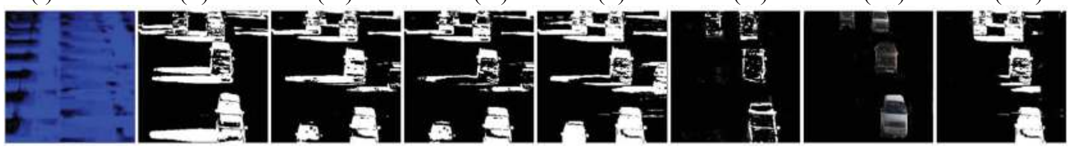
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(b)Frame 150

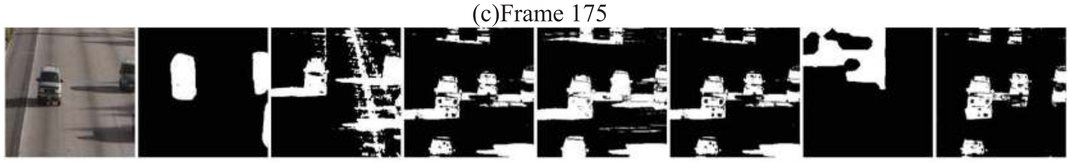


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

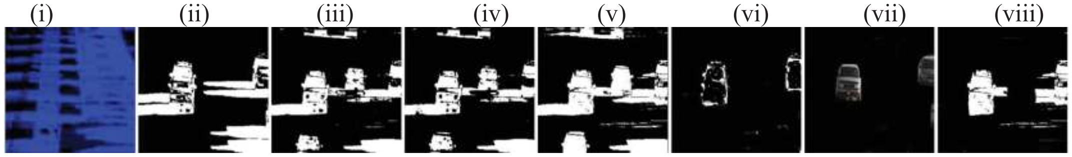


(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(c)Frame 175



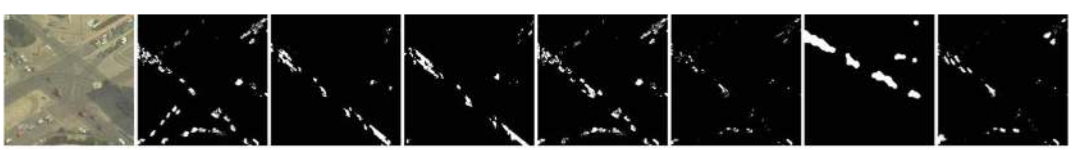
(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)



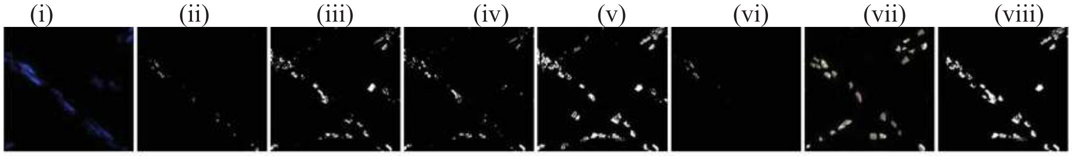
(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

(d)Frame 200

Figure 3.8: Segmentation results for highwayI_raw video sequence [132] corresponding to (a) Frame 125,(b) frame 150, (c) frame 175, (d) frame 200 (i) original frame, and the segmented frame obtained by various methods such as: (ii) the proposed method, (iii)McFarlane and Schofield[27], (iv) Kim *et al.*[33], (v) Zivkovic[32] (vi) Cucchiara *et al.*[111], (vii)Hsia *et al.*[120], (viii) Khare *et al.*[121] (ix) Bradski[98], (x) Liu *et al.* [34], (xi) Wren *et al.*[31], (xii) Kushwaha *et al.* [26], (xiii) Oliver *et al.*[110], (xiv) Meier and Ngan [100], (xv) Kim *et al.* [116], and (xvi) Chien *et al.* [117].

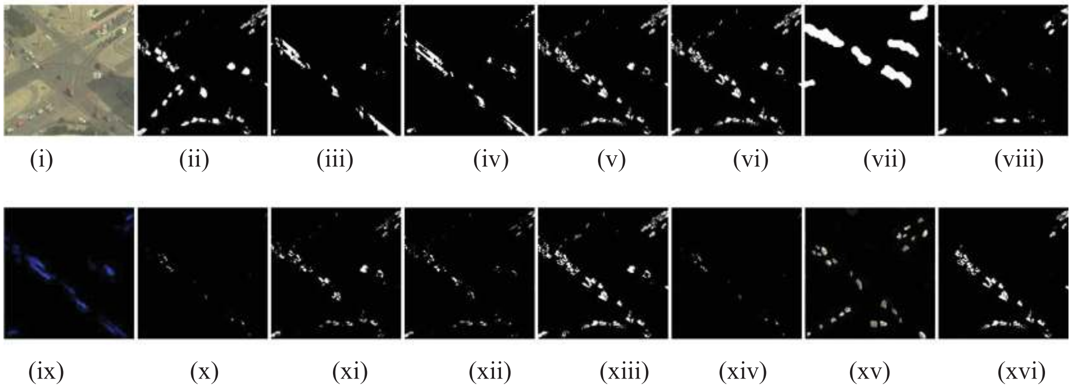


(i) (ii) (iii) (iv) (v) (vi) (vii) (viii)

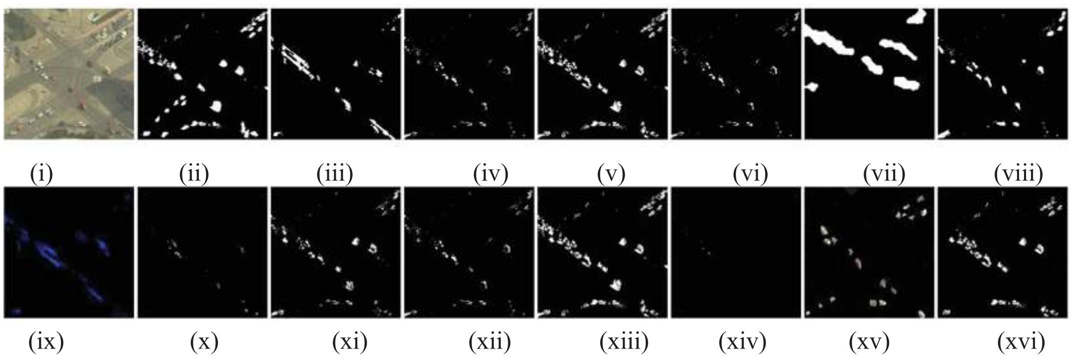


(ix) (x) (xi) (xii) (xiii) (xiv) (xv) (xvi)

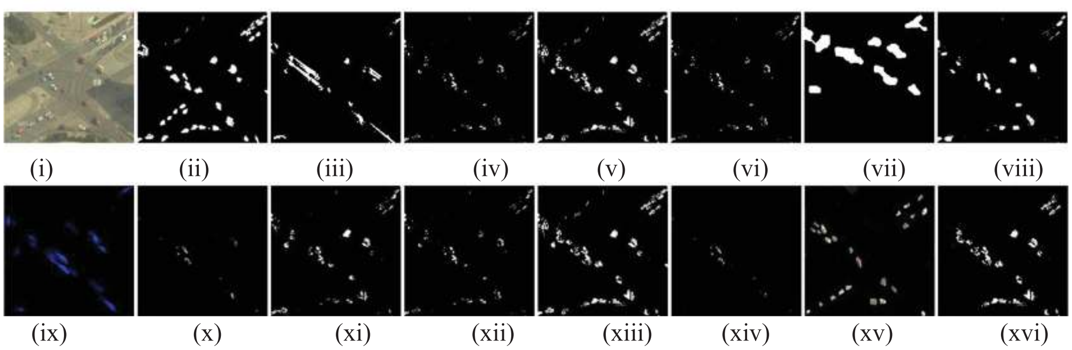
(a)Frame 125



(b)Frame 150



(c)Frame 175



(d)Frame 200

Figure 3.9: Segmentation results for 4917-5_70 video sequence [134] corresponding to(a) Frame 125,(b) frame 150, (c) frame 175, (d) frame 200 (i) original frame, and the segmented frame obtained by various methods such as: (ii) the proposed method, (iii)McFarlane and Schofield[27], (iv) Kim *et al.*[33], (v) Zivkovic[32] (vi) Cucchiara *et al.*[111], (vii)Hsia *et al.*[120], (viii) Khare *et al.*[121] (ix) Bradski[98], (x) Liu *et al.* [34], (xi) Wren *et al.*[31], (xii) Kushwaha *et al.* [26], (xiii) Oliver *et al.*[110], (xiv) Meier and Ngan [100], (xv) Kim *et al.* [116], and (xvi) Chien *et al.* [117].

Table 3.2: Comparison of methods in terms of Relative foreground area measure

F. No.	Category I				Category II			Category III			Category IV				
	Bradski [98]	**Kim *et al.*[33]**	**Liu *et al.* [34]**	**Meier and Ngan [100]**	**McFarlane *et al.* [27]**	**Wren *et al.*[31]**	**Zivkovic[32]**	**Kushwaha *et al.* [26]**	**Cucchiara *et al.*[111]**	**Oliver *et al.*[110]**	**Hsia *et al.*[120]**	**Khare *et al.*[121]**	**Kim *et al.* [116]**	**Chien *et al.* [117]**	**proposed**
A-People Video Sequences [129]															
125	0.7661	0.4220	0.558	0.463	0.8219	0.5969	0.5074	0.8667	0.2674	0.4045	0.5870	0.7126	0.444	0.663	**0.9327**
150	0.7798	0.4665	0.598	0.489	0.7982	0.5575	0.4617	0.8764	0.2607	0.4239	0.6491	0.6005	0.347	0.686	**0.9066**
175	0.7077	0.5719	0.591	0.493	0.8883	0.6335	0.5266	0.8077	0.3221	0.4755	0.7457	0.6190	0.389	0.662	**0.9777**
200	0.7521	0.5084	0.513	0.529	0.9167	0.6683	0.5656	0.8682	0.3181	0.4336	0.6206	0.7508	0.434	0.673	**0.9014**
B-Camera2_070605 Video Sequence [131]															
125	0.7779	0.8631	0.5067	0.471	0.6546	0.9429	0.8645	0.8491	0.8662	0.2269	**0.9514**	0.8757	0.417	0.655	0.9019
150	0.7888	0.9316	0.5531	0.532	0.5601	0.9223	0.7925	0.8618	0.7813	0.489	0.8087	0.8184	0.402	0.651	**0.9878**
175	0.7086	0.9365	0.5935	0.552	0.5033	0.9263	0.8177	0.8625	0.7612	0.6827	0.7920	0.8500	0.446	0.667	**0.9625**
200	0.7561	0.8274	0.5888	0.456	0.5961	0.8455	0.8820	0.8746	0.5615	0.5716	0.8949	0.7261	0.432	0.706	**0.9271**
C-One Step Video Sequence [133]															
125	0.3405	0.7535	0.5132	0.477	0.4773	0.6366	0.4633	0.8726	0.3493	0.7133	0.8651	0.6893	0.384	0.677	**0.8862**
150	0.2637	0.7583	0.4071	0.509	0.4448	0.6162	0.4414	0.8974	0.3409	0.7225	0.7607	**0.8612**	0.381	0.689	0.8137
175	0.3945	0.7866	0.5043	0.515	0.4272	0.6393	0.4501	0.8550	0.3848	0.8142	**0.8249**	0.7893	0.389	0.677	0.7550
200	0.3537	0.7005	0.4678	0.514	0.4770	0.6935	0.4980	0.8671	0.3733	0.4154	0.8199	**0.9752**	0.391	0.659	0.9071
D-Intelligent room Video Sequence [130]															
125	0.7556	0.8215	0.5407	0.520	0.9503	0.8157	0.5216	**0.9351**	0.6180	0.7569	0.8961	0.6415	0.368	0.663	0.9287
150	0.7081	0.8084	0.5653	0.474	0.9092	0.7895	0.5019	0.9375	0.6888	0.7341	0.7992	0.6320	0.364	0.662	**0.9375**
175	0.7285	0.8833	0.52	0.460	0.9851	0.7541	0.5424	**0.9622**	0.6816	0.738	0.6747	0.5326	0.414	0.654	0.9462
200	0.7193	0.8492	0.5635	0.478	0.9031	0.876	0.5171	**0.9126**	0.6809	0.7547	0.7583	0.6739	0.419	0.690	0.8526
E-HighwayI_raw video sequence [132]															
125	0.3476	0.6708	0.4143	0.471	0.6154	0.5193	0.5308	0.7956	0.5678	0.5162	0.7133	0.8302	0.370	0.713	**0.8336**
150	0.3294	0.6200	0.4487	0.483	0.5880	0.4942	0.5944	0.7320	0.8218	0.4876	0.8287	0.8768	0.409	0.694	**0.8830**
175	0.3927	0.7074	0.4122	0.490	0.5901	0.5518	0.4858	0.7971	0.7123	0.5262	0.7645	0.7423	0.416	0.674	**0.8571**
200	0.3123	0.6733	0.4615	0.533	0.5845	0.5761	0.3408	0.7123	0.6773	0.5517	0.7013	0.5521	0.364	0.672	**0.8523**
F-Crowd Video Sequence [134]															
125	0.3149	0.5514	0.5067	0.527	0.3696	0.4723	0.7623	0.7733	0.5744	0.5252	0.4396	0.6581	0.434	0.692	**0.9243**
150	0.3318	0.4445	0.5231	0.532	0.2149	0.4392	0.8307	0.7247	0.5465	0.5696	0.4089	0.6064	0.372	0.709	**0.8987**
175	0.3016	0.4629	0.5785	0.541	0.2140	0.4926	0.7988	0.7143	0.6619	0.4991	0.4762	0.6183	0.413	0.689	**0.9143**
200	0.3511	0.5103	0.5839	0.457	0.3229	0.4727	0.6827	0.7305	0.8735	0.5167	0.4114	0.6835	0.411	0.689	**0.9305**

