UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

NEIGHBORHOOD-LEVEL LEARNING TECHNIQUES FOR NONPARAMETRIC SCENE MODELS

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

NICHOLAS ALLEN MOULD
Norman, Oklahoma
2012

NEIGHBORHOOD-LEVEL LEARNING TECHNIQUES FOR NONPARAMETRIC SCENE MODELS

A DISSERTATION APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

Joseph P. Havlicek

Monte P. Tull

John K. Antonio

Ronald D. Barnes

James J. Sluss Jr.

This dissertation is dedicated to my mother

Carol Ann West

and to the memory of my father

Michael Edward Mould.

Acknowledgements

I would like to express my sincere gratitude to my primary research and dissertation advisor Dr. Joseph Havlicek for his guidance and support throughout this undertaking. Dr. Monte Tull for encouraging me to pursue a Ph.D., Dr. John Antonio for many things, and Drs. Ron Barnes and James Sluss for helping to create a rich, vibrant research atmosphere.

I would also like to thank my co-worker Chuong Nguyen for putting up with my absurd behavior over the course of the past five years.

Table of Contents

Appendix

List of Tables

List of Figures

Abstract

NEIGHBORHOOD-LEVEL LEARNING TECHNIQUES FOR NONPARAMETRIC SCENE MODELS

Nicholas Allen Mould, Ph.D.
The University of Oklahoma, 2012

Supervisor: Joseph P. Havlicek

Scene model based segmentation of video into foreground and background structure has long been an important and ongoing research topic in image processing and computer vision. Segmentation of complex video scenes into binary foreground/background label images is often the first step in a wide range of video processing applications. Examples of common applications include surveillance, Traffic Monitoring, People Tracking, Activity Recognition, and Event Detection.

A wide range of scene modeling techniques have been proposed for identifying foreground pixels or regions in surveillance video. Broadly speaking, the purpose of a scene model is to characterize the distribution of features in an image block or pixel over time. In the majority of cases, the scene model is used to represent the distribution of background features (background modeling) and the distribution of foreground features is assumed to be uniform or Gaussian. In other cases, the model characterizes the distribution of foreground and background values and the segmentation is performed by maximum likelihood.

Pixel-level scene models characterize the distributions of spatio-temporally localized image features centered about each pixel location in video over time. Individual video frames are segmented into foreground and background regions based on a comparison between pixel-level features from within the frame under segmentation and the appropriate elements of the scene model at the corresponding pixel location. Prominent pixel level scene models include the Single Gaussian, Gaussian Mixture Model and Kernel Density Estimation.

Recently reported advancements in scene modeling techniques have been largely based on the exploitation of local coherency in natural imagery based on integration of neighborhood information among nonparametric pixel-level scene models. The earliest scene models inadvertently made use of neighborhood information because they modeled images at the block level. As the resolution of the scene models progressed, textural image features such as the spatial derivative, local binary pattern (LBP) or Wavelet coefficients were employed to provide neighborhood-level structural information in the pixel-level models. In the most recent case, Barnich and Van DroogenBroeck proposed the Visual Background Extractor (ViBe), where neighborhood-level information is incorporated into the scene model in the learning step. In ViBe, the learning function is distributed over a small region such that new background information is absorbed at both the pixel and neighborhood level.

In this dissertation, I present a nonparametric pixel level scene model based on several recently reported stochastic video segmentations algorithms. I propose new stochastic techniques for updating scene models over time that are focused on the incorporation of neighborhood-level features into the model learning process and demonstrate the effectiveness of the system on a wide range of challenging visual tasks. Specifically, I propose a model maintenance policy that is based on the replacement of outliers within each nonparametric pixel level model through kernel density estimation (KDE) and a neighborhood diffusion procedure where information sharing between adjacent models having significantly different shapes is discouraged. Quantitative results are compared using the well known percentage correct classification (PCC) and a new probability correct classification (PrCC) metric, where the underlying models are scrutinized prior to application of a final segmentation threshold. In all cases considered, the superiority of the proposed model with respect to the existing state-of-the-art techniques is well established.

Chapter 1

Introduction

Scene model based segmentation of video into foreground and background structure has long been an important and ongoing research topic in image processing and computer vision. Segmentation of complex video scenes into binary foreground/background label images is often the first step in a wide range of video processing applications. Examples of common applications include surveillance [2, 3, 6, 16, 23, 49], traffic monitoring [26, 47], people tracking [29, 32, 102], activity recognition [12, 29, 94], and event detection [12, 80]. Recently, the need for improved video segmentation algorithms has greatly increased due to growing widespread use of surveillance technology throughout the world. In 2009 the U.S. Air Force recorded 24 years of video from unmanned aerial vehicles and this figure is expected to increase significantly in the next few years [9]. The U.S. National Geospatial Intelligence Agency estimates that 16,000 trained analysts would be required to provide real-time monitoring of the existing airborne surveillance systems deployed at this time [9].

A wide range of scene modeling techniques have been proposed for identifying foreground pixels or regions in surveillance video. Broadly speaking, the purpose of a scene model is to characterize the distribution of features in an image block or pixel over time. Labeled segmentation images are one of many possible outputs that may be drawn from a scene model. Frame segmentations are obtained based on a comparison between recently observed unsegmented frames and the model of the scene. In the majority of cases, the scene model is used to represent the distribution of background features (background modeling) and the distribution of foreground features is assumed to be uniform or Gaussian [3, 16, 22, 46, 83, 95, 102]. In other cases, the model characterizes the distribution of foreground and background values and the segmentation is performed by maximum likelihood.

Pixel-level scene models characterize the distributions of spatio-temporally localized image features centered about each pixel location in video over time. Individual video frames are segmented into foreground and background regions based on a comparison between pixel-level features from within the frame under segmentation and the appropriate elements of the scene model at the corresponding pixel location. Prominent pixel level scene models include the Single Gaussian [1, 6, 8, 20, 37, 69, 75, 97, 102], Gaussian Mixture Model [26, 30, 31, 36, 41, 42, 57, 82, 93, 94, 100, 104, 104, 108–110], Kernel Density Estimation [3, 23, 24, 28, 71, 79, 89, 90] and many of the early reference image comparison techniques [22, 34, 45, 52, 59, 83, 85, 92].

Recently reported advancements in scene modeling techniques have been largely based on the exploitation of local coherency in natural imagery based on integration of neighborhood information among nonparametric pixel-level scene models [3, 90]. The earliest scene models inadvertently made use of neighborhood information because they modeled images at the block level [39]. As the resolution of the scene models progressed, textural image features such as the spatial derivative [37, 41, 69], local binary pattern (LBP) [33] or Wavelet coefficients [47] were employed to provide neighborhood-level structural information in the pixel-level models. In 2002, Elgammal *et al.,* made use of neighboring pixel-level nonparametric models in the segmentation step to eliminate a large majority of false foreground detections due to dynamic components of the background scene [23]. In the most recent case, Barnich and Van DroogenBroeck proposed the Visual Background Extractor (ViBe), where neighborhood-level information is incorporated into the scene model in the learning step. In ViBe, the learning function is distributed over a small region such that new background information is absorbed at both the pixel and neighborhood level [3].

In Chapter 3, I present a nonparametric pixel level scene model based on several recently reported stochastic video segmentations algorithms. I propose new stochastic techniques for updating scene models over time that are focused on the incorporation of neighborhood-level features into the model learning process and demonstrate the effectiveness of the system on a wide range of challenging visual tasks. Specifically, I propose a model maintenance policy that is based on the replacement of outliers within each nonparametric pixel level model through kernel density estimation (KDE) and a neighborhood diffusion procedure where information sharing between adjacent models having significantly different shapes is discouraged. Quantitative results are compared using the well known percentage correct classification (PCC) and a new probability correct classification (PrCC) metric presented in Chapter 4, where the underlying models are scrutinized prior to application of a final segmentation threshold. In all cases considered, the superiority of the proposed model with respect to the existing state-of-the-art techniques is well established.

Chapter 2

Literature Review

A comprehensive review and analysis of existing scene modeling methods is presented in this chapter. Historically, these types of models have been presented as background models and the foreground probabilities were assumed to follow uniform or Gaussian distributions [3, 16, 22, 46, 83, 95, 102]. However, because a large number of models characterize the distributions of both foreground and background image features, I propose the term *scene model* and use it throughout this dissertation to refer to the modeling of image structure in video. With the remainder of this chapter, a review of model based video segmentation literature is presented using a chronological taxonomy based on scene model representation and segmentation functions.

2.1 Reference Image Comparison (1979-2009)

Reference image based scene modeling techniques were the earliest types of video segmentation algorithms to appear [12,22,34,35,39,40,45,52,59,60,83,92, 101]. In these types of scene models a reference image was used to characterize background structure from previously observed video frames. Incoming frames were segmented into foreground and background based on a comparison to the reference image at the corresponding pixel or block locations. A wide variety of image features have been used in reference images including the spatial mean and variance [39, 40], spatial mean, variance and Sobel derivative [38], linear or quadratic regression coefficients [35, 92], RGB or grayscale intensity [12, 22, 83, 85], grayscale intensity and simple temporal difference [45, 52], circular shift moment [60], computational color [34], grayscale intensity and optical flow [101], and principal features [59]. The earliest scene models appearing between 1977 and 1998 divided video into blocks to reduce the memory storage requirements of the reference image and the computational complexity of the comparison function [12, 22, 34, 45, 52, 59, 101]. Later models reported between 1988 and 2009 were able to take advantage of increased computer memory sizes and maintain pixel level reference images. However, the complexity of the comparison functions remained relatively unchanged [35, 38–40, 60, 92].

2.1.1 Block Measurements

Limited by the technology of the time period, the earliest scene models divided video frames into rectangular nonoverlapping blocks because they lacked the computational resources to process pixel level features. A reference image

$$I_{\mathrm{Ref},k}(\mathbf{p}) = \{\phi_1, \phi_2, \phi_3, \ldots, \phi_N\} \tag{2.1}$$

composed of N block features at each block location $\mathbf{p}$ was maintained and used to identify foreground blocks according to

$$L_k(\mathbf{p}) = \begin{cases} \text{Foreground} & : ||I_k(\mathbf{p}) - I_{\mathrm{Ref},k-1}(\mathbf{p})|| > T_k(\mathbf{p}) \\ \text{Background} & : ||I_k(\mathbf{p}) - I_{\mathrm{Ref},k-1}(\mathbf{p})|| \leq T_k(\mathbf{p}) \end{cases} , \tag{2.2}$$

where $||\cdot||$ was a suitable norm on $I_k, I_{\mathrm{Ref},k} \in \mathbb{N}^2 \times \mathbb{R}^N$, and $T_k(\mathbf{p})$ was a possibly time and spatially varying threshold.

In [39] an adaptive reference image was constructed by dividing the image lattice into a 2D array of six by four nonoverlapping blocks called Geopixels and computing the average mean and standard deviation of the grayscale pixel intensities within each block over time. Video frames were segmented by dividing them into Geopixels and then comparing the mean and standard deviation of each block to the average values from the corresponding block in the reference image using the Yakimovsky likelihood ratio proposed in [103]. According to [39], the use of second order statistics appears to be more robust than simple thresholding of the grayscale values when used on real-world imagery.

Later in [38] the segmentations provided by the Geopixel technique were combined with edge information obtained by application of the Sobel operator [10] to the current and previous video frames. A clever ratio of the coincident region boundary and edge points was then used to classify moving foreground regions as leading, trailing or leading /trailing. Finally, the regions were grown or decayed based on their classification to better estimate the true foreground object mask [38]. This method was effective for recovering the masks of objects without holes that were composed of few grayscale intensities. In the case of objects exhibiting high resolution textural features or holes the performance was severely degraded.

In [40] foreground regions were identified by the Geopixel technique proposed in [39] and used to generate a first order difference picture (FODP). An FODP was generated by keeping track of the number of times that a specific Geopixel had been determined to be a component of the foreground. In [40] the monotonicity, fillness and velocities associated with foreground regions within the FODPs and SODPs (second order difference pictures) were estimated and used in the analysis of multiple foreground objects. A drawback of this approach was that foreground objects were required to exhibit smooth motion and maintain a large contrast to the background structure.

In [35] two primitive texture models were introduced and used to construct one of the first reference images in the textural feature space. Video frames were again divided into Geopixels and the textures within each rectangular block were modeled as bivariate linear or quadratic functions over the pixel coordinates. Least squares regression was performed on each block to obtain a vector of coefficients that were averaged over time and compared with unsegmented video frames to identify foreground regions using the Yakimovsky likelihood ratio [103].

In [92] Skifstad proposed two new features for use in video change detection systems based on the linear and quadratic picture functions proposed in [35] and the surface reflectivity study presented in [81]. The derivative model improved on the picture functions introduced in [35] by representing the textures in each Geopixel using the spatial derivative of the linear or quadratic models originally reported in [35]. Foreground regions were detected by thresholding the Manhattan distance measured between the spatial derivative vectors of the current and previous video frames. The shading model characterized pixel intensity as the product of illumination and a shading coefficient based on the object surface reflectivity. In this situation, changes were detected by thresholding the variance of the illumination ratios computed at each pixel location using a small rectangular window.

In [60] the scalar circular shift moment (CSM) feature was proposed as yet another method for representing textures within rectangular image blocks. Changes were detected by thresholding the difference measured between corresponding blocks in consecutive video frames. A global Gaussian noise assumption was used to estimate an appropriate threshold value. The authors reported accurate change detection results in the case of time varying illumination conditions. The CSM continues to be an attractive textural representation for scene modeling applications because it requires very little computational complexity and is robust to variations in lighting conditions.

In [101] a motion saliency image was generated by integrating directionally consistent motion estimated between video frames using a multiresolution version of the Lucas-Kanade optical flow technique [62]. The motion saliency image maintained a collection of counters that indicated the number of frames in which the pixel continued to move in a similar direction. The counters were reset to zero when the direction of the motion changed significantly. Foreground objects were detected by thresholding the motion saliency image and then grouping the pixels into regions. The authors achieved good rejection of dynamic background components exhibiting oscillatory motion. However, the system was unable to detect foreground objects that underwent both straight line and periodic motion.

Compared to other types of scene models the block based techniques generally require the lowest amount of computer resources because the granularity of the blocks are larger than a single pixel. The earliest block level scene models performed poorly against the camouflage problem [40]. Recently, several modern block based techniques have been shown to perform well in the presence of graudal illumination changes [60] and dynamic background components [101]. However, the success of these algorithms has more to do with the types of features used than the block processing strategy.

2.1.2 Temporal Low-Pass Filtering

Temporal low-pass filters have been used in a large number of early reference image based scene models to characterize video background structure according to

$$I_{\mathrm{Ref},k}(\mathbf{p}) = \alpha_k(\mathbf{p}) I_k(\mathbf{p}) + \beta_k(\mathbf{p}) I_{\mathrm{Ref},k}(\mathbf{p}), \tag{2.3}$$

where $\mathbf{p} = (x_1, x_2)$ are the spatial coordinates of a single pixel and $\alpha, \beta \in \mathbb{N}^2 \times \mathbb{R}$ represent the possibly spatiotemporally varying filter weights at time K. Foreground object detection was performed by application of a threshold $T_k(\mathbf{p}) \in \mathbb{N}^2 \times \mathbb{R}$ to a measurement of the difference between the current frame and the reference image according to Eq. (2.2).

In [22], Donohoe proposed the use of an early background reference image generated by applying a temporal low-pass filter with fixed weights to a sequence of video frames. Foreground regions were identified by thresholding pixel-level differences measured between the reference image and unsegmented video frames in the grayscale feature space. Two automatic techniques for identifying a spatially constant time varying threshold were studied and implemented on real-time hardware. One method modeled the noise using a single Gaussian and the other method modeled the noise with a histogram. In both cases, the foreground probabilities were assumed to follow a uniform distribution.

Karmann used a white noise acceleration process to model the variation in individual pixel intensities over time [45]. A reference background image was generated by averaging a predicted background with an observation of the background in a Kalman filtering framework; a careful analysis of the filter reveals that the filter is actually a temporal low-pass filter with fixed gains. Foreground objects were identified by applying a fixed threshold to the absolute difference measured between the reference image and the unsegmented video frames. The authors reported highly accurate object boundary identifications, robustness to object speed and video sample rate, and near immunity to false background detections within homogeneously colored foreground regions.

In [52], Koller used a model similar to Karmann [45] to detect moving objects in traffic. Again, the authors incorrectly claimed to be using a Kalman filter in [52]; however, they did provide the reader with a convenient description of the differences between an actual Kalman filter and their filter in a related technical report [51].

In [83] a temporal low-pass filter similar to [45] was proposed for maintaining an adaptive reference image of the background. The pixel classification process was improved to prevent the model from overadapting to sudden changes in the foreground. Foreground objects were identified by applying a fixed threshold to the difference measured between the reference image and the unsegmented video frames. The algorithm performed well in a human body tracking application in a cloudy outdoor environment with a large amount of unexpected changes in lighting conditions.

The computational color model proposed in [34] represented each pixel as a vector composed of the temporal mean and standard deviation in each RGB band and the variation in chromaticity and brightness distortion. A reference image where each pixel was represented by the computational color vector was updated over time using a temporal low-pass filter. Foreground objects were detected by computing the normalized chromaticity and brightness distortion values from the computational color vectors at each pixel location and then applying a fixed threshold to the difference measured between the normalized values and the observed values in unsegmented video frames. The segmentations were shown to be robust, accurate and efficient when applied to several challenging test videos. This algorithm did not do well in situations where foreground objects became part of the background, or in cases where the background was composed of highly specular surfaces such as mirrors, steel, or water.

In [85] a temporal low-pass filter was used to estimate a background reference image in either the RGB or grayscale color spaces. Foreground objects were detected by application of a fixed threshold to a difference image computed from the reference image and each unsegmented video frame. In this algorithm, each video frame was preprocessed with a 3×3 Gaussian filter to remove the high frequency image components. The system achieved satisfactory results in a real world application where it was used to detect humans in surveillance videos recorded at a high traffic metro station in Nuremberg, Germany.

In [59] a nonparametric model was used to generate a reference image that characterized both the stationary and dynamic background pixels. Stationary background pixels were represented by RGB color and RGB spatial gradient, and dynamic background pixels were represented using a collection of quantized color values. Both the static and dynamic background features were updated over time with a temporal low-pass filter. Foreground regions were identified by comparing observed RGB and RGB spatial gradient features with the reference image using a threshold. The authors achieved good results against the foreground/background segmentation problem on a wide range of scenarios including subway stations, parking lots, airports, public buildings, etc.

The majority of the temporal low-pass filtering technques were performed at the pixel-level and thus they generally require larger amounts of memory than block-based methods. In terms of computational complexity, the thresholds used to compare unsegmented frames with a reference image of the background are generally more complex than the addition and multiplication required to execute the low-pass filter. With respect to the well known scene modeling challenges, the performance of temporal low-pass filtering algorithms is highly dependent upon the learning strategy used to integrate new obervations into the model.

2.1.3 Kurtosis

Kurtosis is classically defined to be the fourth central moment of a statistical distribution. For a random variable y, this is given by according to

$$kurtosis(y) = E\{y^4\} - 3E\{y^2\}^2, \tag{2.4}$$

where $3E\{y^2\}^2$ is a normalizing term used to ensure that the kurtosis vanishes for a Normally distributed random variable. In scene modeling, the sample kurtosis has been used to identify non-Gaussian variations within each grayscale pixel time series according to Eq. (2.2), where

$$I_{\text{Ref},k}(\mathbf{p}) = kurtosis(I_{k-i}(\mathbf{p})) \quad i \in [0, 1, 2, \ldots, N] \tag{2.5}$$

and $\mathbf{p} = (x_1, x_2)$ are the horizontal and vertical pixel coordinates.

Briassouli proposed the use of kurtosis for identifying foreground pixels and grouped them into regions using a motion energy image [12]. Each pixel time series was examined over a small time window and pixel locations with a high kurtosis were assumed to be motion pixels. The authors presented a theoretical argument with many practical examples that demonstrated the use of kurtosis for identifying outliers in statistical distributions and finished with a wide range of examples in video segmentation applications. Briassouli reported robust video segmentations in the presence of occlusions, dynamic backgrounds and shadows when compared to existing difference based methods.

2.2 Digital Filtering (1990-2007)

Digital filtering techniques appeared in the scene modeling literature between 1990 and 2007, where they were used primarily to characterize unimodal distributions of image features. Foreground regions were identified by comparing features from unsegmented video frames with the expected feature values estimated by application of a digital filter to a collection of previously observed frames. The types of digital filters used to estimate the expected values of localized image features include the Kalman filter [46, 107], the Median filter [11, 16, 17], and the Wiener filter [95]. With the exception of [107], where Eigendecomposition was performed on a covariance matrix computed from a collection of complete video frames in the grayscale colorspace, all of the scene models that employed digital filtering modeled pixel level features in the RGB [16,17] or grayscale [11,46,95] color spaces. This section is divided into subsections based on the type of digital filtering technique that was used to estimate the expected background feature image.

2.2.1 Kalman Filtering

The Kalman filter is the minimum mean squared error (MMSE) solution to the problem of estimating the true state $\mathbf{x}_k$ of a linear dynamical system from observed states $\mathbf{z}_k$ under the state transition model

$$\mathbf{x}_k = \mathbf{F}_k \mathbf{x}_{k-1} + \mathbf{w}_k \tag{2.6}$$

and measurement equation

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k, \tag{2.7}$$

where $\mathbf{w}_k$ and $\mathbf{v}_k$ are mutually independent and uncorrelated zero mean white noises [44]. The filter output or posterior state estimate $\hat{\mathbf{x}}_{k|k}$ is given by

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}), \tag{2.8}$$

where $\hat{\mathbf{x}}_{k|k-1}$ is the prior state estimate and $\mathbf{K}_k$ is the Kalman gain applied to the difference between the predicted observation $\mathbf{H}_k \hat{\mathbf{x}}_{k|k-1}$ and the actual observation. The Kalman gain is estimated from the system and measurement noise covariance matrices combined with the state transition and measurement matrices. In scene modeling applications, z_k corresponds to the observed video frames, $\mathbf{x}_k$ is the latent background image, and $\hat{\mathbf{x}}_{k|k}$ is the Kalman filter estimate of the background image. Video frames are segmented by application of a possibly time varying threshold $T_k(\mathbf{p})$ according to Eq. (2.2).

The first true usage of a Kalman filter in a scene model appeared in [46], where an estimate of the grayscale background image was tracked using a Kalman filter with an identity state transition matrix. The system noise was modeled by a temporal low-pass filter and the measurement noise was estimated to be the square of the difference between the current frame and the background image. Foreground regions were detected by thresholding the difference between the background image and the current video frame as in Eq. (2.2). Compared with the authors previous work, they reported an improvement in terms of the number of moving objects that were incorrectly present in the reference image and an elimination of the deadlock situation caused when conservative learning techniques prevent necessary evolution of the background mdoel.

In [107] a dynamic texture model was presented in the context of a Kalman filter, where the measurement equation represented a transformation from the hidden dynamic texture vector to the observable grayscale image space. The state transition matrix, measurement equation, and noise statistics were estimated from a collection of labeled training data and used to segment video by comparing the filter background estimates with unsegmented video frames. The authors reported good results on approximately five well known videos where the distributions of grayscale intensities within the foreground and background objects were mixed.

2.2.2 Median Filtering

A median filter is a nonlinear digital filter where the filter output is defined to be the central value selected from the ordered input set. In the case of even length filter inputs, the two central order statistics are averaged [10]. Multidimensional median filters used in scene modeling applications define a norm and use it to identify the most central input vector from within a collection of previously observed video frames I_{k-i} where $i \in [1, 2, 3, \ldots, N]$ according to

$$l = \underset{i=1,\ldots,N}{\arg\min} \sum_{j=1}^{N} ||\mathbf{I}_{k-i}(\mathbf{p}) - \mathbf{I}_{k-j}(\mathbf{p})||, \tag{2.9}$$

where $I_{\mathrm{Ref},k}(\mathbf{p}) = I_{k-l}(\mathbf{p})$ is the output of the temporal median filter at a single pixel location $\mathbf{p} = (x_1, x_2)$ and $|| \cdot ||$ is a norm. Foreground object detection was performed by application of a threshold $T_k(\mathbf{p})$ to a measurement of the difference between the current frame and the reference image according to Eq. 2.2.

In [17] a reference image of the background was generated by median filtering 50 to 200 video frames at the pixel level. The multidimensional median was computed on each pixel time series in the RGB colorspace and the median value was determined using Eq. 2.9, where $| \cdot |$ was the Manhattan distance [54]. Change detection was performed by thresholding the difference between the current RGB pixel values and the adaptive reference image, under the assumption that all of the pixels were affected by the same globally estimated Gaussian noise. Good results were obtained against both periodic and aperiodic motion of vehicles and humans in several simulated examples.

In [16] a reference image of the background was modeled using a multidimensional median filter in the RGB colorspace. Each pixel time series was filtered with a median filter using the maximum norm distance function, where the median value was determined by computing the minimum sum of the distance measured between all of the filter inputs. Video frames were segmented by first classifying each pixel as foreground, background or shadow using a threshold based comparison to the reference background image. All shadows and foreground objects were further classified as either ghosts or moving objects based on size, saliency, Lucas-Kanade motion [62] and object level connectedness. The algorithm was tested in a wide range of different environments and applications and determined to be a good general purpose approach to foreground, background and shadow segmentation due to the integration of object level knowledge.

The most recent example of the prevalence of the median filter in background modeling algorithms appeared in [11], where a simple pixel level median filter was applied to the grayscale image values to provide a good initial segmentation of foreground and background components. Median filters perform well against sudden illumniation changes, ghosts and dynamic background components because they are robust to outliers in the pixel-level feature distributions. In addition, they adapt well to slowly evolving distributions such as those arising due to gradual illumination changes.

2.2.3 Wiener Filtering

The Wiener filter is the linear time invariant (LTI) minimum mean squared error (MMSE) solution to the problem of removing additive noise $\mathbf{w}_k$ from a signal $\mathbf{x}_k$, where both the signal and the noise are wide sense stationary (WSS) stochastic processes [53, 99]. The filter output $\mathbf{\hat{x}_k}$ is defined according to

$$\mathbf{\hat{x}_k} = A * (\mathbf{x}_k + \mathbf{w}_k), \tag{2.10}$$

where the filter impulse response A is defined in terms of the auto and cross correlation functions of the signal $\mathbf{x}_k$ and noise $\mathbf{w}_k$. In scene modeling applications, $\mathbf{x}_k$ represents the true background state, $\mathbf{w}_k$ is the foreground, and $I_{\mathrm{Ref},k} = \hat{\mathbf{x}}_k$ is an estimate of the background. Foreground object detection is performed by application of a threshold $T_k(\mathbf{p})$ to a measurement of the difference between the current frame and the reference image according to Eq. (2.2).

The only use of a Wiener filter in a scene modeling capacity appeared in the Wallflower background model [95], where a reference image was constructed by applying a 30 tap filter to each pixel time series to predict future background values. Foreground pixels were identified by applying a threshold to the difference measured between unsegmented video frames and a Wiener prediction of the background image. Unfortunately, the Wiener filter coefficients were recalculated at each time step based on the prior observations and thus the general formal assumptions with respect to the Wiener filter, namely that both input signals are wide sense stationary, were completely disregarded. In defense of the Wallflower system however, the authors did provide a comprehensive analysis of several important challenges associated with scene modeling algorithms as well as a unique approach to foreground region isolation. Foreground regions were initially identified by thresholding and then subjected to elimination if connected regions fell below a size threshold. Regions were further subjected to a binary motion mask followed by a region growing process where foreground object masks were grown from internal trusted seed points and then finalized with the histogram back projection algorithm [10].

2.3 Parametric Statistical Modeling (1997-2008)

Parametric statistical models have been used in scene modeling applications to characterize temporal distributions of localized image features using several well known parametric functions. In 1997, parametric models became very popular because their computational needs essentially mirrored the types of computers that were widely available, *i.e.,* a moderately high computational complexity combined with a very low memory storage requirement. Scene models that employed parametric statistical distributions generally adopted the idea that background features accounted for a majority of the density within the model and foreground features were best represented as either low probability regions or outliers. Therefore, foreground regions were identified by comparing image features from unsegmented video frames with the parameters of the corresponding statistical models, under the assumption that either all or a majority of the model is characteristic of the background features. The types of image features that have been modeled with parametric statistical distributions include YUV color [102], YUV color combined with spatial coordinates [102], LUV color combined with microstructural texture response [6], RGB color [31, 36, 42, 57, 93, 94, 104, 109, 110], RGB color combined with grayscale spatial gradient [37,41,69], RG color [100], RGB color combined with optical flow [108], HSV color [63,82], grayscale intensity [20, 26, 29, 30, 57, 75, 109, 110], grayscale intensity combined with the spatial derivative [84, 97], grayscale intensity combined with wavelet coefficients [47] and Eigendecomposition or principal component analysis performed on grayscale intensity blocks (PCA) [73, 77, 88]. These image features were predominately modeled at the pixel [6, 20, 26, 29–31, 36, 37, 41, 42, 57, 63, 69, 75, 93, 94, 97, 100, 104, 108–110] or block [47,73,77,84,88] level, with notable exceptions in the case where the spatial distribution of image features were characterized by the parametric model combined with a binary support map [82,102]. The types of parametric models that have been used to characterize the aforementioned image features are the Normal distribution or single Gaussian [6,20,37,69,73,75,77,88,97,102], Gaussian mixture model (GMM) [26,30,31,36,41,42,57,82,93,94,100,104,108–110], bi-modal distribution [29], hidden Markov model (HMM) [47,84], and the neural network [63].

2.3.1 Normal Distribution

In scene modeling applications, the Normal distribution has been used to characterize statistical distributions of image features according to

$$\eta(\mathbf{x}, \mu, \Sigma) = \frac{1}{(2\pi)^{1/2}|\mathbf{\Sigma}|^{1/2}} e^{-1/2(\mathbf{x}-\mu)^T \mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)}, \tag{2.11}$$

where $\eta(\mathbf{x}, \mu, \Sigma)$ was the probability of observing vector $\mathbf{x}$ in a unimodal distribution with mean and covariance $\mu_{i,k} \in \mathbb{R}^n$ and $\mathbf{\Sigma}_{i,k} \in \mathbb{R}^{nxn}$. Because each distribution was represented by two parameters, a parametric reference image was defined by

$$I_{\text{Ref},k}(\mathbf{p}) = \{\mu_k^{\mathbf{p}}, \Sigma_k^{\mathbf{p}}\} \tag{2.12}$$

where $\mathbf{p} = (x_1, x_2)$ are the horizontal and vertical coordinates of a pixel or region center. Video frames were segmented by estimating the background probabilities of features from the unsegmented video frames and then applying a threshold $T_k(\mathbf{p})$ according to

$$L_k(\mathbf{p}) = \left\{ \begin{array}{ll} Background & : \eta(I_k(\mathbf{p}), \mu_k^{\mathbf{p}}, \Sigma_k^{\mathbf{p}}) > T_k(\mathbf{p}) \\ Foreground & : \eta(I_k(\mathbf{p}), \mu_k^{\mathbf{p}}, \Sigma_k^{\mathbf{p}}) \leq T_k(\mathbf{p}) \end{array} \right. . \tag{2.13}$$

In [102] the PFinder (Person Finder) system modeled the distributions of background colors at each pixel location with a single multivariate Gaussian in the YUV color space. Each foreground blob in the video was modeled with a multivariate Gaussian in a five dimensional space composed of YUV color and the two dimensional spatial location. In addition, a binary object support map was used to strictly define the domain of each foreground blob. Video frames were segmented by classifying each pixel as background or foreground by comparing the color values and locations of each unclassified pixel to the corresponding background model and to each foreground blob according maximum likelihood. A single person object in the video was represented by combining several foreground blobs through a series of morphological operations. The authors reported good results in a wide range of applications including wireless interfaces, video databases and low-bandwidth coding.

In [37] each pixel time series was modeled with a single Gaussian in the RGB colorspace and two univariate Gaussians that characterized the distributions of the spatial gradient in the horizontal and vertical directions. The parameters of all three Gaussians were updated over time according to the online k-means algorithm [61]. Foreground pixels were detected by comparing the color and spatial gradients of pixels in unsegmented video frames with the Gaussian background models at the corresponding locations. Color and gradient foreground detections were combined to produce a final binary segmentation image. According to the authors the algorithm was resistant to clutter, slow illumination changes, camera noise and achieved real time performance on standard computing platforms.

In [69] each pixel time series was modeled by a single Gaussian in the RGB colorspace and the parameters were updated over time using a temporal low-pass filter similar to the online k-means algorithm [61]. Foreground pixels were identified by comparing the RGB values from unsegmented video frames with the corresponding Gaussian distributions using an automatic threshold based on the variances of each model, followed by thresholding of the differences in spatial gradient and chromaticity. Foreground regions were finalized by applying a 3×3 median filter to the initial segmentation image and then performing a connected components labeling [10]. Mckenna claimed that this algorithm was robust in uncontrollable outdoor environments, adaptive to lighting changes and small camera movements, and that it failed when similarly textured objects crossed paths.

In [75] an illumination ratio similar to [81] was thresholded to detect foreground pixels. The threshold was based on the assumption that camera noise followed a Chi-Squared distribution and was empirically determined in the available experiments. The authors reported stable foreground detections over long periods of time due to an appropriate modeling of the camera noise.

Wang introduced a dynamic conditional random field (DCRF) model based on the conditional random field (CFR) model proposed by Lefferty in [56] and used it in the context of a maximum a posteriori (MAP) estimation to segment video into foreground, background and shadow. The foreground was assumed to follow a uniform distribution, the background was modeled as the product of two Gaussians in the grayscale intensity and spatial gradient feature spaces, and shadows were modeled using a linear function corrupted by additive zero mean Gaussian noise. Foreground pixels were identified by comparing pixels from unsegmented video frames with the corresponding background, foreground and shadow distributions and the DCRF was used to propagate the distributions through time. Several example video segmentations were provided to demonstrate the effectiveness of the DCRF model on indoor scenes filmed with monocular grayscale cameras.

In [20] each pixel time series was modeled by a single Gaussian in the grayscale intensity feature space. Initial foreground regions were identified by comparing unsegmented video frames with the corresponding pixel level Gaussian models and then grouped into regions using connected components labeling [10] combined with a size thresholding procedure. A contour saliency map (CSM) was generated from the identified foreground regions and combined with pixel level spatial gradients to better refine the object boundaries. The algorithm achieved promising results on six infrared videos depicting pedestrian traffic.

Benedek proposed a kernel based microstructural texture measurement to characterize local texture at a single pixel location [6]. The pixel level distributions of background features were modeled by a four dimensional Gaussian distribution where the feature vector was composed of the L*U*V* color and the microstructural texture response in the luminance band only. The distributions of shadow features were modeled at each pixel location with a single Gaussian in the L*U*V* feature space. Foreground pixels were detected by comparing unsegmented video frames to the background and shadow distributions and classification was performed according to maximum likelihood, where the foreground distribution was assumed to follow a uniform distribution. A brief numerical evaluation appeared to validate the combination of the shadow and background models proposed, however, a comparison with existing state-of-the-art techniques was absent.

Multivariate normal Gaussian distributions have been used to model a wide variety of foreground and background objects using a myriad of interesting color and texture features. Against gradual illumination changes these models have been extremely effective. However, the normal distribution is not capable of characterizing the multimodal distributions occuring due to dynamic background components and thus it does not perform well in complex outdoor situations.

2.3.2 Principal Component Analysis (PCA)

Principal component analysis (PCA) characterizes a statistical distribution over a vector space by performing an Eigendecomposition on the sample covariance matrix and retaining all or part of the Eigenvectors. In scene modeling applications, Eigendecomposition is performed on a collection of $m \times n$ image blocks by resizing each block to $1 \times mn$, computing the mean centered sample covariance matrix $\mathbf{C}$, and then performing Eigendecomposition according to

$$E = \Phi \mathbf{C} \Phi^T, \tag{2.14}$$

where $\mathbf{E}$ is a diagonal matrix containing mn Eigenvalues and Φ is an $mn \times mn$ matrix of corresponding Eigenvectors. A subset of the Eigenvectors at the location of each block center $\mathbf{p} = (x_1, x_2)$ are retained in a reference image

$$I_{\text{Ref},k}(\mathbf{p}) = \Theta_k^{\mathbf{p}} \subseteq \Phi_k^{\mathbf{p}}, \tag{2.15}$$

where $\Theta_k^{\mathbf{p}}$ represents a subset of the Eigenvectors $\Phi_k^{\mathbf{p}}$ at block location $\mathbf{p}$. The Eigenvector based reference image is used to identify foreground structure by thresholding the distance from feature space (DFFS) according to [72]

$$L_k(\mathbf{p}) = \begin{cases} \text{Foreground} & : ||I_k(\mathbf{p}) - \Theta_k^{\mathbf{p}} I_k(\mathbf{p})|| > T_k(\mathbf{p}) \\ \text{Background} & : ||I_k(\mathbf{p}) - \Theta_k^{\mathbf{p}} I_k(\mathbf{p})|| \leq T_k(\mathbf{p}) \end{cases}, \tag{2.16}$$

where $|| \cdot ||$ is the Euclidean norm.

In [77] Oliver and Pentland introduced the Eigenbackground method for characterizing background structures in video, where a collection of training images were reshaped into one dimensional vectors and used to compute a mean vector and a covariance matrix. Spectral decomposition was performed on the covariance matrix and the Eigenvectors corresponding to the largest Eigenvalues were used to reconstruct an estimate of the background scene. Foreground pixels were detected by thresholding the pixel level differences measured between unsegmented video frames and their projection onto the eigenvectors used to represent the background. Lastly, the authors employed a coupled hidden Markov model (CHMM) to analyze and classify the different types of human interactions that occurred within each foreground region.

In [88] a sequence of video frames were averaged over time and then Eigendecomposition was performed on square nonoverlapping blocks to produce a collection of reference Eigenvectors at each block that characterized the background textures. Foreground regions were identified by thresholding the Mahalanobis [65] distance measured between blocks of grayscale intensities in unsegmented video frames and referential eigenvectors at the corresponding block locations.

In [73] video frames were divided into square nonoverlapping blocks and the incremental principal component analysis (IPCA) algorithm proposed in [98] was applied to each block time series to estimate the Eigendecomposition within each block. Structural changes were detected in unsegmented video frames by projecting the unclassified blocks onto the corresponding referential eigenvectors and thresholding the DFFS [72]. Motion was detected by measuring the sum of the squared error measured between a linear prediction of the expected block structure and the observed block structure projected onto the corresponding referential Eigenvectors. Excellent results were obtained on several real world videos in the presence of highly dynamic background components.

2.3.3 Gaussian Mixture Model (GMM)

Gaussian mixture models characterize statistical distributions of values as as a summation of weighted Normal distributions according to

$$P(\mathbf{x}) = \sum_{i=1}^{K} w_i \eta(\mathbf{x}, \mu_i, \mathbf{\Sigma}_i), \tag{2.17}$$

where $P(\mathbf{x})$ is the probability of observing vector $\mathbf{x}$, K is the number of Gaussian components, and w_i is the mixing probability of the ith Gaussian function $\eta(\mathbf{x}, \mu_i, \mathbf{\Sigma}_i)$ parameterized by a mean vector and covariance matrix. In scene modeling applications, GMMs have been used to represent the distributions of foreground and background image features in arbitrarily shaped spatial regions, in addition to those at the pixel and block level. Foreground object detection was performed by comparing pixels from unsegmented video frames with the background components $i \in [1, 2, 3, \ldots, K_b]$ of the corresponding GMM according to

$$L_k(\mathbf{p}) = \begin{cases} \text{Background} & : \sum_{i=1}^{K_b} w_i \eta(I_k(\mathbf{p}), \mu_i, \mathbf{\Sigma}_i) > T_k(\mathbf{p}) \\ \text{Foreground} & : \sum_{i=1}^{K_b} w_i \eta(I_k(\mathbf{p}), \mu_i, \mathbf{\Sigma}_i) \leq T_k(\mathbf{p}) \end{cases}, \tag{2.18}$$

where $\mathbf{p} = (x_1, x_2)$ are the horizontal and vertical coordinates of a single pixel, K_b is the number of background components, and $T_k(\mathbf{p})$ is a possibly time varying threshold.

In [26] Friedman and Russell reported the first use of Gaussian mixture models (GMM) for representing distributions of pixel level features in a traffic monitoring application. Each grayscale pixel time series was modeled by a mixture of three Gaussian functions that represented the road, vehicle, and shadow colors. Pixels in unsegmented video frames were classified according to maximum likelihood by comparison with the mixture models at the corresponding pixel locations. The parameters of each mixture model were updated over time using the EM algorithm [21]. The authors achieved adequate segmentations in the case of highly constrained traffic surveillance videos and claimed that there was significant room for improvement in terms of initialization mechanisms and the integration of neighborhood information into the classification procedure.

In [82], Raja proposed modeling the distributions of color in background and foreground objects using a multivariate GMMs in the HSV color space. Each model was initialized using the EM algorithm [21] and a cross-validation procedure to estimate the number of Gaussians required for each object. The parameters of each mixture model were updated over time using a temporal low-pass filter with a learning rate parameter. Pixels were classified by comparing HSV features from unsegmented video frames with the existing color distribution models for each object according to maximum likelihood. Experimental results demonstrated the effectiveness of the algorithm in the presence of highly variable lighting conditions.

Stauffer and Grimson improved upon the applicability of GMMs to scene modeling in [93,94], where they proposed the use of mixture modeling to characterize arbitrary distributions of pixel level features in video. Each pixel time series was modeled by a GMM composed of three to five single Gaussian functions in the RGB or grayscale feature spaces, the parameters of which were updated according to the online k-means algorithm [61]. Foreground pixels were detected by classifying each Gaussian function as foreground or background based on mixing probability and variance, and then comparing pixel level features from unsegmented video frames with the labeled components according to maximum likelihood. The system was stable and robust to slow lighting changes, shadows and a wide range of dynamic background components. The authors reported that the system could be improved by using a full covariance matrix in the RGB feature space, and by the inclusion of a procedure to determine the optimal number of Gaussians to use at each pixel location.

In [41], each pixel time series was modeled by a slightly modified version of the GMM proposed by Stauffer and Grimson [93], where the classification of each Gaussian was based on an analysis of each component individually rather than as a whole. Foreground pixels were detected according to [93] and combined with a foreground edge detection mechanism that compared spatial gradient measurements from the unsegmented video frame with a representation of the spatial gradient structure of the background components. The system achieved good results on several indoor and outdoor test scenarios using the same fixed thresholds throughout.

In [30] GMMs were used to model the distribution of grayscale background values at each pixel location. The parameters of the GMMs were updated using a control system approach where foreground pixels were ignored and in some cases the number of Gaussians was altered either by adding additional components or by merging the existing components. Foreground pixels were detected by comparing grayscale intensities from unsegmented video frames with GMMs at the corresponding locations, as well as a uniform model of the foreground according to maximum likelihood. The authors demonstrated the effectiveness of their higher level control modules in the background model maintenance procedure against several challenging surveillance videos.

In [100] the GMM of Stauffer and Grimson was modified to reduce the effects of shadows and lighting changes using the normalized red and green colorspace. In addition, the model parameters were updated by the EM algorithm rather than the with online k-means algorithm [61].

In [31] the GMMs of Stauffer and Grimson were used to segment video recorded by an instrumented camera unit mounted on a mobile robotic platform. Knowledge of the camera motion was used to index a significantly larger world image that was updated regionally based on the camera field of view. A logical framework was developed to handle initialization of the model in the presence of foreground objects that occluded important background structures.

In [109] and [110], Zivkovic improved the Stauffer and Grimson [93] model with the addition of a method for adjusting the number of Gaussian components in each pixel model dynamically. The EM algorithm [21] was used to update the model parameters and several different techniques for determining the order of each model were explored. For a detailed review of well known techniques for determining the number of modes in a statistical distribution, the reader is directed to [70]. The adaptive GMM proposed by Zivkovic achieved great success on a wide variety of video segmentation challenges and is generally considered to be the most effective among the pixel level GMM algorithms in existence.

In [104] the GMM of Stauffer and Grimson was used to detect initial foreground pixels and a morphological procedure was used to remove gaps in object masks by integrating the foreground regions over time. In addition, a method for removing shadows based on brightness and chromaticity was proposed.

Zhou combined initial foreground pixel detections from the Stauffer and Grimson GMMs with a simple temporal derivative and an estimate of optical flow [108]. Initial foreground detections were grouped into regions and optical flow was computed according to [62] on regions exhibiting significant temporal motion estimated by Otsu's method [78]. In other words, foreground regions were identified by the method of Stauffer and Grimson and further subjected to a threshold based on a simple temporal derivative and then filtered according to the magnitude and direction of their optical flow vectors. The algorithm achieved only satisfactory performance on real world image sequences and appeared to suffer from an abundance of unrealistic assumptions.

In [57] an alternative to the online k-means [61] and EM algorithms was proposed for maintaining the parameters of GMMs over time. A learning model was introduced based on the incremental EM algorithm considered in [87] that achieved an increase in convergence speed and maintained the stability of the original model. In contrast to the fixed global learning rate of [93], Lee proposed a learning rate that varied with time based on the historical parameters of each individual GMM. The approach was verified on a wide range of simulations where it achieved both improved convergence and estimation accuracy.

Huang employed the Stauffer and Grimson scene modeling algorithm and used the Bhattacharyya [7] distance to identify foreground pixels in unsegmented video frames [36]. In [42] the Stauffer and Grimson model was again used to identify foreground pixels which were then further analyzed in terms of intensity, color distortion, and edge magnitude and direction to extract shadow regions from the segmentation.

Overall, pixel-level GMMs have been verified as an excellent method for characterizing multimodal statistical distributions occurring due to gradual illumination changes and or dynamic components of the background. However, GMMs suffer from the bootstrapping problem in the sense that it is difficult to predetermine the number of modes and initialize the models.

2.3.4 Hidden Markov Model (HMM)

A hidden Markov model (HMM) is a dynamic system model where the true state of the system $\mathbf{x}_k$ is generally represented by a collection of discrete hidden states governed by a stochastic state transition process

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}) \tag{2.19}$$

where the true state is only

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k). \tag{2.20}$$

In (2.19) and (2.20) $f(\cdot)$ and $h(\cdot)$ are possibly nonlinear functions and the statistics of the system and measurement noises $\mathbf{w}_k$ and $\mathbf{v}_k$ are not restricted in general [4, 5]. In scene modeling applications, the measurement noise $\mathbf{v}_k$ has been used to characterize the so called *emission* distributions over the observable image features for a finite set of discrete hidden states $\mathbf{x}_k$. Segmentation was performed by analyzing a sequence of observed states $\mathbf{z}_k$ to determine the most likely underlying sequence of hidden states using the well known forward algorithm [4, 5].

Rittscher proposed a three state hidden Markov model (HMM) that characterized the distributions of grayscale intensity and Sobel edge features in the discrete cases of foreground, background and shadows [84] in 3×3 square nonoverlapping video blocks. In this case, the state transition f and measurement function h were assumed to be linear systems corrupted with additive noise signals. The HMM state transition matrix was estimated from assumptions regarding the amount of time that was likely to be spent in each of the three hidden states as well as the improbability of certain state transitions. Pixels from unsegmented video frames were classified by comparison to the emission distributions at the corresponding HMM region using the forward algorithm [4, 5]. The authors reported promising results in a single car tracking application.

Kato and Rittscher improved on their work in [84] by using the variance of the LL, HL and HH length two 2D separable Daubechies [19] wavelet coefficients to characterize texture in 3×3 blocks rather than the previously used Sobel edge features [47]. Each block time series was modeled with the three state HMM proposed by Rittscher in [84] that used multivariate Gaussians to model the emission distributions in the grayscale and textural feature spaces. Several experimental results were provided in a low-level car tracking application that appeared to validate the claims of the authors.

2.3.5 Artificial Neural Networks (ANN)

Artificial Neural Networks (ANN) estimate possibly nonlinear functions using an interconnected network of nodes that each perform simple tasks. The inputs and outputs of the network are specified by the interconnection scheme and the types of functions that are used in each network node.

In [63] a neural network was proposed for modeling the distribution of pixel level features in the HSV colorspace. A two-dimensional neuronal map structure similar to a self organizing map (SOM) [50] was used to characterize foreground, background and shadow color distributions in each pixel time series. Pixels in unsegmented video frames were classified by comparison to the weights of the corresponding neural network. The authors reported accurate segmentations at a relatively low computational complexity on a range of examples exhibiting background motion, gradual illumination changes and camouflaged foreground structures.

2.4 Nonparametric Statistical Modeling (2000-2010)

Nonparametric statistical models have been used in scene modeling applications to characterize temporal distributions of localized image features using collections of previously observed features or measurements thereof. Nonparametric scene models identify foreground regions by comparing image features from unsegmented video frames with collections or characterizations of previously observed feature vectors at the corresponding image locations. A wide variety of features have been employed by nonparametric scene models including RGB edges [67], Local Binary Pattern [32,33], RGB color [3,23,24,48,49,80], RGB color combined with spatial coordinates [90,106], spatiotemporal derivative [2], grayscale intensity [3,23,24], Eigendecomposition [28,89], and normalized RGB combined with optical flow [71]. Similar to parametric scene models, the majority of the nonparametric models characterize features at the pixel level [2, 3, 23, 24, 48, 49, 71, 79, 80] or block level [28, 32, 33, 67, 89], with the exception of [90,106] where the spatial distributions of features are included in the model. The types of nonparametric statistical models used in video segmentation include histogramming [2, 32, 33, 67, 80], kernel density estimation (KDE) [3, 23, 24, 71, 79, 89, 90], Codebooks [48, 49], and binned kernel density estimation (BKDE) [106].

2.4.1 Histogramming

Histograms characterize statistical distributions by dividing the observation space into equally sized bins and then maintaining a count for each bin that corresponds to the number of times that a vector lying within the boundaries of the bin has been observed. This may be interpreted as an empirical discrete estimate of the true underlying probability density. Scene models have employed N bin histograms to characterize the distributions of background features at the pixel and block level according to

$$M_k(\mathbf{p}) = \{\phi_1, \phi_2, \phi_3, \ldots, \phi_N\}, \tag{2.21}$$

where $\mathbf{p}$ indicates the pixel or block position and $\{\phi_i\}_{i=1}^{N}$ represent the individual bin counts. Typically, video frames are segmented by applying a threshold $T_k(\mathbf{p})$ to the estimated background probability according to

$$L_k(\mathbf{p}) = \begin{cases} \text{Background} & : f(I_k(\mathbf{p}), M_k(\mathbf{p})) < T_k(\mathbf{p}) \\ \text{Foreground} & : f(I_k(\mathbf{p}), M_k(\mathbf{p})) \leq T_k(\mathbf{p}) \end{cases}, \tag{2.22}$$

where $f(\cdot)$ is a function that uses the histogram $M_k(\mathbf{p})$ to estimate the probability of observing vector $I_k(\mathbf{p})$.

In [67], Mason compared two histogram based background modeling techniques in the RGB and RGB edge feature spaces. In both cases, the background model was initialized by computing histograms in the RGB and RGB edge feature spaces using the first video frame. Foreground pixels were identified using the Chi-Squared similarity metric [74] to compare histograms from unsegmented video frames with the corresponding histograms from the background model. RGB color histograms bin widths were computed by quantizing the color space while RGB edge histogram bin widths were based on quantizing the edge directions. The heights of the RGB edge histogram bins were estimated from the magnitudes of the observed RGB edges. The authors reported superior detections of humans in the case of RGB edges.

In [33], the textural video structure was characterized on square partially overlapping blocks using a collection of weighted histograms in the local binary pattern (LBP) feature space introduced in [76]. In [32] the model was computed on a per pixel basis at the expense of an increased computational requirement. The scene model maintenance and the foreground detections were performed according to the method of Stauffer and Grimson [93]. The LBP scene model was demonstrably verified as an accurate, modern method for characterizing local textural structure in a wide range of theoretical scene models.

In [80], video frames were segmented individually by comparison to an existing layer-based segmentation where each layer represented a single foreground object and a single remaining layer represented the background image. The sampling expectation (SE) algorithm of [105] was used to methodically separate the layers using the Kullback-Leibler divergence criterion [55]. Because each each layer was a representation of the spatial distribution of grayscale intensities for a single object or for the background, the entire collection was essentially a nonparametric representation of the distribution of intensities across the object space. Due to the procedure used to segment the layers, where each layer was treated as a histogram, this method is discussed here in the histogramming subsection of this dissertation. The algorithm produced highly accurate segmentations given a good initialization of the model.

In [2], Adam employed the earth movers distance (EMD) [86] to identify similar regions in unsegmented video frames by comparing reference histograms computed on each region. The reference histograms were generated from manually identified seed regions corresponding to backgrounds or objects of interest and regions within future video frames were segmented by comparison. The features characterized by the histograms were the spatiotemporal derivatives computed over time in arbitrarily shaped seed regions.

2.4.2 Kernel Density Estimation (KDE)

Kernel density estimation (KDE) techniques represent statistical distributions with variably sized collections of samples. Probability estimates are computed by applying a kernel function to the sample collection centered at the value in question according to

$$P(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{\Sigma}}(\mathbf{x} - \mathbf{x}_i^{\mathbf{p}}), \tag{2.23}$$

where $K_{\mathbf{\Sigma}}(\cdot)$ is a kernel function parameterized by a bandwidth $\mathbf{\Sigma}$, $\{x_i^{\mathbf{p}}\}_{i=1}^{N}$ is the collection of N samples at pixel location $\mathbf{p}$, and $\mathbf{x}$ is the point at which the probability is to be estimated. Scene models characterize a distribution of background values by maintaining a collection of samples $\{x_i^{\mathbf{p}}\}_{i=1}^{N}$ at each pixel location $\mathbf{p} = (x_1, x_2)$ and then using KDE to segment video frames according to

$$L_k(\mathbf{p}) = \begin{cases} \text{Background} & : \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{\Sigma_p}}(\mathbf{I}_k(\mathbf{p}) - \mathbf{x}_i^{\mathbf{p}}) > T_k(\mathbf{p}) \\ \text{Foreground} & : \frac{1}{N}\sum_{i=1}^{N} K_{\mathbf{\Sigma_p}}(\mathbf{I}_k(\mathbf{p}) - \mathbf{x}_i^{\mathbf{p}}) \leq T_k(\mathbf{p}) \end{cases}, \tag{2.24}$$

where $T_k(\mathbf{p})$ is a possibly time varying threshold and $\Sigma_{\mathbf{p}}$ is a location dependent kernel bandwidth matrix.

In [23, 24], Elgammal presented the first nonparametric pixel level density modeling technique using Gaussian kernel density estimation (KDE). The distribution of grayscale intensities arising due to background structures were characterized by a collection of values for each pixel time series. Potential foreground pixels from within unsegmented video frames were identified by applying a threshold to an estimate of the background probability calculated using a variable bandwidth Gaussian KDE technique. The kernel bandwidth was estimated from the data by setting the variance of each color channel to the absolute median deviation within each band. False foreground detections were suppressed by thresholding the probability that the potential foreground pixel detections were due to a neighboring background component. In the case of highly camouflaged foreground regions, the false foreground suppression algorithm was significantly throttled to reduce the number of false negatives. The nonparametric background models were updated over time by replacing the oldest value within the collection with new observations determined to be background values. The authors reported highly accurate segmentations with a very low false alarm rate on a wide range of examples with highly dynamic background structures.

In [89], the distributions of textural features in terms of principal components were modeled using a collection of prior eigendecompositions on square nonoverlapping video blocks. Foreground blocks were identified by comparing the eigendecompositions of image blocks from unsegmented video frames both with a linear combination of the neighboring eigendecompositions and with a collection of previously observed decompositions at the corresponding location [96]. Experimental results verified the effectiveness of the proposed method against a small number of short video sequences.

Mittal proposed a variable bandwidth KDE technique for characterizing background probability distributions at the pixel level in the normalized RGB and optical flow [91] feature spaces [71]. Each pixel time series was modeled with a collection of vectors composed of the normalized RGB color and the Simoncelli optical flow in the horizontal and vertical directions [91]. Foreground pixels were identified by comparison to the background model at the corresponding location using a Gaussian kernel where the bandwidth of the kernel was equal to the sum of the estimated covariance measured at both the observation and the background sample point. The covariance of the observation was a block diagonal matrix that represented the uncertainty of the measurement and was composed of the covariance in the normalized color, the standard deviation of the illumination and the covariance of the optical flow. The effectiveness of the algorithm was demonstrated on a well known person detection video with a highly dynamic background.

Han [28] introduced a method for propagating the important modes of a statistical distribution based on the variable bandwidth mean shift algorithm [14, 15]. The procedure was illustrated by an example where the IPCA algorithm [98] was used to characterize the textural background features of a video in the RGB feature space. The procedure produced excellent results on several well known benchmark surveillance videos.

In [90], Sheikh presented the first joint domain-range scene model, where the background distributions of RGB features were modeled by five dimensional vectors composed of both color and horizontal and vertical image coordinates. The domain of each distribution was described by a statistical membership region in the spatial coordinates and the foreground was assumed to follow a uniform distribution. Foreground pixels were identified using a likelihood ratio where the background probability was estimated by KDE [27] between pixels in unsegmented video frames and the corresponding collections of background feature vectors. The method proposed by Sheikh has enjoyed widespread attention in the literature due to it being the first nonparametric joint domain-range background model.

In [79], a generic nonparametric scene modeling technique called RealBoost was introduced. The RealBoost system proposed a pixel level method for the selection of arbitrary image features to be used in the segmentation of video binary foreground and background structure. KDE was used to identify foreground and background pixels in video frames by comparison to feature collections at the corresponding locations.

In [3], Barnich proposed the most recent pixel level background model where the distributions of grayscale intensities were characterized with collections of previously observed background values. Foreground pixels were identified by comparing pixels from unsegmented video frames with the background model at the corresponding location using KDE with a spherical cutoff kernel. Because the foreground was assumed to follow a uniform distribution, pixels were essentially classified using a globally static threshold. Background maintenance was performed according to a stochastic process using uniformly distributed random variables to determine which values to replace in corresponding and neighboring sample collections. Pixels that were determined to belong to the background were integrated into the model at the same pixel location by randomly selecting a sample to replace. In addition, a neighboring distribution was randomly selected and the background value was propagated to a random location within the neighboring model. The Barnich scene model outperformed all existing scene modeling techniques [3, 13].

KDE techniques are an excellent method for characterizing multimodal statistical distributions occurring due to gradual illumination changes and or dynamic components of the background. By comparison to GMMs, KDE methods do not require that the number of modes in the underlying distribution be predefined, only that storage for a sufficient number of samples exist to capture the structure of the distribution.

2.4.3 Codebook

Codebooks characterize statistical distributions using an ad-hoc collection of variables such as the locations of important modes, the mean, the maximum and minimum values, etc. In scene modeling applications, codebooks have been used to represent pixel level background distributions in terms of a finite collection of N variables according to

$$M_k(\mathbf{p}) = \{\phi_1, \phi_2, \phi_3, \ldots, \phi_N\}, \tag{2.25}$$

where $\mathbf{p} = (x_1, x_2)$ are the horizontal and vertical coordinates of a single pixel. Although Eq. (2.25) may appear to be identical to Eq. (2.21), the reader should note that in Eq. (2.21) $\{\phi_1, .., \phi_N\}$ represent the heights of histogram bins and in Eq. (2.25) they represent an ad-hoc collection of variables.

Foreground pixels were detected by thresholding a comparison between observed grayscale intensities $I_k(\mathbf{p})$ and the corresponding codebooks using a possibly nonlinear function f according to

$$L_k(\mathbf{p}) = \left\{ \begin{array}{ll} \text{Background} & : f(I_k(\mathbf{p}), M_k(\mathbf{p})) > T_k(\mathbf{p}) \\ \text{Foreground} & : f(I_k(\mathbf{p}), M_k(\mathbf{p})) \leq T_k(\mathbf{p}) \end{array} \right. , \tag{2.26}$$

where $T_k(\mathbf{p})$ is a possibly time varying threshold.

In [29], Haritaoglu proposed the W^4 surveillance system that employed the only background model that was specifically bimodal. The distribution of values in each pixel time series was characterized by the minimum, maximum, and maximum difference in grayscale intensity. Foreground pixels were identified by applying a threshold to the minimum intensity difference measured between pixel intensities from unsegmented video frames, and the minimum or maximum intensity values in the corresponding pixel level background model. In this way, the minimum and maximum values in the background model represented the two modes of the distribution. The threshold used for comparison was determined automatically from the median intensity difference computed over the entire background image. The W^4 algorithm achieved widespread success in a variety of outdoor surveillance applications focused on monitoring people and their activities.

Kim proposed the second Codebook background model in [48,49] where each pixel time series was modeled with a collection of common modes described by vectors composed of the RGB color, minimum and maximum brightness values, frequency of occurrence, maximum negative run length and the first and last access times. The codebook models were updated by learning vector quantization (LVQ) [50] and foreground pixels were identified by comparison to the codebooks at the corresponding locations in the video stream.

2.4.4 Binned Kernel Density Estimation (BKDE)

Binned kernel density estimation (BKDE) is hybrid technique that models statistical distributions using a parameterized histogram to reduce the amount of memory required to store large sample collections. Probability estimates are computed by applying a kernel function to the histogram parameters centered at the value in question.

In [106] Zhong proposed the only BKDE background model for representing localized image intensity features in the joint domain-range feature space originally proposed in [90]. The distribution of grayscale intensities in local rectangular regions surrounding each pixel were represented with a parameterized histogram. Foreground pixels were detected by comparing pixel intensities in unsegmented video frames with the parameters of the background histograms at the corresponding locations using a KDE technique. Although the authors achieved good results, they did not compare their model with state-of-the-art techniques and instead opted for a favorable comparison to the Stauffer and Grimson GMMs [93].

Chapter 3

Stochastic Scene Modeling

In the Fall of 2011, I implemented ViBe and immediately observed its superiority to several other well known scene modeling techniques, namely, the GMM of Stauffer and Grimson [93,94], the multidimensional median filter of [16], the temporal low-pass filter of [22] and the KDE technique proposed by Elgammal, Harwood and Davis in [23,24]. In [3], Barnich demonstrated the effectiveness of the ViBe model against the Zivkovic GMM [109], the Codebook proposed in [49], a pixel level single Gaussian model with adaptive variance, and several other lesser known techniques such as the $\Sigma - \Delta$ model [66], a Bayesian histogramming algorithm [58], an alternative GMM [43], and a simple temporal low-pass filter similar to [22]. In addition, Brutzer [13] independently verified the claims of Barnich by comparison to another collection of well known scene models that included a classical median filter [68], the Stauffer and Grimson GMM [93,94], the Oliver and Pentland Eigenbackground subtraction method [77], the single Gaussian model proposed in [69], a Bayesian histogram [58], the Codebook of [49], the Zivkovic GMM [109], and a self organizing map (SOM) [63].

Due to the preponderance of evidence in support of the ViBe model, I focused my attention on understanding the mechanisms within the Barnich system that allowed it to achieve superior results despite the fact that the

Table 3.1: Prominent Background Modeling Techniques

Author(s)	Model Description	Feature Vector	Feature V
Donohoe, Hush and Ahmed [22]	Temporal Low-Pass	Grayscale	Pixel
McKenna, Jabri, Duric, *et al.* [69]	Multivariate Normal	RGB/Sobel	Neighborh
Oliver, Rosario and Pentland [77]	PCA	Grayscale	Frame
Stauffer and Grimson [93, 94]	GMM	Grayscale/RGB	Pixel
Elgammal [23, 24]	Nonparametric	Grayscale/RGB	Pixel
Cucchiara, Piccardi and Prati [16]	Median Filter	RGB	Pixel
Zivkovic [109, 110]	GMM	Grayscale/RGB	Pixel
Kim, Thanarat, Chalidabbhognse, *et al.* [49]	Codebook	RGB	Pixel

algorithm is relatively simple and nearly parameterless. As I began to fully understand the inner workings of the algorithm, I became dissatisfied with many of the assumptions that were made by the authors with respect to the stochastic update policy, and began to form and evaluate my own theories that would eventually lead to several major improvements in stochastic scene modeling.

The ViBe scene model is a pixel level nonparametric background model that operates in the grayscale or RGB colorspaces and uses KDE to classify pixels in unsegmented video frames. The number of previously observed samples that are used to characterize the distributions of background values at each pixel location is fixed at twenty. The background probabilities of each pixel in an unsegmented frame are estimated by performing KDE using a spherical cutoff kernel with a fixed radius of twenty pixels. If the background probability is less than or equal to $\frac{1}{10}$, then the pixel is classified as foreground, otherwise it is classified as background and integrated into the system at the pixel level and possibly at the neighborhood level.

The ViBe model is unique in that it is the first and only scene model that uses a completely stochastic maintenance algorithm to integrate new information into the system. Pixels that are classified as background are automatically inserted into the sample collection at the corresponding pixel location. In contrast to existing nonparametric models where the oldest value in the sample collection would be replaced by the new value, ViBe uses a uniformly distributed random variable to determine the index of the sample to be replaced. In [3], the authors show that this policy ensures that the expected lifespan of each sample decays exponentially and that the probability of a sample being preserved is independent of time, and therefore the system is memoryless. In this dissertation, I propose a different update policy that replaces the most significant outlier in the sample collection. I argue that the outlying value is both the least important sample in a statistical sense and the most likely sample to represent a foreground component that has been included in the background model. In Chapter 4, I show that the outlier replacement policy has no negative impact on the performance of the algorithm and that it nearly eliminates the previously observed "everlasting ghost problem" which I will describe later in this section.

In addition to integrating background pixels into the corresponding pixel level models, the ViBe system may propagate new background values to a single neighboring distribution to promote spatial consistency throughout the scene [3]. A uniformly distributed random variable with a fixed one in sixteen chance is used to determine if the background model is to be propagated to a neighboring sample collection. In the case that the sample is selected for propagation, one of the eight neighboring sample collections is randomly selected using another uniformly distributed random variable. Selection of the sample within the neighboring distribution to be replaced is also performed by a stochastic process where a uniformly distributed random variable is used to determine the replacement index. The ViBe neighborhood diffusion process is based on an assumption that is composed of two contradictory premises, *viz.* that the structures of the neighboring distributions are similar enough that information can be randomly swapped without fear of corrupting the sample collections, and yet disparate enough that the swapping of information improves the diversity of each model in a constructive sense [3]. In the case where neighboring pixel level models lie on different sides of an edge boundary, the assumption that adjacent distributions are similar is clearly incorrect and will lead to unpredictable corruption of the two models through the Barnich diffusion algorithm as given in [3]. Because the neighboring substitution index is chosen at random, the potential for severe damage to the model is greatly increased as important and unimportant values are equally likely to be replaced. Indeed, all of my simulations with the Barnich model revealed unjustifiably high foreground probabilities along the edges of the background structure when examined prior to application of the final segmentation threshold. To combat the negative effects associated with the ViBe neighborhood diffusion process, I propose a modified algorithm that selects an adjacent sample collection using a stochastic process that favors the selection of similarly shaped distributions. In Chapter 4, I demonstrate the effectiveness of the modified spatial diffusion process on several challenging test videos and measure a significant reduction in the number of false foreground detections both before and after application of the segmentation threshold.

With the remainder of this chapter, I present a new stochastic scene model based on the ViBe algorithm that significantly improves upon the conservative update policy introduced by Barnich in [3]. I identify the following four scene modeling components and use them to describe the theoretical aspects of my algorithm and to compare and contrast the model with a wide variety of existing techniques.

Model representation: The collection of static and dynamic system parameters combined with data storage elements that represent the model at a single discrete time instant k.

Model initialization: The method by which the elements of the scene model are initialized at time $k = 0$.

Frame segmentation: The procedure used to compare an unsegmented video frame to the current instance of the model to arrive at a segmented video frame.

Model maintenance: The algorithm or update policy used to integrate new information into the existing scene model. The maintenance strategy may or may not make use of the segmented frame, but in general it will make use of the image features observed within the observed unsegmented video frame.

In addition, I provide the reader with a descriptive list of the challenging problems and definitions thereof that have been historically encountered in the field of video segmentation by aggregating the work of Toyama and Brutzer [13, 95] in the following comprehensive collection. From this point forward, I will use these terms to analyze both the theoretical aspects of my proposed algorithm as well as the simulation results that are presented in the following chapters.

Bootstrapping: In many situations, the scene model must be initialized from a single video frame in the presence of foreground objects, and because a trusted model of the scene does not yet exist, it is impossible to determine the difference between foreground and background objects. In the video segmentation literature, this procedure is known as bootstrapping, although the actual statistical term "bootstrap" is at best only loosely related to this process.

Gradual illumination changes: Reasonable changes in lighting conditions such as those that are naturally occurring and expected in outdoor environments.

Sudden illumination changes: Unexpected variations in lighting conditions that occur frequently in indoor settings, but are generally unpredictable.

Dynamic background components: Swaying tree branches, rippling water, and uninteresting components of the scenery are all common examples of dynamic background components. Unfortunately, the definitions of background and foreground are not completely straightforward, and thus, the term background may refer to any elements of the scenery that are unimportant to the application at hand. These are a subset of the broader class of image features that have been historically been referred to as *clutter*.

Camouflaged foreground components: Foreground objects that share very similar color and textural appearance with the background, making detection difficult if not impossible.

Shadows: Shadows may be cast by either foreground or background objects and they pose a significant challenge to video segmentation systems because they generally appear different from the known background components and thus they are incorrectly identified as foreground objects. Shadows have sometimes been considered as *clutter* in the classical literature.

Ghosts/waking person: When background objects suddenly become a part of the foreground such as in the case of a parked car leaving its space, the region uncovered by the object is, in many cases, incorrectly identified as a foreground object. If the incorrectly classified region is not quickly identified as part of the background in the model update step, then the object may linger for a long period of time and continue to appear as an everlasting ghost.

Foreground aperture: The situation in which homogeneously colored or textured regions within a moving foreground object are incorrectly identified as background structure because they do not appear to be in motion.

3.1 Model Representation

I employ a pixel level nonparametric model to characterize the temporal distributions of background image features according to [3, 23, 24]

$$M(\mathbf{p}) = \{\phi_1, \phi_2, \phi_3, \dots, \phi_N\}, \tag{3.1}$$

where M is a nonparametric model of the background scene represented by a collection of N previously observed values in the grayscale intensity feature space, and $\mathbf{p} = (x_1, x_2)$ are the horizontal and vertical coordinates of a single pixel. The reader should be aware that Eq. (3.1) is very similar to several equations presented in the previous section. The equations are similar in the sense that they each represent characterizations of pixel level distributions using a finite collection of variables. The difference between the equations is that in Eq. (3.1) $\{\phi_1, .., \phi_N\}$ are previously observed grayscale samples and in Eq. (2.25) and Eq. (2.25) they represent either histogram bin heights or a collection of ad-hoc variables. In terms of versatility, nonparametric models are unique in that they are well suited to the representation of multimodal statistical distributions where the number of modes is unknown and likely to change over time.

Historically, nonparametric models have been shown to provide excellent characterizations of highly dynamic background components and gradual variations in lighting conditions [3, 23, 24, 29, 48, 49, 71, 89, 90]. Naturally occurring changes in lighting conditions have been easily modeled with unimodal techniques; however, it is impossible to model dynamic background components simultaneously undergoing changes in lighting conditions with single modal models. Thus, nonparametric techniques have been generally accepted as a powerful tool in the modeling of complex outdoor environments [23].

With respect to the shadow identification problem, a wide variety of scene model representation techniques have been proposed that employ certain image features to assist shadow detection. Although I do not explicitly consider the shadow problem in this dissertation, I chose the grayscale feature space in part due to the overwhelming prior use of these types of features. In [34], the computational color model uses the variation in chromaticity and brightness distortion to segregate shadow components. In [97] the shadow values are predicted using a linear model in the grayscale feature space. Both [6] and [84] use multivariate Gaussian distributions to characterize shadows in the L*U*V* (CIELUV 1976) and grayscale Sobel derivative feature spaces, respectively. In [26], shadows are modeled using a single component of a GMM with *a priori* defined statistics in grayscale intensity. Zang modeled shadows in chromaticity and brightness [104], Joshi [42] employed edge features, and Elgammal [23, 24] used chromaticity (normalized r/g colorspace) coordinates alone.

3.2 Model Initialization

I performed a blind initialization of the model over N frames by assigning each grayscale value directly according to

$$\begin{aligned} M(\mathbf{p}) &= \{\phi_1, \phi_2, \phi_3, \ldots, \phi_N\} \\ &= \{I_{k-(N-1)}(\mathbf{p}), I_{k-(N-2)}(\mathbf{p}), I_{k-(N-3)(\mathbf{p}}, \ldots, I_k(\mathbf{p})\}, \end{aligned} \tag{3.2}$$

where I_k represents a single video frame at time k. Because descriptive information about the foreground and background structures is not generally available during the initialization process, and because the presence of moving foreground objects is both likely to occur and unlikely to be detected accurately, I elected to use a naive initialization strategy. With this approach, the effects of a moving object are spread over several spatial locations rather than concentrated at a single location as in the case of the single frame bootstrapping techniques.

In the ViBe model [3], initialization is performed by single frame bootstrapping and the samples are randomly selected from a 3×3 neighborhood centered about the model location using a uniformly distributed random variable. Unfortunately, this tactic increases the degree to which moving foreground objects corrupt the initial background model, because entire regions within the model will contain only foreground values. When the video processing begins, these moving foreground regions will begin to uncover the true background structure, resulting in both a true foreground detection due to the moving object, and a false foreground detection or ghost in the place of the objects original position. In addition, the random selection of values from a neighborhood may cause neighboring values from significantly different image regions to dominate or unfairly cripple the initial model of the background scene. For these reasons, I have adopted a simpler initialization method that avoids the accidental creation of a ghost and delays the neighborhood diffusion process until sufficient models of the foreground and background structure are available for use in the information sharing process.

3.3 Frame Segmentation

Segmentation was performed by thresholding the estimated background probabilities of each observed pixel value $I_k(\mathbf{p})$ within the unsegmented frame I_k

according to

$$L_k(\mathbf{p}) = \left\{ \begin{array}{lcl} \text{Foreground} & : & P(I_k(\mathbf{p})) < T \\ \text{Background} & : & \text{Otherwise} \end{array} \right. , \tag{3.3}$$

where T is a fixed threshold and $P(I_k(\mathbf{p}))$ is the background probability of a single observed pixel $I_k(\mathbf{p})$ estimated by

$$P(I_k(\mathbf{p})) = \frac{1}{N} \sum_{i=1}^{N} K(I_k(\mathbf{p}), \phi_i^{\mathbf{p}}). \tag{3.4}$$

In (3.4), $\phi_i^{\mathbf{p}}$ represents the i'th sample from the background model M at pixel location $\mathbf{p}$, and K is a uniform spherical cutoff kernel of radius R given by [3]

$$K(a, b) = \left\{ \begin{array}{lcl} 1 & : & |a - b| \leq R \\ 0 & : & \text{Otherwise} \end{array} \right. , \tag{3.5}$$

where $a, b \in \mathbb{R}$.

Pixel level segmentation techniques produce high resolution binary classification of foreground and background structures within video. In terms of the foreground aperture problem, these rich segmentations make it possible to use post segmentation algorithms to identify foreground details that penetrate the occluding background structures and use them to reconstruct a more accurate estimate of the object shape. Popular pixel level scene models that have featured post segmentation algorithms for dealing with the foreground aperture problem are the GMM of Stauffer and Grimson [93, 94], where foreground detections are combined through a connected components algorithm, and the nonparametric models of Elgammal [23, 24], where foreground regions are refined through a probabilistic analysis of the neighboring pixels. Not surprisingly, the advantage of high resolution segmentations is not completely without a few drawbacks, namely the susceptibility of pixel level algorithms to the foreground aperture problem. To combat the foreground aperture problem, a wide variety of post segmentation procedures have been proposed, such as a region growing operation by back-projection [95], morphological operations combined with a binary support map to strictly define the support of each foreground object [102, 104], and a probabilistic region growing algorithm [23, 24]. In the model that I propose in this dissertation, I do not perform any post segmentation processing. However, because information is shared among compatible neighboring models through the model update policy, the effects of foreground aperture and camouflage on the final segmentations are significantly reduced.

In one interesting case, shadows have also been detected through post segmentation processing by Cucchiara [16], where foreground regions were subjected to a gauntlet of size, saliency and motion thresholds to identify portions of the object believed to be shadows.

3.4 Model Maintenance

This section describes how new information is used to update the existing nonparametric models over time. I have divided this section into two distinct subsections that correspond to the primary contributions presented within this dissertation, namely, pixel and neighborhood learning algorithms.

3.4.1 Pixel-Level Learning

In the past, several different methods have been proposed for updating scene models over time. Integration of new observations into the existing scene model has generally been characterized as either *blind* or *conservative* based on the degree to which observed information is scrutinized prior to its incorporation in the model [3]. *Blind* learning techniques allow all of the observed information to be used to update the model, while *conservative* approaches apply a filter to the observed data to avoid the inclusion of information that would significantly disturb the existing model. In practice, the most conservative update strategy is to exclude foreground values from the model update and include background values; however, there are cases where other types of objects such as shadows and ghosts have been considered [16].