Table 3.3: Comparison of methods in terms of Misclassification Penalty

F. No.	Category I				Category II			Category III			Category IV				
	Bradski [98]	**Kim *et al.*[33]**	**Liu *et al.* [34]**	**Meier and Ngan [100]**	**McFarlane *et al.* [27]**	**Wren *et al.*[31]**	**Zivkovic[32]**	**Kushwaha *et al.* [26]**	**Cucchiara *et al.*[111]**	**Oliver *et al.*[110]**	**Hsia *et al.*[120]**	**Khare *et al.*[121]**	**Kim *et al.* [116]**	**Chien *et al.* [117]**	**proposed**
A-People Video Sequences [129]															
125	0.1061	0.0378	0.201	0.184	0.0011	0.0131	0.0018	0.0656	0.0167	0.0011	0.0060	0.0011	0.0674	0.020631	**1.1546 e-004**
150	0.1028	0.0040	0.102	0.179	5.0e-006	9.45E-04	0.0622	0.0109	0.0112	0.0027	0.0030	1.4e-003	0.0736	0.018506	**6.3695 e-006**
175	0.1168	0.0067	0.104	0.178	2.0797 e-004	0.0012	0.0125	0.0037	0.0063	0.0027	0.0020	8.4168 e-003	0.0731	0.016423	**6.4861 e-006**
200	0.1221	0.0189	0.023	0.162	2.1236 e-004	0.0145	0.0082	0.0121	0.0168	4.56E-04	0.0095	1.3720 e-004	0.0698	0.02126	**6.6059 e-006**
B-Camera2_070605 Video Sequence [131]															
125	0.0123	0.0035	0.1088	0.212	0.0021	0.0063	0.0134	0.0362	0.0088	0.0362	2.0025 e-004	4.3669 e-004	0.06945	0.015142	**0.0046**
150	0.03	1.4146 e-004	0.124	0.256	0.0032	0.0051	0.0101	0.0291	0.0105	0.0123	0.0032	0.0010	0.06989	0.021289	**1.1815 e-004**
175	0.0071	3.6778 e-004	0.1778	0.229	0.0305	0.0058	0.0074	0.0083	0.0115	0.0195	0.0034	5.4249 e-004	0.07134	0.014267	**1.3563 e-004**
200	0.0027	0.0013	0.1162	0.221	9.3381 e-004	0.0114	0.0096	0.0167	0.0280	0.0186	0.0066	**2.2011 e-004**	0.07012	0.01588	4.0812 e-004
C-One Step Video Sequence [133]															
125	0.1194	0.1502	0.2556	0.228	0.0739	0.1337	0.1883	0.0016	0.1366	0.1161	0.0069	0.0275	0.066	0.019	**2.9030 e-004**
150	0.2086	0.1846	0.2226	0.179	0.0114	0.1562	0.1235	0.0731	0.1752	0.0218	0.0034	0.0043	0.069	0.014	**0.0014**
175	0.3111	0.1581	0.1979	0.208	0.0063	0.1009	0.1800	0.0935	0.1095	0.0512	0.0094	**0.0046**	0.064	0.021	0.0059
200	0.2492	0.1988	0.2247	0.182	0.0086	0.1475	0.1763	0.0081	0.1875	0.067	0.0078	0.0400	0.073	0.017	**1.7832 e-004**
D-Intelligent room Video Sequence [130]															
125	0.1045	0.1064	0.0536	0.226	0.0070	0.0193	0.1017	0.0571	0.1742	0.0406	0.0031	0.0035	0.07234	0.01756	**1.9051 e-004**
150	0.1086	0.0052	0.0109	0.229	8.0105 e-004	0.0087	0.0128	0.0044	0.0484	0.0281	6.0313 e-004	8.4038 e-004	0.06667	0.016352	**5.0712 e-005**
175	3.57E-004	0.0107	0.0153	0.185	2.8677 e-004	0.0026	0.0084	0.0052	0.0318	0.0115	5.0712 e-005	1.6023 e-004	0.06589	0.016777	**4.9444 e-005**
200	4.29E-004	0.1097	0.0255	0.241	0.0026	0.0025	0.0250	0.0631	0.0454	0.0066	2.0408 e-004	0.0024	0.06823	0.020179	**5.1020 e-005**
E-HighwayI_raw video sequence [132]															
125	0.1271	0.1242	0.1672	0.220	0.0621	0.0263	0.1302	0.0016	0.1512	0.1836	0.0975	0.1463	0.068	0.013	**0.0028**
150	0.1792	0.1307	0.1529	0.218	0.1044	0.0192	0.1932	0.0169	0.1387	0.1731	0.0136	0.1498	0.072	0.016	**0.0112**
175	0.1641	0.1217	0.1853	0.209	0.0366	0.0416	0.0110	0.0027	0.0345	0.1285	0.0285	0.1444	0.073	0.018	**0.0020**
200	0.1862	0.1894	0.1992	0.174	0.0131	0.0322	0.1900	0.0031	0.1232	0.1166	0.0104	0.1424	0.064	0.020	**0.0013**
F-Crowd Video Sequence [134]															
125	0.1617	0.0021	0.2937	0.203	0.0342	0.1482	0.0653	0.0023	0.0675	0.1782	0.0698	0.1256	0.071	0.017	**0.0035**
150	0.1249	0.0201	0.3724	0.249	0.0395	0.2836	0.0571	0.0193	0.0342	0.1539	0.0970	0.1195	0.072	0.019	**0.0139**
175	0.1591	0.0085	0.1936	0.162	0.0719	0.1729	0.0069	0.0062	0.0563	0.1846	0.0096	0.1456	0.070	0.020	**0.0035**
200	0.1814	0.0120	0.2933	0.221	0.0382	0.1936	0.0071	0.0142	0.0762	0.1592	0.0178	0.1864	0.073	0.015	**0.0042**

Table 3.4: Comparison of methods in terms of Relative position based measure (RPM)

F. No.	Category I				Category II			Category III			Category IV				
	Bradski [98]	**Kim *et al.*[33]**	**Liu *et al.* [34]**	**Meier and Ngan [100]**	**McFarlane *et al.* [27]**	**Wren *et al.*[31]**	**Zivkovic[32]**	**Kushwaha *et al.* [26]**	**Cucchiara *et al.*[111]**	**Oliver *et al.*[110]**	**Hsia *et al.*[120]**	**Khare *et al.*[121]**	**Kim *et al.* [116]**	**Chien *et al.* [117]**	**proposed**
A-People Video Sequences [129]															
125	0.7585	0.9537	0.896	0.686	0.9214	0.8546	0.8091	0.8791	0.6569	0.8983	0.7840	0.9268	0.723	0.865	**0.9647**
150	0.8475	0.8440	0.846	0.658	0.9016	0.922	0.8867	0.8691	0.7324	0.846	0.8616	0.8602	0.777	0.843	**0.9941**
175	0.7688	0.8328	0.820	0.715	0.9576	0.9276	0.8941	0.8725	0.8195	0.8537	0.9053	0.8842	0.751	0.825	**0.9944**
200	0.8817	0.8775	0.947	0.679	0.9734	0.8403	0.7939	0.8846	0.6877	0.947	0.7890	0.9789	0.734	0.820	**0.9941**
B-Camera2_070605 Video Sequence [131]															
125	0.9279	**0.9525**	0.7196	0.651	0.9358	0.9416	0.9262	0.8817	0.9196	0.876	0.9800	0.9125	0.777	0.854	0.9507
150	0.8968	0.9900	0.7851	0.703	0.9550	0.9525	0.9397	0.8819	0.9248	0.7874	0.9490	0.9576	0.771	0.840	**0.9929**
175	0.8864	0.9775	0.751	0.654	0.8966	0.9446	0.9466	0.8230	0.9195	0.3787	0.9495	0.9107	0.723	0.834	**0.9900**
200	0.9616	0.9754	0.7016	0.647	0.9604	0.9269	0.9374	0.8861	0.8879	0.8918	0.9359	0.9690	0.729	0.844	**0.9861**
C-One Step Video Sequence [133]															
125	0.726	0.7854	0.536	0.682	0.8482	0.6887	0.6145	0.8688	0.5941	0.6203	0.9244	0.9002	0.760	0.823	**0.9863**
150	0.712	0.7802	0.547	0.756	0.9505	0.7439	0.6626	0.8893	0.6646	0.8747	0.9411	0.9667	0.779	0.864	**0.9804**
175	0.7845	0.8277	0.6075	0.733	0.9610	0.8056	0.7274	0.8829	0.7305	0.9071	0.9269	**0.9904**	0.752	0.848	0.9568
200	0.7686	0.8004	0.5677	0.683	0.9505	0.8432	0.7782	0.8772	0.7275	0.767	0.9240	0.9705	0.777	0.837	**0.9900**
D-Intelligent room Video Sequence [130]															
125	0.8903	0.6608	0.7672	0.743	0.9216	0.8492	0.6020	0.8626	0.3886	0.7868	0.9403	0.9515	0.734	0.824	**0.9889**
150	0.8707	0.8503	0.7049	0.655	0.9234	0.7293	0.6717	0.8897	0.3489	0.5464	0.9459	0.9395	0.745	0.856	**0.9869**
175	0.9746	0.7864	0.7435	0.771	0.9590	0.8714	0.7667	0.8954	0.5572	0.7405	0.9876	0.9735	0.739	0.860	**0.9886**
200	0.8877	0.4521	0.6718	0.762	0.8949	0.884	0.6366	0.8886	0.4677	0.8647	0.9652	0.9245	0.743	0.846	**0.9826**
E-HighwayI_raw video sequence [132]															
125	0.5562	0.8773	0.6882	0.688	0.9467	0.7645	0.8815	0.8896	0.7813	0.7284	0.8910	0.7437	0.733	0.825	**0.9856**
150	0.5458	0.8549	0.7281	0.647	0.9714	0.7821	0.8370	0.8827	0.8235	0.7119	0.8804	0.6589	0.730	0.852	**0.9858**
175	0.5382	0.9521	0.7192	0.711	0.9344	0.8194	0.9649	0.8699	0.7856	0.7302	0.9471	0.6896	0.756	0.846	**0.9799**
200	0.5281	0.8325	0.6901	0.712	0.9014	0.7293	0.8559	0.8786	0.7919	0.7521	0.8670	0.7230	0.720	0.814	**0.9756**
F-Crowd Video Sequence [134]															
125	0.3376	0.8894	0.5721	0.759	0.7399	0.6713	0.5403	0.7972	0.8576	0.5823	0.9149	0.7202	0.753	0.868	**0.9170**
150	0.3145	0.8659	0.5518	0.722	0.7631	0.6691	0.5369	0.7857	0.8126	0.5317	0.9145	0.7447	0.765	0.840	**0.9753**
175	0.3297	0.8475	0.5592	0.692	0.7132	0.672	0.5572	0.7928	0.7845	0.5673	0.9451	0.7647	0.732	0.827	**0.9666**
200	0.3453	0.8484	0.5782	0.704	0.7259	0.6218	0.5687	0.7891	0.8263	0.5584	0.9594	0.7474	0.763	0.830	**0.9818**

Table 3.5: Comparisons of methods in terms of Normalized Cross Correlation (NCC)

A-People Video Sequences [129]															
F. No.	**Category I**				**Category II**			**Category III**			**Category IV**				
	Bradski [98]	**Kim *et al.*[33]**	**Liu *et al.* [34]**	**Meier and Ngan [100]**	**McFarlane *et al.* [27]**	**Wren *et al.*[31]**	**Zivkovic[32]**	**Kushwaha *et al.* [26]**	**Cucchiara *et al.*[111]**	**Oliver *et al.*[110]**	**Hsia *et al.*[120]**	**Khare *et al.*[121]**	**Kim *et al.* [116]**	**Chien *et al.* [117]**	**proposed**
125	0.3163	0.3598	0.306	0.415	0.8995	0.881	0.8913	0.9655	0.7946	0.3147	0.8739	0.9505	0.516	0.778	**0.9837**
150	0.3069	0.4371	0.391	0.448	0.8975	0.8085	0.8711	0.9712	0.8488	0.4045	0.7991	0.8785	0.453	0.747	**1**
175	0.3013	0.3692	0.391	0.401	0.8747	0.9346	0.9622	0.9465	0.8184	0.3266	0.8980	0.8799	0.472	0.727	**0.9598**
200	0.3224	0.3890	0.371	0.412	0.7894	0.8403	0.7150	**0.9516**	0.6923	0.483	0.6379	0.7121	0.469	0.711	0.8516
B-Camera2_070605 Video Sequence [131]															
125	0.7257	0.7980	0.3379	0.454	0.6960	0.5418	0.6669	**0.8859**	0.4489	0.4754	0.6215	0.7799	0.482	0.728	0.8455
150	0.7273	0.7621	0.3901	0.393	0.6247	0.7563	0.8277	0.9215	0.6786	0.5096	0.7411	0.9070	0.472	0.693	**0.9505**
175	0.7011	0.7034	0.3265	0.429	0.4145	0.7562	0.8386	0.9071	0.6801	0.5083	0.7446	0.8030	0.495	0.775	**0.9428**
200	0.7444	0.7245	0.3144	0.410	0.4033	0.6077	0.7393	0.9023	0.4911	0.5395	0.7053	0.8091	0.475	0.691	**0.9253**
C-One Step Video Sequence [133]															
125	0.3369	0.4162	0.3073	0.407	0.4054	0.5026	0.5791	0.9534	0.7236	0.612	0.9407	0.8755	0.491	0.763	**0.9871**
150	0.2178	0.3435	0.3299	0.434	0.4761	0.5102	0.6278	0.9782	0.7513	0.6056	0.9928	0.9298	0.524	0.728	**1**
175	0.2733	0.3762	0.3016	0.454	0.4915	0.6033	0.6521	0.9018	0.8532	0.6198	0.8847	**0.9594**	0.486	0.751	0.9068
200	0.3455	0.3029	0.3456	0.421	0.5757	0.672	0.7781	0.9812	0.9116	0.6583	0.9093	0.9726	0.514	0.688	**0.9992**
D-Intelligent room Video Sequence [130]															
125	0.7206	0.8477	0.4838	0.452	0.9268	0.8178	0.8567	0.9761	0.8396	0.7604	0.8769	0.7154	0.477	0.753	**1**
150	0.7672	0.8928	0.4926	0.433	0.9519	0.9365	0.9737	0.9816	0.4945	0.7438	0.9672	0.7182	0.490	0.720	**0.9956**
175	0.7803	0.8033	0.4197	0.403	0.9098	0.8525	0.9016	0.9668	0.9574	0.7836	0.9475	0.6529	0.530	0.712	**1**
200	0.7655	0.8038	0.4628	0.417	0.8831	0.864	0.9253	0.9425	0.9693	0.7245	0.9368	0.7302	0.500	0.684	**1**
E-HighwayI_raw video sequence [132]															
125	0.6276	0.7631	0.3772	0.459	0.8819	0.5293	0.4379	0.9540	0.5631	0.6614	0.4543	0.3263	0.473	0.718	**0.9998**
150	0.7193	0.7174	0.3991	0.421	0.8873	0.5829	0.3316	0.9762	0.3458	0.6491	0.2374	0.3394	0.454	0.728	**1**
175	0.6863	0.7590	0.3183	0.408	0.8770	0.5425	0.5741	0.9418	0.4128	0.6825	0.5364	0.3830	0.530	0.732	**0.9998**
200	0.6659	0.7215	0.3637	0.414	0.7772	0.5814	0.4169	0.9423	0.4847	0.6285	0.3583	0.4596	0.531	0.754	**0.9983**
F-Crowd Video Sequence [134]															
125	0.2391	0.2496	0.1829	0.416	0.1968	0.4182	0.7604	0.8821	0.3452	0.3691	0.3128	0.4027	0.475	0.767	**0.9819**
150	0.1838	0.2029	0.2836	0.405	0.2332	0.4492	0.7395	0.8918	0.4356	0.3519	0.3093	0.4217	0.461	0.757	**0.9918**
175	0.2192	0.1837	0.2947	0.417	0.2479	0.4378	0.7275	0.8863	0.3583	0.3861	0.2944	0.4133	0.493	0.748	**0.9893**
200	0.2482	0.1865	0.2168	0.442	0.1921	0.4891	0.7575	0.8717	0.3891	0.3126	0.3578	0.4126	0.529	0.716	**0.9971**