In parametric modeling, the most prominent learning technique is certainly the online k-means algorithm made popular in the Stauffer and Grimson GMM [93]. In the case of unimodal parametric models, new samples are integrated into the model by averaging them with the existing statistics using a learning rate parameter [69, 102]. In multimodal models, the first step is to associate the new sample with a mode in the existing model by maximum-likelihood estimation and then to update the parameters of the corresponding statistical structure by averaging the sample and the existing parameters using a learning rate [93]. In the Stauffer and Grimson GMM, the Gaussian function with the lowest mixing probability is replaced if the new sample does not match any of the existing modes. In some cases, adaptive learning rates have been used to greatly improve the performance of the model [109, 110].

Because nonparametric models characterize statistical distributions with fixed size sample collections, learning is generally conducted by replacement of the oldest value within the sample collection [24]. In the ViBe system, Barnich proposed a random replacement strategy and argued that it guaranteed a uniform decay of the model over time. I argue that it is not possible to detect how the underlying distribution of values at each pixel location is evolving over time and therefore importance cannot be assigned to the samples based on their age. In addition it cannot be assumed that all samples are of equal importance. Thus, I propose a scene model update policy where pixels that have been identified as background in the segmentation step are integrated into the existing pixel level models by replacing the most significant outlying samples. The proposed method does not assign importance to the samples based on their age. Instead, I assume that the samples are a reasonably good characterization of the underlying distribution of background image features and assign importance to the samples based on their role in the model. This replacement strategy is similar to the online k-means algorithm in that low probability regions within the model are more likely to be discarded and replaced with newer observations. In terms of scene modeling, this approach is reasonable because low probability regions within the background models are more likely to be caused by foreground variations in the video surface.

I define the location of the outlier l within each pixel level background model $M(\mathbf{p})$ to be the least probable value by estimating the probability of each sample with respect to the entire sample collection using KDE according to

$$l = \operatorname*{arg\,min}_{i=1,\ldots,N} \frac{1}{N} \sum_{j=1}^{N} K(\phi_i^{\mathbf{p}}, \phi_j^{\mathbf{p}}), \tag{3.6}$$

where $\phi_i^{\mathbf{p}}$ and $\phi_j^{\mathbf{p}}$ are samples from the model $M(\mathbf{p})$ and K is a spherical cutoff kernel given by Eq. (3.5). In (3.6), the radius of the kernel is computed from the data using a technique originally presented by Elgammal in [24], where the bandwidth is set to the median absolute deviation measured between all of the possible unique sample pairs and where pairs composed of identical samples are excluded. In the case where no unique outlier exists, the sample to be replaced is selected at random from the collection of minumum probability values identified using Eq. (3.6).

3.4.2 Neighborhood Information Sharing

The use of neighborhood information has appeared in all aspects of scene modeling and it is discussed here in the maintenance section because that is where it appears in the proposed model. In terms of representation, spatially localized image features such as block statistics [39], PCA [28, 73, 88], spatial gradients [37, 41, 69], textural properties [6, 33, 47], and statistical representations of domain and range components using multivariate Gaussians [102] or nonparametric models [90, 106] have been used to characterize neighborhood structures in video. In [90], Sheikh proposed the most successful of the representation-based spatially-conscious models, where each pixel level observation was represented by a five dimensional vector composed of the spatial coordinates combined with the RGB color value. The entire background scene was modeled by a single five-dimensional distribution, characterized with a nonparametric model, and probability estimates were obtained by KDE, where the spatial and color components of the bandwidth matrix were block diagonal.

In some cases, neighborhood information is only considered in the segmentation phase where, in general, pixel level models are combined to estimate probabilities associated with new observations. Obviously, any models that contain spatial information within the representation must consider this data in the segmentation procedure. In addition, most scene models perform post segmentation region grouping on pixel or block level detections to refine the initial segmentations. Common examples include connected components labeling, region growing, morphological removal of small regions, spatial median filtering, etc. The proposed method does not perform post processing of pixel level segmentations.

Prior to the work of Barnich, learning algorithms in pixel level scene models had not yet been distributed over a neighborhood in the model update step. Because the nonparametric pixel level models used in ViBe do not represent the spatial coordinates of the observations and the segmentation step is performed on independent pixel level models, samples from the background distributions are randomly injected into neighboring models in the update step [3]. I propose an improvement to the ViBe neighborhood diffusion algorithm that inhibits information sharing between significantly dissimilar background models. For a given pixel level model $M(\mathbf{p})$, I form a probability mass function by assigning a weight to each of the eight-connected neighboring background models $M(\mathbf{q})$ based on a measurement of the similarity between $M(\mathbf{p})$ and $M(\mathbf{q})$. Here, $\mathbf{q} \in \Lambda(\mathbf{p})$ where $\Lambda(\mathbf{p})$ represents the set of background pixels that are considered to be in the neighborhood of $\mathbf{p}$. The similarity metric w is computed by measuring the L^2 norm between histograms of the two sample distributions and then exponentiating the result according to

$$w_i(\mathbf{p}, \mathbf{q}) = \exp\left[-\left(\sum_{i=1}^{256} [h(M(\mathbf{p}))_i - h(M(\mathbf{q}))_i]^2 \right)^{(1/2)} \right], \qquad (3.7)$$

where $h(\cdot)$ is a function that takes a collection of values and produces a 256 bin histogram and $\mathbf{q} \in \Lambda(\mathbf{p})$. The neighboring distribution that the new background value will be inserted into is selected by drawing at random from the distribution defined by the normalized neighborhood similarity weights $\{w_i\}_{i \in |\Lambda(\mathbf{p})|}$. Once a neighboring distribution is selected, the value is integrated into the model using the outlier replacement strategy described in (3.6).

This update policy achieves excellent results against the ghost problem, because the image features associated with ghosts generally correspond to outliers in the background sample collections. In Chapter 4, I will demonstrate the effectiveness of the strategy in a classical everlasting ghost scenario.

With respect to the overarching problem of preventing misclassified foreground information from corrupting the background model, the proposed improvements to the stochastic neighborhood diffusion process significantly reduce the chances of model corruption in cases where the distributions are incompatible. By reducing the probability of sharing information between adjacent background models with significantly different shapes, the previously observed problem that resulted in high foreground probabilities along the edges of stationary background structures is nearly eliminated with the new strategy proposed in this section. In addition, the outlier replacement policy ensures that the neighboring distributions are only minimally transformed by the diffusion procedure, which is of utmost importance in cases where the adjacent model has been poorly chosen.

Chapter 4

Experiments

I selected four well known videos that have been frequently used in the literature to evaluate video segmentation algorithms. Two videos are from the performance evaluation of tracking and surveillance workshop [18] and the other two are from the University of California San Diego (UCSD) background subtraction dataset [64]. Table 4.1 summarizes the details of each video subsequence with respect to source, length, frame size, and literature appearances. The PETS 1 video corresponds to the PETS 2001 dataset 1 testing camera 1, and the PETS 2 video corresponds to the PETS 2001 dataset 3 testing camera 2 [18]. Table 4.2 summarizes the challenges that are present within each video sequence with respect to the list provided in Chapter 3.

4.1 Scene Model Evaluation

I manually created ground truth data for each video subsequence by carefully inspecting each frame over a period of several weeks. An example of a sin-

Table 4.1: Selected test video details.

Source	Name	Length	Frame Size	Appearances
UCSD	Rain	229	308×228	[23, 24]
UCSD	Beach	250	320×200	[71, 73]
PETS	PETS 1	200	768×576	None
PETS	PETS 2	225	768×576	[3]

Table 4.2: Test video details.

Challenge	Rain	Beach	PETS 1	PETS 2
Gradual Illumination Changes	x	x	x	x
Dynamic Background Components	x	x	x	x
Camouflage	x	x	x	x
Shadows	x	x	x	x
Ghosts			x	
Foreground Aperture	x	x		

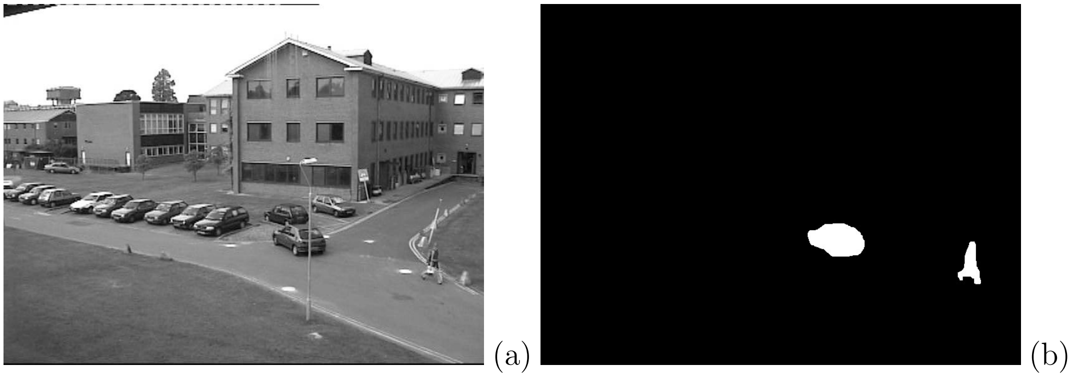

Figure 4.1: Frame 625 of the PETS 1 sequence depicting the original grayscale image (a) and the manually generated ground truth image (b).

gle ground truth frame for the PETS video is shown in Fig. 4.1. Each video was processed by the ViBe system and the proposed algorithm using Matlab. The results were compared using the well known percentage correct classification (PCC) [3, 25] and a new probability correct classification (PrCC) metric proposed here for the first time.

PCC is computed according to

$$\mathrm{PCC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}, \tag{4.1}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

To better identify the differences in the two scene models, I propose the probability of correct classification (PrCC) measurement and use it to evaluate each algorithm prior to application of the final segmentation threshold. I argue that the pixel-level foreground and background probabilities allow for a richer analysis of the scene models when compared to the alternative binary classification results that have been traditionally used to evaluate video segmentation systems. The PrCC is computed according to

$$\mathrm{PrCC} = \frac{\mathrm{TP}_{prob} + \mathrm{TN}_{prob}}{\mathrm{TP}_{prob} + \mathrm{TN}_{prob} + \mathrm{FP}_{prob} + \mathrm{FN}_{prob}} \tag{4.2}$$

where TP_{prob} is the sum of the foreground probabilities at the ground truth foreground pixel locations, TN_{prob} is the sum of the background probabilities at the ground truth background locations, FP_{prob} is the sum of the foreground probabilities at the ground truth background locations and FN_{prob} is the sum of the background probabilities at the ground truth foreground location.

My implementation of the ViBe algorithm is based on the pseudocode provided in the Barnich paper, and the results appear to be nearly identical

Table 4.3: Percentage Correct Classification (ViBe)

Sequence	TP	TN	FP	FN	PCC
Rain	142,565	14,386,446	21,838	55,743	99.5%
Beach People	108,355	13,293,313	869,206	65,126	93.5%
PETS 1	1,086,901	86,809,564	320,213	256,922	99.3%
PETS 2	179,163	90,053,770	372,580	79,927	**99.5%**

Table 4.4: Percentage Correct Classification (Proposed Algorithm)

Sequence	TP	TN	FP	FN	PCC
Rain	147,303	14,379,476	28,808	51,005	99.5%
Beach People	116,720	13,523,431	639,088	56,761	**95.2%**
PETS 1	1,112,343	86,911,973	217,804	231,480	**99.5%**
PETS 2	193,552	89,850,090	576,260	65,538	99.3%

to those presented in [3]. Because the algorithm is stochastic the results are expected to vary slightly between each application. Tables 4.3 and 4.4 summarize the results that were obtained by segmenting each video subsequence with the ViBe and proposed algorithms, respectively. The TP, TN, FP, and FN columns correspond to the total number of instances over the entire video sequence, and therefore the values are quite large.

With respect to the differences in performance, the proposed algorithm achieved an overall reduction of false positives (FP) of 7.70% and an increase in true positives (TP) of 3.49%. The proposed algorithm achieved an overall PCC of 99.1% compared to the ViBe result of 99.0%. These quantities were calculated by combining the PCC data from all of the videos.

In Table 4.5, I provide the PrCC results that highlight somewhat larger differences in the two models. I believe that the results shown here in Table 4.5 agree more with visually observed segmentation results because they identify the true differences in the underlying scene models prior to application

Table 4.5: Probability Correct Classification

Sequence	ViBe	Proposed Algorithm
Rain	96.9%	**99.4%**
Beach People	85.6%	**87.5%**
PETS 1	93.5%	**99.2%**
PETS 2	90.6%	**98.3%**

of the final classification threshold. In addition, I believe that it is important to measure the underlying models because it is not always possible to select appropriate segmentation thresholds, and because dependence upon a threshold is not a prerequisite design feature of scene models.

4.2 Selected Video Frames

In this section, four collections of video frames illustrate the differences between the proposed algorithm and ViBe. A detailed analysis of each video with respect to the previously identified challenges is provided in each of the following subsections. Each figure is organized as follows:

(a) Grayscale video frame,

(b) Ground truth frame,

(c) ViBe foreground probability image,

(d) ViBe segmentation image,

(e) Proposed method foreground probability image,

(f) Proposed method segmentation image.

4.2.1 Rain Sequence

The rain video sequence features gradual illumination changes, dynamic background components, camouflaged foreground objects, shadows and the foreground aperture condition. I processed 229 frames of the video, which represents the entirety of the sequence, and selected ten frames for presentation including the first frame and each 25th frame beginning with frame number 22. In this sequence, a large truck, a small car, and a walking person are depicted traveling through an intersection in a rainy and somewhat windy outdoor setting. Throughout the sequence, gradual illumination changes as well as dynamic appearance of the background structures due to the outdoor environment are present in the scene.

A relatively static outdoor scene is depicted in frame 22 (Fig. 4.2) of the rain sequence. The observable imagery is devoid of foreground objects and composed entirely of static background structure. Tables 4.6 and 4.7 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Evaluation by PCC indicates a 100% reduction of false positives and a 0.01% increase in true negatives. PrCC measurements indicate an identical 100% reduction in false positives and a 2.92% increase in true negatives. In terms of overall performance the proposed algorithm improves on ViBe by 0.01% and 2.92% measured by PCC and PrCC, respectively.

Fig. 4.3 begins to expose some of the drawbacks of ViBe in an outdoor scene composed of only static background structures. Tables 4.8 and 4.9 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In terms of PCC, the proposed algorithm outperforms ViBe by 0.01% due to a 0.01% increase in true negatives and a 100% reduction of false positives. PrCC measurements indicate an overall improvement of 2.89% based on a 2.89% increase in true negatives and a

99.97% reduction in false positives.

Frame 50 (Fig. 4.4) further emphasizes the tendency of the ViBe algorithm to produce false foreground detections along edge boundaries. In this frame the windy outdoor conditions cause movement of the tree branches on the left side of the frame, resulting in nonzero foreground probabilities in the proposed system (Fig. 4.4(e)). Tables 4.10 and 4.11 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. PCC measurements indicate a 0.01% increase in true negatives and a 100% reduction in false positives, resulting in an overall improvement of 0.01%. Evaluation by PrCC results in an overall performance improvement of 2.72% based on a 2.72% increase in true negatives and a 99.52% reduction of false positives.

In Frame 75 4.5 a large moving truck enters the scene from the right at a high rate of speed. The truck is easily detected by both ViBe and the proposed algorithm. Tables 4.12 and 4.13 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In terms of PCC, the proposed algorithm achieves an 4.92% increase in true positives and an 11.92% reduction in false positives, while measurements taken by PrCC produce only a minor true positive increase of 0.47% and a much larger decrease in false positives of 84.98%. Overall, the performance of the proposed algorithm is superior to ViBe in terms of both PCC (0.09%) and PrCC (2.59%).

After rounding the corner in the road, the truck exits the scene in frame 100 of the rain sequence (Fig. 4.6). Tables 4.14 and 4.15 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In this frame, an unexpected camera motion has caused significant disturbances to both models, leading to larger than normal false positive detections in both algorithms. Nonetheless, the performance of the two models remains comparable as they are both affected by the motion of the camera. Overall performance results indicate a 0.14% reduction in PCC and a 3.02% increase in PrCC.

Frame 125 (Fig. 4.7) depicts a scene composed of static and dynamic background components and no foreground objects. Tables 4.16 and 4.17 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. According to the PCC metric the proposed algorithm suffers from a 33.33% increase in false positives, resulting in a minor performance decline of 0.01%. By PrCC, the proposed algorithm achieves a 2.58% increase in true negatives and a 98.98% decrease in false positives. This results in an overall PrCC increase of 2.58%.

In frame 150 (Fig. 4.8) a person enters the scene and walks down and to the left. Tables 4.18 and 4.19 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Based on PCC the proposed technique suffers from a small decline in true negatives of 0.01%, a large increase in false positives of 133.33% and a false negative reduction of 5.77%. In terms of PrCC the proposed algorithm increases true negatives by 2.46% and reduces false positives by 98.49%. Overall, the proposed algorithm outperforms ViBe in both PCC and PrCC by 0.01% and 2.46%, respectively.

In Fig. 4.9, the walking person continues to travel towards the bottom left corner of the image. Tables 4.20 and 4.21 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. By PCC the proposed system decreases false positives by 4.8% and increases true negatives by 0.01%, resulting in an unchanged performance. The PrCC metric indicates a 2.44% increase in performance when compared to ViBe owing to a 2.45% increase in true negatives and a 98.66% decrease in false positives.

By frame 200 (Fig. 4.10), the walking person has exited the viewable region and a fast moving car appears in the center of the frame. The car entered the frame on the right side in the same location as the truck and it is traveling to the left. Once again the camera has sustained a large unexpected motion that has resulted in many false foreground detections in both ViBe and the proposed algorithm. Tables 4.22 and 4.23 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. The PCC and PrCC performance differences for the two scene models indicate a reduction in PCC between ViBe and the proposed algorithm of 0.44%. According to PrCC the proposed scene model outperforms ViBe by 2.16%.

Frame 225 (Fig. 4.11) depicts yet another scene without foreground objects. Tables 4.24 and 4.25 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. PCC measurements indicate that the proposed algorithm suffers from a decrease in false negatives of 0.02%, resulting in an overall minor performance decline of 0.02%. According to the PrCC metric the proposed technique achieves an improvement in true negative detection of 2.46% and a false positive reduction of 97.42%, resulting in an overall performance improvement of 2.46%.

Table 4.6: Rain Frame 22 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	0	0	0.00%
TN	70,219	70,224	0.01%
FP	5	0	-100.00%
FN	0	0	0.00%
PCC	99.99%	100.00%	0.01%

Table 4.7: Rain Frame 22 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	0.00	0.00	0.00%
TN_{prob}	67,964.16	69,949.69	2.92%
FP_{prob}	1,985.52	0.00	-100.00%
FN_{prob}	0.00	0.00	0.00%
PrCC	97.16%	100.00%	2.92%

Table 4.8: Rain Frame 25 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	0	0	0.00%
TN	70,218	70,224	0.01%
FP	6	0	-100.00%
FN	0	0	0.00%
PCC	99.99%	100.00%	0.01%

Table 4.9: Rain Frame 25 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	0.00	0.00	0.00%
TN_{prob}	67,986.53	69,949.09	2.89%
FP_{prob}	1,963.16	0.59	-99.97%
FN_{prob}	0.00	0.00	0.00%
PrCC	97.19%	100.00%	2.89%

Table 4.10: Rain Frame 50 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	0	0	0.00%
TN	70,218	70,224	0.01%
FP	6	0	-100.00%
FN	0	0	0.00%
PCC	99.99%	100.00%	0.01%

Table 4.11: Rain Frame 50 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	0.00	0.00	0.00%
TN_{prob}	68,085.68	69,940.77	2.72%
FP_{prob}	1,864.01	8.92	-99.52%
FN_{prob}	0.00	0.00	0.00%
PrCC	97.34%	99.99%	2.72%

Table 4.12: Rain Frame 75 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	1,566	1,643	4.92%
TN	67,752	67,740	-0.02%
FP	260	272	4.62%
FN	646	569	-11.92%
PCC	98.71%	98.80%	0.09%

Table 4.13: Rain Frame 75 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	1,669.26	1,677.14	0.47%
TN_{prob}	65,703.13	67,439.39	2.64%
FP_{prob}	2,043.20	306.93	-84.98%
FN_{prob}	534.10	526.22	-1.48%
PrCC	96.32%	98.81%	2.59%

Table 4.14: Rain Frame 100 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	4,673	4,752	1.69%
TN	64,454	64,280	-0.27%
FP	428	602	40.65%
FN	669	590	-11.81%
PCC	98.44%	98.30%	-0.14%

Table 4.15: Rain Frame 100 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	4,757.88	4,758.21	0.01%
TN_{prob}	61,813.63	63,824.99	3.25%
FP_{prob}	2,814.93	803.57	-71.45%
FN_{prob}	563.26	562.93	-0.06%
PrCC	95.17%	98.05%	3.02%

Table 4.16: Rain Frame 125 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	0	0	0.00%
TN	70,221	70,220	0.00%
FP	3	4	33.33%
FN	0	0	0.00%
PCC	100.00%	99.99%	0.00%

Table 4.17: Rain Frame 125 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	0.00	0.00	0.00%
TN_{prob}	68,171.72	69,930.60	2.58%
FP_{prob}	1,777.96	19.09	-98.93%
FN_{prob}	0.00	0.00	0.00%
PrCC	97.46%	99.97%	2.58%

Table 4.18: Rain Frame 150 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	93	102	9.68%
TN	69,972	69,968	-0.01%
FP	3	7	133.33%
FN	156	147	-5.77%
PCC	99.77%	99.78%	0.01%

Table 4.19: Rain Frame 150 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	111.43	110.95	-0.43%
TN_{prob}	68,003.14	69,675.96	2.46%
FP_{prob}	1,698.52	25.70	-98.49%
FN_{prob}	136.60	137.08	0.35%
PrCC	97.38%	99.77%	2.46%

Table 4.20: Rain Frame 175 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	141	147	4.26%
TN	69,957	69,952	-0.01%
FP	1	6	500.00%
FN	125	119	-4.80%
PCC	99.82%	99.82%	0.00%

Table 4.21: Rain Frame 175 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	153.10	152.37	-0.48%
TN_{prob}	67,997.52	69,662.15	2.45%
FP_{prob}	1,687.21	22.58	-98.66%
FN_{prob}	111.86	112.59	0.65%
PrCC	97.43%	99.81%	2.44%

Table 4.22: Rain Frame 200 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	672	756	12.50%
TN	68,597	68,205	-0.57%
FP	114	506	343.86%
FN	841	757	-9.99%
PCC	98.64%	98.20%	-0.44%

Table 4.23: Rain Frame 200 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	788.84	786.52	-0.29%
TN_{prob}	66,210.07	67,661.79	2.19%
FP_{prob}	2,232.53	780.80	-65.03%
FN_{prob}	718.25	720.57	0.32%
PrCC	95.78%	97.85%	2.16%

Table 4.24: Rain Frame 225 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	0	0	0.00%
TN	70,224	70,212	-0.02%
FP	0	12	0.00%
FN	0	0	0.00%
PCC	100.00%	99.98%	-0.02%

Table 4.25: Rain Frame 225 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	0.00	0.00	0.00%
TN_{prob}	68,225.68	69,905.13	2.46%
FP_{prob}	1,724.01	44.56	-97.42%
FN_{prob}	0.00	0.00	0.00%
PrCC	97.54%	99.94%	2.46%

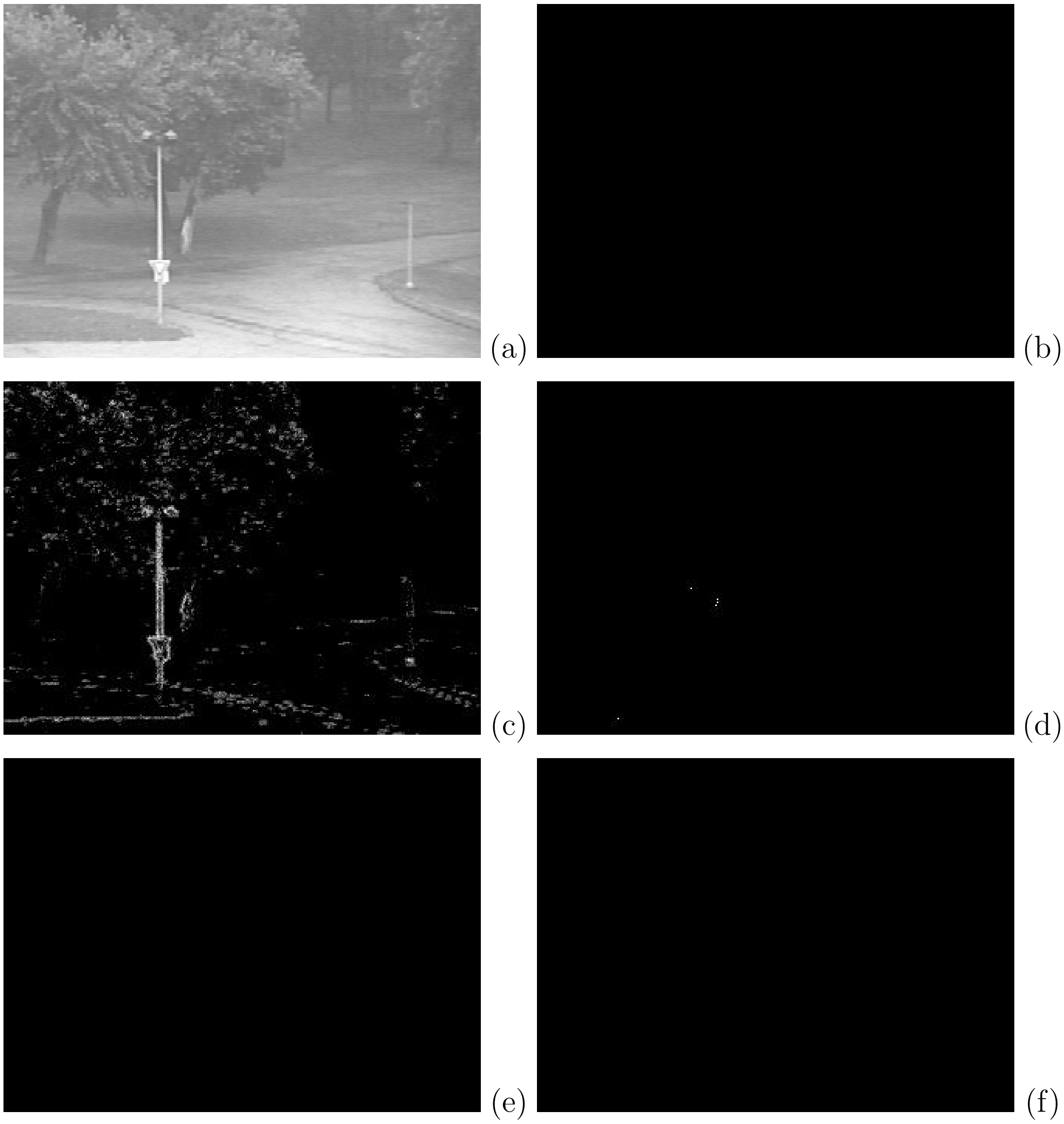

Figure 4.2: Frame 22 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=100%)(c), ViBe final segmentation (PCC=99.99%)(d), the foreground probability image for the proposed algorithm (PrCC=97.16%)(e), and the final segmentation for the proposed algorithm (PCC=100%)(f).

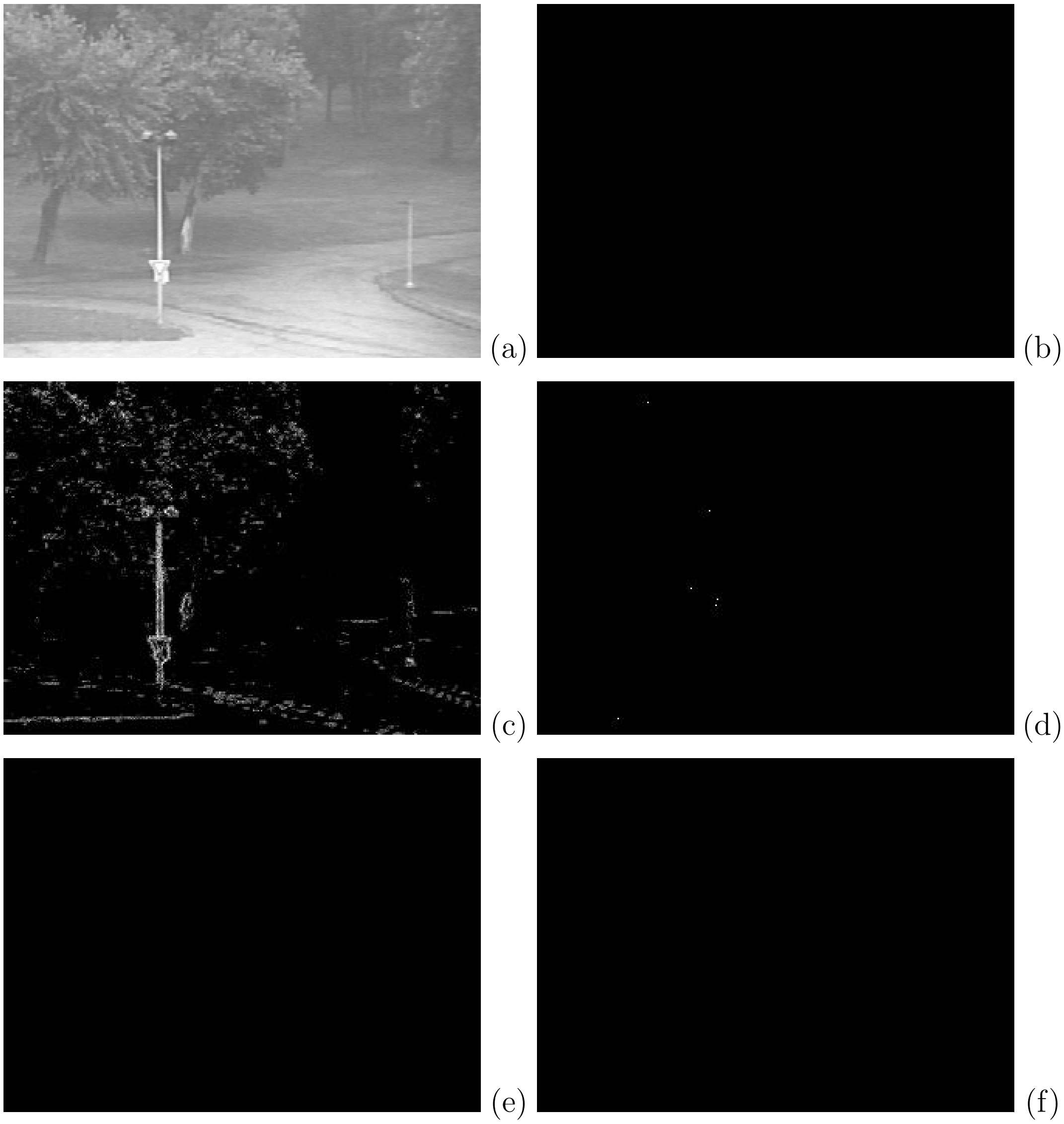

Figure 4.3: Frame 25 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=97.19%)(c), ViBe final segmentation (PCC=99.99%)(d), the foreground probability image for the proposed algorithm (PrCC=99.99%)(e), and the final segmentation for the proposed algorithm (PCC=100%)(f).

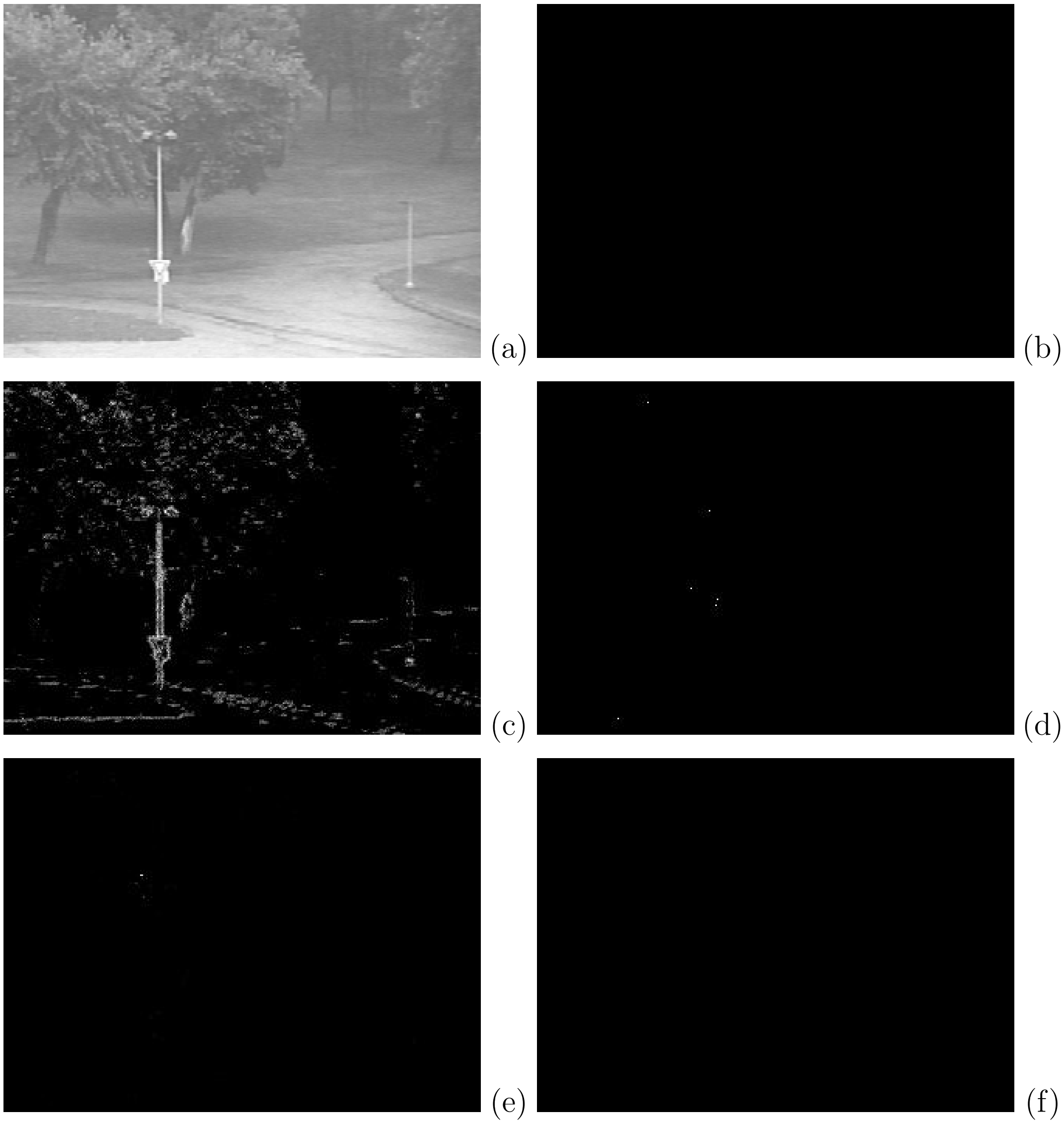

Figure 4.4: Frame 50 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=97.34%)(c), ViBe final segmentation (PCC=99.99%)(d), the foreground probability image for the proposed algorithm (PrCC=99.99%)(e), and the final segmentation for the proposed algorithm (PCC=100%)(f).

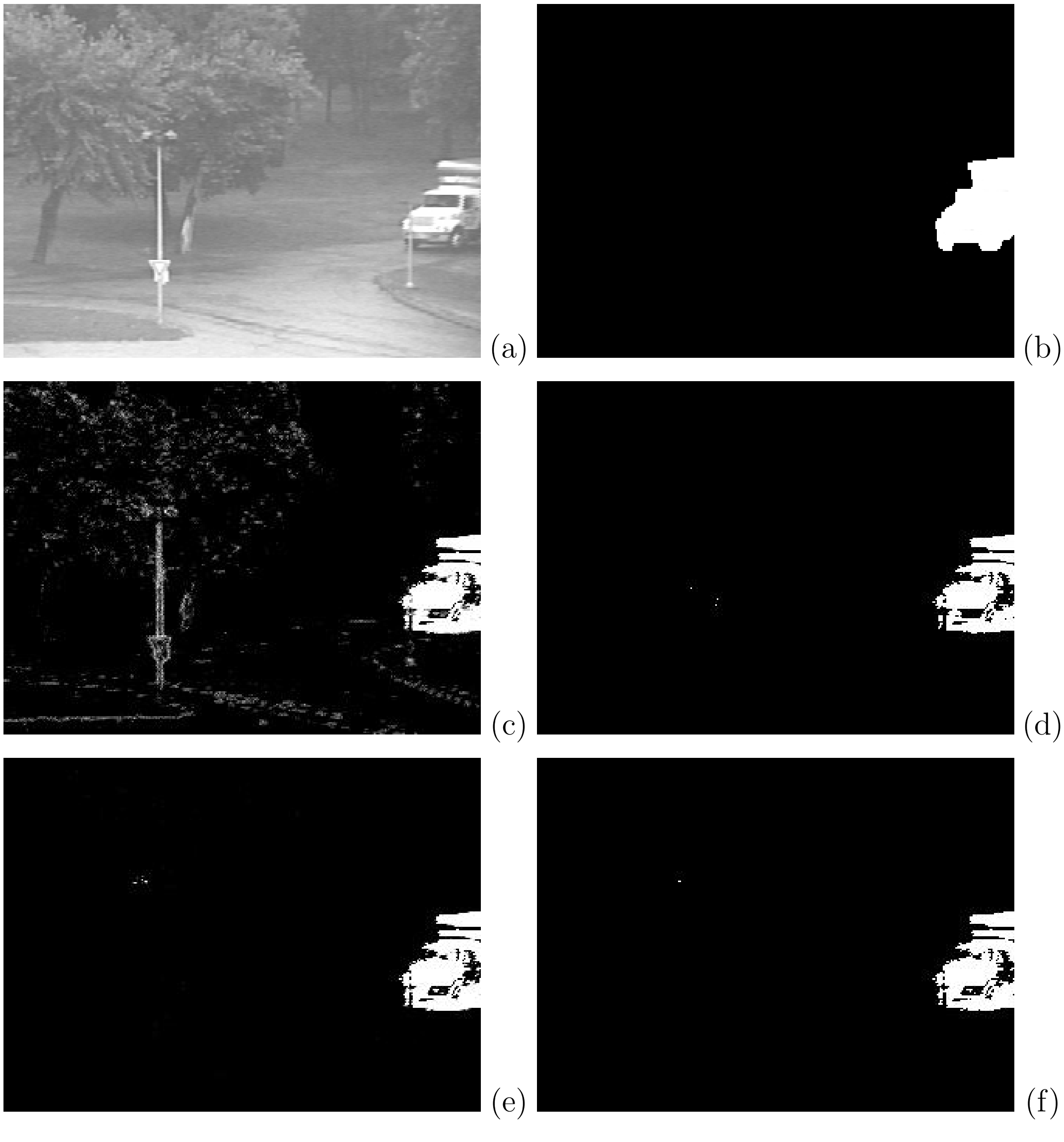

Figure 4.5: Frame 75 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=96.32%)(c), ViBe final segmentation (PCC=98.71%)(d), the foreground probability image for the proposed algorithm (PrCC=98.80%)(e), and the final segmentation for the proposed algorithm (PCC=98.80%)(f).