Table 3.6: Comparison of methods in terms of Peak Signal-to-Noise Ratio (PSNR)

A-People Video Sequences [129]															
F. No .	**Category I**				**Category II**			**Category III**			**Category IV**				
	Bradski [98]	**Kim *et al.*[33]**	**Liu *et al.* [34]**	**Meier and Ngan [100]**	**McFarlane *et al.* [27]**	**Wren *et al.*[31]**	**Zivkovic[32]**	**Kushwaha *et al.* [26]**	**Cucchiara *et al.*[111]**	**Oliver *et al.*[110]**	**Hsia *et al.*[120]**	**Khare *et al.*[121]**	**Kim *et al.* [116]**	**Chien *et al.* [117]**	**proposed**
125	67.228	62.262	67.16	47.31	70.532	67.085	65.948	70.830	61.731	67.183	66.853	69.670	52.14	65.07	**76.388**
150	67.191	62.262	66.18	44.86	69.262	65.112	64.274	66.662	60.869	66.248	66.035	67.258	58.03	63.06	**75.559**
175	65.184	62.250	65.67	42.35	69.599	66.794	65.393	68.172	61.393	65.788	67.887	67.056	56.99	66.36	**75.072**
200	65.283	62.304	66.19	45.63	70.407	64.960	64.448	65.838	61.331	66.237	64.404	68.592	53.92	67.07	**72.512**
B-Camera2_070605 Video Sequence [131]															
125	59.653	62.022	63.844	41.27	67.690	63.203	62.447	67.759	63.167	62.716	64.560	63.239	57.37	63.50	**66.918**
150	64.719	66.786	63.698	44.31	67.188	67.468	65.821	64.998	66.981	63.720	68.052	67.878	56.99	66.99	**73.041**
175	64.688	64.857	63.629	48.01	65.851	67.448	66.289	66.613	67.153	63.526	68.323	66.859	52.04	67.56	**71.710**
200	64.494	64.420	63.801	45.62	64.760	65.570	65.312	64.825	64.956	63.732	66.470	65.924	53.74	69.34	**69.800**
C-One Step Video Sequence [133]															
125	63.970	61.287	62.97	46.97	63.75	61.083	60.01	66.984	59.222	62.966	68.655	64.571	55.28	64.67	**71.093**
150	63.725	61.011	63.18	41.88	65.24	61.66	60.10	63.18	59.295	63.405	67.969	68.354	57.28	65.36	**69.488**
175	63.398	61.181	62.9	45.44	65.09	61.66	60.16	64.122	60.245	63.035	66.526	67.551	53.13	68.98	**67.022**
200	63.721	61.763	63.35	44.04	65.54	62.070	60.42	63.629	60.681	63.36	67.307	68.457	54.51	64.57	**73.077**
D-Intelligent room Video Sequence [130]															
125	67.584	65.109	68.835	48.51	75.980	71.232	68.159	70.994	60.173	69.039	73.292	72.752	54.74	69.70	**79.994**
150	68.552	66.554	70.328	43.45	77.442	69.774	66.727	72.213	60.025	69.668	75.311	74.359	58.66	62.17	**81.543**
175	69.032	67.926	68.882	44.03	76.731	71.255	65.949	75.811	58.719	68.795	71.445	71.721	57.77	67.99	**81.543**
200	70.559	63.978	69.692	41.72	74.554	73.721	69.525	77.254	59.857	69.917	73.329	73.909	51.37	66.47	**77.394**
E-HighwayI_raw video sequence [132]															
125	53.582	58.050	57.593	44.72	59.495	55.620	53.863	64.038	56.153	59.836	54.841	55.636	58.12	61.04	**65.782**
150	54.394	58.912	58.204	49.03	57.429	53.734	54.040	65.103	51.357	59.937	55.699	54.183	56.04	68.74	**65.808**
175	56.872	58.439	57.845	46.60	57.448	56.926	54.378	62.429	53.135	58.835	56.276	54.190	51.70	64.99	**64.929**
200	55.384	58.579	55.936	47.63	55.412	53.485	53.589	64.254	55.671	59.282	56.190	54.521	53.31	67.65	**65.994**
F-Crowd Video Sequence [134]															
125	55.395	63.144	58.493	46.97	61.396	52.938	68.202	62.864	64.936	62.592	62.647	58.887	51.66	66.56	**71.169**
150	56.946	62.240	55.294	44.92	60.962	57.294	65.677	64.909	62.826	61.924	62.344	57.153	53.96	68.10	**70.109**
175	58.748	61.544	56.846	47.03	60.585	54.193	65.178	63.125	65.395	60.163	61.454	56.677	55.44	63.79	**70.100**
200	60.293	61.744	55.628	45.79	60.112	55.357	64.551	64.275	63.673	58.295	62.140	57.036	50.10	67.22	**71.760**

Table 3.7: Comparison of methods in terms of Normalized absolute error (NAE)

F. No.	Category I				Category II			Category III			Category IV				
	Bradski [98]	**Kim *et al.*[33]**	**Liu *et al.* [34]**	**Meier and Ngan [100]**	**McFarlane *et al.* [27]**	**Wren *et al.*[31]**	**Zivkovic[32]**	**Kushwaha *et al.* [26]**	**Cucchiara *et al.*[111]**	**Oliver *et al.*[110]**	**Hsia *et al.*[120]**	**Khare *et al.*[121]**	**Kim *et al.* [116]**	**Chien *et al.* [117]**	**proposed**
A-People Video Sequences [129]															
125	0.8734	0.7239	0.886	0.722	0.4082	0.7027	1.1728	0.6821	1.0967	0.8826	0.9522	0.4978	0.464	0.433	**0.1060**
150	0.7253	0.7559	0.914	0.685	0.4501	0.7705	1.4197	0.5434	1.1092	0.9011	0.9463	0.7141	0.497	0.353	**0.1056**
175	0.7138	0.7925	0.905	0.719	0.3668	0.6998	0.9661	0.2381	1.4271	0.8822	0.9441	0.6588	0.540	0.344	**0.1040**
200	0.7944	0.7732	0.887	0.699	0.3364	0.7789	1.3264	0.5827	1.7189	0.907	1.3399	0.5109	0.506	0.423	**0.2071**
B-Camera2_070605 Video Sequence [131]															
125	0.7797	0.7751	0.8381	0.697	0.7458	0.6715	0.562	0.3734	0.7797	1.0867	**0.4107**	0.5836	0.487	0.339	0.4130
150	0.7697	0.7781	0.9737	0.724	0.7359	0.4086	0.5971	0.5133	0.7572	0.9686	0.3572	0.3718	0.455	0.349	**0.1133**
175	0.7808	0.7510	0.9965	0.663	0.7973	0.4136	0.5401	0.4550	0.7426	1.0205	0.3381	0.4736	0.553	0.336	**0.1550**
200	0.7818	0.7953	0.9171	0.735	0.7353	0.6102	0.6475	0.3304	0.7029	0.9318	0.4960	0.5624	0.501	0.426	**0.2304**
C-One Step Video Sequence [133]															
125	0.8011	0.4856	1.008	0.705	0.5416	1.557	1.9899	0.2554	1.7905	1.0094	0.2724	0.6975	0.516	0.354	**0.1554**
150	0.8683	0.6223	0.9824	0.711	0.6122	1.4351	1.9007	0.3273	1.7083	0.9349	0.3268	0.2991	0.495	0.422	**0.2303**
175	0.906	0.5092	1.0156	0.721	0.6128	1.3498	1.9078	**0.3386**	1.8723	0.9844	0.4408	0.3482	0.510	0.434	0.3932
200	0.9129	0.4329	0.9921	0.670	0.5995	1.3349	1.9481	0.3253	1.8379	0.9909	0.3997	0.3655	0.452	0.350	**0.1059**
D-Intelligent room Video Sequence [130]															
125	1.4567	1.3988	1.4171	0.690	0.1963	0.5857	1.1885	0.8178	1.8735	0.9704	0.3645	0.3029	0.498	0.428	**0.0779**
150	1.5252	1.4158	1.5131	0.730	0.1969	0.551	1.3217	0.9316	1.8643	1.1794	0.3217	0.2914	0.543	0.337	**0.0766**
175	1.423	1.0197	1.359	0.729	0.1738	0.6131	1.0803	0.9584	1.9933	1.0803	0.5869	0.3750	0.471	0.365	**0.0574**
200	1.4941	1.8276	1.3268	0.703	0.3352	0.4061	1.0670	0.8293	1.8870	0.9751	0.4444	0.2929	0.524	0.403	**0.1743**
E-HighwayI_raw video sequence [132]															
125	1.3371	1.3876	1.327	0.713	0.8579	0.783	1.1378	0.6341	1.7356	0.989	1.5055	1.0860	0.459	0.407	**0.2017**
150	1.263	1.6452	1.183	0.685	0.9217	0.882	1.0113	0.6339	1.2563	1.144	1.3726	1.9462	0.481	0.423	**0.1339**
175	1.739	1.4875	1.293	0.725	0.9365	0.761	1.8987	0.7672	1.6571	1.728	1.2266	1.9827	0.522	0.360	**0.1673**
200	1.482	1.4619	1.217	0.701	0.9324	0.883	1.0923	0.6383	1.5427	0.925	1.6993	1.4953	0.506	0.355	**0.1778**
F-Crowd Video Sequence [134]															
125	1.492	0.7742	1.428	0.684	1.283	1.381	0.2416	0.9605	0.5143	1.758	0.8681	1.0635	0.492	0.397	**0.1220**
150	1.338	0.8175	1.582	0.725	1.638	1.881	0.3704	0.8102	0.9484	1.375	0.7981	1.6374	0.513	0.388	**0.1335**
175	1.826	0.8542	1.726	0.736	1.184	0.372	0.3699	0.9831	0.7563	1.274	0.8721	1.6194	0.554	0.435	**0.1191**
200	1.394	0.8321	1.381	0.716	1.346	1.853	0.4360	0.9179	0.8945	1.836	0.7595	1.4604	0.538	0.383	**0.0829**

Table 3.8: Comparison of methods in terms of Pixel Classification Measure (PCM)