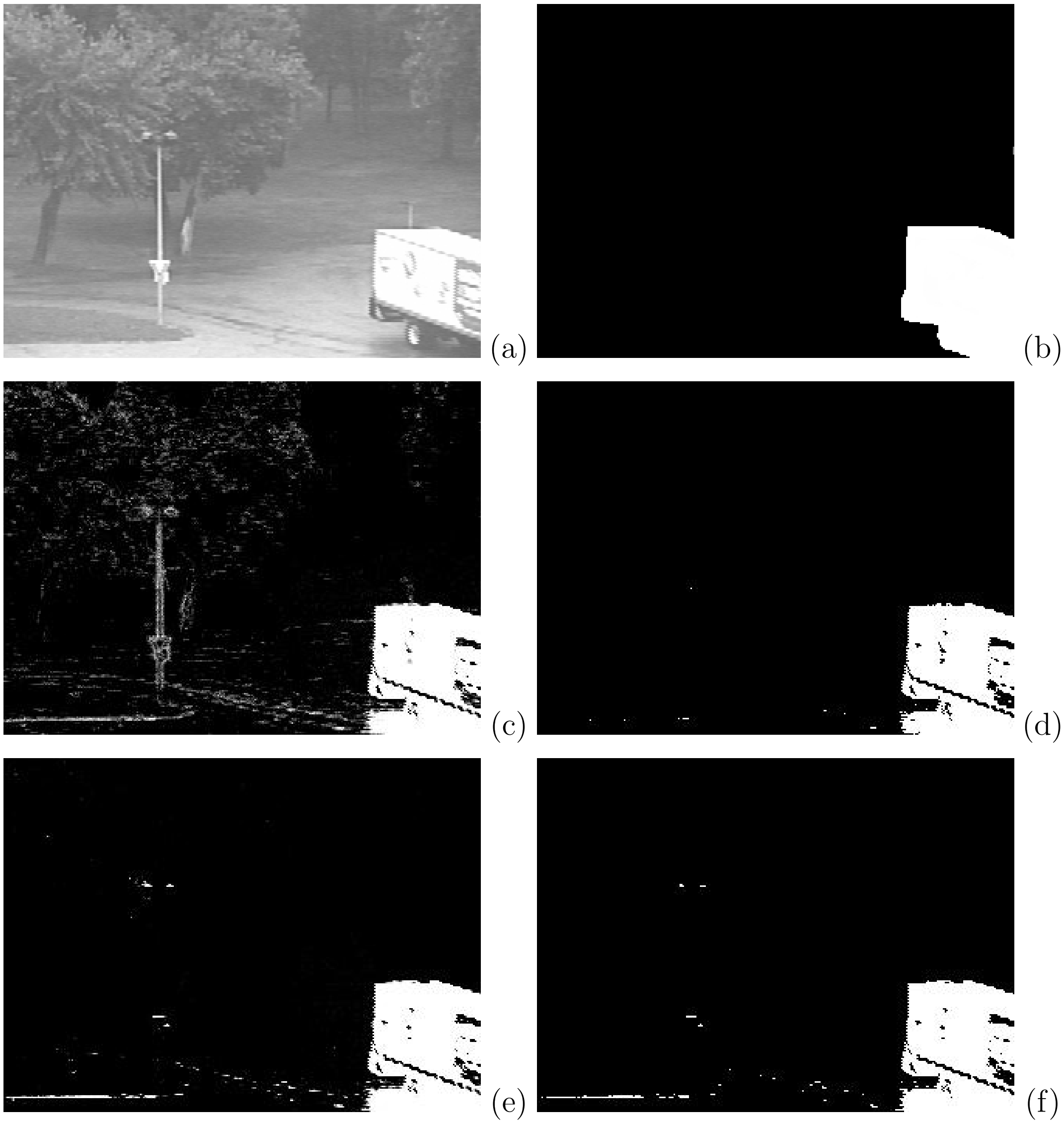

Figure 4.6: Frame 100 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=95.17%)(c), ViBe final segmentation (PCC=98.44%)(d), the foreground probability image for the proposed algorithm (PrCC=98.05%)(e), and the final segmentation for the proposed algorithm (PCC=98.30%)(f).

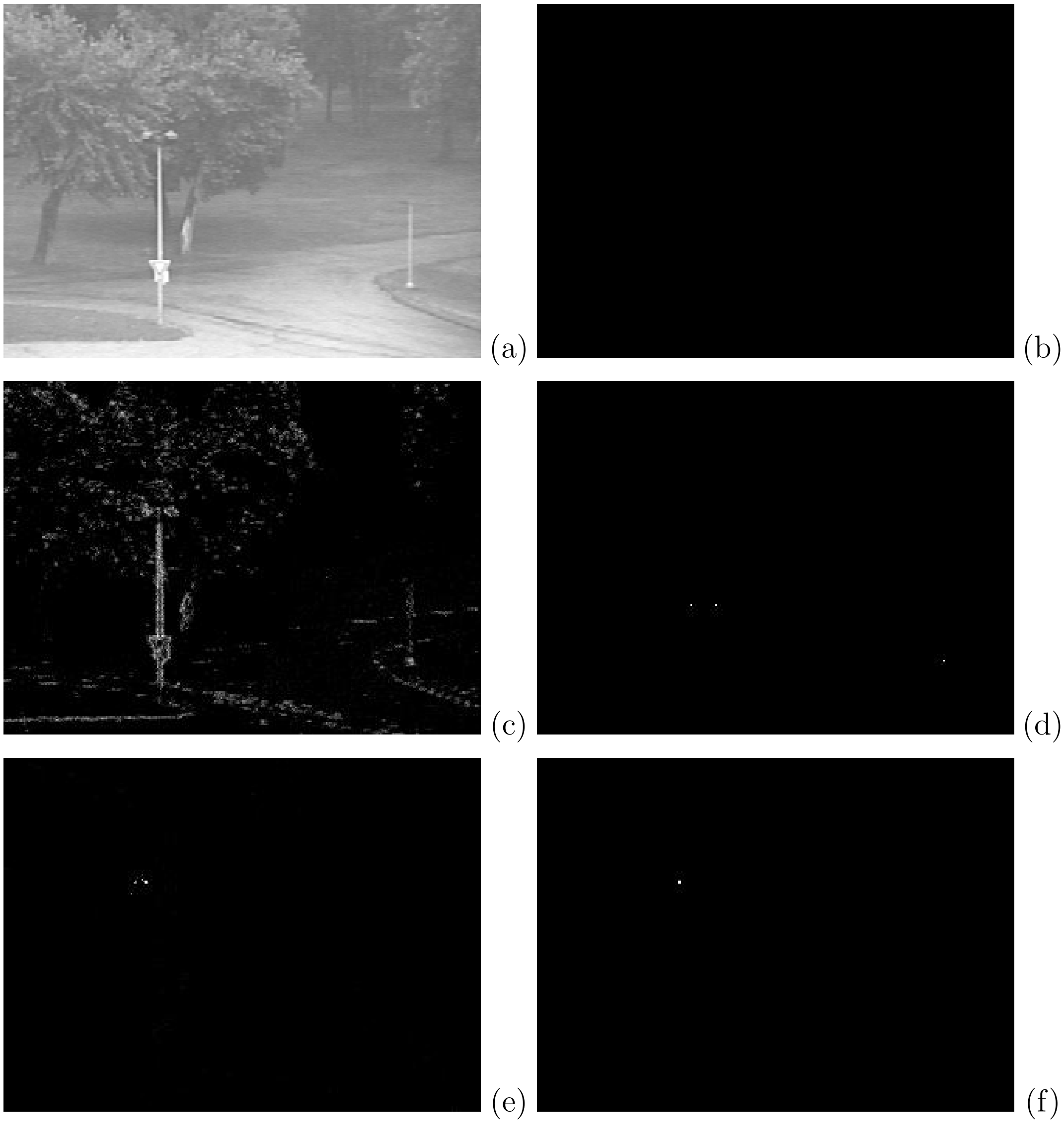

Figure 4.7: Frame 125 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=97.46%)(c), ViBe final segmentation (PCC=100%)(d), the foreground probability image for the proposed algorithm (PrCC=99.97%)(e), and the final segmentation for the proposed algorithm (PCC=99.99%)(f).

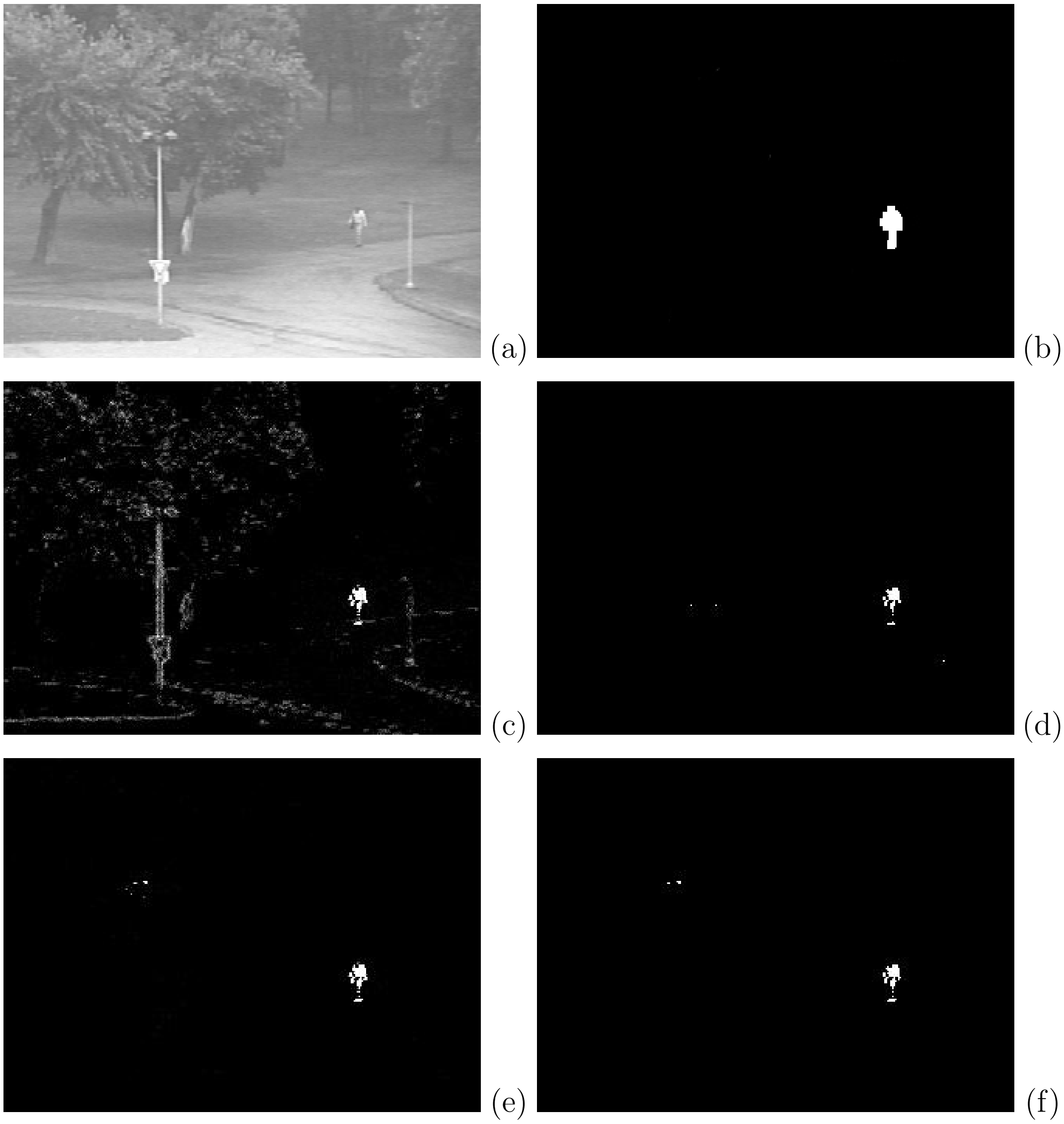

Figure 4.8: Frame 150 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=97.38%)(c), ViBe final segmentation (PCC=97.77%)(d), the foreground probability image for the proposed algorithm (PrCC=99.77%)(e), and the final segmentation for the proposed algorithm (PCC=99.78%)(f).

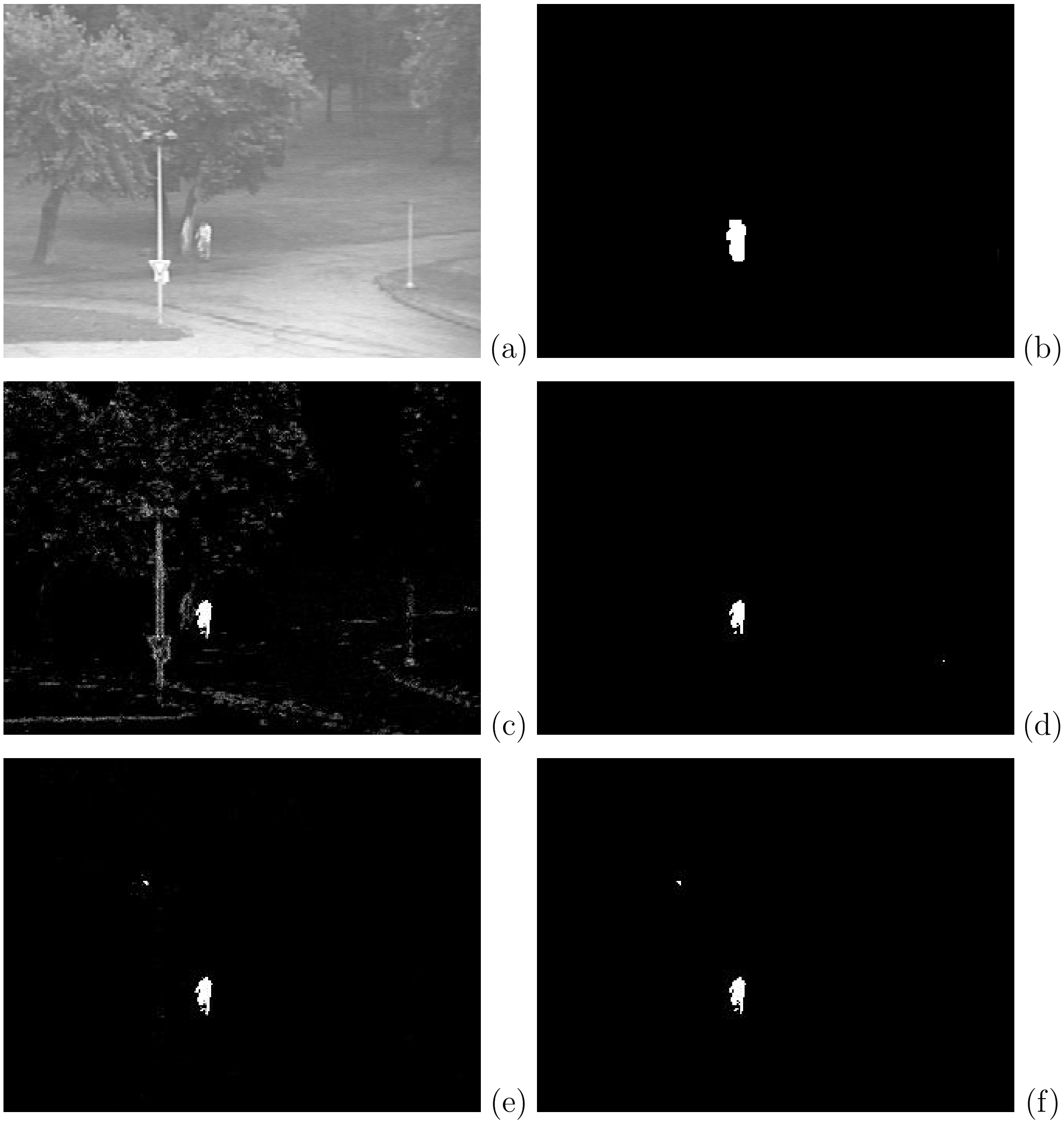

Figure 4.9: Frame 175 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=97.43%)(c), ViBe final segmentation (PCC=99.82%)(d), the foreground probability image for the proposed algorithm (PrCC=99.81%)(e), and the final segmentation for the proposed algorithm (PCC=99.82%)(f).

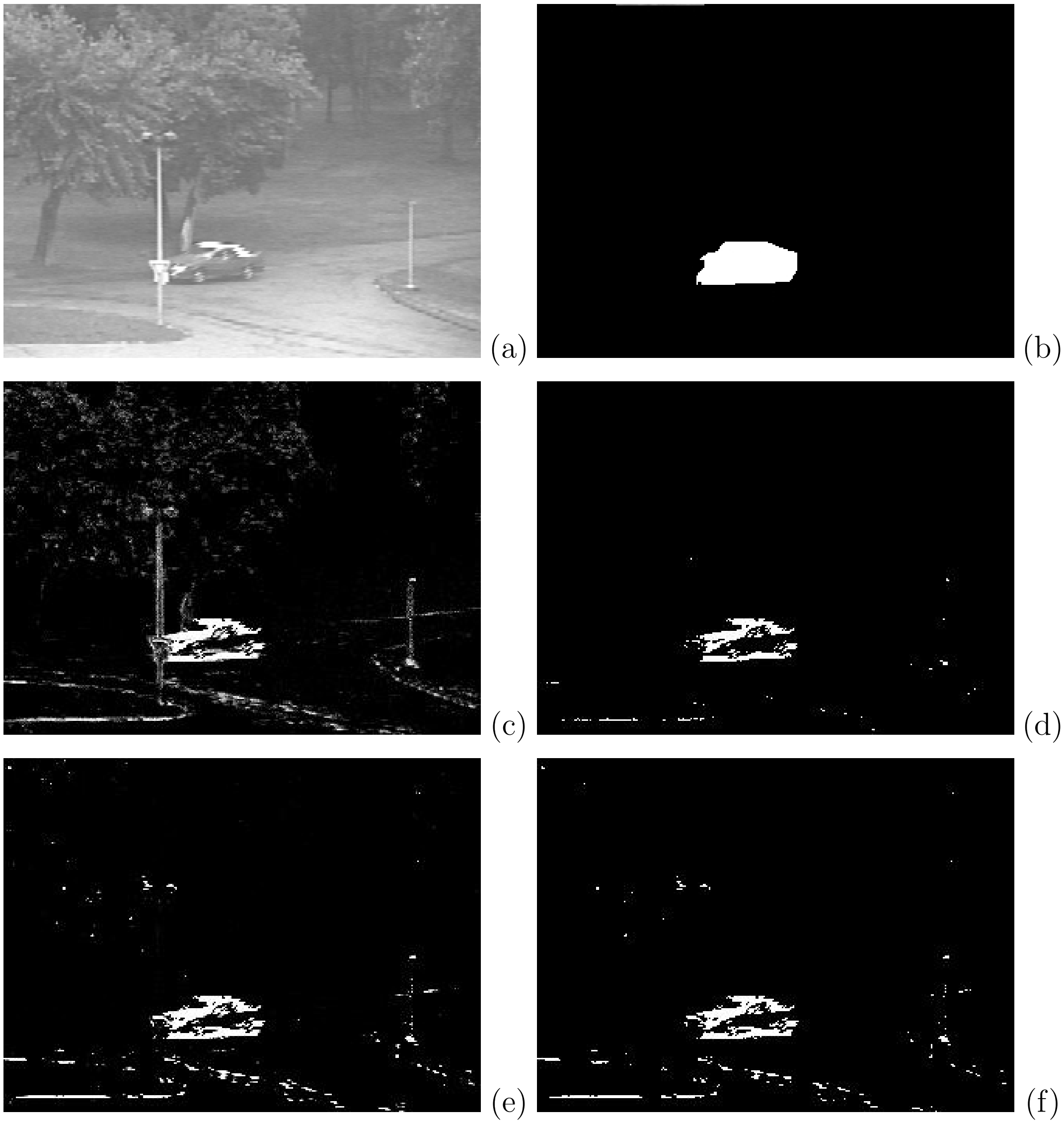

Figure 4.10: Frame 200 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=95.78%)(c), ViBe final segmentation (PCC=98.64%)(d), the foreground probability image for the proposed algorithm (PrCC=97.85%)(e), and the final segmentation for the proposed algorithm (PCC=98.20%)(f).

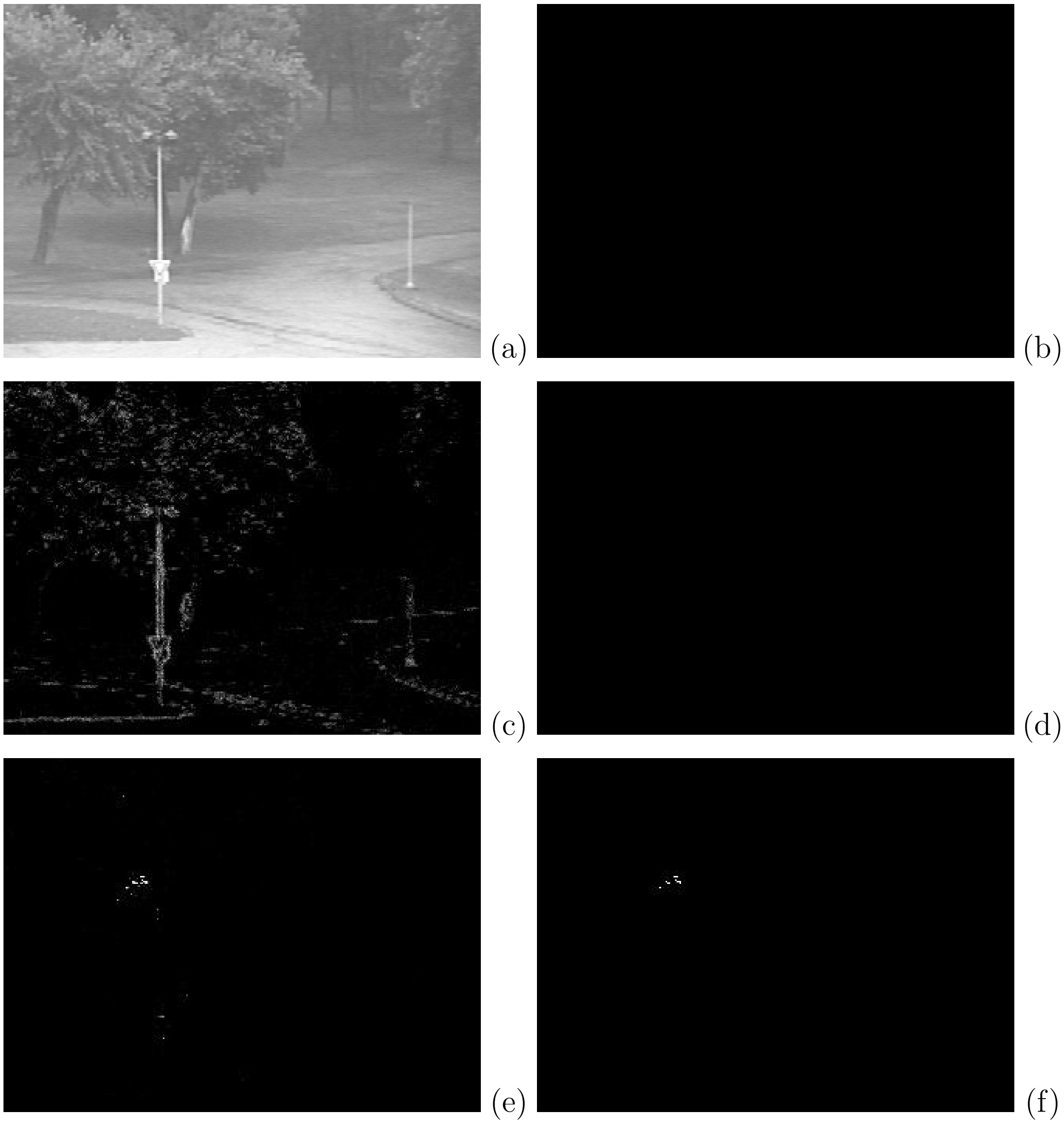

Figure 4.11: Frame 225 of the Rain sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=97.54%)(c), ViBe final segmentation (PCC=100%)(d), the foreground probability image for the proposed algorithm (PrCC=99.93%)(e), and the final segmentation for the proposed algorithm (PCC=99.98%)(f).

4.2.2 Beach Sequence

The beach video sequence features gradual illumination changes, dynamic background components, camouflaged foreground objects, shadows and the foreground aperture condition. I processed 250 frames of the video, which represents the entirety of the sequence, and selected ten frames for presentation including the first and last frames and each 25th frame beginning with frame number 50. In this video, two people enter the frame on the lower right side and walk across a beach from right to left before leaving the frame. A long wooden fence in the center of the frame occludes the people when they cross and ocean waves undergo complex motion in the top portion of the scene. Gradual illumination changes and dynamic background components dominate large regions within the beach sequence. The outdoor environment coupled with a slight wind result in a video where the majority of the background structure is constantly undergoing highly dynamic motion. Illumination changes are present in the sand and the water reflection, while the wind only seems to affect the brush in the bottom right of the scene.

Fig. 4.12 depicts frame 27 of the beach sequence composed of static and dynamic background components and no foreground objects. The most notable features are the crashing ocean waves that dominate the uppermost region in the frame, a long vertical fence that divides the frame and the vegetation in the lower right corner that sways in the wind. Tables 4.26 and 4.27 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Based on PCC the performance of the two algorithms is identical on frame 27. In terms of PrCC, the proposed algorithm achieves an increase in true negatives of 7.08% and a decrease in false positives of 41.83%, resulting in an overall improvement of 7.08%.

In frame 50 (Fig. 4.13), two people have entered the frame in the lower right corner and they are traveling from right to left. In this frame, the people are occluded by a small tree in the lower right area of the image. Tables 4.28 and 4.29 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. PCC measurements indicate increases in true positives and true negatives of 11.37% and 5.70%, and decreases in false positives and false negatives of 72.46% and 15.99%. PrCC measurements based on the foreground probability images indicate minor improvements in all areas. Overall, PCC and PrCC measurements indicate a performance increase of 5.75% and 5.42%.

As the people continue to walk from right to left in the video they emerge from behind the small tree and near the right side of the long vertical fence in frame 75 (Fig. 4.14). Tables 4.30 and 4.31 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Based on PCC the proposed scene model improves true positive and true negative detections by 8.22% and 5.77%, and reduces false positives and false negatives by 61.07% and 13.67%. According the PrCC, true positives decline by 0.89%, true negatives improve by 5.49%, false positives are reduced by 29.11% and false negatives increase 2.19%. Overall, the performance results obtained by PCC and PrCC are very similar with a reported PCC improvement of 5.80% and PrCC increase of 5.39%, both favoring the proposed algorithm.

In frame 100 (Fig. 4.15) the walking people are both partially occluded by the fence that runs perpendicular to the ocean. Tables 4.32 and 4.33 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In terms of PCC the proposed algorithm achieves an increase in true positives of 7.53% and a decrease in true negatives of 59.90%, resulting in an overall improvement of 4.34%. According to PrCC, the proposed model outperforms ViBe by 4.77% based on a false positive reduction of 27.79%.

Frame 125 (Fig. 4.16) depicts the same two people in an apparent conversation on the right side of the fence. In this frame, the people have stopped walking and their motion is consistent with that of two humans engaging in normal conversation. As such, their positions change continually and one of the people is more visible than the other. Tables 4.34 and 4.35 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. According to both PCC and PrCC the proposed technique improves on ViBe by 0.76% and 1.22%, respectively. PCC measurements indicate an 11.42% increase in true positives and a 28.16% reduction in false positives. In terms of PrCC the differences are much more subtle with a true positive improvement of 1.40% and a false positive reduction of 6.90%.

In Fig. 4.17 (frame 150) the people have moved to the right of the fence and they are more visible than in frames 125 and 100. Tables 4.36 and 4.37 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. According to PCC the proposed scene model improves true positive detections by 7.22% and reduces false positives by 37.83%, resulting in an overall improvement of 2.56%. By PrCC the proposed algorithm improves true negatives by 2.69% and reduces false positives by 15.82%, yielding an overall performance increase of 2.65%.

Frame 175 (Fig. 4.18) sees the two people continue to travel from right to left and they are just beginning to cross through what appears to be an opening in the fence. Tables 4.38 and 4.39 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. According to PCC measurements the proposed scene model improves true positives by 8.90%, reduces false positives by 21.37% and reduces false negatives by 10.99%. In terms of PrCC the main highlight of the proposed technique is a false positive reduction of 6.56%. Overall the proposed method achieves improvements of 0.70% and 1.07% when measured according to PCC and PrCC.

In frame 200 (Fig. 4.19) one person has emerged walking from right to left on the left side of the fence and the other person is still completely occluded by the fence. Tables 4.40 and 4.41 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Using the PCC metric we observe a 7.76% increase in true positives and a 28.63% increase in true negatives, leading to an overall reduction in performance of 1.43%. The PrCC metric yields similar measurement results with a performance reduction of only 0.22% based largely on a 4.22% reduction in true positives and a 1.25% increase in false positives.

Frame 225 (Fig. 4.20) depicts both people walking from right to left after crossing the fence. In comparison to previous frames from the beach sequence the distance between the two walking people and the sizes of their respective shadows have significantly increased. Tables 4.42 and 4.43 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Based on PCC the proposed algorithm achieves a 3.69% increase in true positives and an unfortunate 27.96% increase in false

Table 4.26: Beach Frame 27 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	0	0	0.00%
TN	62,756	62,756	0.00%
FP	1,244	1,244	0.00%
FN	0	0	0.00%
PCC	98.06%	98.06%	0.00%

Table 4.27: Beach Frame 27 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	0.00	0.00	0.00%
TN_{prob}	54,521.21	58,381.67	7.08%
FP_{prob}	9,228.79	5,368.33	-41.83%
FN_{prob}	0.00	0.00	0.00%
PrCC	85.52%	91.58%	7.08%

positives, resulting in an overall performance decline of 1.16%. By PrCC the proposed algorithm suffers a 0.37% performance reduction when compared to ViBe based largely on a 4.07% reduction in true positives and an 8.27% increase in false negatives.

In the last frame of the sequence (frame 250), one of the people has walked past the left edge of the frame and all that remains visible is the remaining person and part of the first person's shadow (Fig. 4.21). Tables 4.44 and 4.45 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. The proposed scene model is outperformed by ViBe based on both PCC and PrCC, where minor performance declines are observed at 3.80% and 1.56%, respectively. These results are based on minor reductions in the true positive category and large increases false positives.

Table 4.28: Beach Frame 50 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	554	617	11.37%
TN	58,455	61,786	5.70%
FP	4,597	1,266	-72.46%
FN	394	331	-15.99%
PCC	92.20%	97.50%	5.75%

Table 4.29: Beach Frame 50 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	642.80	643.64	0.13%
TN_{prob}	53,817.71	56,767.60	5.48%
FP_{prob}	8,988.00	6,038.11	-32.82%
FN_{prob}	301.50	300.66	-0.28%
PrCC	85.43%	90.06%	5.42%

Table 4.30: Beach Frame 75 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	766	829	8.22%
TN	57,357	60,664	5.77%
FP	5,416	2,109	-61.06%
FN	461	398	-13.67%
PCC	90.82%	96.08%	5.80%

Table 4.31: Beach Frame 75 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	867.62	859.86	-0.89%
TN_{prob}	52,604.25	55,493.14	5.49%
FP_{prob}	9,923.55	7,034.65	-29.11%
FN_{prob}	354.59	362.34	2.19%
PrCC	83.88%	88.40%	5.39%

Table 4.32: Beach Frame 100 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	332	357	7.53%
TN	59,234	61,793	4.32%
FP	4,272	1,713	-59.90%
FN	162	137	-15.43%
PCC	93.07%	97.11%	4.34%

Table 4.33: Beach Frame 100 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	381.34	362.93	-4.83%
TN_{prob}	53,882.94	56,488.02	4.83%
FP_{prob}	9,374.99	6,769.91	-27.79%
FN_{prob}	110.73	129.14	16.62%
PrCC	85.12%	89.18%	4.77%

Table 4.34: Beach Frame 125 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	429	478	11.42%
TN	58,634	59,032	0.68%
FP	4,763	4,365	-8.36%
FN	174	125	-28.16%
PCC	92.29%	92.98%	0.76%

Table 4.35: Beach Frame 125 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	470.14	476.73	1.40%
TN_{prob}	53,668.40	54,322.77	1.22%
FP_{prob}	9,480.95	8,826.58	-6.90%
FN_{prob}	130.50	123.92	-5.04%
PrCC	84.92%	85.96%	1.22%

Table 4.36: Beach Frame 150 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	748	802	7.22%
TN	59,003	60,478	2.50%
FP	3,899	2,424	-37.83%
FN	350	296	-15.43%
PCC	93.36%	95.75%	2.56%

Table 4.37: Beach Frame 150 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	812.77	811.36	-0.17%
TN_{prob}	53,554.54	54,994.36	2.69%
FP_{prob}	9,101.75	7,661.93	-15.82%
FN_{prob}	280.94	282.35	0.50%
PrCC	85.28%	87.54%	2.65%

Table 4.38: Beach Frame 175 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	281	306	8.90%
TN	59,988	60,385	0.66%
FP	3,614	3,217	-10.99%
FN	117	92	-21.37%
PCC	94.17%	94.83%	0.70%

Table 4.39: Beach Frame 175 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	308.77	308.14	-0.20%
TN_{prob}	54,447.60	55,031.57	1.07%
FP_{prob}	8,905.96	8,321.98	-6.56%
FN_{prob}	87.68	88.30	0.72%
PrCC	85.89%	86.81%	1.07%

Table 4.40: Beach Frame 200 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	219	236	7.76%
TN	60,463	59,578	-1.46%
FP	3,091	3,976	28.63%
FN	227	210	-7.49%
PCC	94.82%	93.46%	-1.43%

Table 4.41: Beach Frame 200 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	257.42	246.56	-4.22%
TN_{prob}	54,677.52	54,569.43	-0.20%
FP_{prob}	8,628.22	8,736.32	1.25%
FN_{prob}	186.84	197.70	5.81%
PrCC	86.17%	85.99%	-0.22%

Table 4.42: Beach Frame 225 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	569	590	3.69%
TN	60,409	59,680	-1.21%
FP	2,607	3,336	27.96%
FN	415	394	-5.06%
PCC	95.28%	94.17%	-1.16%

Table 4.43: Beach Frame 225 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	656.74	629.99	-4.07%
TN_{prob}	54,548.45	54,373.34	-0.32%
FP_{prob}	8,221.39	8,396.51	2.13%
FN_{prob}	323.41	350.17	8.27%
PrCC	86.60%	86.28%	-0.37%

Table 4.44: Beach Frame 250 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	423	419	-0.95%
TN	60,198	57,899	-3.82%
FP	3,262	5,561	70.48%
FN	117	121	3.42%
PCC	94.72%	91.12%	-3.80%

Table 4.45: Beach Frame 250 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	435.51	423.89	-2.67%
TN_{prob}	54,291.90	53,451.92	-1.55%
FP_{prob}	8,920.21	9,760.19	9.42%
FN_{prob}	102.38	114.00	11.35%
PrCC	85.85%	84.51%	-1.56%

4.2.3 PETS 1 Sequence

The PETS 1 video sequence features gradual illumination changes, dynamic background components, camouflaged foreground objects, shadows, ghosts, and the foreground aperture condition. I processed 225 frames of the video, which is a subset of the original 2688 frames, and selected nine frames for presentation including the first and last frames and each 25th frame beginning with frame number 426. In this sequence, a person walking from left to right crosses paths with a small car traveling from right to left. The person begins motion from the center of the frame leading to the ghost problem, while the vehicle enters the frame post initialization. The PETS 1 sequence depicts an outdoor scene containing gradual illumination changes in all frames and only minor dynamic background components in the way of moving vegetation in the distant background.

In the first frame of the PETS 1 (Fig. 4.22) sequence a person can be seen in front of a row of parked cars walking from left to right along the primary roadway. Tables 4.46 and 4.47 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In terms of PCC the proposed model achieves a 192.59% increase in the number of true positives, a 96.83% reduction in false positives and a 40.08% decrease in false negatives. By PrCC we observe an 18.67% increase in true positives, a 95.50% reduction in false positives and a 22.69% decrease in false negatives. Overall, the proposed technique outperforms ViBe by 0.32% PCC and 7.41% PrCC.