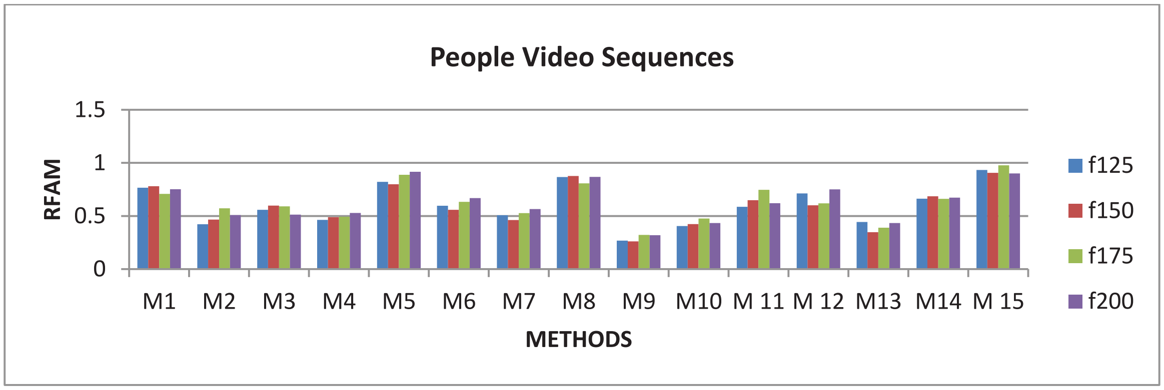
F. No.	Category I				Category II			Category III			Category IV				
	Bradski [98]	**Kim *et al.*[33]**	**Liu *et al.* [34]**	**Meier and Ngan [100]**	**McFarlane *et al.* [27]**	**Wren *et al.*[31]**	**Zivkovic[32]**	**Kushwaha *et al.* [26]**	**Cucchiara *et al.*[111]**	**Oliver *et al.*[110]**	**Hsia *et al.*[120]**	**Khare *et al.*[121]**	**Kim *et al.* [116]**	**Chien *et al.* [117]**	**proposed**
A-People Video Sequences [129]															
125	0.6081	0.6731	0.569	0.567	0.9517	0.8183	0.7318	0.8788	0.7215	0.9142	0.8076	0.8454	0.640	0.729	**0.9824**
150	0.5852	0.6098	0.553	0.536	0.9323	0.7437	0.7604	0.8531	0.7432	0.8699	0.7610	0.7790	0.650	0.743	**0.9335**
175	0.5578	0.6218	0.543	0.572	0.9304	0.7302	0.7547	0.8491	0.6925	0.8791	0.7443	0.7598	0.666	0.730	**0.9693**
200	0.5595	0.5780	0.595	0.528	0.9468	0.7667	0.7115	0.8549	0.6876	0.8614	0.7640	0.8127	0.610	0.693	**0.9349**
B-Camera2_070605 Video Sequence [131]															
125	0.7148	0.8377	0.7195	0.511	0.8985	0.7846	0.7993	0.8559	0.7512	0.8493	0.8125	0.8076	0.641	0.749	**0.9288**
150	0.7879	0.8284	0.7629	0.564	0.8550	0.7862	0.7176	0.8482	0.7634	0.8396	0.8419	0.8180	0.676	0.725	**0.8808**
175	0.8066	0.8294	0.7756	0.521	0.8645	0.7873	0.7317	0.8273	0.7688	0.843	0.8462	0.8230	0.615	0.738	**0.8988**
200	0.8093	0.8375	0.7653	0.533	0.8581	0.7733	0.7427	0.8658	0.7529	0.858	0.8177	0.7960	0.659	0.767	**0.8613**
C-One Step Video Sequence [133]															
125	0.6569	0.6709	0.695	0.537	0.9373	0.6411	0.5014	0.8691	0.8496	0.9581	0.5762	0.6810	0.648	0.723	**0.9592**
150	0.667	0.6803	0.6697	0.575	0.9498	0.6379	0.4949	0.8546	0.8527	**0.9358**	0.5646	0.7645	0.650	0.764	0.9546
175	0.6561	0.6797	0.645	0.544	0.9492	0.6381	0.4874	0.8550	0.8984	**0.9875**	0.4817	0.7462	0.662	0.734	0.9637
200	0.6468	0.7732	0.6811	0.528	0.9385	0.655	0.4877	0.8429	0.8782	0.9416	0.5317	0.7449	0.624	0.715	**0.9873**
D-Intelligent room Video Sequence [130]															
125	0.7848	0.7716	0.9566	0.520	0.9566	0.8269	0.5894	0.8882	0.7328	0.9288	0.9015	0.6810	0.624	0.698	**0.9654**
150	0.8218	0.7271	0.9598	0.563	0.9611	0.7421	0.5952	0.8572	0.6492	0.9069	0.9155	0.7645	0.642	0.749	**0.9663**
175	0.8368	0.8343	0.9624	0.554	0.9693	0.8435	0.6002	0.8558	0.7942	0.9176	0.9041	0.7462	0.655	0.714	**0.9717**
200	0.8251	0.7441	0.9685	0.575	0.9538	0.8742	0.6778	0.8895	0.7223	0.9503	0.9315	0.7583	0.631	0.728	**0.9485**
E-HighwayI_raw video sequence [132]															
125	0.538	0.5707	0.592	0.509	0.6419	0.783	0.7817	0.8488	0.6757	0.794	0.5245	0.8490	0.660	0.690	**0.8718**
150	0.564	0.6316	0.562	0.545	0.6150	0.692	0.7067	0.8582	0.5414	0.757	0.4859	0.7911	0.653	0.728	**0.8084**
175	0.568	0.6371	0.544	0.574	0.6211	0.685	0.7878	0.8748	0.5876	0.726	0.4354	0.7363	0.644	0.701	**0.8745**
200	0.581	0.6333	0.572	0.514	0.6607	0.648	0.7041	0.8752	0.5294	0.753	0.4747	0.7494	0.627	0.731	**0.8581**
F-Crowd Video Sequence [134]															
125	0.725	0.8451	0.524	0.577	0.7368	0.656	0.6545	0.8469	0.7553	0.472	0.7755	0.9106	0.675	0.701	**0.9519**
150	0.636	0.8392	0.537	0.561	0.7279	0.685	0.6423	0.7384	0.7114	0.583	0.7718	0.8490	0.669	0.716	**0.9520**
175	0.634	0.8332	0.521	0.510	0.7192	0.613	0.6220	0.7642	0.7016	0.527	0.7546	0.8322	0.658	0.727	**0.9273**
200	0.735	0.8372	0.532	0.523	0.7145	0.639	0.6476	0.7793	0.7812	0.585	0.7490	0.8470	0.618	0.740	**0.9213**

Table 3.9: Computational Time and Consumption Memory for One step video [133]

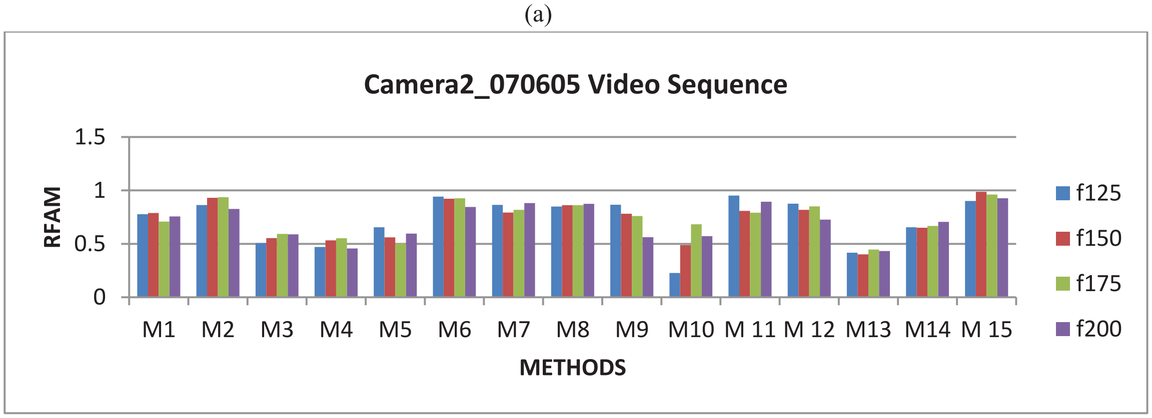
S.no.	Methods	Computational Time (in frame/second)	Memory Consumption (MB)
1	McFarlane *et al.*[27]	1.376	8.68
2	Kim *et al.*[33]	0.722	22.92
3	Zivkovic[32]	1.864	9.40
4	Cucchiara *et al.*[111]	1.625	24.95
5	Hsia *et al.*[120]	1.912	8.64
6	Khare *et al.*[121]	1.753	7.08
7	Kim *et al.* [116]	1.325	11.37
8	Chien *et al.* [117]	1.687	13.35
9	Bradski [98]	0.912	17.62
10	Liu *et al.* [34]	1.412	30.92
11	Wren *et al.*[31]	1.443	25.17
12	Kushwaha *et al.* [26]	0.824	15.26
13	Oliver *et al.*[110]	1.392	20.62
14	Meier and Ngan [100]	1.427	18.35
15	The Proposed Method	1.232	3.90

In Fig. 3.10-3.16(a-f), Y-axis shows the different quantitative measure such as RFAM, MP, RPM, NCC, PSNR, NAE, PCM and X-axis shows the frame number. From Fig. 3.10-3.16(a-f), one can conclude that proposed method performed better than other methods in different quantitative measures such as RFAM, MP, RPM, NCC, PSNR, NAE, and PCM.

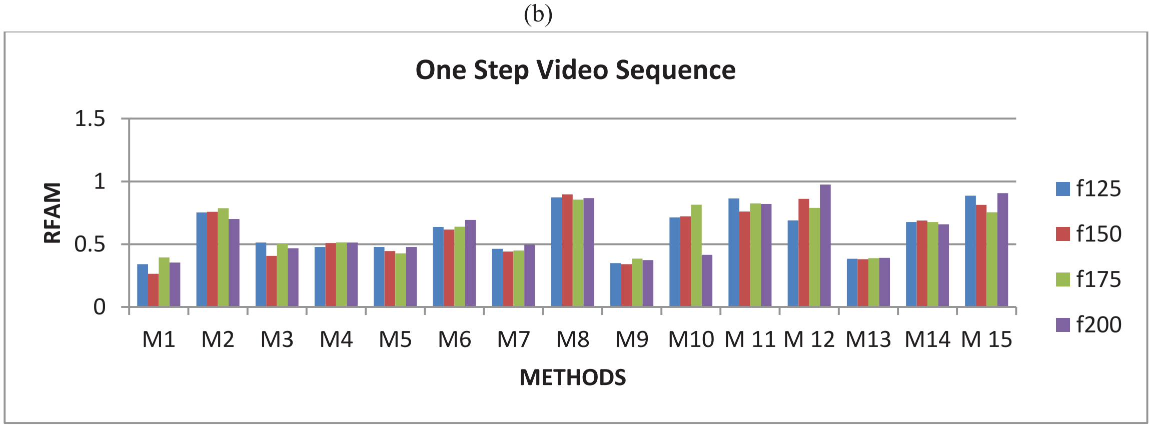
M1: Bradski[98]; **M2:** Kim *et al.*[33]; **M3**: Liu *et al.* [34]; **M4**: Meier and Ngan [100]; **M5**: McFarlane *et al.* [27]; **M6**: Wren *et al.*[31]; **M7**: Zivkovic[32];**M8:** Kushwaha *et al.* [26] ; **M9:** Cucchiara *et al.*[111]; **M10:** Oliver *et al.*[110]; **M11:** Hsia *et al.*[120]; **M12:** Khare *et al.*[121]; **M13**: Kim *et al.* [116]; **M14**: Chien *et al.* [117]; **M15**:Proposed Method



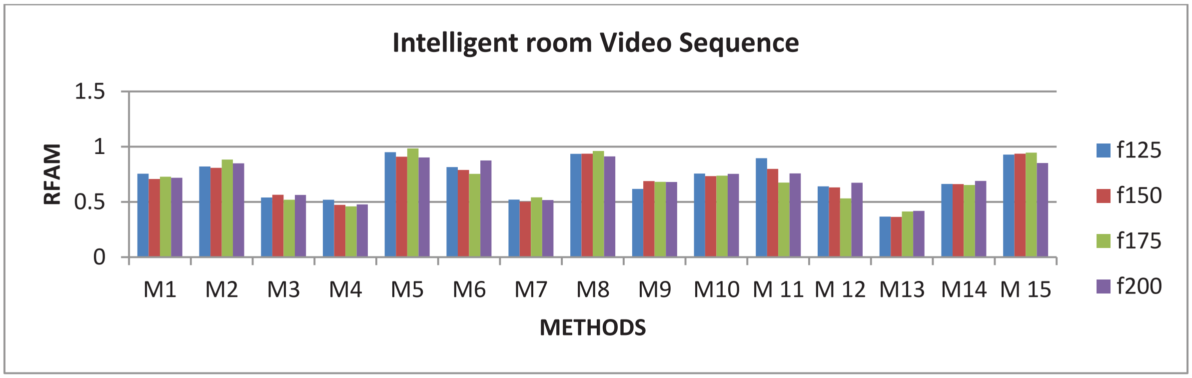
(a)



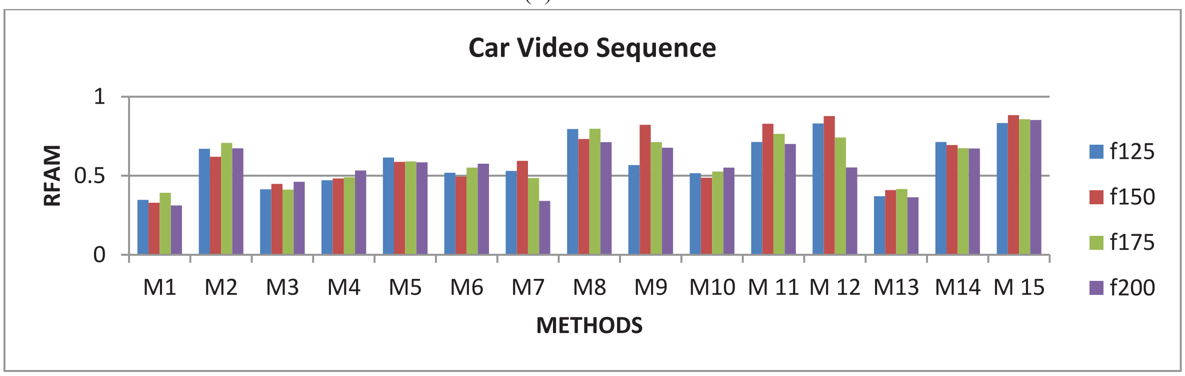
(b)