Frame 450 (Fig. 4.23) continues to depict a single person walking towards the right side of the frame. Tables 4.48 and 4.49 summarize the performance differences measured between ViBe and the proposed algorithm in

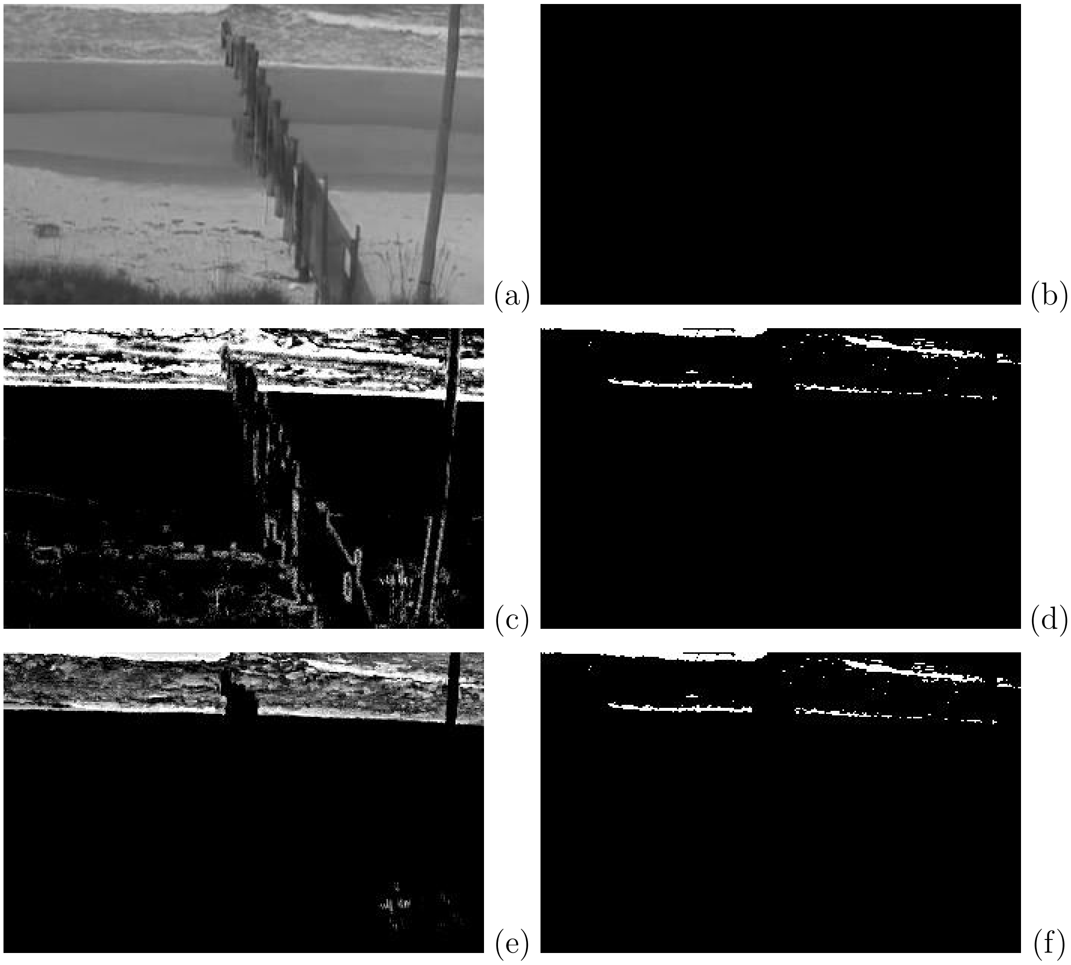

Figure 4.12: Frame 27 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=85.52%)(c), ViBe final segmentation (PCC=98.06%)(d), the foreground probability image for the proposed algorithm (PrCC=91.04%)(e), and the final segmentation for the proposed algorithm (PCC=98.06%)(f).

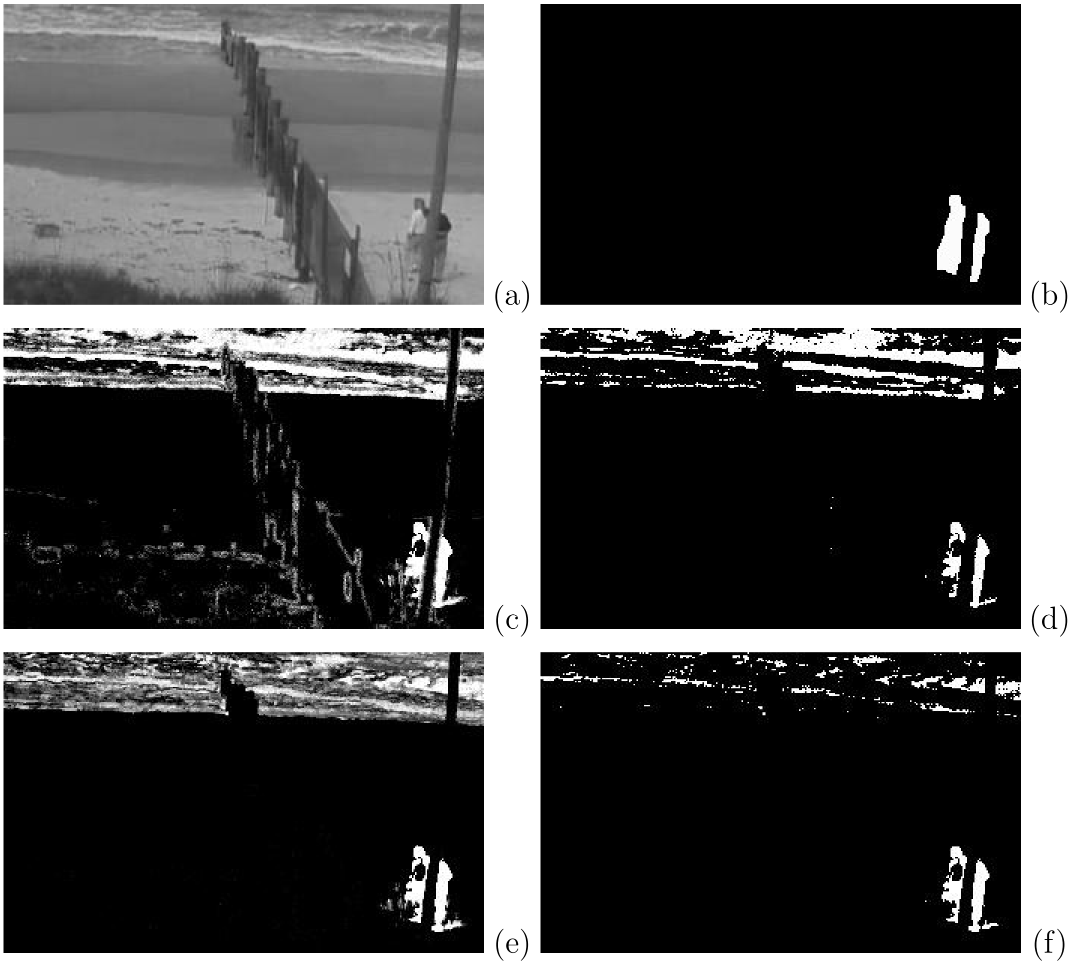

Figure 4.13: Frame 50 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=85.43%)(c), ViBe final segmentation (PCC=92.20%)(d), the foreground probability image for the proposed algorithm (PrCC=89.57%)(e), and the final segmentation for the proposed algorithm (PCC=97.50%)(f).

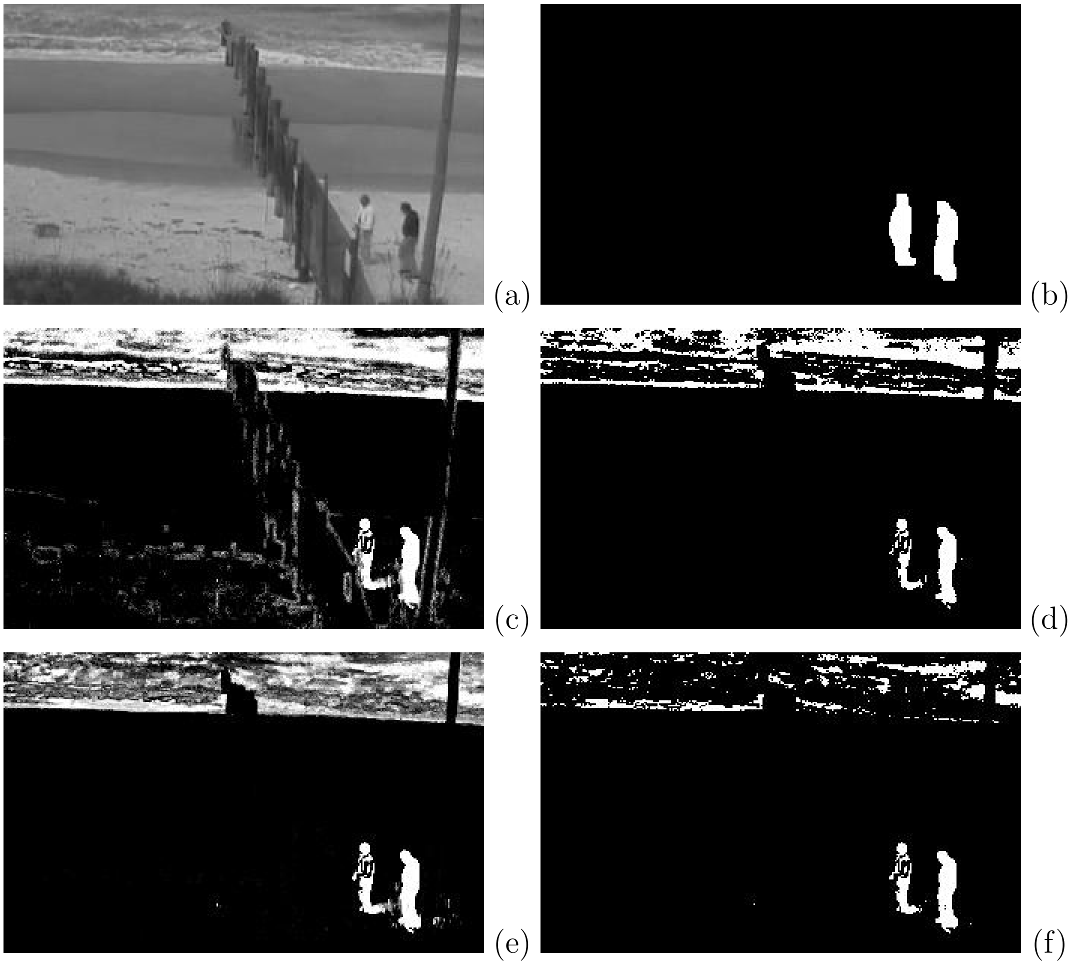

Figure 4.14: Frame 75 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=83.88%)(c), ViBe final segmentation (PCC=90.82%)(d), the foreground probability image for the proposed algorithm (PrCC=87.86%)(e), and the final segmentation for the proposed algorithm (PCC=96.10%)(f).

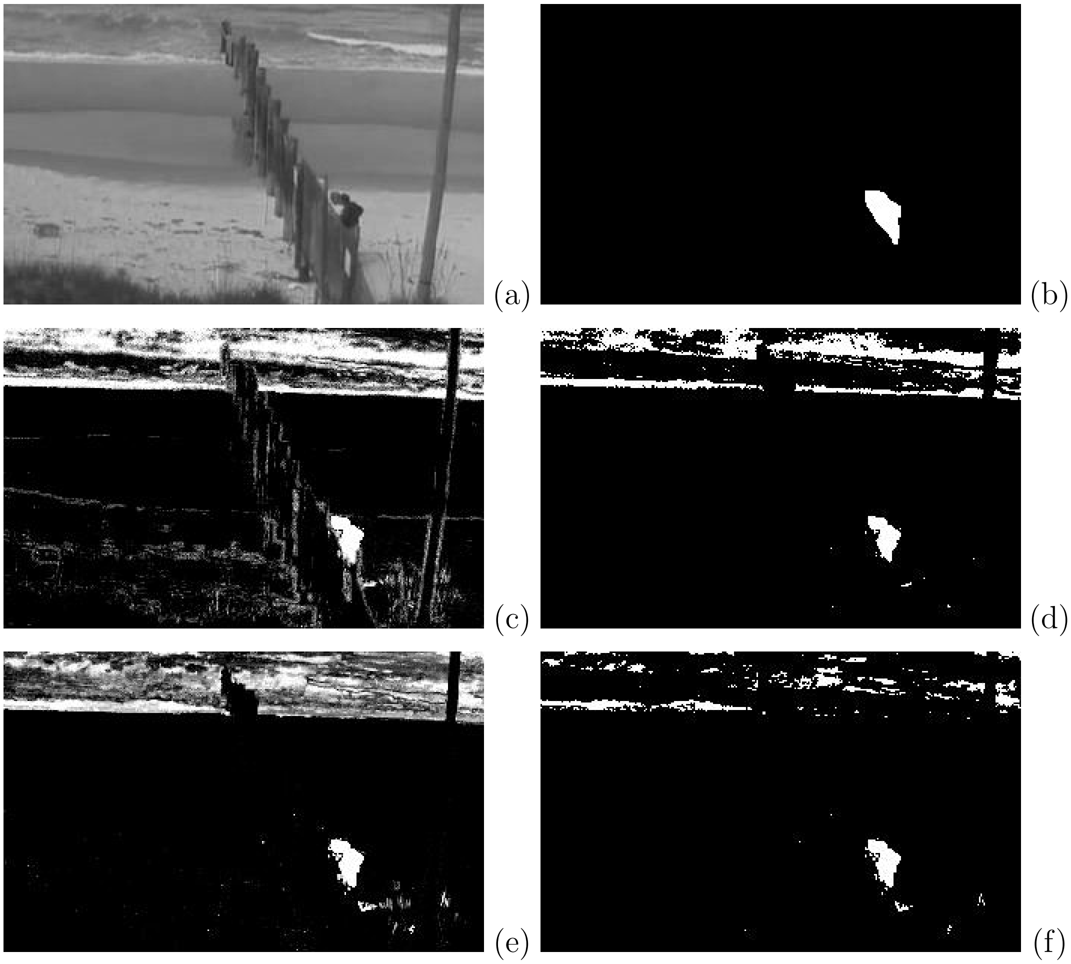

Figure 4.15: Frame 100 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=85.12%)(c), ViBe final segmentation (PCC=93.07%)(d), the foreground probability image for the proposed algorithm (PrCC=88.74%)(e), and the final segmentation for the proposed algorithm (PCC=97.11%)(f).

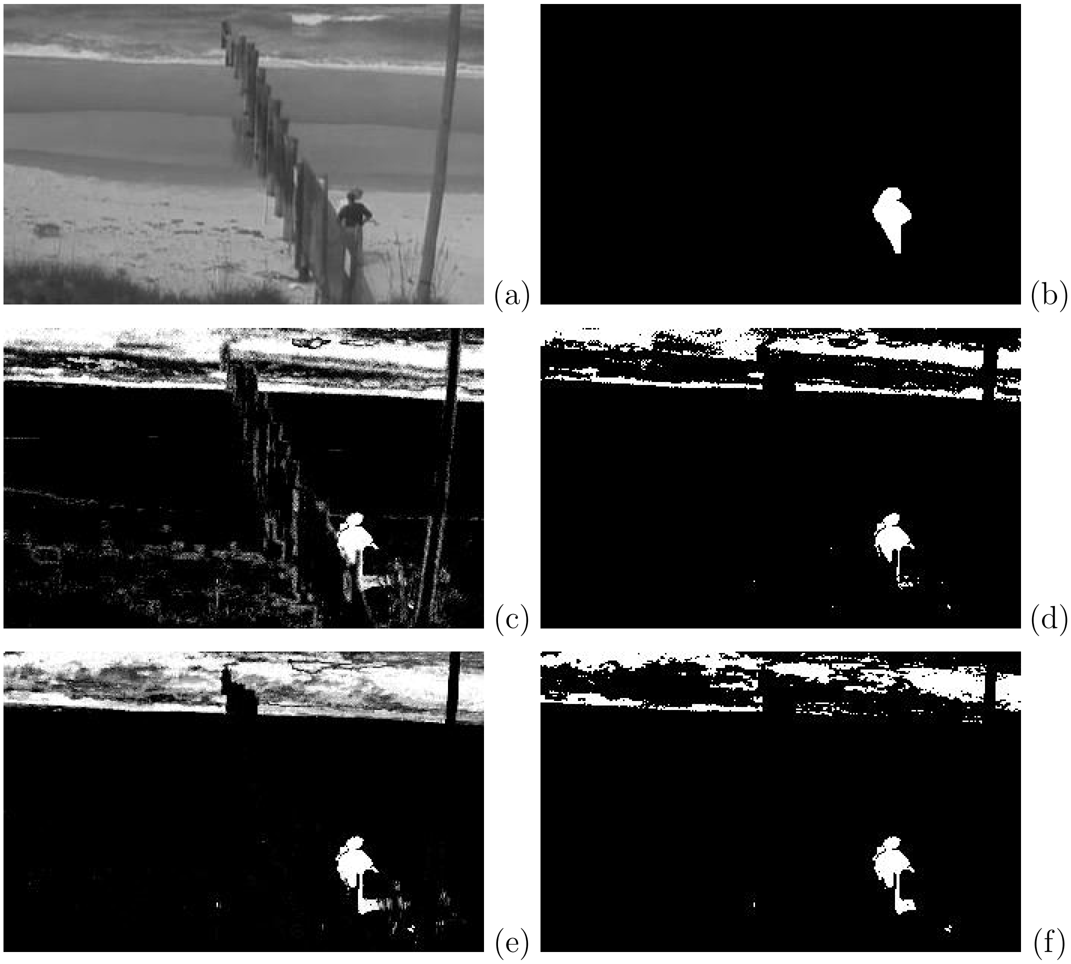

Figure 4.16: Frame 125 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=84.92%)(c), ViBe final segmentation (PCC=92.29%)(d), the foreground probability image for the proposed algorithm (PrCC=85.81%)(e), and the final segmentation for the proposed algorithm (PCC=92.98%)(f).

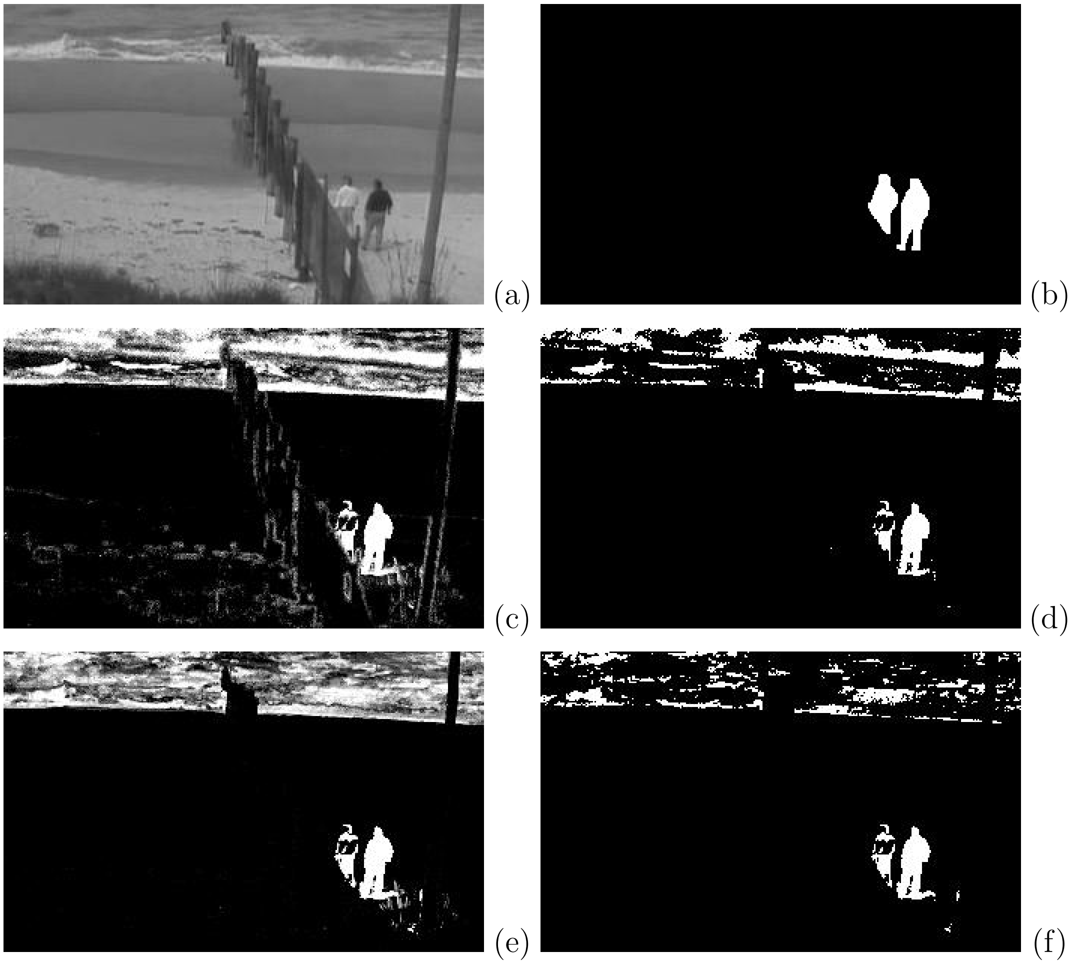

Figure 4.17: Frame 150 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=85.28%)(c), ViBe final segmentation (PCC=93.36%)(d), the foreground probability image for the proposed algorithm (PrCC=87.25%)(e), and the final segmentation for the proposed algorithm (PCC=95.75%)(f).

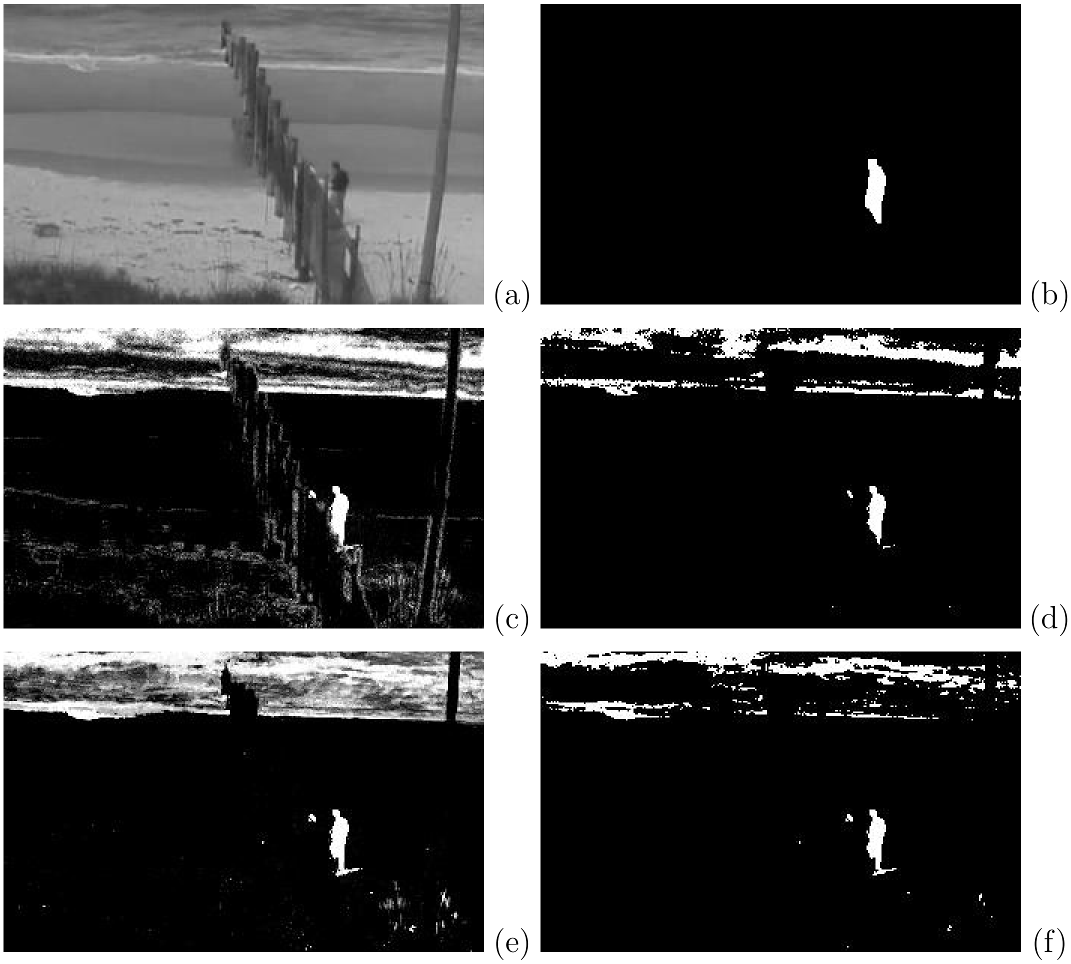

Figure 4.18: Frame 175 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=85.89%)(c), ViBe final segmentation (PCC=94.17%)(d), the foreground probability image for the proposed algorithm (PrCC=86.69%)(e), and the final segmentation for the proposed algorithm (PCC=94.83%)(f).

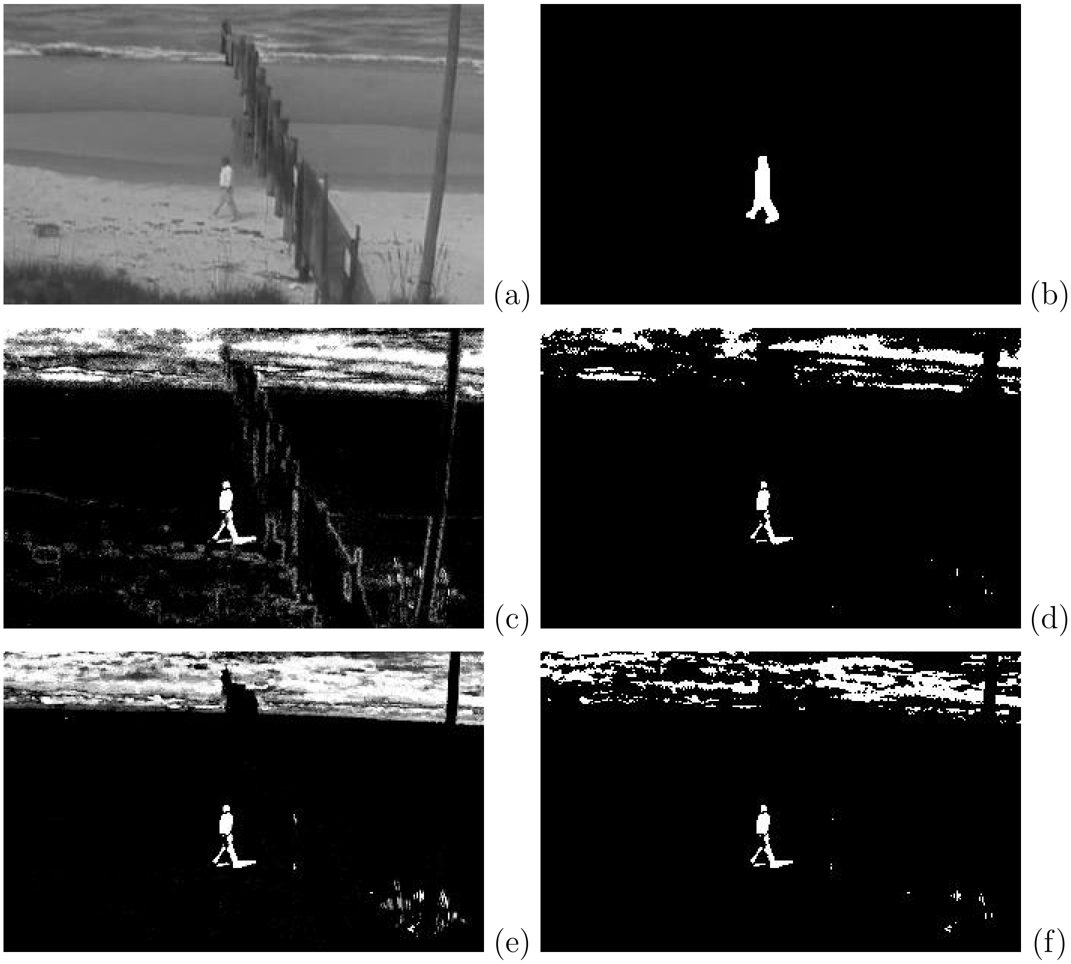

Figure 4.19: Frame 200 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=86.17%)(c), ViBe final segmentation (PCC=94.82%)(d), the foreground probability image for the proposed algorithm (PrCC=86.02%)(e), and the final segmentation for the proposed algorithm (PCC=93.46%)(f).

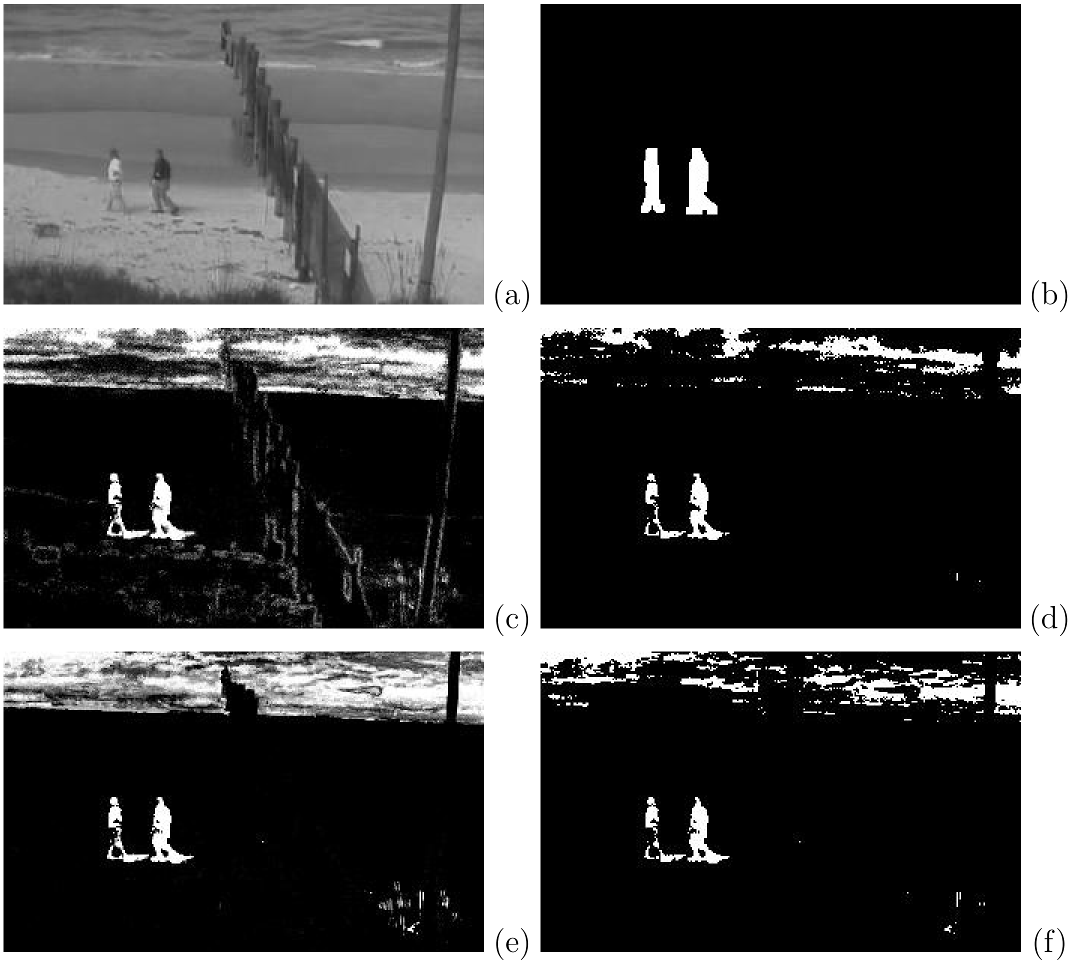

Figure 4.20: Frame 225 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=86.60%)(c), ViBe final segmentation (PCC=95.28%)(d), the foreground probability image for the proposed algorithm (PrCC=86.35%)(e), and the final segmentation for the proposed algorithm (PCC=94.17%)(f).

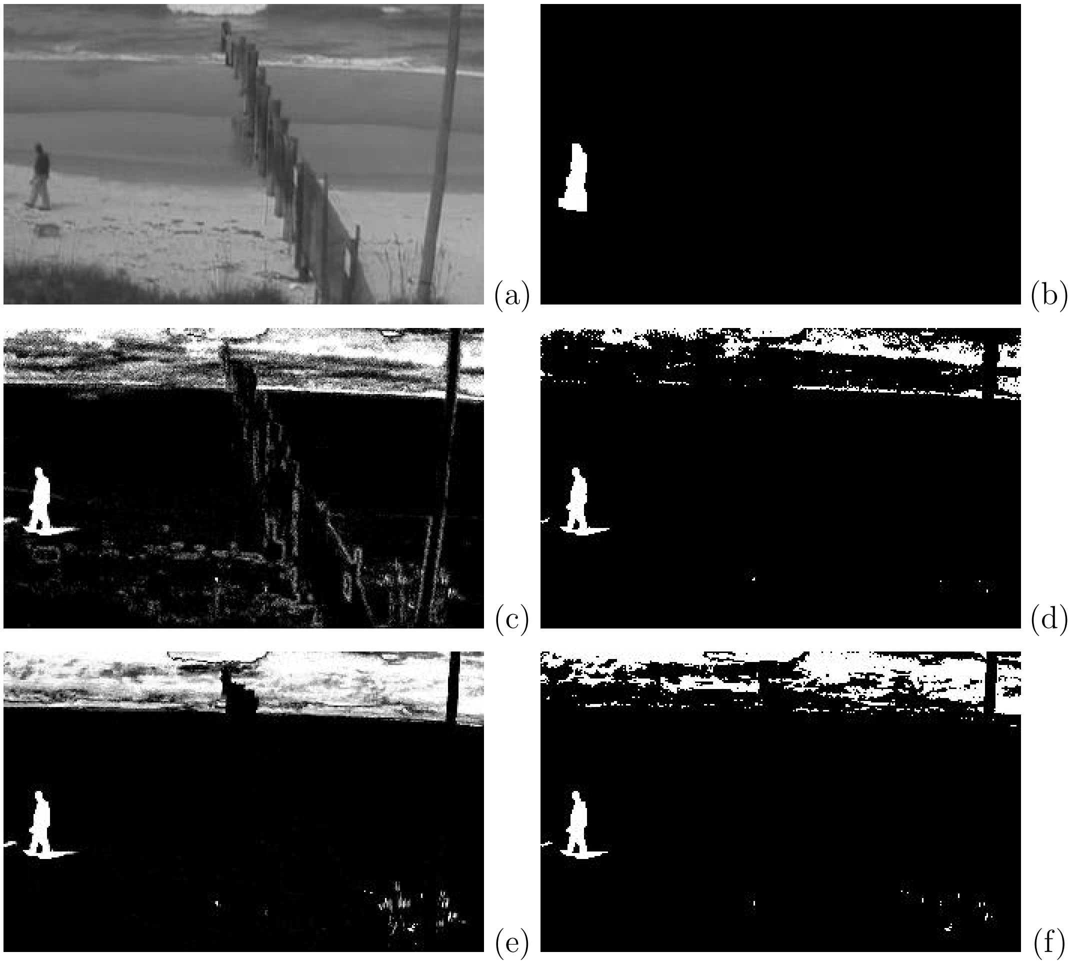

Figure 4.21: Frame 250 of the Beach sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=85.85%)(c), ViBe final segmentation (PCC=94.72%)(d), the foreground probability image for the proposed algorithm (PrCC=84.73%)(e), and the final segmentation for the proposed algorithm (PCC=91.12%)(f).

terms of PCC and PrCC. By PCC we observe a 6.85% increase in true positives amnd a 78.02% decrease in false positives, resulting in an overall performance improvement of 0.27% over the ViBe system. According to PrCC the proposed algorithm improves on ViBe by 6.98% based on a 93.83% reduction in the potential to classify false positives.

In Fig. 4.24 (frame 475) a moving car traveling from right to left enters the frame in the lower right corner. In this frame only the front of the vehicle has entered the frame. Tables 4.50 and 4.51 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In terms of PCC and PrCC the proposed scene model improves on the performance of ViBe by 0.21% and 6.59% respectively. These measurements are based on PCC and PrCC false positives reductions of 42.09% and 91.21%.

In frame 500 (Fig. 4.25) the moving car has become completely visible and the the person and the car appear to be heading towards each other Tables 4.52 and 4.53 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Yet again, we observe significant reductions in false positives in terms of both PCC and PrCC measurements, resulting in overall performance improvements of 0.17% and 6.26%. By PCC false positives were reduced by 23.62% and by PrCC they were reduced by 89.12%. In addition, we observe an increase in PrCC true negatives of 6.41% and an increase in PCC true positives of 2.18%.

In frame 525 (Fig. 4.26) the car and the person continue to approach one another near the middle right region of the video frame. Tables 4.54 and 4.55 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. According to PCC the proposed algorithm increases the number of true positives by 1.82% and decreases the number of false positives by 18.13%. PrCC measurements indicate a minor decrease in true positives of 0.19% and a larger reduction in false positives of 89.07%. Overall, the PCC and PrCC metrics indicate improvements of the ViBe algorithm of 0.12% and 5.96%.

Finally, the car and the person begin to cross paths in frame 550 (Fig. 4.27) with the car positioned between the camera and the person. Tables 4.56 and 4.57 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Both PCC and PrCC metrics indicate significant reductions in false positives of 13.33% and 89.59% when comparing the proposed model to ViBe. In addition, we observe a PCC true positive increase of 1.60% and a PrCC true negative increase of 5.84%. Overall evaluation by PCC and PrCC indicate performance gains of 0.08% and 5.74%, respectively.

In Fig. 4.28 (frame 575) the vehicle is occluding the lower unit of the person as they continue to pass each other traveling in opposite directions. Tables 4.58 and 4.59 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Based on PCC the proposed technique achieves a 16.77% reduction in false positives and a 7.76% reduction in false negatives, resulting in an overall performance improvement over ViBe of 0.07%. In terms of PrCC, we observe an overall improvement of 5.56% that is largely based on a 90.85% reduction in potential false positives.

In frame 600 (Fig. 4.29) the car and the person have completed their observable interaction and both the person and the vehicle appear unobscured by one another. Tables 4.60 and 4.61 summarize the performance differences mea-

Table 4.46: PETS 1 Frame 426 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	108	316	192.59%
TN	440,510	441,702	0.27%
FP	1,231	39	-96.83%
FN	519	311	-40.08%
PCC	99.60%	99.92%	0.32%

Table 4.47: PETS 1 Frame 426 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	342.62	406.59	18.67%
TN_{prob}	408,359.66	438,592.15	7.40%
FP_{prob}	31,655.79	1,423.30	-95.50%
FN_{prob}	281.93	217.96	-22.69%
PrCC	92.75%	99.63%	7.41%

sured between ViBe and the proposed algorithm in terms of PCC and PrCC. By PCC the proposed algorithm improves on ViBe only slightly with a 0.06% increase in PCC based on a 12.79% reduction in false negatives. According to PrCC the proposed technique improves on ViBe by 5.31% due to a large reduction in potential false positives of 90.89%.

In frame 625 (Fig. 4.30) the person continues to travel from left to right nearing the rightmost edge of the frame and the vehicle appears to begin a parking maneuver. Tables 4.62 and 4.63 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Similar to frame 600, we observe a minor improvement of 0.08% PCC and a larger improvement of 5.11% PrCC. The differences in overall performance are largely based on the differences in false positives measured by the two metrics, where PCC false positives are reduced by 17.15% and PrCC false positives are reduced by 91.17%.

Table 4.48: PETS 1 Frame 450 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	686	733	6.85%
TN	439,894	441,019	0.26%
FP	1,442	317	-78.02%
FN	346	299	-13.58%
PCC	99.60%	99.86%	0.27%

Table 4.49: PETS 1 Frame 450 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	758.29	760.00	0.23%
TN_{prob}	409,129.41	437,731.95	6.99%
FP_{prob}	30,482.62	1,880.09	-93.83%
FN_{prob}	269.68	267.97	-0.63%
PrCC	93.02%	99.51%	6.98%

Table 4.50: PETS 1 Frame 475 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	6,175	6,297	1.98%
TN	432,619	433,431	0.19%
FP	1,929	1,117	-42.09%
FN	1,645	1,523	-7.42%
PCC	99.19%	99.40%	0.21%

Table 4.51: PETS 1 Frame 475 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	6,355.30	6,332.81	-0.35%
TN_{prob}	403,246.66	430,247.68	6.70%
FP_{prob}	29,603.89	2,602.87	-91.21%
FN_{prob}	1,434.15	1,456.64	1.57%
PrCC	92.96%	99.08%	6.59%

Table 4.52: PETS 1 Frame 500 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	8,991	9,187	2.18%
TN	429,266	429,821	0.13%
FP	2,350	1,795	-23.62%
FN	1,761	1,565	-11.13%
PCC	99.07%	99.24%	0.17%

Table 4.53: PETS 1 Frame 500 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	9,197.63	9,179.75	-0.19%
TN_{prob}	401,072.88	426,791.14	6.41%
FP_{prob}	28,857.13	3,138.86	-89.12%
FN_{prob}	1,512.37	1,530.25	1.18%
PrCC	93.11%	98.94%	6.26%

Table 4.54: PETS 1 Frame 525 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	7,732	7,873	1.82%
TN	430,973	431,349	0.09%
FP	2,074	1,698	-18.13%
FN	1,589	1,448	-8.87%
PCC	99.17%	99.29%	0.12%

Table 4.55: PETS 1 Frame 525 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	7,900.06	7,885.34	-0.19%
TN_{prob}	403,798.27	428,342.11	6.08%
FP_{prob}	27,557.14	3,013.30	-89.07%
FN_{prob}	1,384.53	1,399.25	1.06%
PrCC	93.43%	99.00%	5.96%

Table 4.56: PETS 1 Frame 550 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	7,056	7,169	1.60%
TN	432,129	432,362	0.05%
FP	1,748	1,515	-13.33%
FN	1,435	1,322	-7.87%
PCC	99.28%	99.36%	0.08%

Table 4.57: PETS 1 Frame 550 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	7,156.91	7,156.91	0.00%
TN_{prob}	405,719.74	429,428.51	5.84%
FP_{prob}	26,462.43	2,753.66	-89.59%
FN_{prob}	1,300.92	1,300.93	0.00%
PrCC	93.70%	99.08%	5.74%

Table 4.58: PETS 1 Frame 575 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	5,811	5,892	1.39%
TN	434,116	434,341	0.05%
FP	1,397	1,172	-16.11%
FN	1,044	963	-7.76%
PCC	99.45%	99.52%	0.07%

Table 4.59: PETS 1 Frame 575 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	5,857.17	5,862.33	0.09%
TN_{prob}	408,452.51	431,491.02	5.64%
FP_{prob}	25,359.27	2,320.75	-90.85%
FN_{prob}	971.05	965.89	-0.53%
PrCC	94.02%	99.25%	5.56%

Table 4.60: PETS 1 Frame 600 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	4,487	4,666	3.99%
TN	435,344	435,440	0.02%
FP	1,138	1,042	-8.44%
FN	1,399	1,220	-12.79%
PCC	99.43%	99.49%	0.06%

Table 4.61: PETS 1 Frame 600 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	4,709.38	4,677.10	-0.69%
TN_{prob}	410,497.10	432,565.62	5.38%
FP_{prob}	24,279.89	2,211.38	-90.89%
FN_{prob}	1,153.63	1,185.91	2.80%
PrCC	94.23%	99.23%	5.31%

Table 4.62: PETS 1 Frame 625 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	3,347	3,617	8.07%
TN	436,407	436,502	0.02%
FP	1,040	945	-9.13%
FN	1,574	1,304	-17.15%
PCC	99.41%	99.49%	0.08%

Table 4.63: PETS 1 Frame 625 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	3,710.45	3,660.86	-1.34%
TN_{prob}	412,353.50	433,673.34	5.17%
FP_{prob}	23,384.73	2,064.88	-91.17%
FN_{prob}	1,191.33	1,240.91	4.16%
PrCC	94.42%	99.25%	5.11%

4.2.4 PETS 2 Sequence

The PETS 2 video sequence features gradual illumination changes, dynamic background components, camouflaged foreground objects and shadows. I processed 225 frames of the video, which is a subset of the original 5336 frames, and selected ten frames for presentation including the first and last frames and each 25th frame beginning with frame number 1056. In this sequence, two people enter the frame from the bottom and walk upwards until they are almost completely obscured by a large tree undergoing periodic motion due to a windy outdoor environment. Throughout the PETS 2 sequence, gradual illumination changes and a large swaying tree contribute to a rich dynamic background scene.

In frame 1056 (Fig. 4.31) of the PETS 2 sequence a large static background scene is visible with a swaying tree in the lower center of the frame. No foreground objects are present in this frame. Tables 4.64 and 4.65 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In terms of PCC we observe a 94.17% reduction in the number of false positives when comparing the proposed technique to ViBe. By PrCC the reduction in false positives is only 91.12%, however, a 10.72% increase in true negatives is also reported. Overall, the PCC and PrCC metrics indicate that the proposed algorithms outperforms ViBe by 0.41% and 10.75%, respectively.

In frame 1075 (Fig. 4.32) the head of a person enters the frame to the left of the central tree. The person is traveling upwards towards the parked cars. Tables 4.66 and 4.67 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Based on PCC

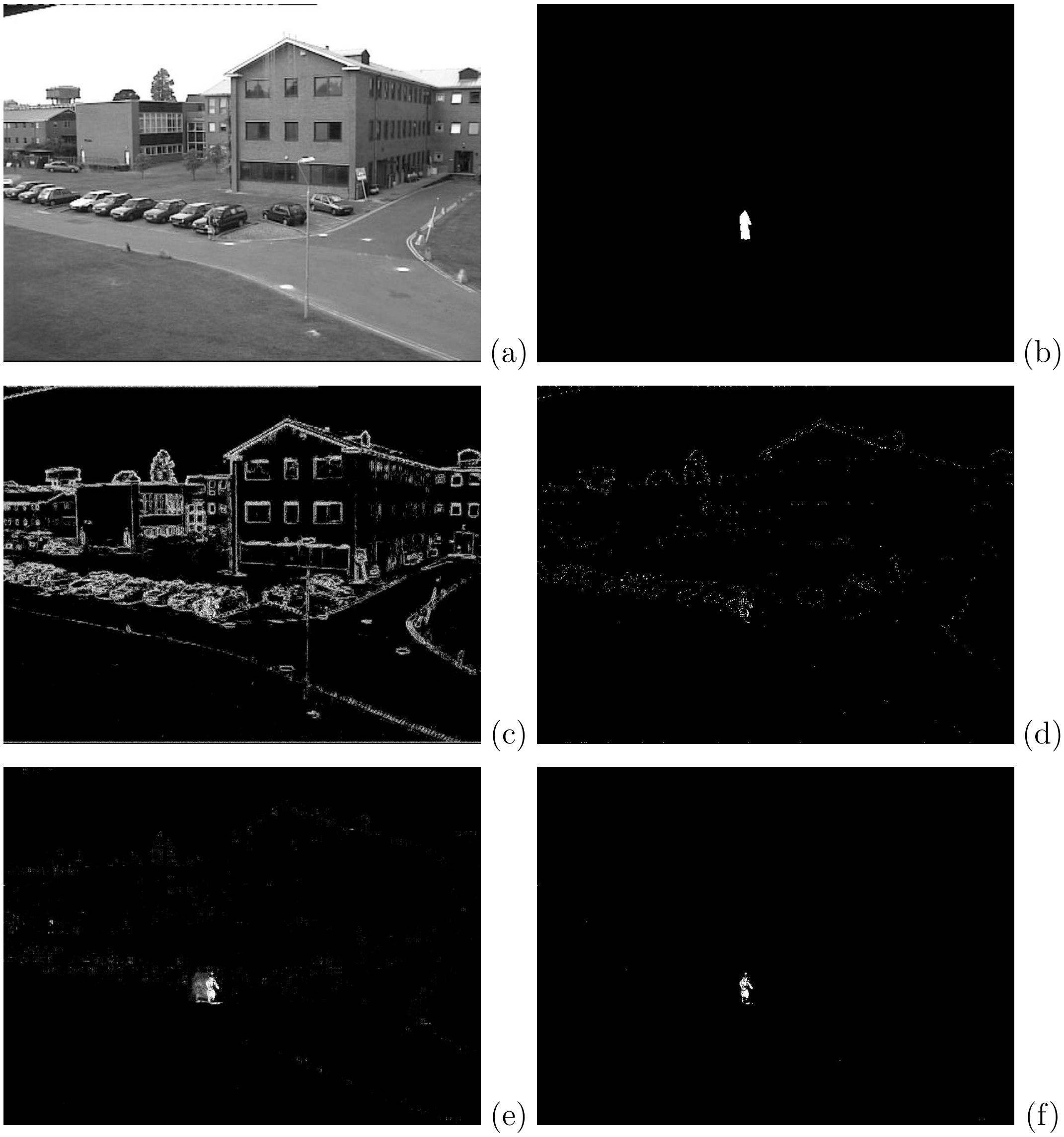

Figure 4.22: Frame 426 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=92.75%)(c), ViBe final segmentation (PCC=99.60%)(d), the foreground probability image for the proposed algorithm (PrCC=99.58%)(e), and the final segmentation for the proposed algorithm (PCC=99.92%)(f).

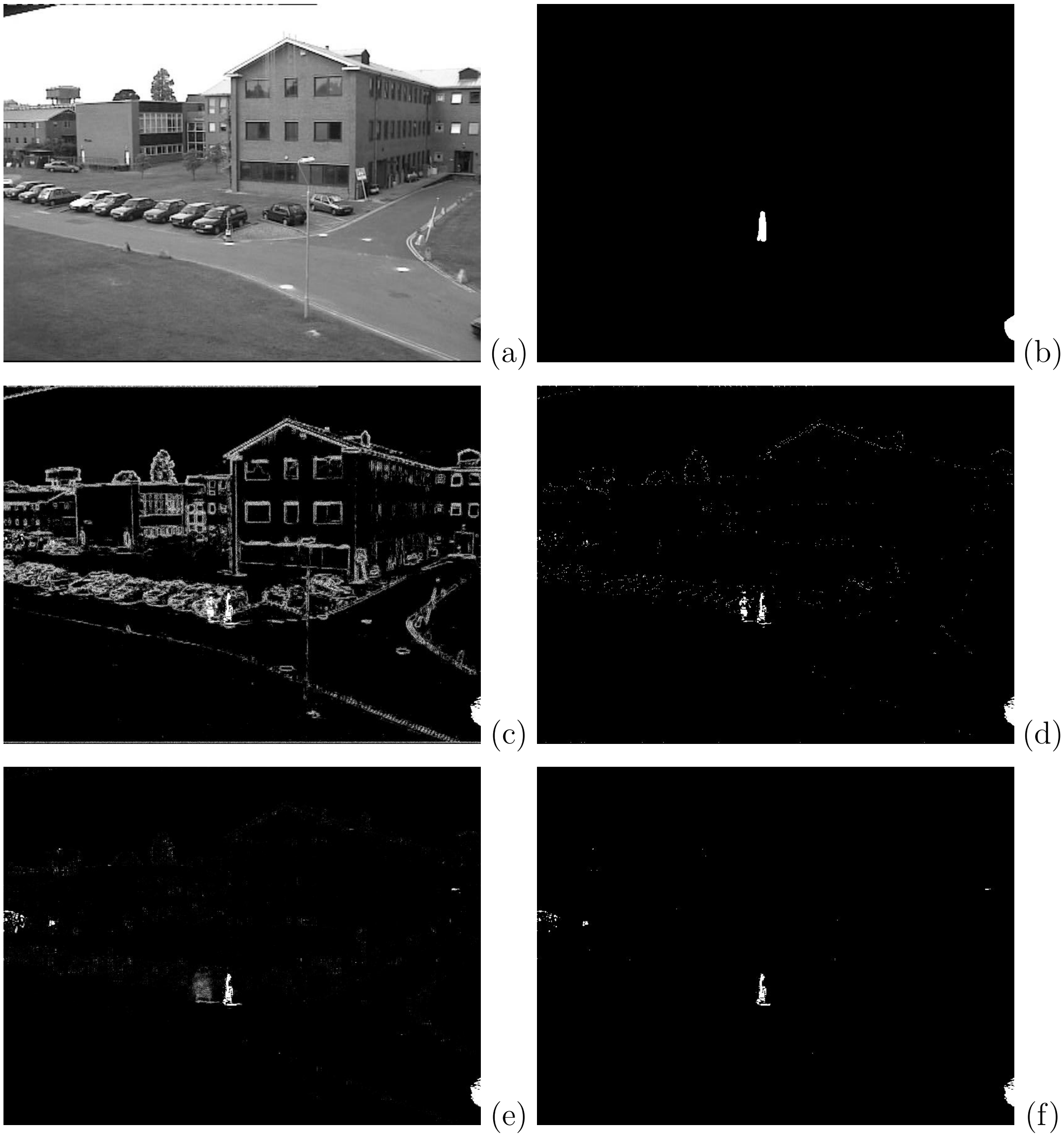

Figure 4.23: Frame 450 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=93.02%)(c), ViBe final segmentation (PCC=99.60%)(d), the foreground probability image for the proposed algorithm (PrCC=99.48%)(e), and the final segmentation for the proposed algorithm (PCC=99.86%)(f).

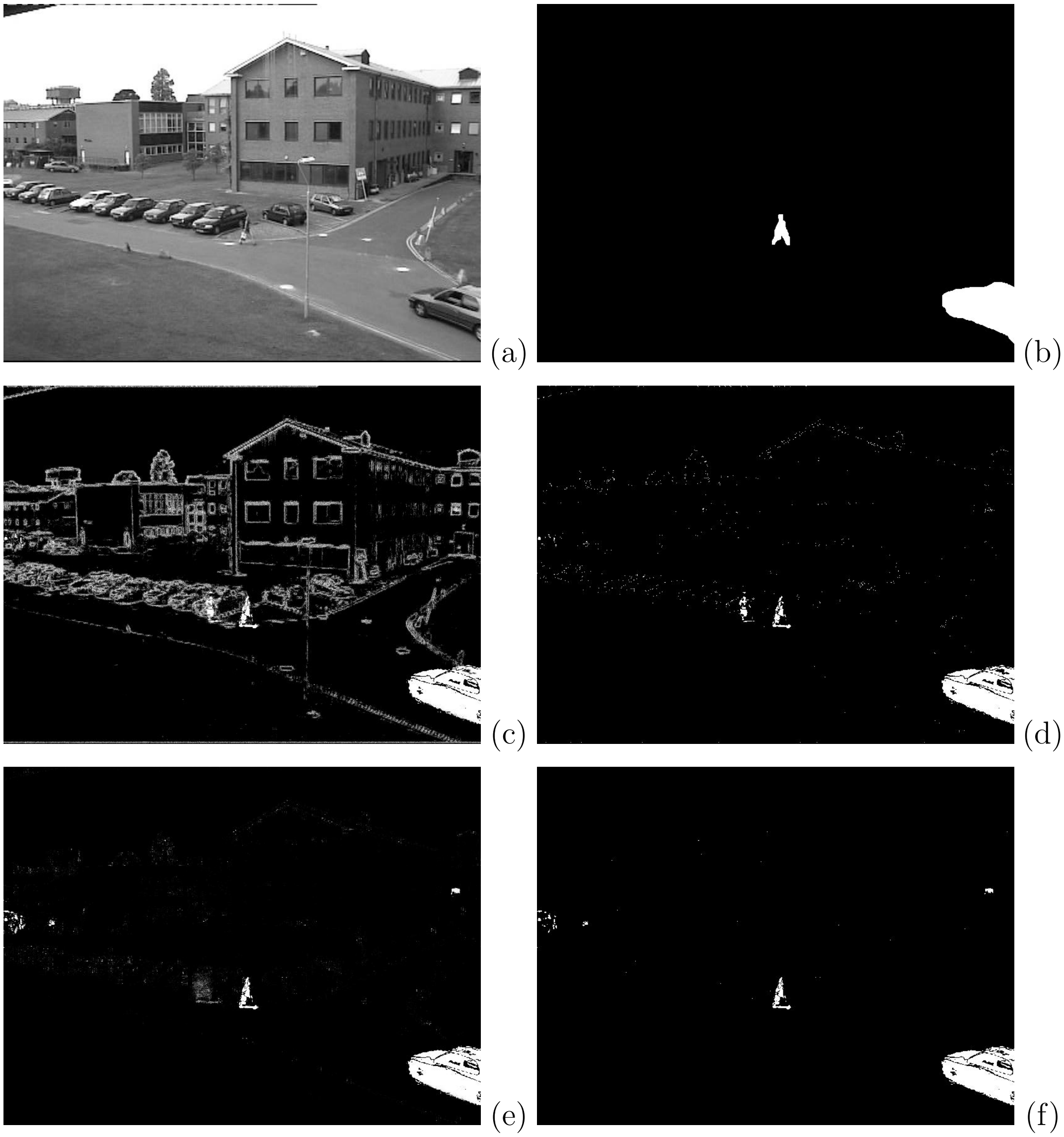

Figure 4.24: Frame 475 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=92.96%)(c), ViBe final segmentation (PCC=99.19%)(d), the foreground probability image for the proposed algorithm (PrCC=99.02%)(e), and the final segmentation for the proposed algorithm (PCC=99.40%)(f).

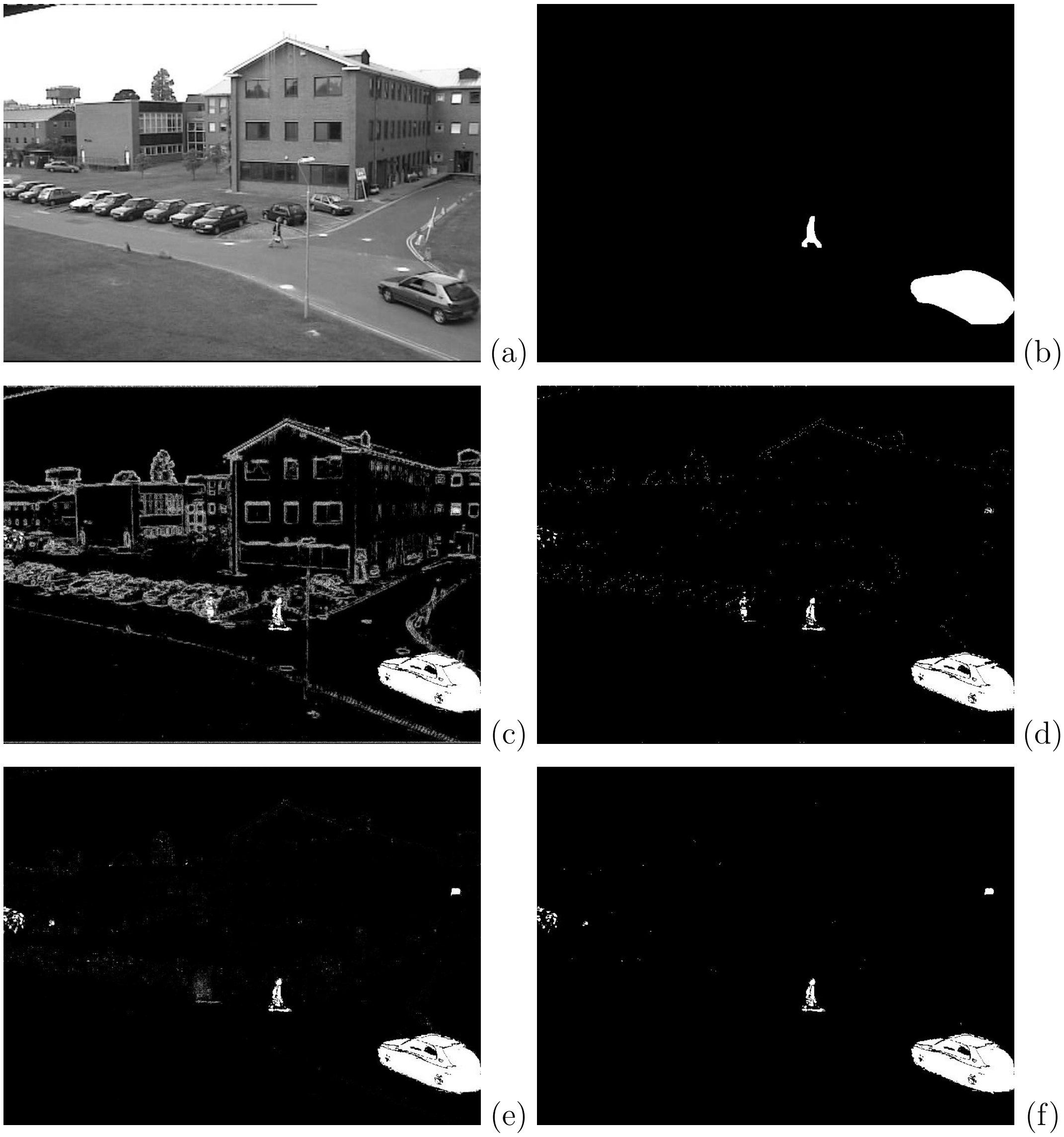

Figure 4.25: Frame 500 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=93.11%)(c), ViBe final segmentation (PCC=99.07%)(d), the foreground probability image for the proposed algorithm (PrCC=98.88%)(e), and the final segmentation for the proposed algorithm (PCC=99.24%)(f).

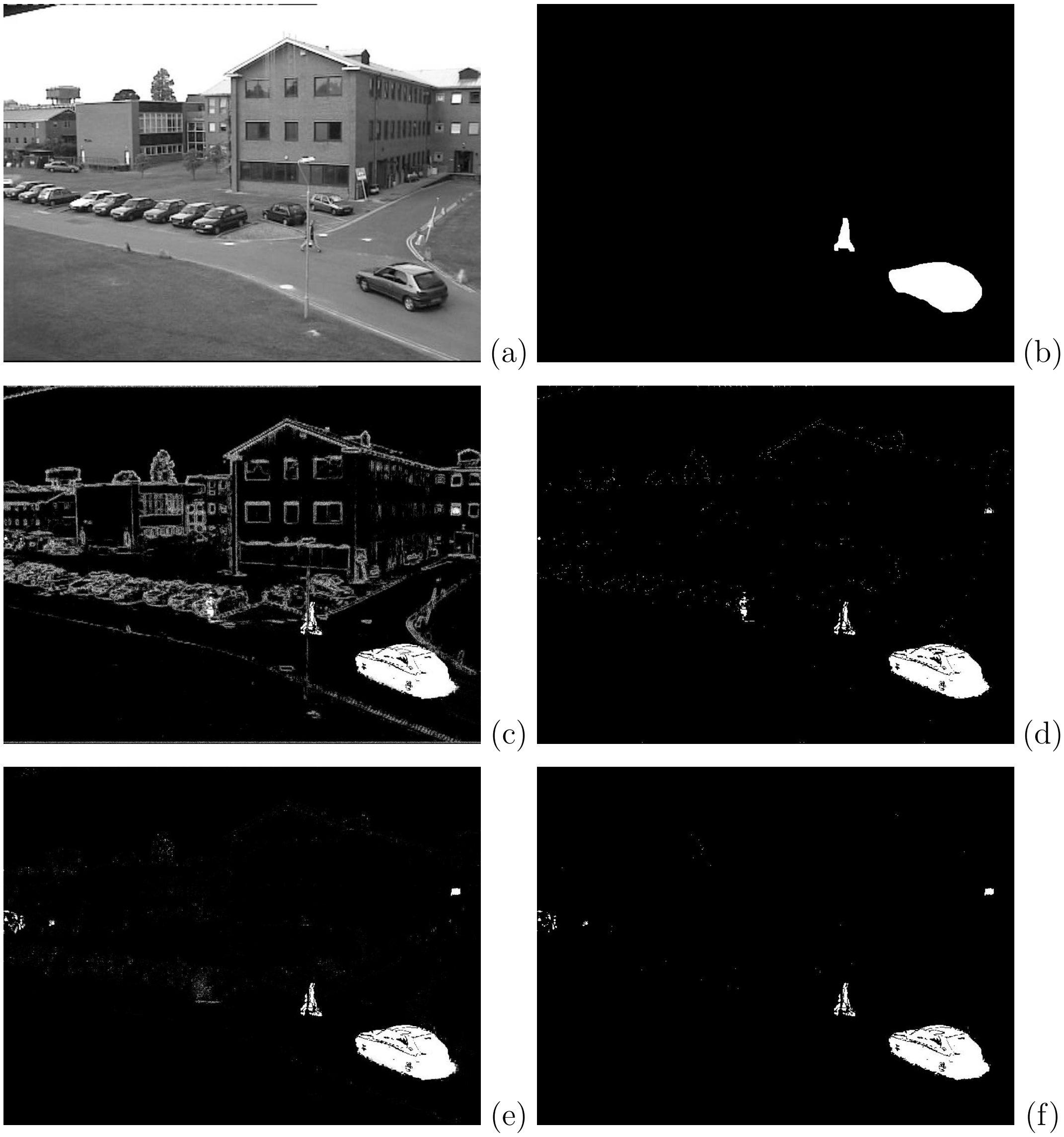

Figure 4.26: Frame 525 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=93.43%)(c), ViBe final segmentation (PCC=99.17%)(d), the foreground probability image for the proposed algorithm (PrCC=98.94%)(e), and the final segmentation for the proposed algorithm (PCC=99.29%)(f).

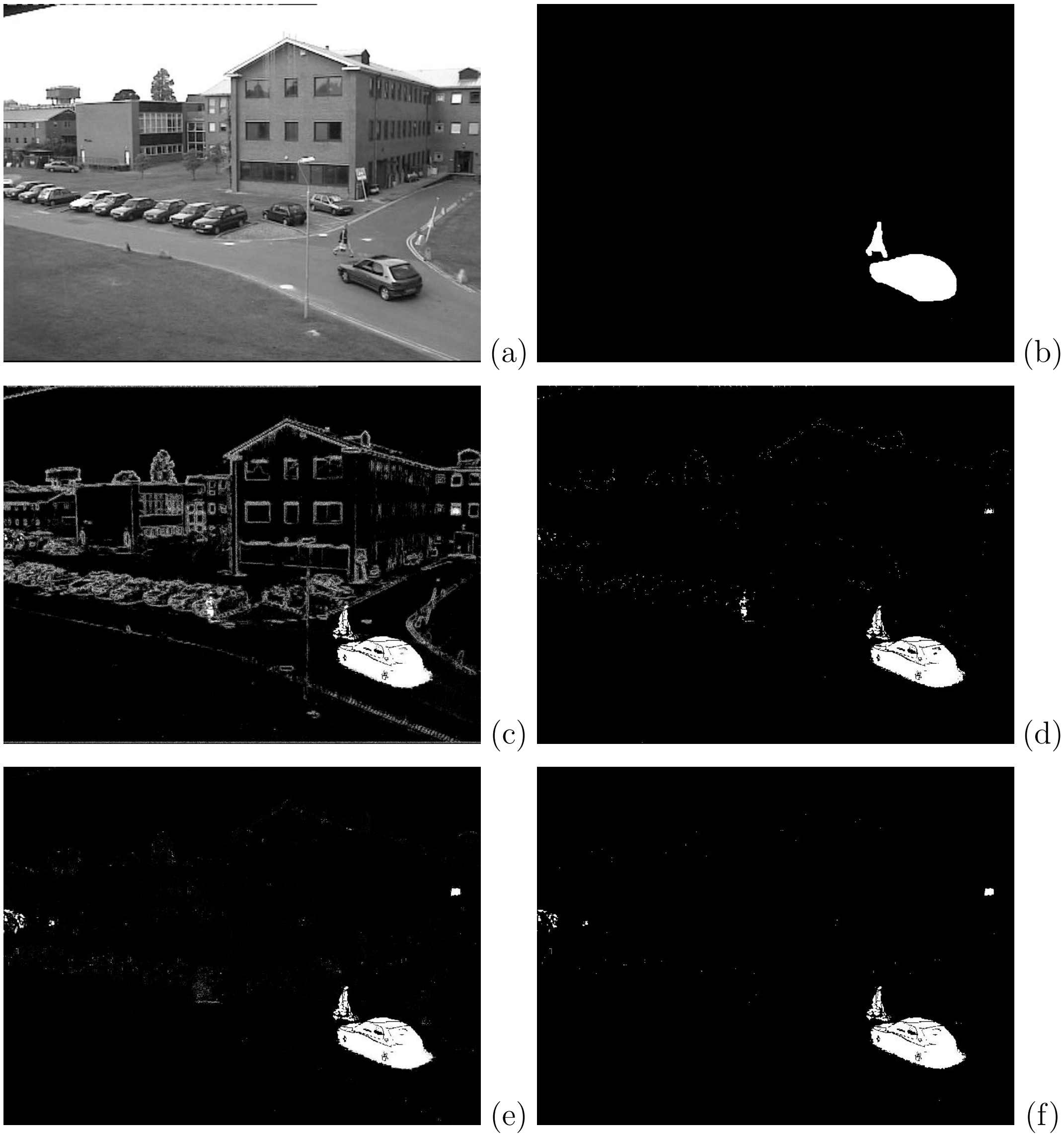

Figure 4.27: Frame 550 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=93.70%)(c), ViBe final segmentation (PCC=99.28%)(d), the foreground probability image for the proposed algorithm (PrCC=99.03%)(e), and the final segmentation for the proposed algorithm (PCC=99.36%)(f).

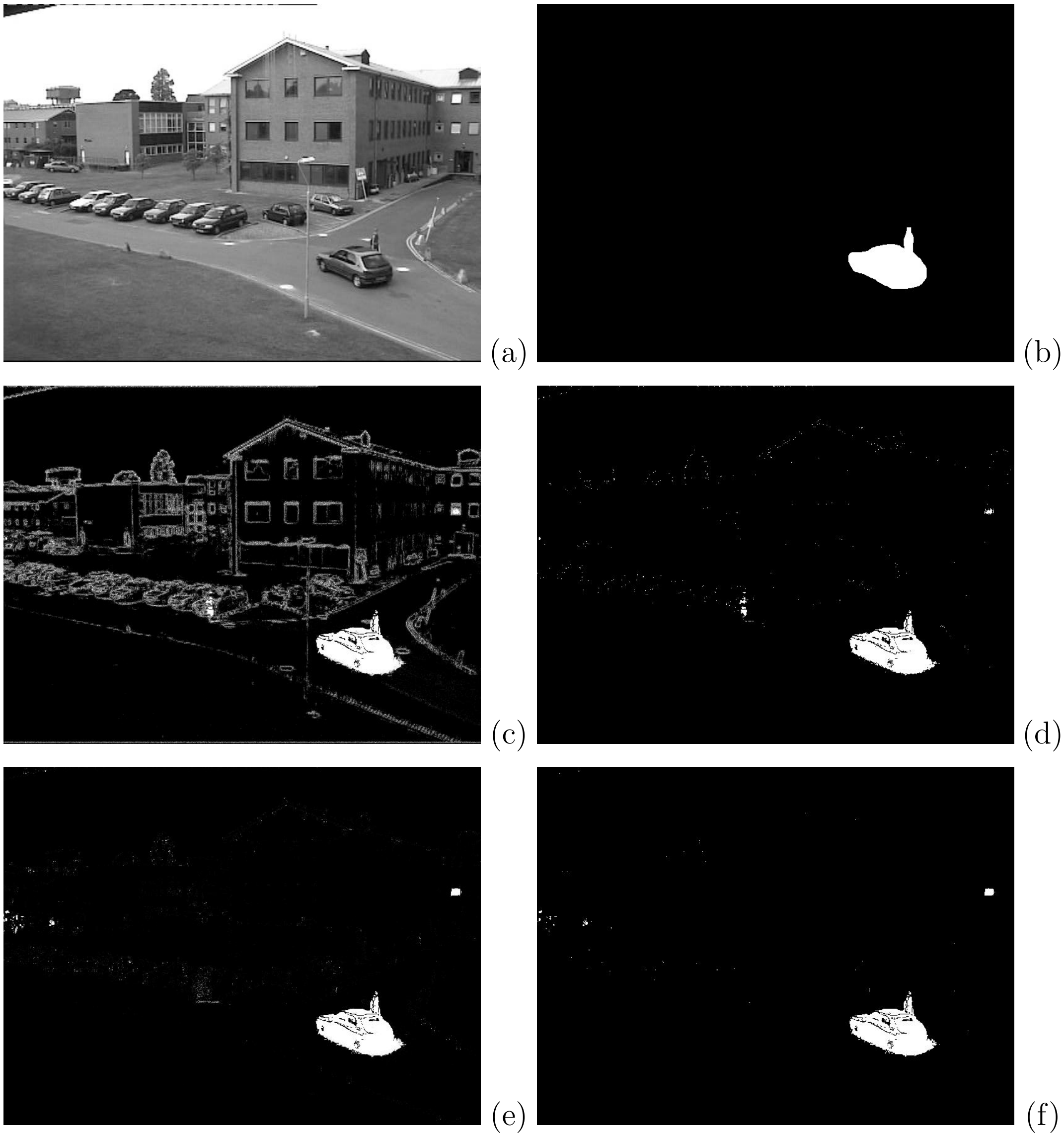

Figure 4.28: Frame 575 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=94.02%)(c), ViBe final segmentation (PCC=99.45%)(d), the foreground probability image for the proposed algorithm (PrCC=99.21%)(e), and the final segmentation for the proposed algorithm (PCC=99.52%)(f).

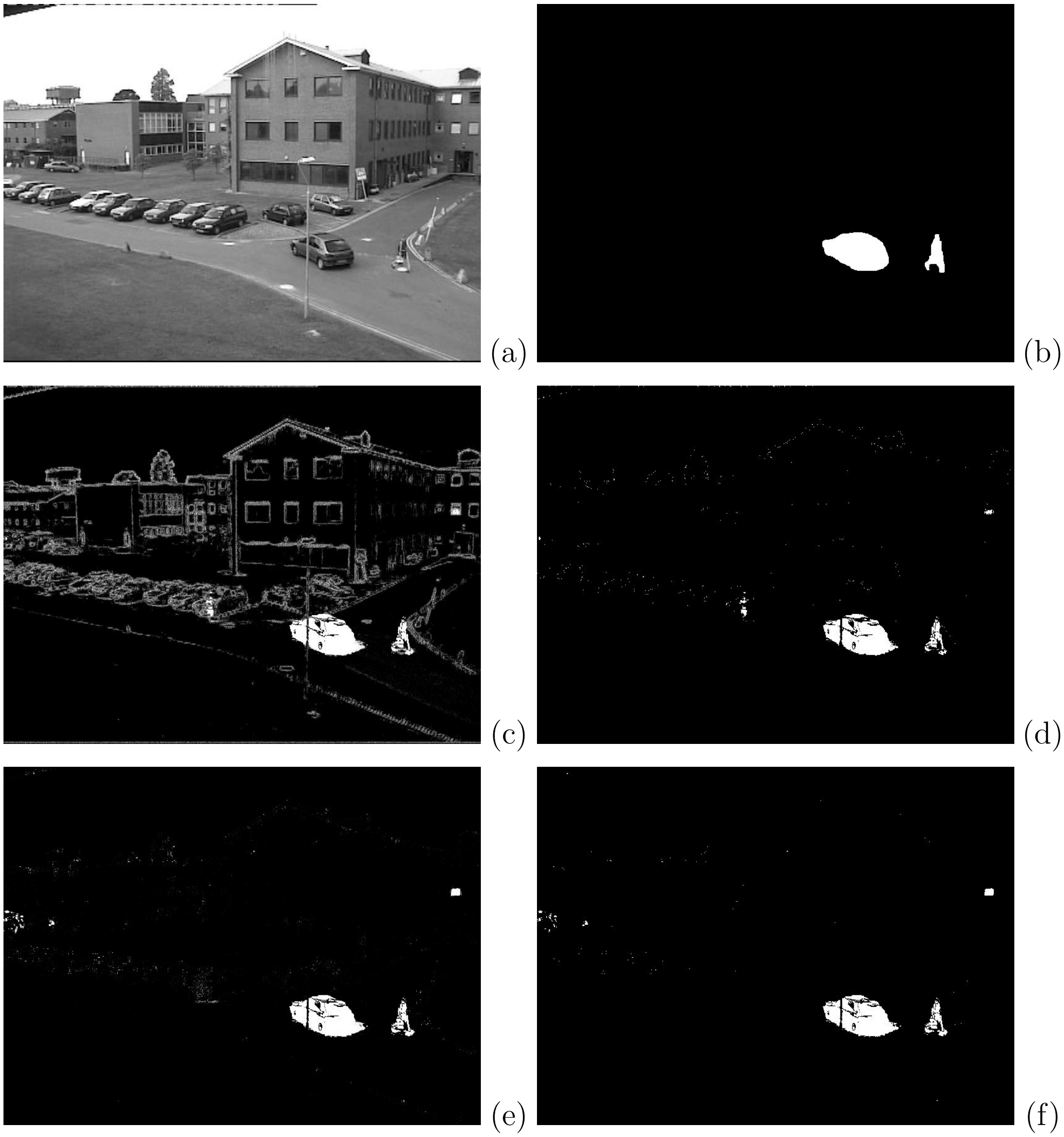

Figure 4.29: Frame 600 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=94.23%)(c), ViBe final segmentation (PCC=99.43%)(d), the foreground probability image for the proposed algorithm (PrCC=99.20%)(e), and the final segmentation for the proposed algorithm (PCC=99.50%)(f).