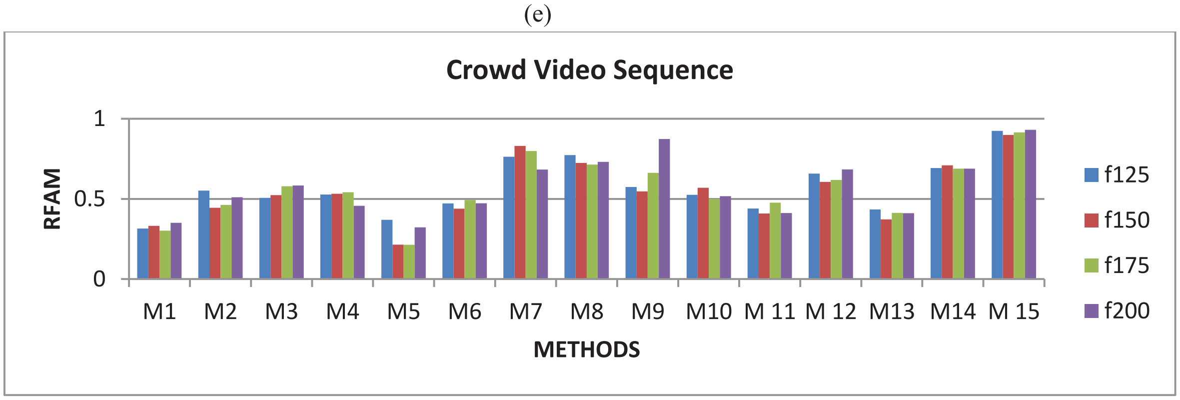
(c)



(d)

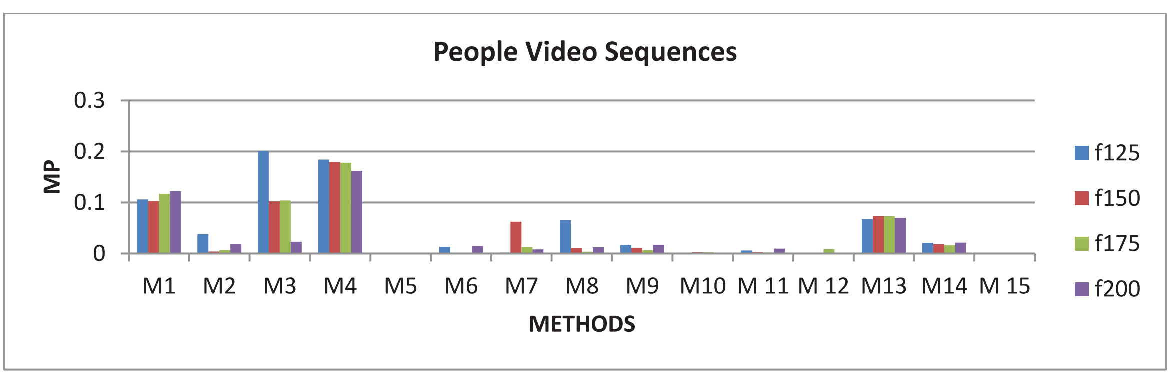


(e)

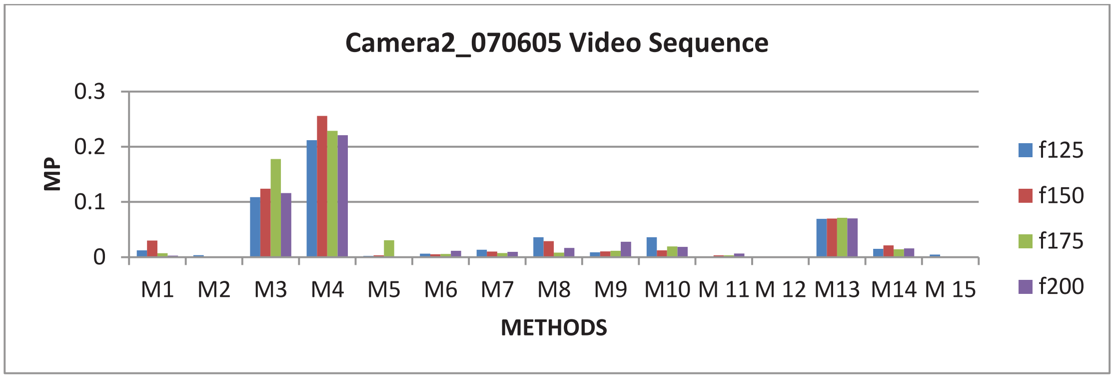


(f)

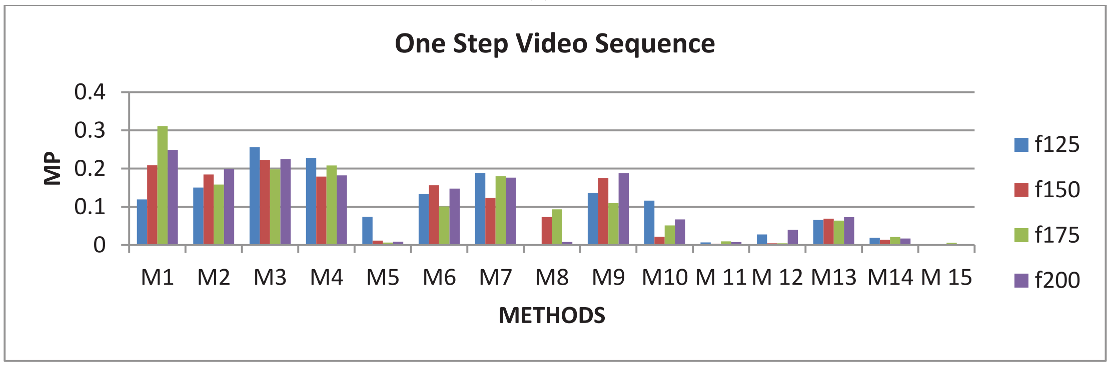
Figure 3.10: (a-f) RFAM variations with respect to frame no. for different Test cases



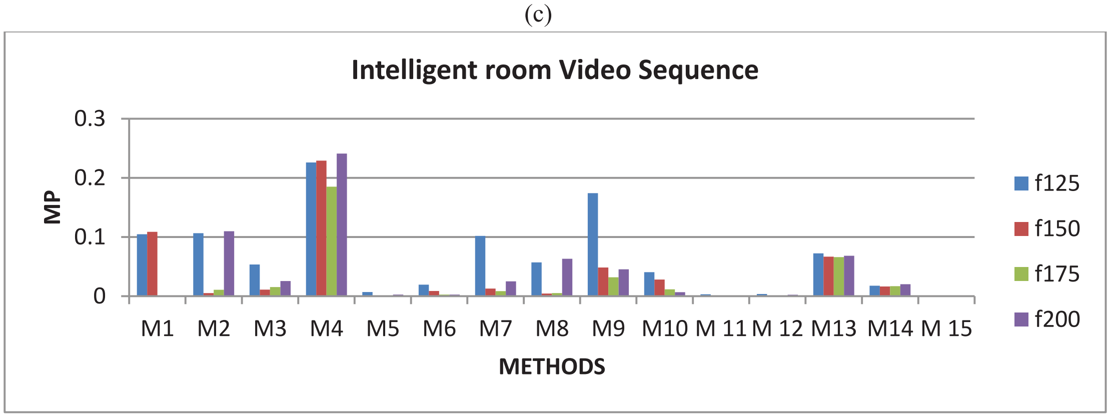
(a)


Camera2_070605 Video Sequence
MP
0.3
0.2
0.1
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

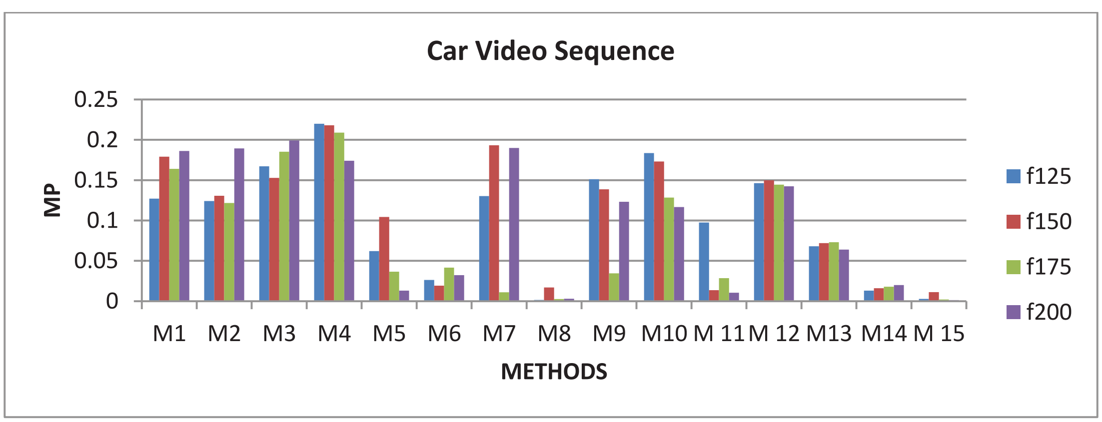
(b)


One Step Video Sequence
MP
0.4
0.3
0.2
0.1
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

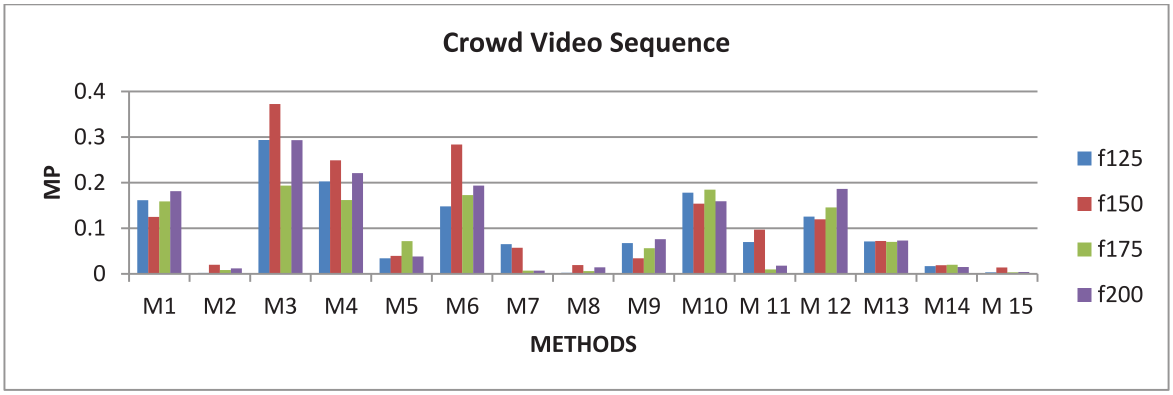
(c)


Intelligent room Video Sequence
MP
0.3
0.2
0.1
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(d)

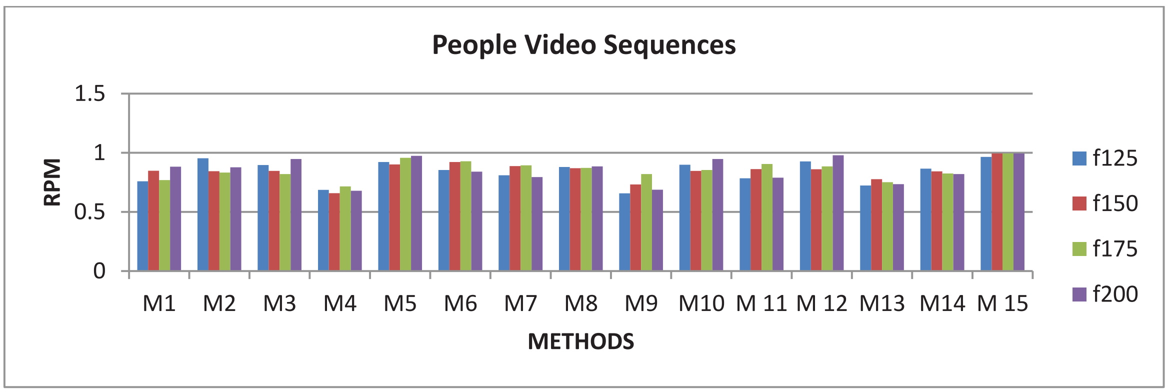

Car Video Sequence
MP
0.25
0.2
0.15
0.1
0.05
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(e)

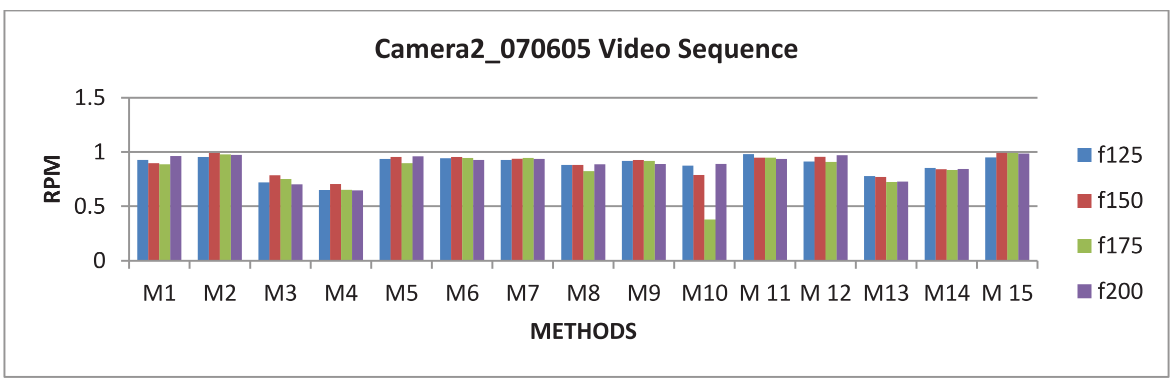


(f)

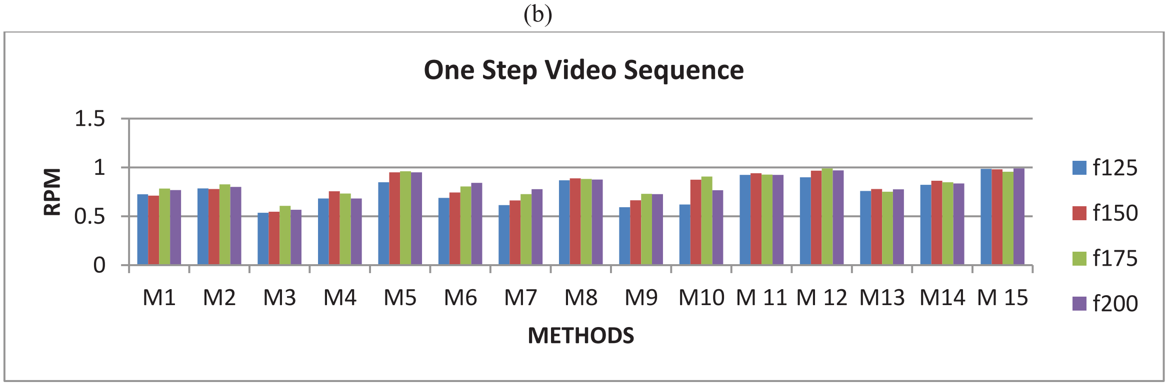
Figure 3.11: (a-f) MP variations with respect to frame no. for different Test cases



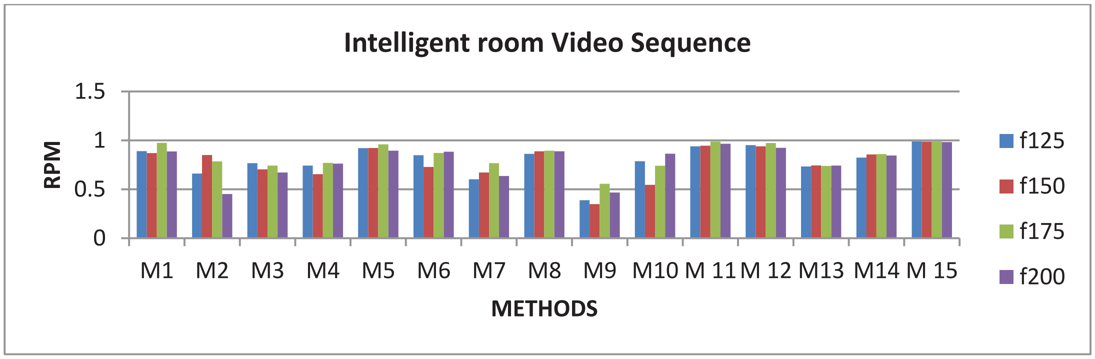
(a)



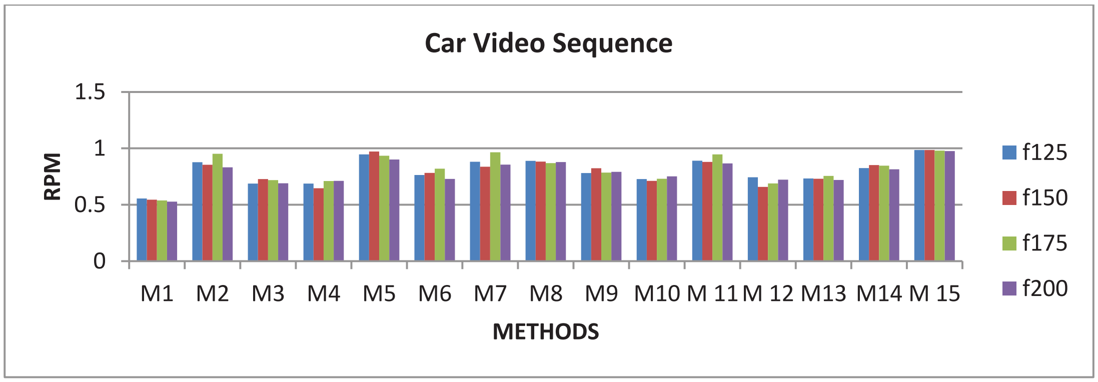
(b)



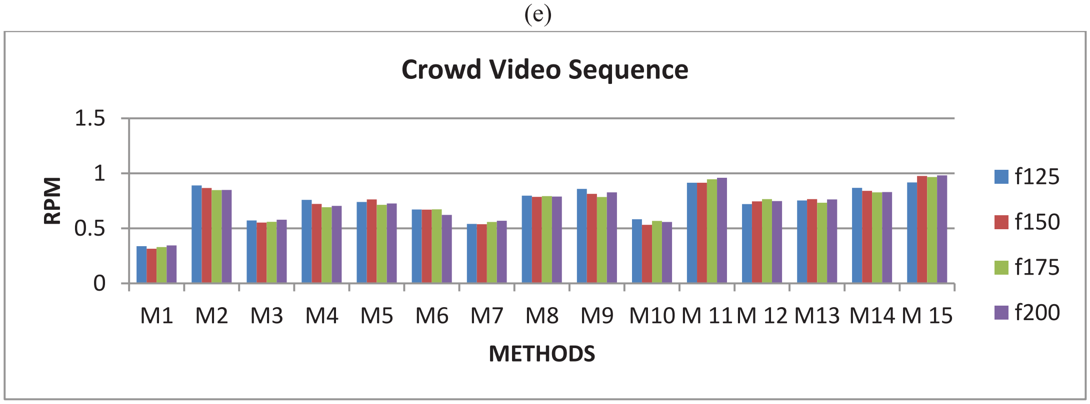
(c)



(d)

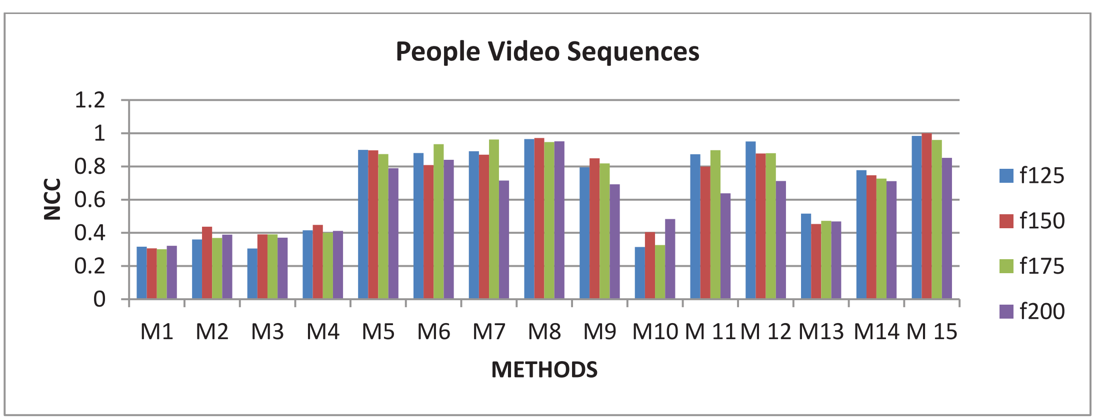


(e)

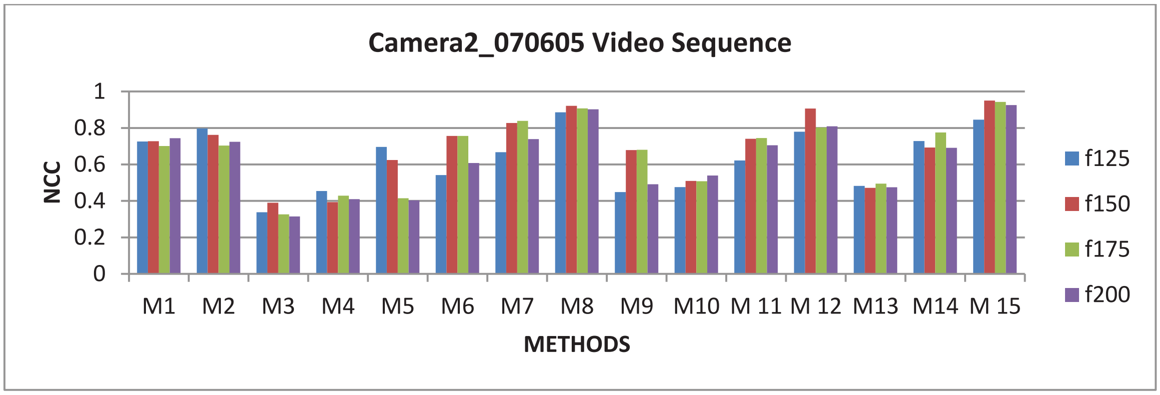


(f)

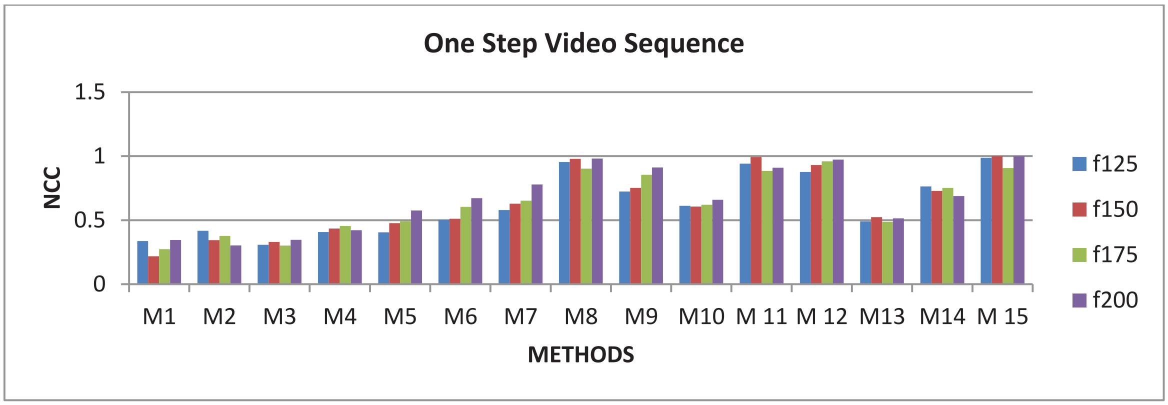
Figure 3.12: (a-f) RPM variations with respect to frame no. for different Test cases



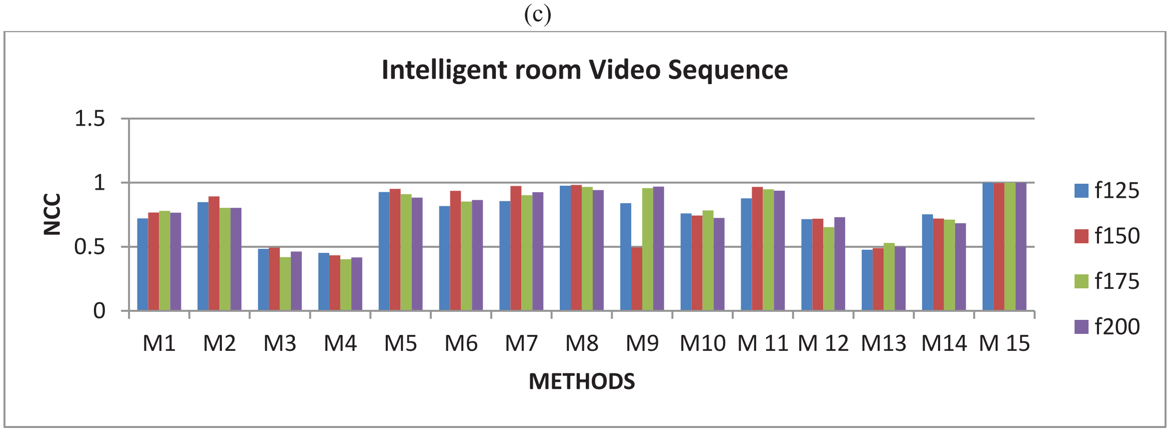
(a)


Camera2_070605 Video Sequence
NCC
1
0.8
0.6
0.4
0.2
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

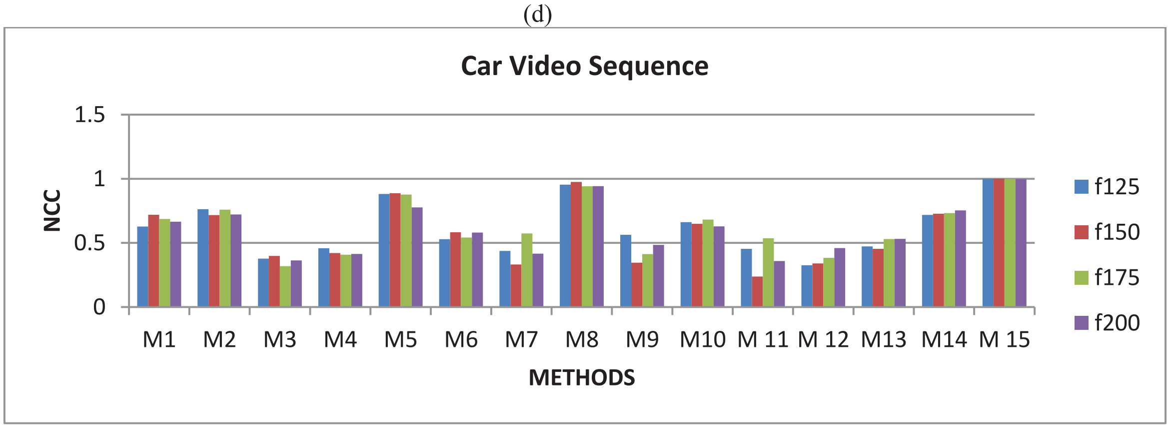
(b)


One Step Video Sequence
NCC
1.5
1
0.5
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

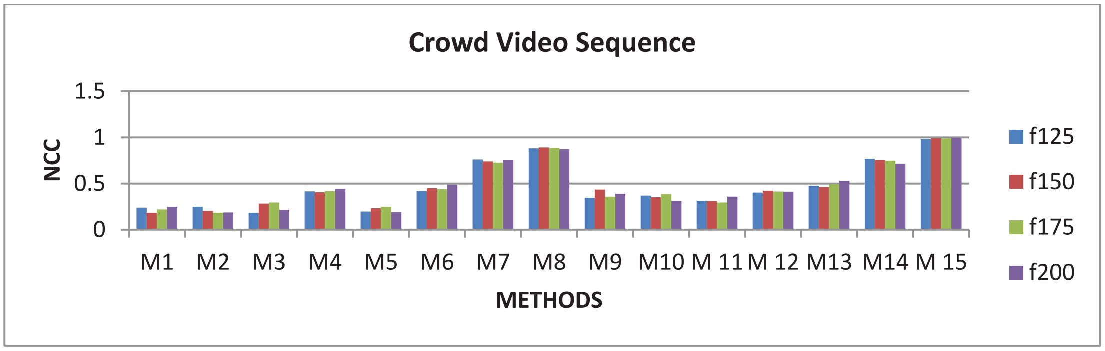
(c)