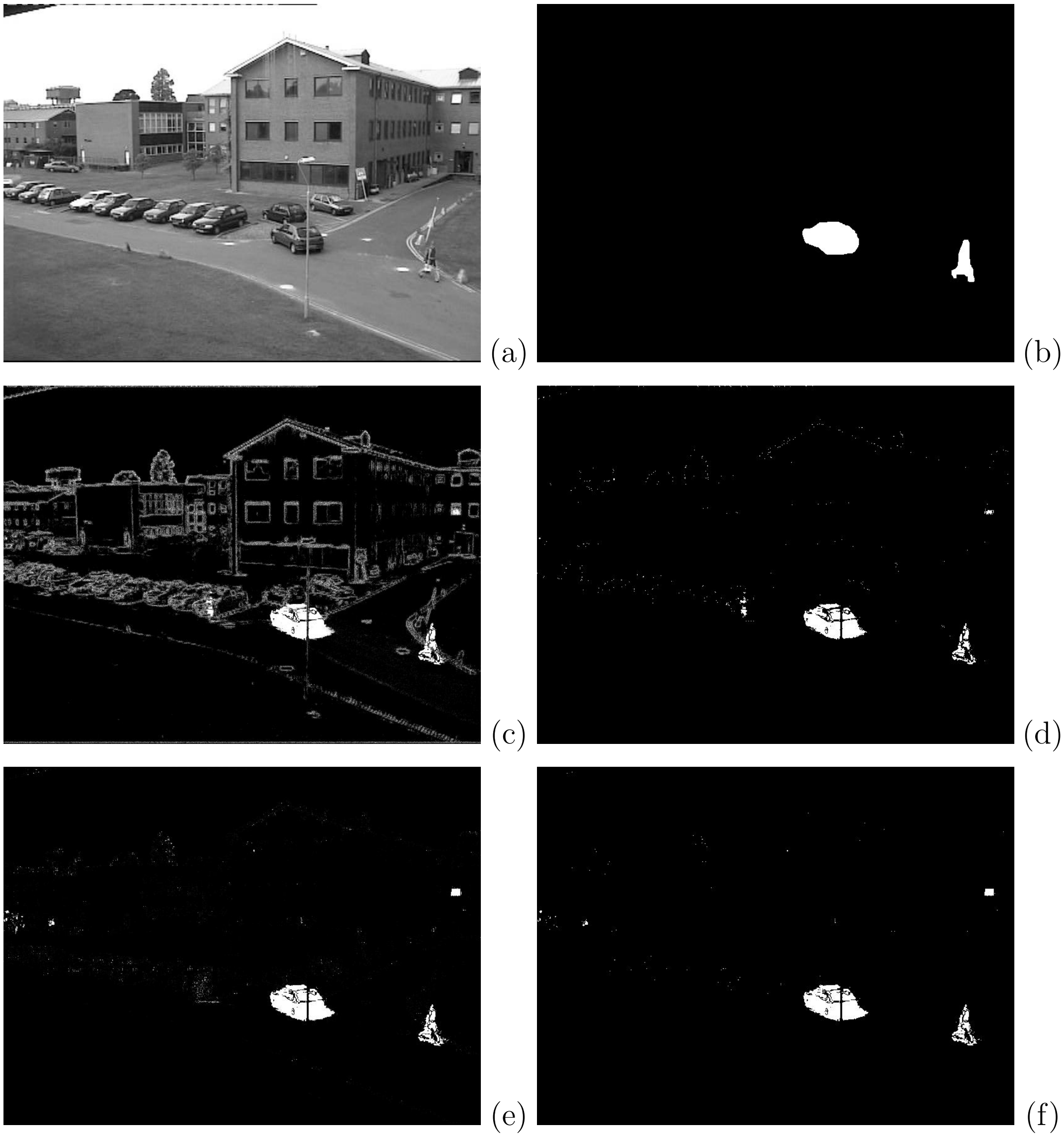

Figure 4.30: Frame 625 of the PETS 1 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=94.42%)(c), ViBe final segmentation (PCC=99.41%)(d), the foreground probability image for the proposed algorithm (PrCC=99.22%)(e), and the final segmentation for the proposed algorithm (PCC=99.50%)(f).

the proposed algorithm achieves 4.37% increase in true positives and a 14.12% increase in false positives, resulting in an unfavorable performance reduction of 0.07%. In terms of PrCC however, we observe an 83.01% reduction in potential false positives that elevate the performance of the proposed algorithm above ViBe by 9.64%.

By frame 1100 (Fig. 4.33) two people have entered the frame, they are traveling upwards on each side of the tree. The person on the right side of the tree has just entered the frame and all that can be observed is the back of his head. The person on the left can be seen from the waist upwards and is clearly wearing a backpack. Tables 4.68 and 4.69 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. According to PCC the proposed algorithm is outperformed by ViBe due to an 82.13% increase in false positives that results in an overall performance decline of 0.36%. Because PrCC is unaffected by the segmentation threshold that leads to the binary label image, we observe an overall improvement in PrCC of 8.89%, where the false positives are observed to decline by 79.85%.

In frame 1125 (Fig. 4.34) the person on the left has walked past the tree and is now partly occluded by the tree branches. The person on the right is more visible but remains between the camera and the tree. Tables 4.70 and 4.71 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. PCC fails to measure any performance difference between the two algorithms, where a 4.21% increase in true positives is essentially offset by a 6.43% increase in false positives. In terms of PrCC a large 85.85% reduction in false positives and a 9.08% increase in true negatives leads to an overall performance improvement of 9.02%.

In frame 1150 (Fig. 4.35) the person on the left is completely occluded by the tree and the person on the right is now partly occluded. Tables 4.72 and 4.73 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Based on PCC the proposed scene model suffers from an 82.84% increase in false positives that yields an overall performance reduction of 0.26%. By PrCC evaluation the proposed technique reduces false positives by 82.17% resulting in an improvement of 8.34% over ViBe.

In frame 1175 (Fig. 4.36) both people are nearly completely occluded by the tree and all that we can observe is the arm of the person on the right. Tables 4.74 and 4.75 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Again, we observe a large discrepancy in the false positive category between the PCC and PrCC metrics which lead to conflicting performance differences of 0.32% and 7.98%, respectively. PCC false positives increase by 120.88% and PrCC false positives decrease by 82.02%.

By frame 1200 (Fig. 4.37) both people are hidden by the tree and we can now see the shadow of the person on the right. Tables 4.76 and 4.77 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. By PCC the performance of the proposed algorithm declines by 0.65% compared to ViBe. Careful analysis reveals that the PCC true positives increased by 20.19% and the false positives also increased by 112.94%. This is a good example of a situation where a simple threshold does not lead to a desirable binary segmentation. In terms of PrCC we observe a 75.42% reduction in the number of false positives which yields a good performance improvement of 7.44% over the ViBe model.

In frame 1225 4.38 we continue to observe a large shadow of the person on the right while the physical bodies of the two people remain occluded by the tree. Tables 4.78 and 4.79 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. Again, the difficulties associated with thresholding can be clearly observed in the PCC results where a 7.14% increase in true positives is offset by an 88.03% increase in false positives that results in a performance decline of 0.39%. PrCC evaluation however, does not suffer from the problems associated with thresholding and an overall improvement of 7.57% is reported due to a 79.44% reduction in false positives.

In frame 1250 (Fig. 4.39) the person on the right begins to emerge from the tree. Tables 4.80 and 4.81 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. PCC measurements indicate a 0.34% reduction in performance based on a 90.34% increase in the number of false positives. Evaluation prior to application of the segmentation threshold by PrCC indicates a performance improvement of 7.47% over ViBe based on a 80.70% reduction in false positives and 7.48% increase in true negatives.

Between frames 1260 (Fig. 4.40) and 1250 (Fig. 4.39) very little change has taken place. Tables 4.82 and 4.83 summarize the performance differences measured between ViBe and the proposed algorithm in terms of PCC and PrCC. In terms of PCC the performance of the proposed technique declines by 0.44% compared to ViBe. By PrCC, the proposed algorithm outperforms ViBe by 7.28%. The large discrepancy is due to the numbers of false positives and

Table 4.64: PETS 2 Frame 1056 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	0	0	0.00%
TN	440,430	442,255	0.41%
FP	1,938	113	-94.17%
FN	0	0	0.00%
PCC	99.56%	99.97%	0.41%

Table 4.65: PETS 2 Frame 1056 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	0.00	0.00	0.00%
TN_{prob}	394,147.86	436,509.61	10.75%
FP_{prob}	46,492.14	4,130.39	-91.12%
FN_{prob}	0.00	0.00	0.00%
PrCC	89.45%	99.06%	10.75%

true positives that vary nonlinearly based on application of the segmentation threshold.

Table 4.66: PETS 2 Frame 1075 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	252	263	4.37%
TN	439,834	439,531	-0.07%
FP	2,146	2,449	14.12%
FN	136	125	-8.09%
PCC	99.48%	99.42%	-0.07%

Table 4.67: PETS 2 Frame 1075 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	273.67	265.61	-2.94%
TN_{prob}	394,403.56	432,461.98	9.65%
FP_{prob}	45,849.96	7,791.54	-83.01%
FN_{prob}	112.82	120.87	7.14%
PrCC	89.57%	98.20%	9.64%

Table 4.68: PETS 2 Frame 1100 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	1,418	1,491	5.15%
TN	438,395	436,731	-0.38%
FP	2,026	3,690	82.13%
FN	529	456	-13.80%
PCC	99.42%	99.06%	-0.36%

Table 4.69: PETS 2 Frame 1100 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	1,516.52	1,495.11	-1.41%
TN_{prob}	394,569.88	429,808.45	8.93%
FP_{prob}	44,130.73	8,892.16	-79.85%
FN_{prob}	422.88	444.29	5.06%
PrCC	89.89%	97.88%	8.89%

Table 4.70: PETS 2 Frame 1125 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	2,611	2,721	4.21%
TN	437,548	437,455	-0.02%
FP	1,446	1,539	6.43%
FN	763	653	-14.42%
PCC	99.50%	99.50%	0.00%

Table 4.71: PETS 2 Frame 1125 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	2,745.65	2,734.87	-0.39%
TN_{prob}	395,439.48	431,360.64	9.08%
FP_{prob}	41,839.70	5,918.54	-85.85%
FN_{prob}	615.17	625.95	1.75%
PrCC	90.37%	98.51%	9.02%

Table 4.72: PETS 2 Frame 1150 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	1,553	1,674	7.79%
TN	438,657	437,378	-0.29%
FP	1,544	2,823	82.84%
FN	614	493	-19.71%
PCC	99.51%	99.25%	-0.26%

Table 4.73: PETS 2 Frame 1150 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	1,738.32	1,709.03	-1.69%
TN_{prob}	397,871.97	431,240.82	8.39%
FP_{prob}	40,609.49	7,240.64	-82.17%
FN_{prob}	420.21	449.51	6.97%
PrCC	90.69%	98.25%	8.34%

Table 4.74: PETS 2 Frame 1175 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	520	692	33.08%
TN	439,922	438,330	-0.36%
FP	1,317	2,909	120.88%
FN	609	437	-28.24%
PCC	99.56%	99.24%	-0.32%

Table 4.75: PETS 2 Frame 1175 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	774.61	738.74	-4.63%
TN_{prob}	400,419.23	432,484.25	8.01%
FP_{prob}	39,096.18	7,031.16	-82.02%
FN_{prob}	349.98	385.85	10.25%
PrCC	91.05%	98.32%	7.98%

Table 4.76: PETS 2 Frame 1200 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	317	381	20.19%
TN	438,952	436,019	-0.67%
FP	2,597	5,530	112.94%
FN	502	438	-12.75%
PCC	99.30%	98.65%	-0.65%

Table 4.77: PETS 2 Frame 1200 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	506.34	426.03	-15.86%
TN_{prob}	400,185.80	430,082.98	7.47%
FP_{prob}	39,638.40	9,741.22	-75.42%
FN_{prob}	309.46	389.77	25.95%
PrCC	90.93%	97.70%	7.44%

Table 4.78: PETS 2 Frame 1225 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	294	315	7.14%
TN	440,039	438,296	-0.40%
FP	1,980	3,723	88.03%
FN	55	34	-38.18%
PCC	99.54%	99.15%	-0.39%

Table 4.79: PETS 2 Frame 1225 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	314.93	315.70	0.25%
TN_{prob}	401,971.64	432,415.05	7.57%
FP_{prob}	38,320.72	7,877.31	-79.44%
FN_{prob}	32.71	31.93	-2.38%
PrCC	91.30%	98.21%	7.57%

Table 4.80: PETS 2 Frame 1250 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	264	384	45.45%
TN	439,939	438,338	-0.36%
FP	1,772	3,373	90.35%
FN	393	273	-30.53%
PCC	99.51%	99.18%	-0.34%

Table 4.81: PETS 2 Frame 1250 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	424.82	421.70	-0.73%
TN_{prob}	402,668.32	432,781.75	7.48%
FP_{prob}	37,317.25	7,203.82	-80.70%
FN_{prob}	229.62	232.73	1.36%
PrCC	91.48%	98.31%	7.47%

Table 4.82: PETS 2 Frame 1260 Percentage Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP	216	320	48.15%
TN	439,935	437,898	-0.46%
FP	1,812	3,849	112.42%
FN	405	301	-25.68%
PCC	99.50%	99.06%	-0.44%

Table 4.83: PETS 2 Frame 1260 Probability Correct Classification Details

	ViBe	Proposed Algorithm	Percentage Change
TP_{prob}	392.13	366.09	-6.64%
TN_{prob}	403,009.30	432,401.34	7.29%
FP_{prob}	37,012.12	7,620.09	-79.41%
FN_{prob}	226.44	252.48	11.50%
PrCC	91.55%	98.21%	7.28%

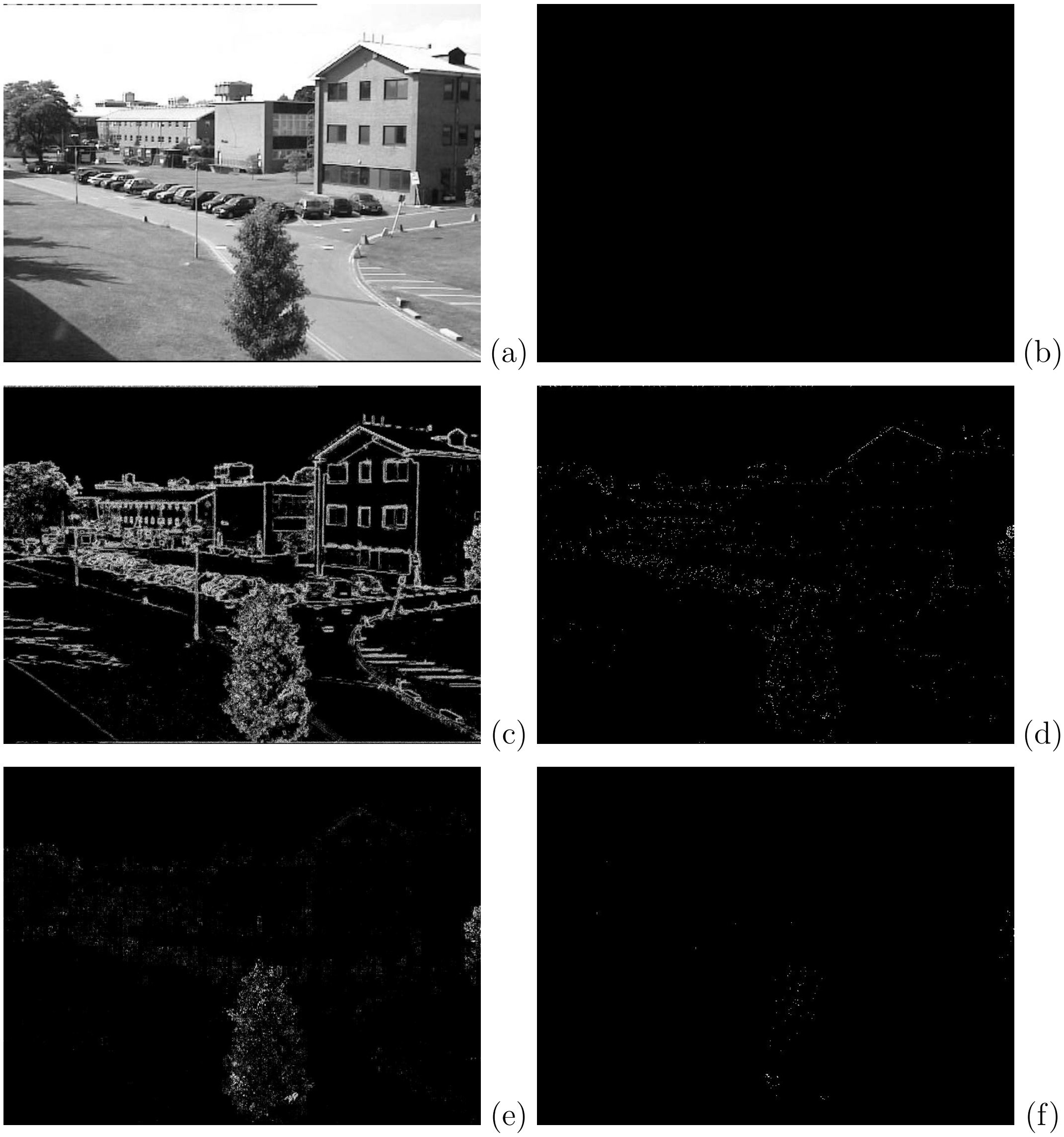

Figure 4.31: Frame 1056 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=89.45%)(c), ViBe final segmentation (PCC=99.56%)(d), the foreground probability image for the proposed algorithm (PrCC=98.96%)(e), and the final segmentation for the proposed algorithm (PCC=99.97%)(f).

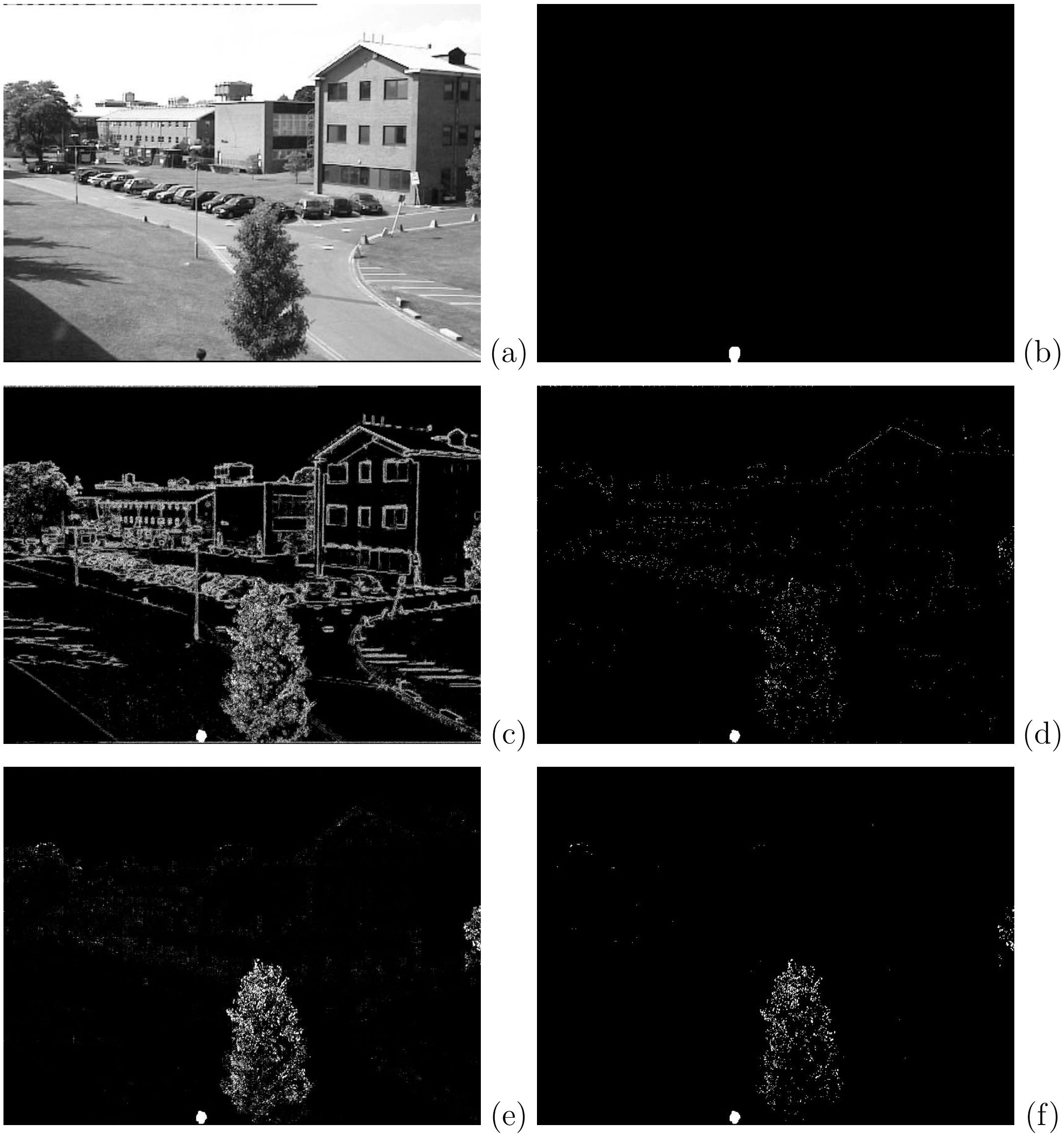

Figure 4.32: Frame 1075 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=89.57%)(c), ViBe final segmentation (PCC=99.48%)(d), the foreground probability image for the proposed algorithm (PrCC=98.04%)(e), and the final segmentation for the proposed algorithm (PCC=99.42%)(f).

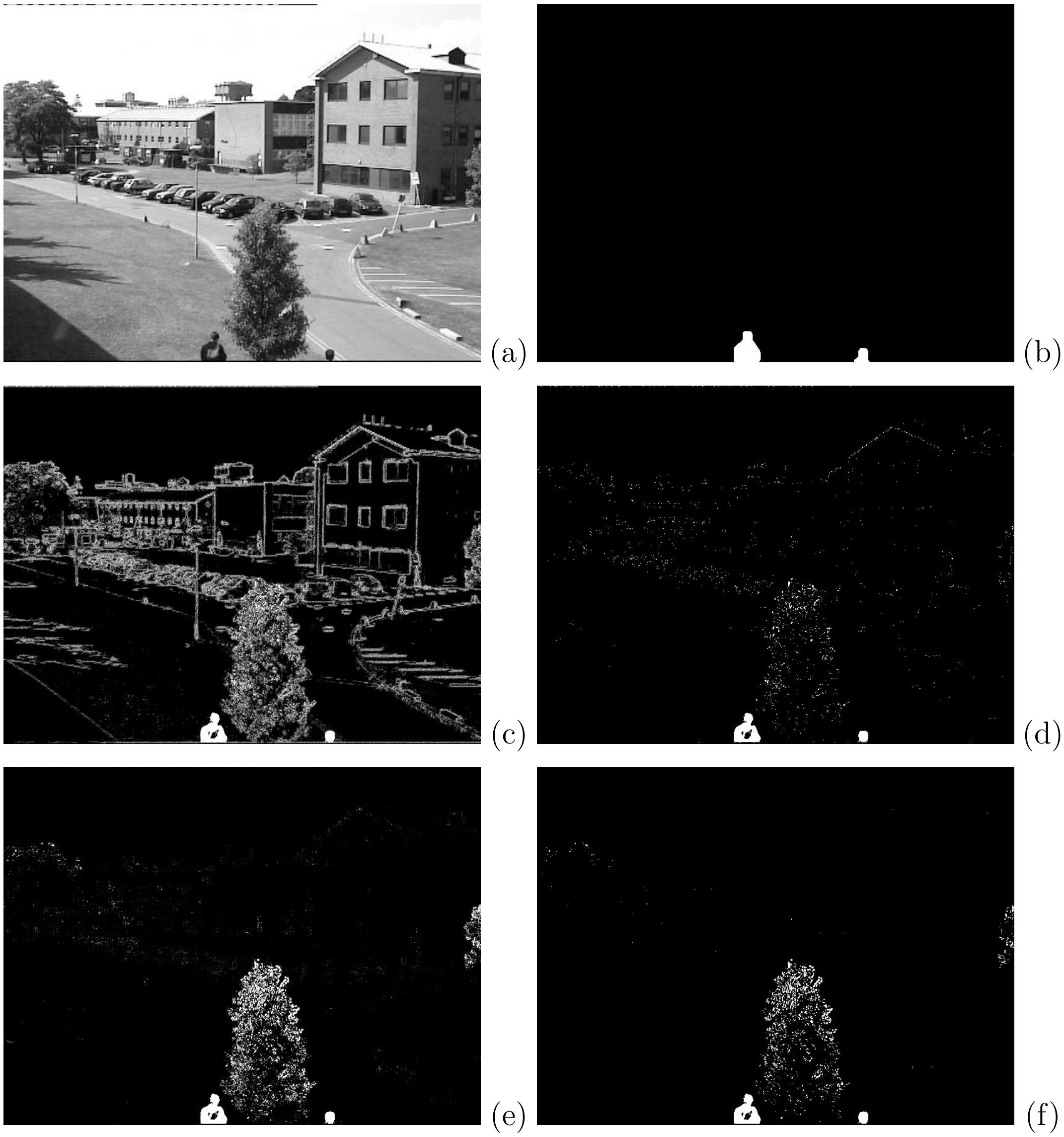

Figure 4.33: Frame 1100 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=89.89%)(c), ViBe final segmentation (PCC=99.42%)(d), the foreground probability image for the proposed algorithm (PrCC=97.70%)(e), and the final segmentation for the proposed algorithm (PCC=99.06%)(f).

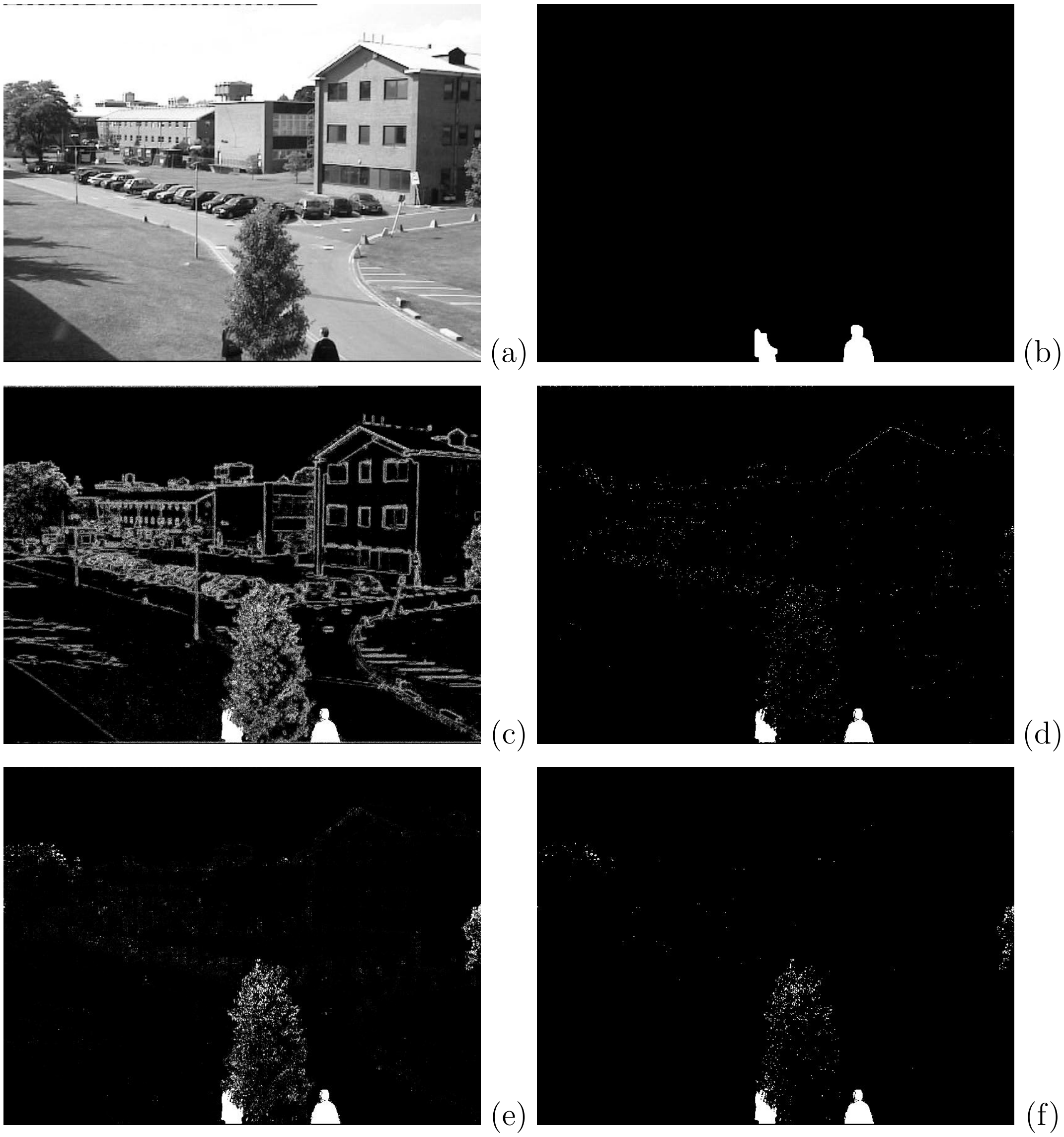

Figure 4.34: Frame 1125 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=90.37%)(c), ViBe final segmentation (PCC=99.50%)(d), the foreground probability image for the proposed algorithm (PrCC=98.39%)(e), and the final segmentation for the proposed algorithm (PCC=99.50%)(f).

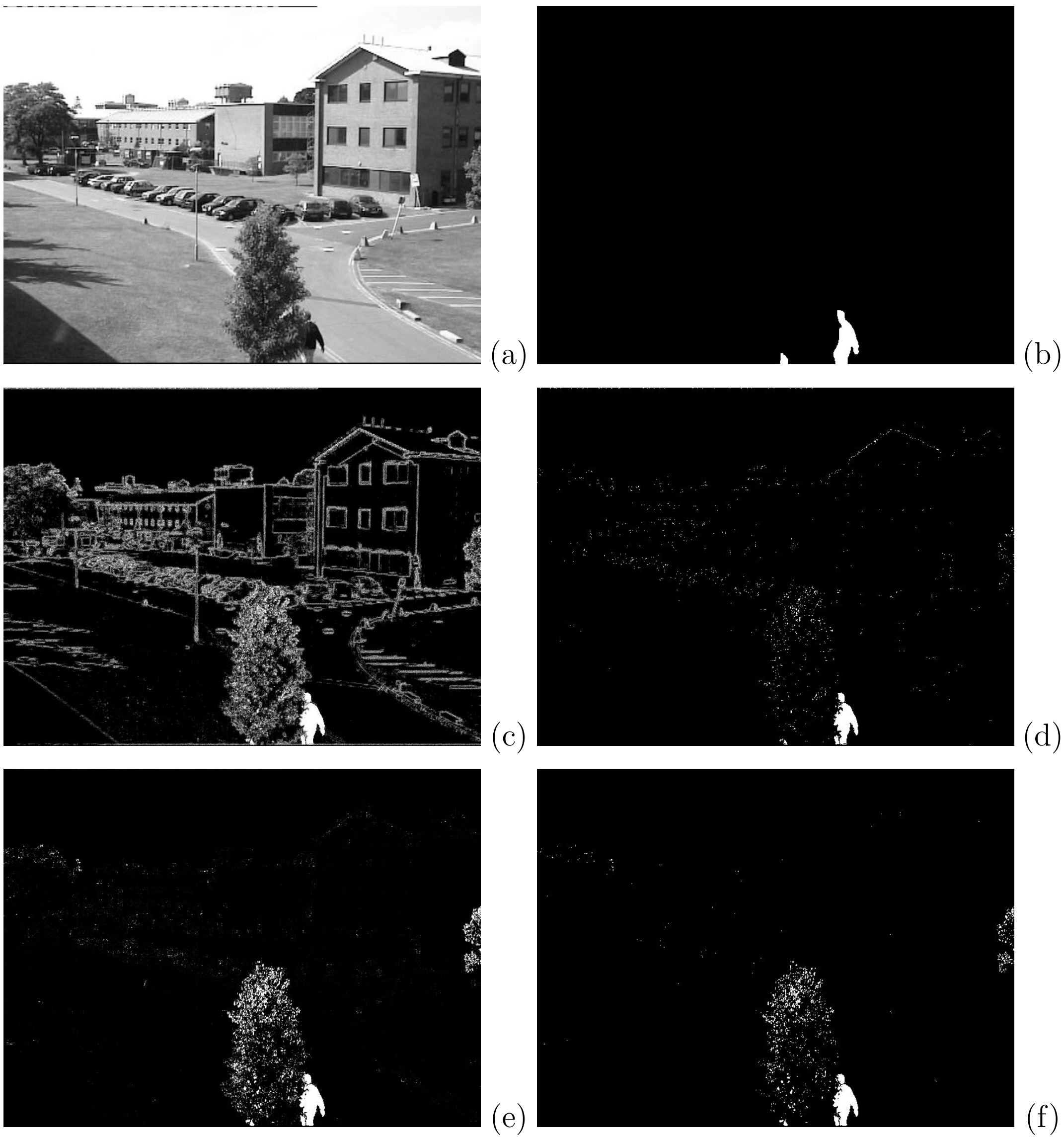

Figure 4.35: Frame 1150 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=90.69%)(c), ViBe final segmentation (PCC=99.51%)(d), the foreground probability image for the proposed algorithm (PrCC=98.12%)(e), and the final segmentation for the proposed algorithm (PCC=99.25%)(f).

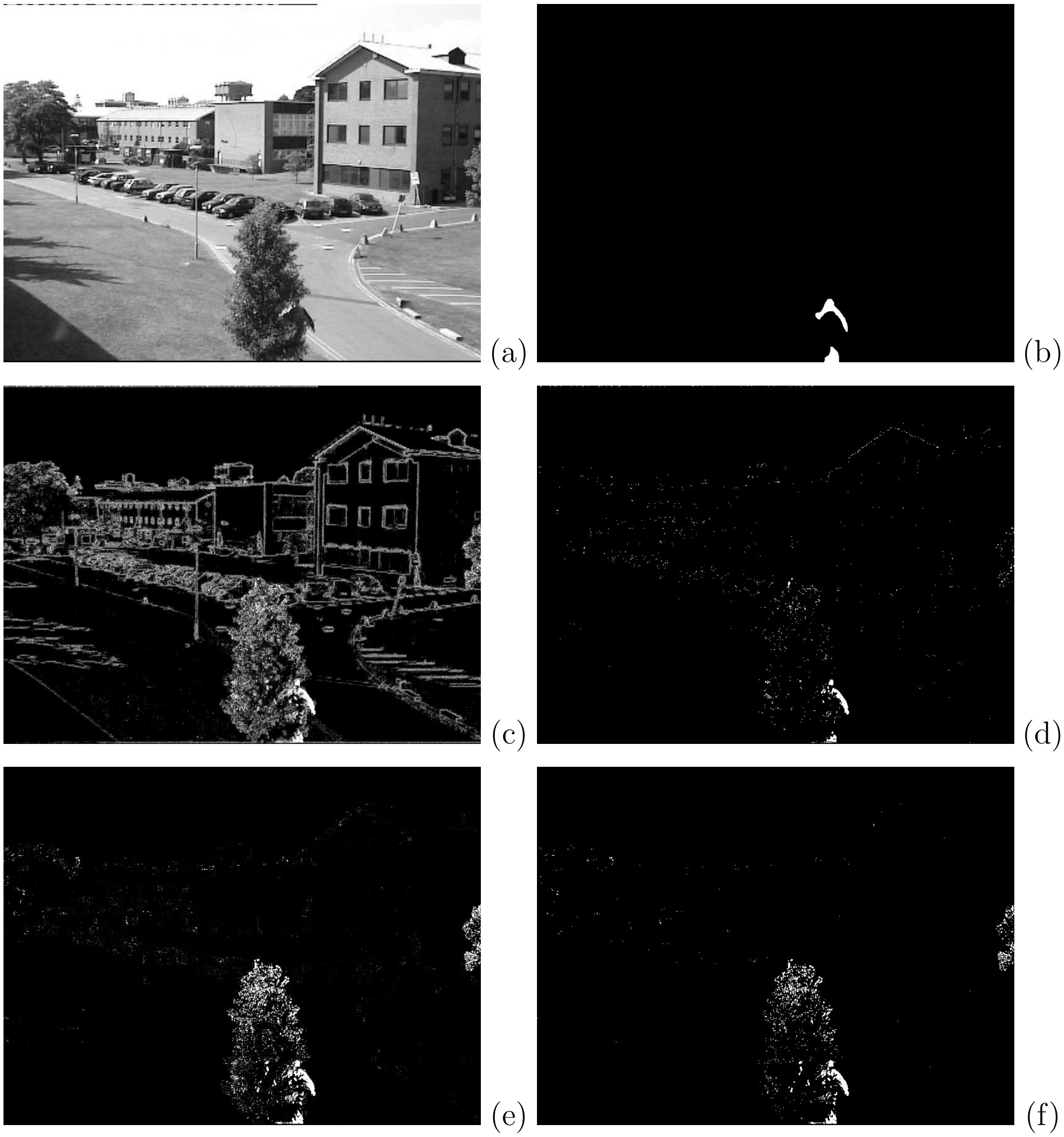

Figure 4.36: Frame 1175 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=91.05%)(c), ViBe final segmentation (PCC=99.56%)(d), the foreground probability image for the proposed algorithm (PrCC=98.19%)(e), and the final segmentation for the proposed algorithm (PCC=99.24%)(f).