Intelligent room Video Sequence
NCC
1.5
1
0.5
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(d)

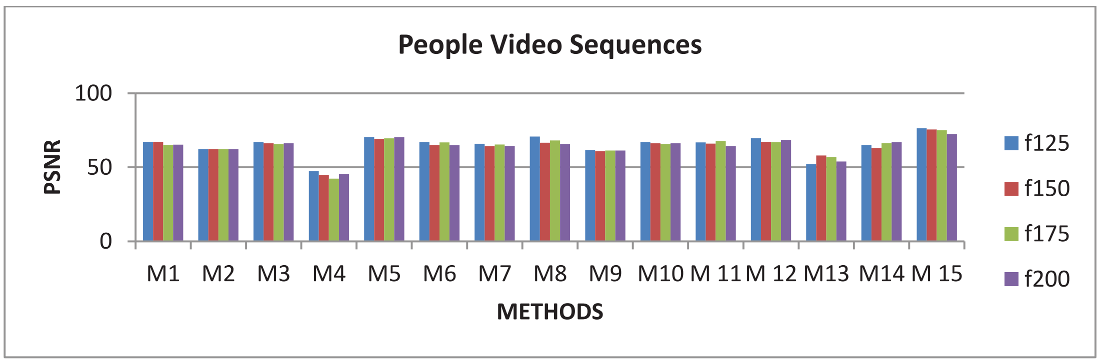

Car Video Sequence
NCC
1.5
1
0.5
0
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(e)

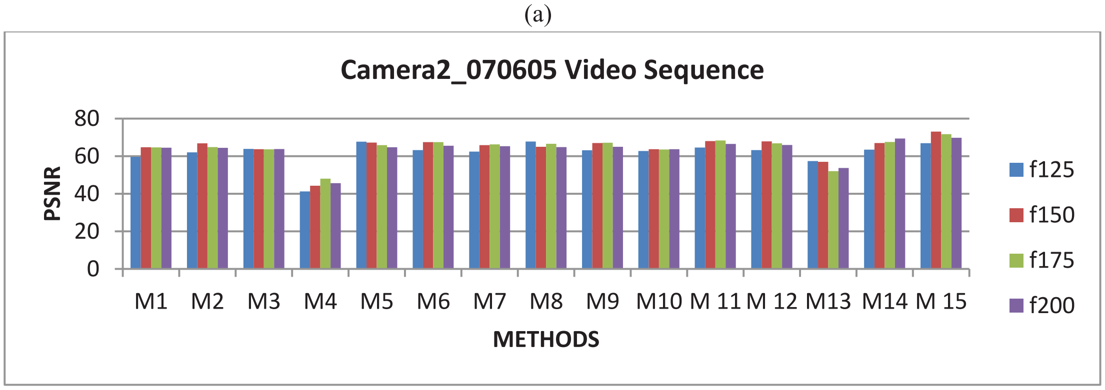


(f)

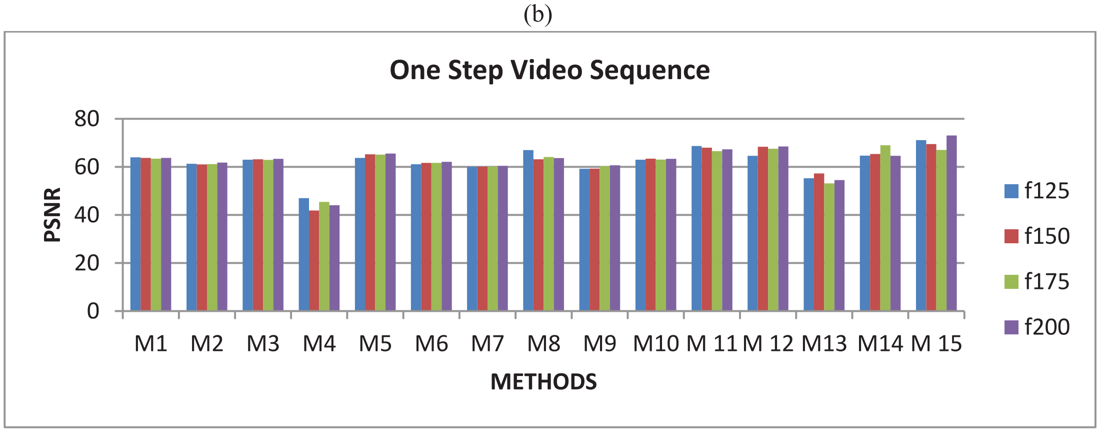
Figure 3.13: (a-f) NCC variations with respect to frame no. for different Test cases



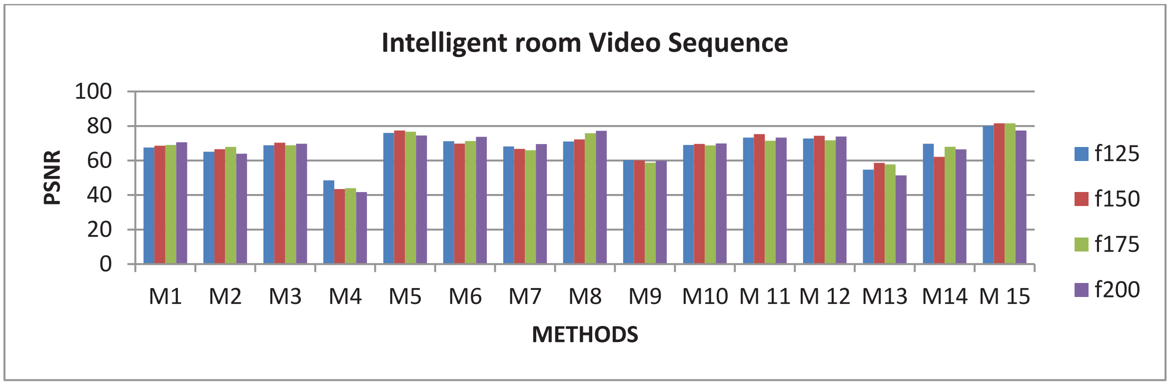
(a)



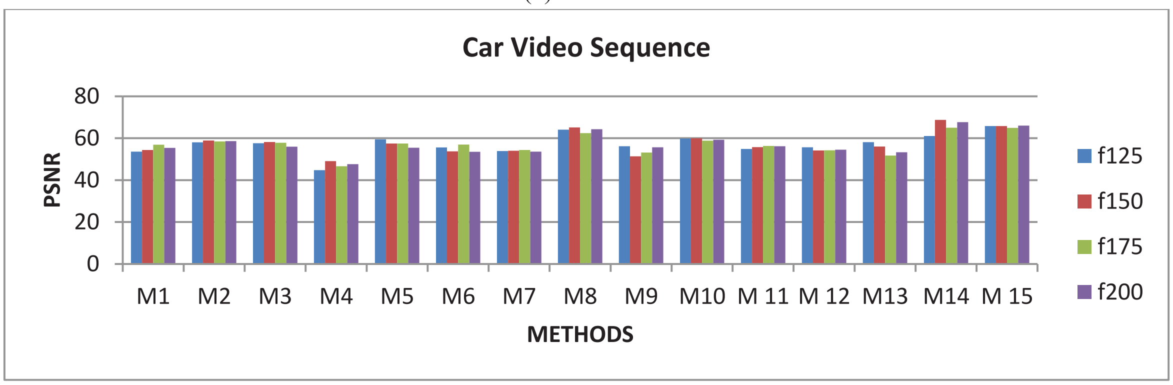
(b)



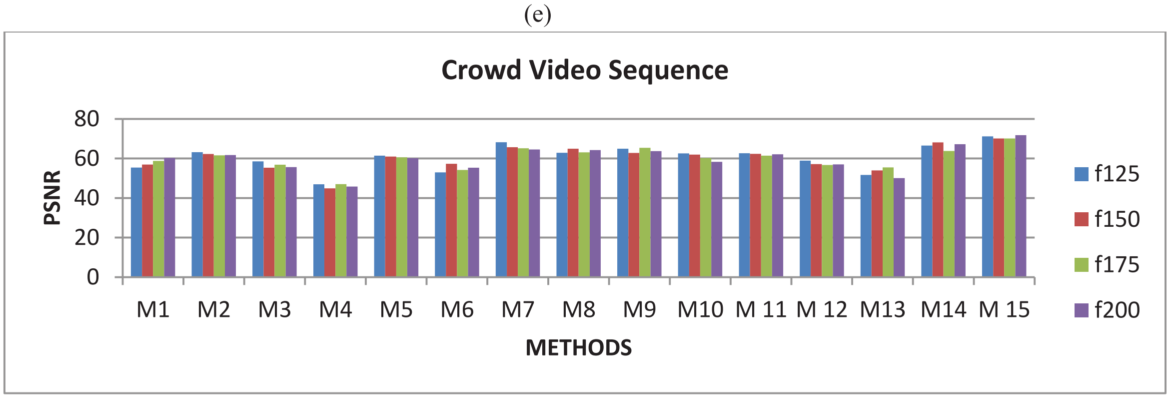
(c)



(d)

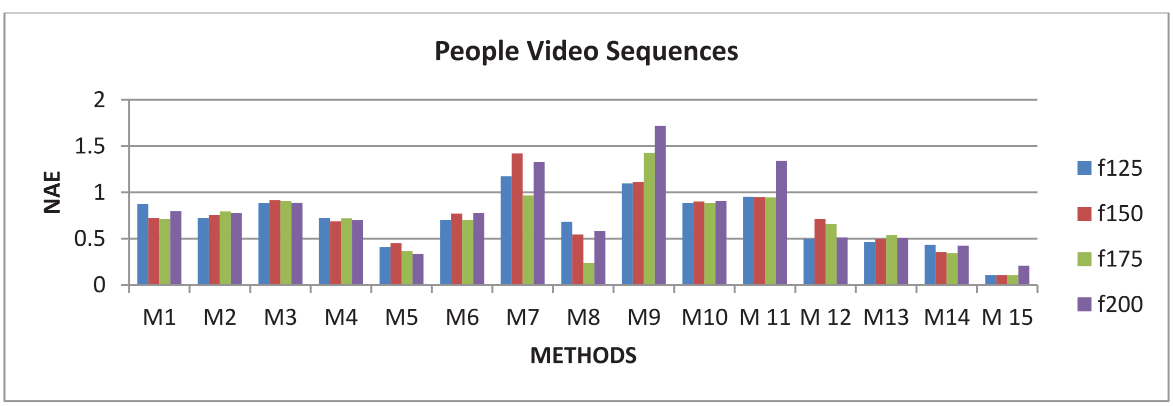


(e)

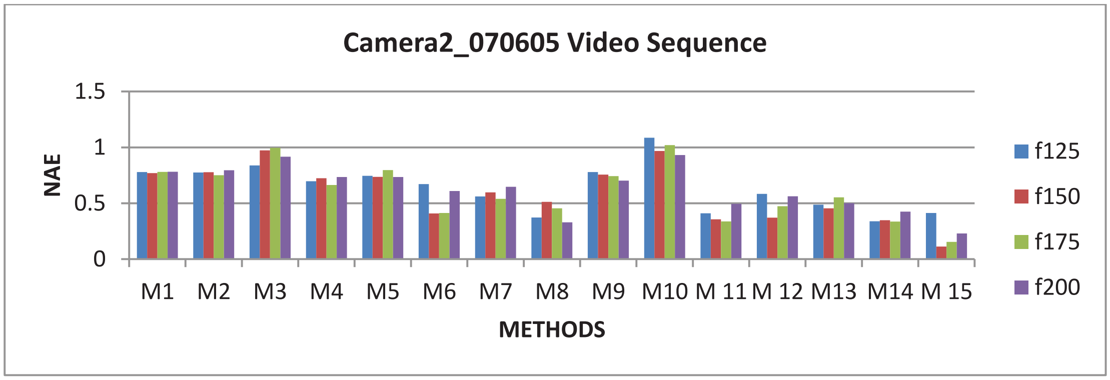


(f)

Figure 3.14: (a-f) PSNR variations with respect to frame no. for different Test cases

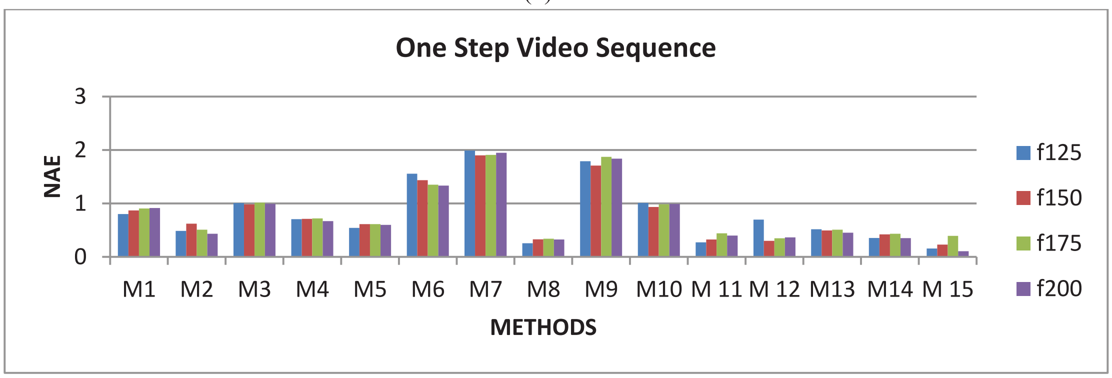


(a)



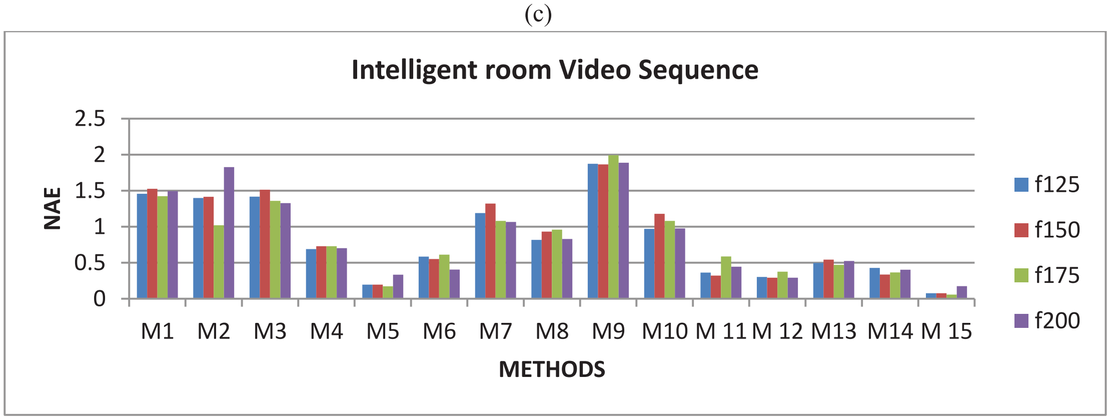
Camera2_070605 Video Sequence
1.5
1
0.5
0
NAE
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(b)



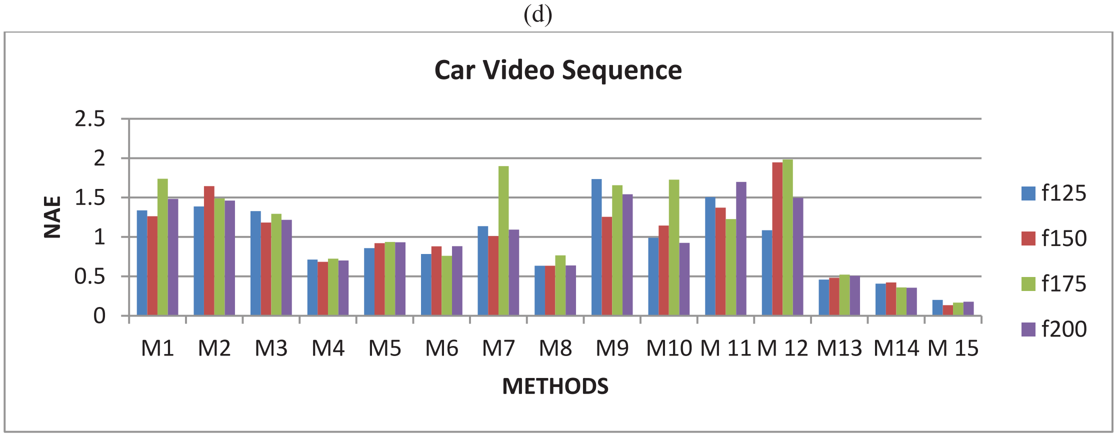
One Step Video Sequence
3
2
1
0
NAE
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(c)



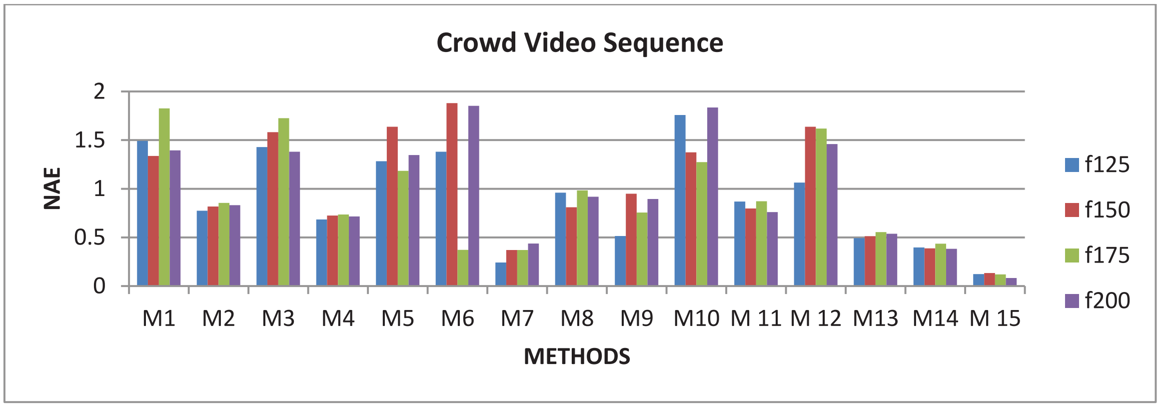
Intelligent room Video Sequence
2.5
2
1.5
1
0.5
0
NAE
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(d)



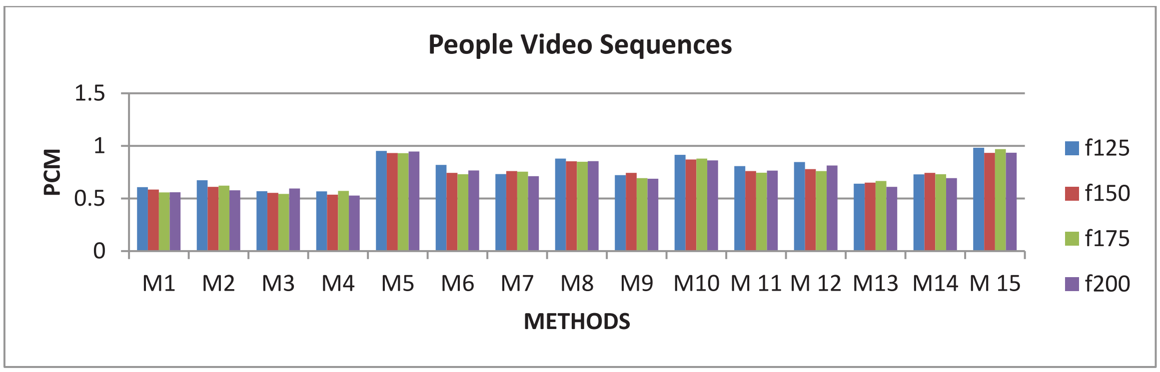
Car Video Sequence
2.5
2
1.5
1
0.5
0
NAE
M1 M2 M3 M4 M5 M6 M7 M8 M9 M10 M 11 M 12 M13 M14 M 15
METHODS
f125
f150
f175
f200

(e)

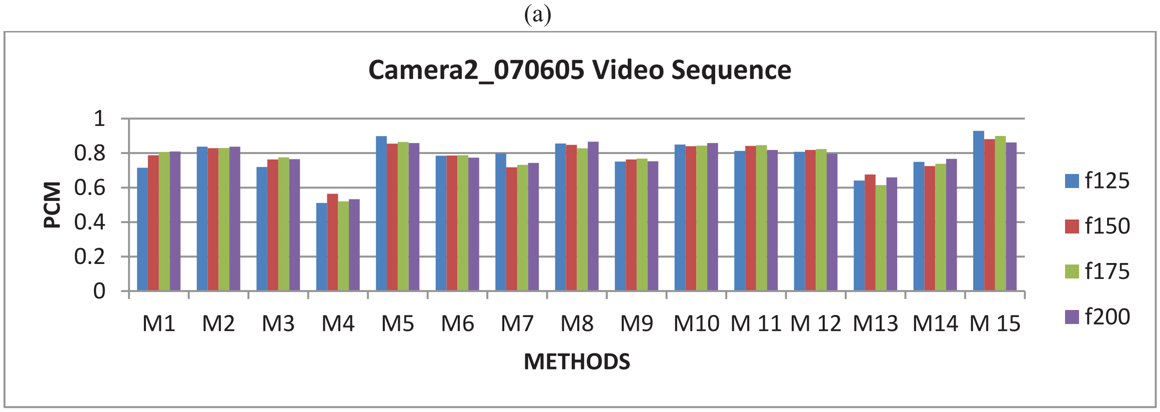


(f)

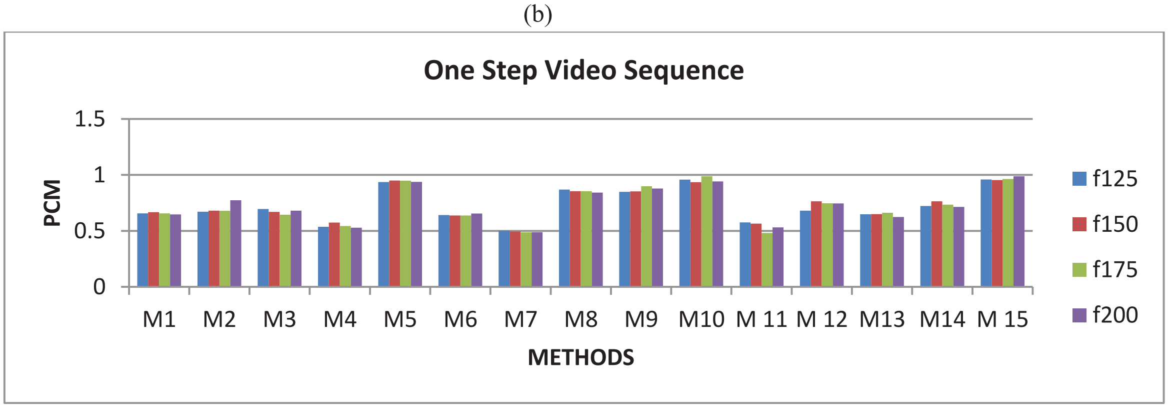
Figure 3.15: (a-f) NAE variations with respect to frame no. for different Test cases



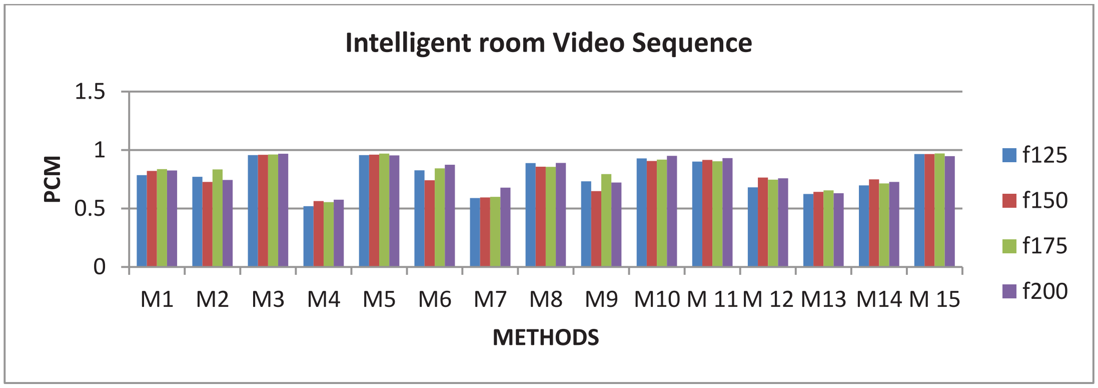
(a)



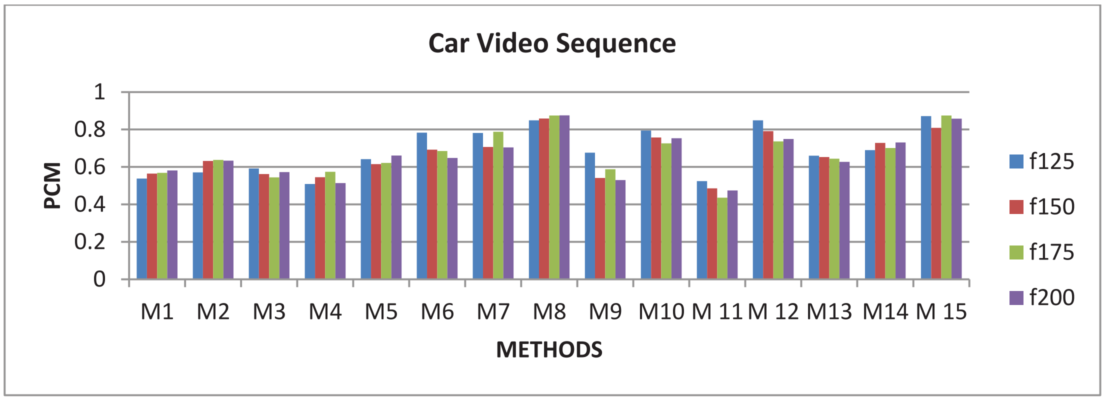
(b)



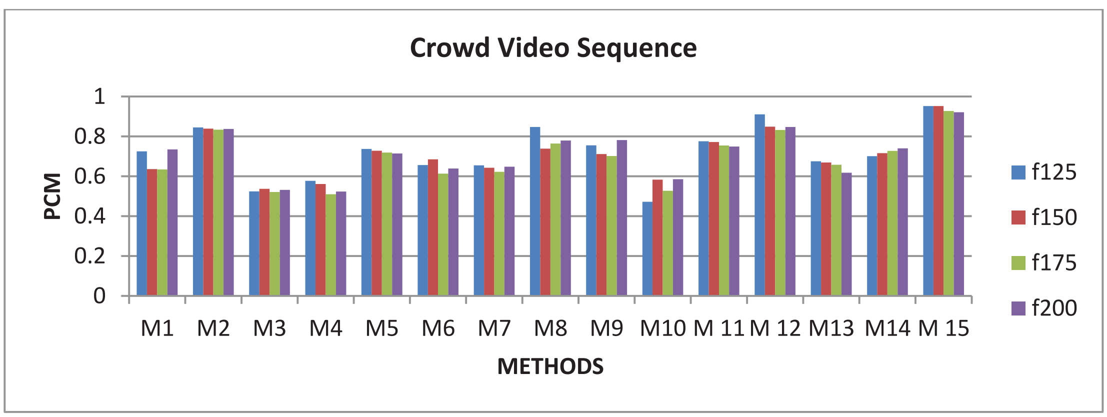
(c)



(d)



(e)



(f)

Figure 3.16: (a-f) PCM variations with respect to frame no. for different Test cases