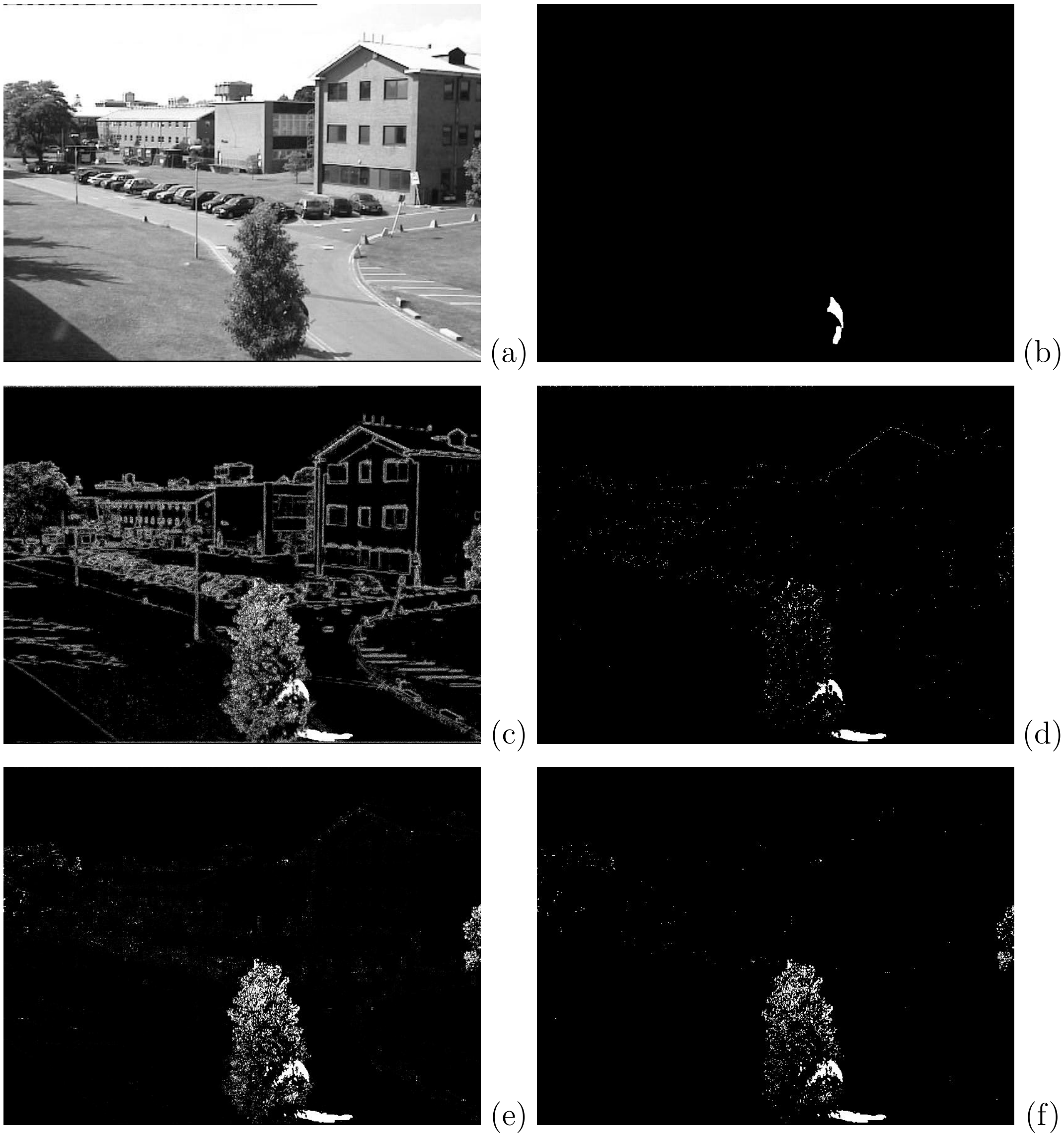

Figure 4.37: Frame 1200 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=90.93%)(c), ViBe final segmentation (PCC=99.30%)(d), the foreground probability image for the proposed algorithm (PrCC=97.55%)(e), and the final segmentation for the proposed algorithm (PCC=98.65%)(f).

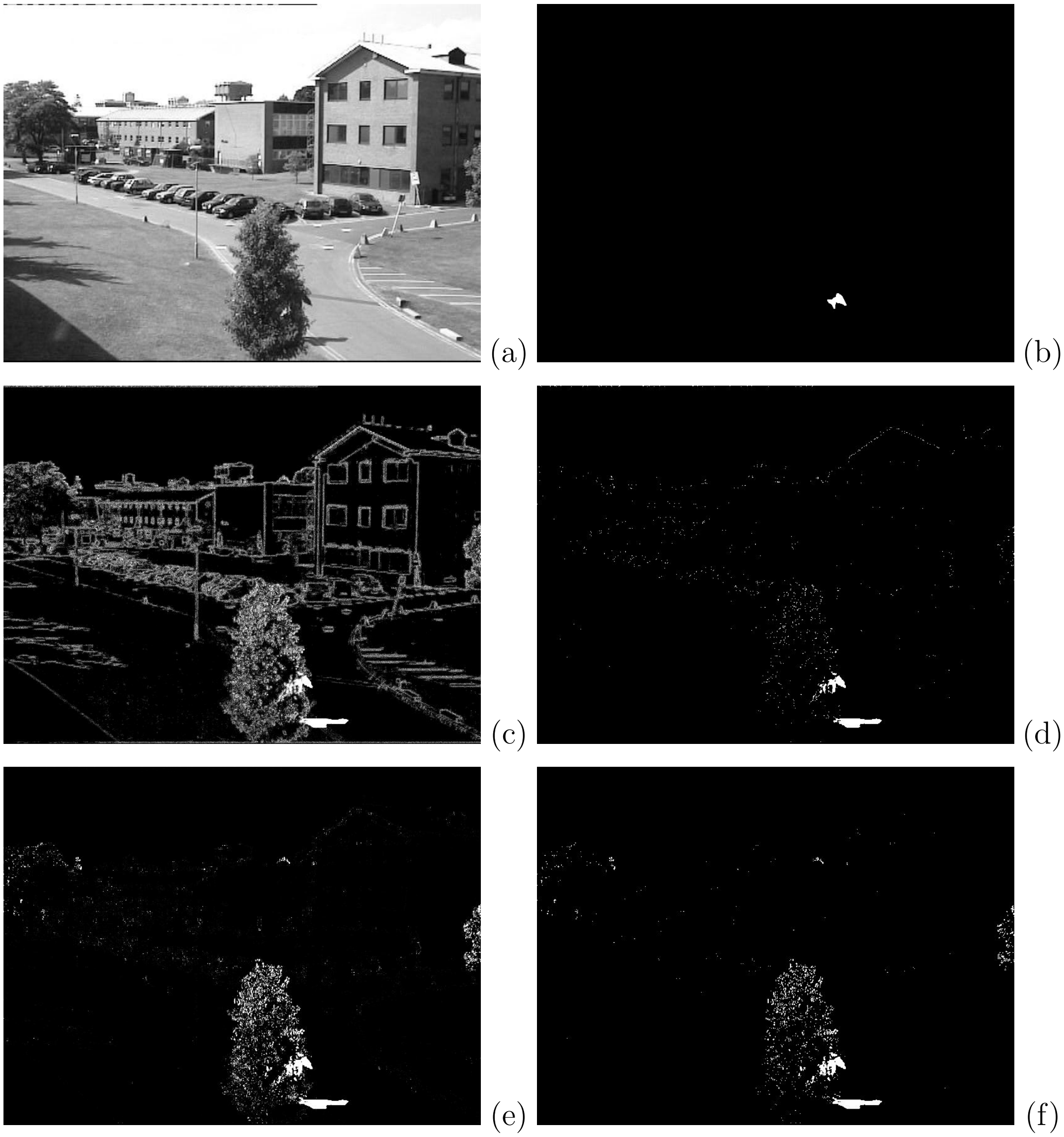

Figure 4.38: Frame 1225 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=91.30%)(c), ViBe final segmentation (PCC=99.54%)(d), the foreground probability image for the proposed algorithm (PrCC=98.07%)(e), and the final segmentation for the proposed algorithm (PCC=99.15%)(f).

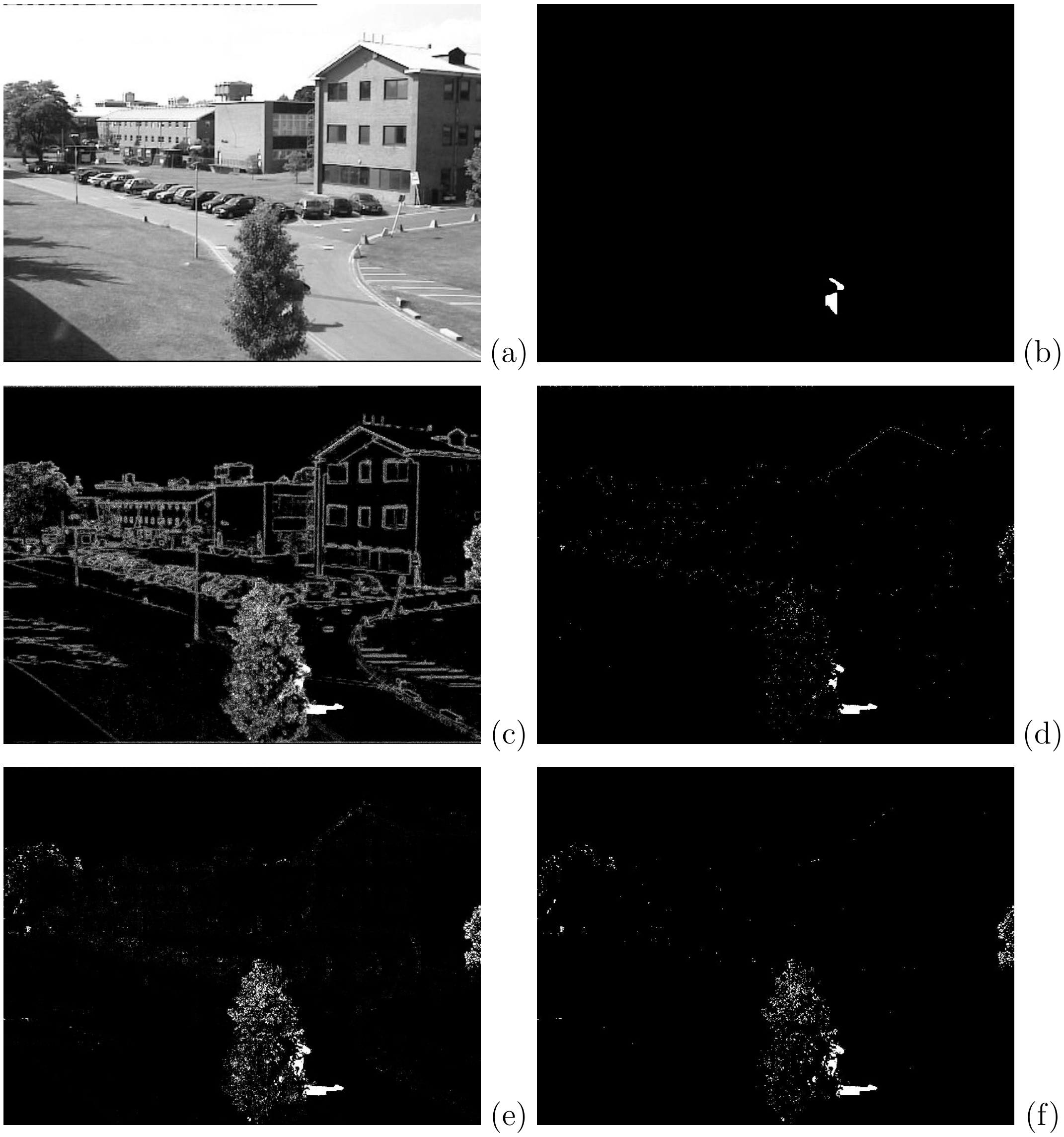

Figure 4.39: Frame 1250 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=91.50%)(c), ViBe final segmentation (PCC=99.51%)(d), the foreground probability image for the proposed algorithm (PrCC=98.19%)(e), and the final segmentation for the proposed algorithm (PCC=99.18%)(f).

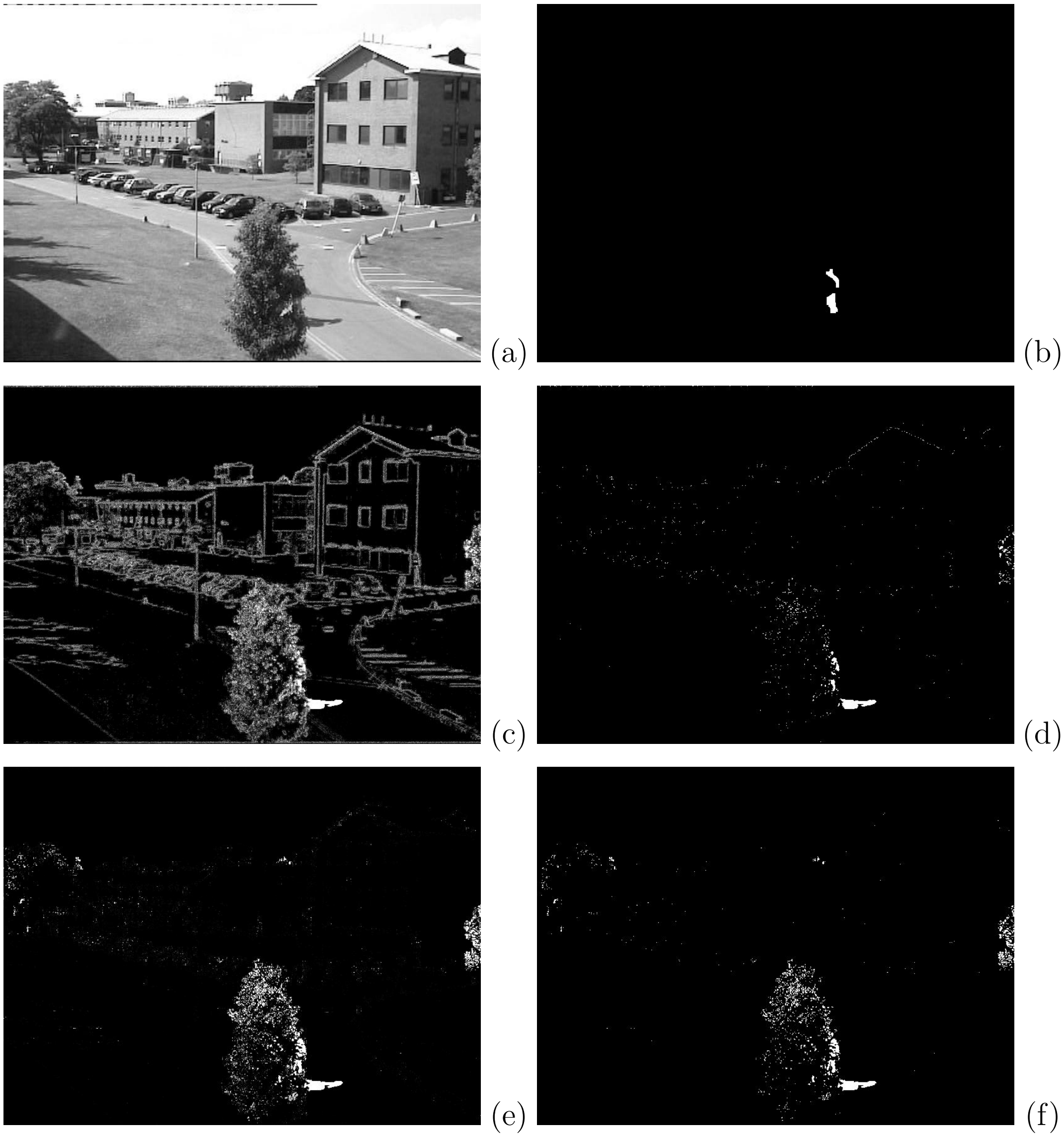

Figure 4.40: Frame 1260 of the PETS 2 sequence depicting the original grayscale image (a), manually generated ground truth image (b), ViBe foreground probability image (PrCC=91.55%)(c), ViBe final segmentation (PCC=99.50%)(d), the foreground probability image for the proposed algorithm (PrCC=98.10%)(e), and the final segmentation for the proposed algorithm (PCC=99.06%)(f).

Chapter 5

Conclusion

A new unsupervised pixel level nonparametric scene model was proposed in this dissertation for segmenting video into foreground and background regions. The architecture of the proposed model was based on valuable research results obtained over a long period of time by Elgammal [23,24] and Barnich [3]. The nonparametric model representation and the algorithm used for estimating the bandwidth of the kernel from a sample collection were originally proposed by Elgammal in [24]. The nondeterministic neighborhood information sharing process and the use of spherical cutoff kernels to reduce computational complexity were both proposed by Barnich in [3]. The original contributions of this dissertation include a conservative update policy based on outlier identification and replacement, an intelligent neighborhood information sharing algorithm, and the PrCC performance metric used to compare the scene models prior to application of the final classification threshold.

The proposed algorithm was compared to the state-of-the-art ViBe algorithm using four well known videos that have been frequently used to evaluate segmentation algorithms. The two models were compared using both the PCC and PrCC performance metrics. Because the PCC metric is computed after application of an empirically determined threshold, I developed the PrCC metric as a better way to evaluate the underlying data models as I argued in

Table 5.1: Comparison of the proposed algorithm and ViBe.

Challenge	ViBe	Proposed
Gradual Illumination Changes	x	x
Sudden Illumination Changes		
Dynamic Background Components	x	x
Camouflage		x
Shadows		x
Ghosts		x
Foreground Aperture		

Chapter 4. In both cases, the proposed algorithm significantly outperformed the ViBe system except when using the PCC metric on the PETS 2 sequence. By visual inspection, I believe that the PrCC measurement provides a better characterization of the disparity in the two underlying models, which can be clearly observed in the foreground probability images prior to thresholding. However, the proposed method yielded large improvements in terms of both a reduction in false positives as well as an increase in true positives irrespective of the measure used in the evaluation.

With respect to the well known challenges that have been reported in video segmentation systems, the proposed method improved on the ViBe model by providing solutions to the ghost and false foreground detection problems and improving detections in the case of camouflage. The proposed method does not address the problems of sudden illumination changes, shadows or foreground aperture. I believe that detection of sudden illumination changes and mitigation of the shadow and foreground aperture problems could be dealt with using any number of existing techniques in the grayscale color space. Table 5.1 summarizes the major differences between the ViBe model and the proposed algorithm.

In terms of computational complexity, the proposed algorithm is not as efficient as the ViBe system. The differences in complexity between the two models are best highlighted by the increased computational requirements necessary to identify outliers using KDE and to evaluate the similarity in neighboring background models. The proposed method must perform KDE on each sample distribution to identify outliers and estimate the similarity between neighboring models by computing sparse histograms and several large inner products. Each of these tasks represents an increase in computation when compared to simply drawing a sample from a uniformly distributed random variable. However, I believe that the theoretical ideas that were proposed and subsequently verified within this dissertation justify the relatively unavoidable increases in computational complexity.

Although the proposed method as well as several existing methods have achieved very good segmentation results, there are many unexplored topics in scene modeling. Until very recently in [90] spatial correlations between neighboring pixel or block level models were completely ignored, most likely due to insufficient computational resources. As a consequence, the techniques used to identify neighboring models remain relatively immature throughout the literature. For example, why do we limit ourselves to 4 or 8 connected neighborhoods? It may be desireable to redefine neighborhood connections based on an analysis of the background models. The idea to propagate foreground pixels throughout the neighboring models was certainly unique [3], but why do we only share the information with one neighboring model? Perhaps we could share the information with a larger number of neighbors, perhaps based on a measurement of their similarity to the central model?

Bibliography

[1] T. Aach and A. Kaup, “Statistical model-based change detection in moving video,” *Signal Process.*, vol. 31, no. 2, pp. 165–180, 1993.

[2] A. Adam, R. Kimmel, and E. Rivlin, “On scene segmentation and histograms-based curve evolution,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 31, no. 9, pp. 1708–1714, Sep. 2009.

[3] O. Barnich and M. V. Droogenbroeck, “Vibe: A universal background subtraction algorithm for video sequences,” *IEEE Trans. Image Process.*, vol. 20, no. 6, pp. 1709–1724, June 2011.

[4] L. Baum and J. Eagon, “An inequality with applications to statistical estimations for probabilistic functions of Markov processes and to a model of ecology,” *Amer. Math. Soc. Bull.*, vol. 73, pp. 360–363, 1967.

[5] L. Baum, T. Petrie, G. Soules, and N. Weiss, “A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains,” *Annals of Math. Statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[6] C. Benedek and T. Sziranyi, “Bayesian foreground and shadow detection in uncertain frame rate surveillance videos,” *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 608–621, Apr. 2008.

[7] A. Bhattacharyya, “On a measure of divergence between two multinomial populations,” *Sankhya: The Indian Journal of Statistics*, vol. 7, no. 4, pp. 401–406, July 1946.

[8] M. Bichsel, “Segmenting simply connected moving objects in a static scene,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 16, no. 11, pp. 1135–1142, Nov. 1994.

[9] A. Bleicher, “Eyes in the sky that see too much,” *IEEE Spectrum*, vol. 47, no. 10, pp. 16–16, Oct. 2010.

[10] A. Bovik, *Handbook of Image and Video Process.* Orlando, FL, USA: Academic Press, Inc., 2011.

[11] A. Briassouli and N. Ahuja, “Extraction and analysis of multiple periodic motions in video sequences,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 29, no. 7, pp. 1244–1261, July 2007.

[12] A. Briassouli and I. Kompatsiaris, “Robust temporal activity templates using higher order statistics,” *IEEE Trans. Image Process.*, vol. 18, no. 12, pp. 2756–2768, Dec. 2009.

[13] S. Brutzer, B. Hoferlin, and G. Heidemann, “Evaluation of background subtraction techniques for video surveillance,” in *Proc. IEEE Int'l. Conf. on Comp. Vision, Pattern Recog.*, Colorado Springs, CO, June 2011, pp. 1937–1944.

[14] D. Comaniciu, “An algorithm for data-driven bandwidth selection,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 25, no. 2, pp. 281–288, Feb. 2003.

[15] D. Comaniciu, V. Ramesh, and P. Meer, “The variable bandwidth mean shift and data-driven scale selection,” in *Proc. IEEE Int'l. Conf. Computer Vision*, vol. 1, Vancouver, BC, CA, July 7-14 2001, pp. 438–445.

[16] R. Cucchiara, M. Piccardi, and A. Prati, “Detecting moving objects, ghosts, and shadows in video streams,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 25, no. 10, pp. 1337–1342, Oct. 2003.

[17] R. Cutler and L. Davis, “View-based detection and analysis of periodic motion,” in *Proc. IEEE Int'l. Conf. Pattern Recog.*, Brisbane, Qld., Australia, Aug. 16-20 1998, pp. 495–500.

[18] C. V. P. R. (CVPR), “Performance evaluation of tracking and surveillance (PETS),” *Website: ftp://ftp.pets.rdg.ac.uk/PETS2001/*, 2001.

[19] I. Daubechies, *Ten Lectures on Wavelets.* Philadelphia: SIAM, 1992.

[20] J. Davis and V. Sharma, “Background-subtraction in thermal imagery using contour saliency,” *Int'l. J. Computer Vision*, vol. 71, no. 2, pp. 161–181, 2007.

[21] A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society*, vol. 39 (Series B), pp. 1–38, 1977.

[22] G. Donohoe, D. Hush, and N. Ahmed, "Change detection for target detection and classification in video sequences," in *Proc. IEEE Int'l. Conf. Acoustic Speech and Signal Process.*, 1988, pp. 1084–1087.

[23] A. Elgammal, R. Duraiswami, D. Harwood, and L. Davis, "Background and foreground modeling using nonparametric kernel density estimation for visual surveillance," *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1163, July 2002.

[24] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *Proc. European Conf. Computer Vision*, vol. 1843, 2000, pp. 751–767.

[25] S. Elhabian, K. El-Sayed, and S. Ahmed, "Moving object detection in spatial domain using background removal techniques - state-of-art," *Recent Patents on Computer Science*, vol. 1, pp. 32–54, 2008.

[26] N. Friedman and S. Russell, "Image segmentation in video sequences: A probabilistic approach," in *Proc. Thirteenth Conf. Uncertainty in Artificial Intelligence*, Aug. 1-3 1997.

[27] K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," *IEEE Trans. Information Theory*, vol. IT-21, no. 1, pp. 32–40, Jan. 1975.

[28] B. Han, D. Comaniciu, and L. Davis, "Sequential kernal density approximation through mode propogation," in *Proc. Asian Conf. Computer Vision*, Jeju Island, Korea, 2004.

[29] I. Haritaoglu, D. Harwood, and L. Davis, "W4: Real-time surveillance of people and their activities," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 22, no. 8, pp. 809–830, Aug. 2000.

[30] M. Harville, "A framework for high-level feedback to adaptive, per-pixel, mixture-of-Gaussian background models," in *Proc. European Conf. Computer Vision*, 2002, pp. 37–49.

[31] E. Hayman and J.-O. Eklundh, "Statistical background subtraction for a mobile observer," in *Proc. IEEE Int'l. Conf. Computer Vision*, Nice, France, Oct. 13-16 2003, pp. 67–74.

[32] M. Heikkila and M. Pietikainen, “A texture-based method for modeling the background and detecting moving objects,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 28, no. 4, pp. 657–662, Apr. 2006.

[33] M. Heikkila, M. Pietikainen, and J. Heikkila, “A texture-based method for detecting moving objects,” in *Proc. British Machine Vision Conf.*, vol. 1, 2004, pp. 187–196.

[34] T. Horprasert, D. Harwood, and L. Davis, “A statistical approach for real-time background subtraction and shadow detection,” in *Proc. IEEE Int'l. Conf. Computer Vision*, 1999, pp. 1–19.

[35] Y. Hsu, H. Nagel, and G. Rekers, “New likelihood test methods for change detection in image sequences,” *Computer Vision, Graphics, Image Process.*, vol. 26, pp. 73–106, 1984.

[36] S.-S. Huang, L.-C. Fu, and P.-Y. Hsiao, “Region-level motion-based background modeling and subtraction using MRFs,” *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1446–1456, May 2007.

[37] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld, “Detection and location of people in video images using adaptive fusion of color and edge information,” in *Proc. IEEE Int'l. Conf. on Pattern Recog.*, vol. 4, 2000, pp. 627–630.

[38] R. Jain, W. Martin, and J. Aggarwal, “Segmentation through the detection of changes due to motion,” *Computer Graphics and Image Process.*, vol. 11, no. 1, pp. 13–34, 1979.

[39] R. Jain, D. Militzer, and H. Nagel, “Separating non-stationary from stationary scene components in a sequence of real world tv-images,” in *Proc. Int'l. Joint Conf. Artificial Intel.*, Cambridge, MA, USA, Aug. 1977, pp. 612–618.

[40] R. Jain and H.-H. Nagel, “On the analysis of accumulative difference pictures from image sequences of real worl scenes,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 1, no. 2, pp. 206–214, Apr. 1979.

[41] O. Javed, K. Shafique, and M. Shah, “A hierarchical approach to robust background subtraction using color and gradient information,” in *Proc. IEEE Workshop on Motion, Video Computing*, Dec. 5-6 2002, pp. 22–27.

[42] A. Joshi and N. Papanikolopoulos, "Learning to detect moving shadows in dynamic environments," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 30, no. 11, pp. 2055–2063, Nov. 2008.

[43] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Proc. European Workshop Advanced Video Based Surveillance Systems*, 2001.

[44] R. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME J. Basic Engr.*, vol. 82 Series D, pp. 35–45, 1960.

[45] K.-P. Karmann and A. von Brandt, "Moving object recognition using adaptive background memory," in *Time-Varying Image Process. and Moving Object Recog. 2: Proc. Third Int'l. Workshop*, Florence, Italy, May 29-31 1989, pp. 289–296.

[46] K.-P. Karmann, A. von Brandt, and R. Gerl, "Moving object segmentation based on adaptive reference images," in *Proc. Fifth European Signal Process. Conf.*, Barcelona, Spain, Sep. 18-21 1990, pp. 951–954.

[47] J. Kato, T. Watanabe, S. Joga, J. Rittscher, and A. Blake, "An HMM-based segmentation method for traffic monitoring movies," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 24, no. 9, pp. 1291–1296, Sep. 2002.

[48] K. Kim, T. Chalidabhongse, D. Harwood, and L. Davis, "Background modeling and subtraction by codebook construction," in *Proc. IEEE Int'l. Conf. Image Process.*, vol. 5, Oct. 24-27 2004, pp. 3061–3064.

[49] K. Kim, T. Thanarat, H. Chalidabbhognse, D. Harwood, and L. Davis, "Real time foreground-background segmentation using codebook model," *Real-Time Imaging*, vol. 11, pp. 172–185, 2005.

[50] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, no. 9, pp. 1464–1480, Sept. 1990.

[51] D. Koller, J. Weber, and J. Malik, "Robust multiple car tracking with occlusion reasoning," *Technical Report UCB/CSD-93-780, University of California at Berkeley*, 1993.

[52] D. Koller, J. Weber, J. Malik, G. Ogasawara, B. Rao, and S. Russell, "Towards robust automatic traffic scene analysis in real-time," in *Proc. IEEE Int'l. Conf. on Pattern Recog.*, Jerusalem, Israel, Oct. 9 1994, pp. 126–131.

[53] A. Kolmogorov, “Interpolation and extrapolation von station ärenzufälligen Folgen,” *Bull. Acad. Sci, U.S.S.R., Ser. Math.*, vol. 5, pp. 3–14, 1941.

[54] E. F. Krause, *TAXICAB GEOMETRY: An adventure in Non-Euclidean Geometry.* Mineola, NY: Dover, 1986.

[55] S. Kullback and R. Leibler, “On information and sufficiency,” *Annals of Math. Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[56] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in *Int’l. Conf. Machine Learning*, 2001, pp. 282–289.

[57] D.-A. Lee, “Effective gaussian mixture learning for video background subtraction,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 27, no. 5, pp. 827–832, May 2005.

[58] L. Li, W. Huang, I. Gu, and Q. Tian, “Foreground object detection from videos containing complex background,” in *ACM Int’l. Conf. Multimedia*, Berkeley, CA, Nov. 2003, pp. 2–10.

[59] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, “Statistical modeling of complex backgrounds for foreground object detection,” *IEEE Trans. Image Process.*, vol. 13, no. 11, pp. 1459–1472, Nov. 2004.

[60] S.-C. Liu, C.-W. Fu, and S. Chang, “Statistical change detection with moments under time-varying illumination,” *IEEE Trans. Image Process.*, vol. 7, no. 9, pp. 1258–1268, Sep. 1998.

[61] S. Lloyd, “Least squares quantization in PCM,” *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–137, March 1982.

[62] B. Lucas and T. Kanade, “An iterative image regsitration technique with an application to stereo vision,” in *Proc. Image Understanding Workshop*, 1981, pp. 121–130.

[63] L. Maddalena and A. Petrosino, “A self-organizing approach to background subtraction for visual surveillance applications,” *IEEE Trans. Image Process.*, vol. 17, no. 7, pp. 1168–1177, July 2008.

[64] V. Mahadevan and N. Vasconcelos, “Spatiotemporal saliency in dynamic scenes,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 32, no. 1, pp. 171–177, 2010.

[65] P. Mahalanobis, “On the generalised distance in statistics,” in *Proc. National Institute of Sciences of India*, vol. 2, 1936, pp. 49–55.

[66] A. Manzanera, “$\Sigma - \Delta$ background subtraction and the Zipf law,” *Progress in Pattern Recog., Image Analysis and Applications, Springer*, vol. 4756, pp. 42–51, Nov. 2007.

[67] M. Mason and Z. Duric, “Using histograms to detect and track objects in color video,” in *Proc. Applied Imagery Pattern Recog. Workshop*, 2001, pp. 154–159.

[68] N. McFarlane and C. Schofield, “Segmentation and tracking of piglets in images,” in *Machine Vision and Applications*, vol. 8, no. 3, 1995, pp. 187–193.

[69] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler, “Tracking groups of people,” in *Computer Vision and Image Understanding*, vol. 80, no. 1, Oct 2000, pp. 42–56.

[70] G. McLachlan and D. Peel, *Finite Mixture Models.* John Wiley and Sons, 2000.

[71] A. Mittal and N. Paragios, “Motion-based background subtraction using adaptive kernel density estimation,” in *Proc. IEEE Int’l. Conf. on Comp. Vision, Pattern Recog.*, vol. 2, Washington, DC, USA, June 27 - July 2 2004, pp. 302–309.

[72] B. Moghaddam and A. Pentland, “Probabilistic visual learning for object detection,” in *Proc. IEEE Int’l. Conf. Computer Vision*, vol. 1, June 20-23 1995, p. 786.

[73] A. Monnet, A. Mittal, N. Paragios, and R. Visvanathan, “Background modeling and subtraction of dynamic scenes,” in *Proc. IEEE Int’l. Conf. Computer Vision*, vol. 2, Nice, France, Oct. 13-16 2003, pp. 1305–1312.

[74] M. Nikulin, “Chi-squared test for normality,” in *Proc. Int’l. Vilnius Conf. Probability Theory and Mathematical Statistics*, vol. 2, 1973, pp. 119–122.

[75] N. Ohta, “A statistical approach to background subtraction for surveillance systems,” in *Proc. IEEE Int’l. Conf. Computer Vision*, vol. 2, Vancouver, BC, CA, July 7-14 2001, pp. 481–486.

[76] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 24, no. 7, pp. 971–987, July 2002.

[77] N. Oliver, B. Rosario, and A. Pentland, “A Bayesian computer vision system for modeling human interactions,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 22, no. 8, pp. 831–843, Aug. 2000.

[78] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. Sys. Man, Cybernetics*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[79] T. Parag, A. Elgammal, and A. Mittal, “A framework for feature selection for background subtraction,” in *Proc. IEEE Int’l. Conf. on Comp. Vision, Pattern Recog.*, vol. 2, New York, NY, USA, June 17-22 2006, pp. 1916–1923.

[80] K. Patwardhan, G. Sapiro, and V. Morellas, “Robust foreground detection in video using pixel layers,” *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 30, no. 4, pp. 746–751, Apr. 2008.

[81] B. Phong, “Illumination for computer generated pictures,” *Communications ACM*, vol. 18, no. 6, pp. 311–317, 1975.

[82] Y. Raja, S. McKenna, and S. Gong, “Colour model selection and adaptation in dynamic scenes,” in *Proc. European Conf. Computer Vision*, 1998, pp. 460–474.

[83] C. Ridder, O. Munkelt, and H. Kirchner, “Adaptive background estimation and foreground detection using Kalman-filtering,” in *Proc. Int’l. Conf. Recent Advances in Mechatronics*, 1995, pp. 193–199.

[84] J. Rittscher, J. Kato, S. Joga, and A. Blake, “A probabilistic background model for tracking,” in *Proc. European Conf. Computer Vision*, 2000.

[85] N. Rota and M. Thonnat, "Video sequence interpretation for visual surveillance," in *Proc. IEEE Workshop Visual Surveillance*, Dublin, IL, Aug. 6 2000, pp. 325–332.

[86] Y. Rubner, C. Tomasi, and L. Guibas, "The earth mover's distance as a metric for image retrieval," *Int'l. J. Comp. Vision*, vol. 40, no. 2, pp. 91–121, 2000.

[87] M. Sato and S. Ishii, "Online EM algorithm for the normalized Gaussian network," *Neural Computation*, vol. 12, pp. 407–432, 1999.

[88] M. Seki, H. Fujiwara, and K. Sumi, "A robust background subtraction method for changing background," in *Proc. IEEE Workshop Applications of Computer Vision*, 2000, pp. 207–213.

[89] M. Seki, T. Wada, H. Fuliwara, and K. Sumi, "Background subtraction based on cooccurrence of image variations," in *Proc. IEEE Int'l. Conf. on Comp. Vision, Pattern Recog.*, vol. 2, Madison, WI, USA, June 18-20 2003, pp. 65–72.

[90] Y. Sheikh and M. Shah, "Bayesian modeling of dynamic scenes for object detection," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 27, no. 11, pp. 1778–1792, Nov. 2005.

[91] E. Simoncelli, "Bayesian multi-scale differential optical flow," *Handbook of Computer Vision and Applications, Academic Press*, vol. 2, pp. 397–422, 1999.

[92] K. Skifstad and R. Jain, "Illumination independent change detection from real world image sequence," *Computer Vision, Graphics, Image Process.*, vol. 46, pp. 387–399, 1989.

[93] C. Stauffer and W. Grimson, "Adaptive mixture models for real-time tracking," in *Proc. IEEE Int'l. Conf. on Comp. Vision, Pattern Recog.*, vol. 2, Fort Collins, CO, USA, June 23-25 1999.

[94] ——, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 22, no. 8, pp. 747–757, Aug. 2000.

[95] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: Principles and practice of background maintenance," in *Proc. IEEE Int'l. Conf. Computer Vision*, Kerkyra, Greece, Sep. 20-27 1999, pp. 255–261.

[96] H. Ukida, K. Konishi, T. Wada, and T. Matsuyama, "Recovering shape of unfolded book surface from a scanner image using eigenspace method," in *Proc. of IAPR Workshop on Machine Vision and Applications*, 2000, pp. 463–466.

[97] Y. Wang, K.-F. Loe, and J.-K. Wu, "A dynamic conditional random field model for foreground and shadow segmentation," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 28, no. 2, pp. 279–289, Feb. 2006.

[98] J. Weng, Y. Zhang, and W.-S. Hwang, "Candid covariance-free incremental principal component analysis," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 25, no. 8, pp. 1034–1040, Aug. 2003.

[99] N. Wiener, *Extrapolation, interpolation, and smoothing of stationary time series, with engineering applications*, 1st ed., ser. 9. Cambridge, MA: The M.I.T. Press, Aug. 1964.

[100] P. Withagen, K. Schutte, and F. Groen, "Likelihood-based object detection and object tracking using color histograms and EM," in *Proc. IEEE Int'l. Conf. Image Process.*, 2002, pp. 589–592.

[101] L. Wixson, "Detecting salient motion by accumulating directionally-consistent flow," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 22, no. 8, pp. 774–780, Aug. 2000.

[102] C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 19, no. 7, pp. 780–785, July 1997.

[103] Y. Yakimovsky, "Boundary and object detection in real world images," *J. ACM*, vol. 23, pp. 599–618, 1976.

[104] Q. Zang and R. Klette, "Robust background subtraction and maintenance," in *Proc. IEEE Int'l. Conf. on Pattern Recog.*, vol. 2, 2004, pp. 90–93.

[105] L. Zhao and L. Davis, "Iterative figure-ground discrimination," in *Proc. IEEE Int'l. Conf. on Pattern Recog.*, Aug. 23-24 2004, pp. 67–70.

[106] B. Zhong, S. Liu, and H. Yao, "Local spatial co-occurrence for background subtraction via adaptive binned kernel estimation," in *Proc. Asian Conf. Computer Vision*, vol. 5996, 2010, pp. 152–161.

[107] J. Zhong and S. Sclaroff, "Segmenting foreground objects from a dynamic textured background via a robust Kalman filter," in *Proc. IEEE Int'l. Conf. Computer Vision*, Nice, France, Oct. 13-16 2003, pp. 44–50.

[108] D. Zhou and H. Zhang, "Modified GMM background modeling and optical flow for detection of moving objects," in *Proc. IEEE Int'l. Conf. on Systems, Man, Cybernetics*, vol. 3, Jan. 2006, pp. 2224–2229.

[109] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. IEEE Int'l. Conf. on Pattern Recog.*, vol. 2, 2004, pp. 28–31.

[110] Z. Zivkovic and F. van der Heijden, "Recursive unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal., Machine Intel.*, vol. 26, no. 5, pp. 651–656, May 2004.

Appendix

Appendix A

PseudoCode

A.1 Representation

The following list of data storage arrays and constants are used in the pseudocode to describe the scene model proposed in this dissertation.

$I_k[rows][columns]$ - A single video frame at time k.

$M[rows][columns][samples]$ - The background model.

$samples$ - The total number of samples.

$frames$ - The total number of frames.

R - Spherical kernel radius.

T - Segmentation threshold.

$L_k[rows][columns]$ - The labeled image.

A.2 Initialization

Initialization is performed by iterating over a short number of frames at the beginning of the video sequence and blindly assigning the values to the sample collections at the corresponding locations in the model M. The length of the frame initialization sequence is equal to the number of samples used to characterize the distribution of values at each location within the background model. The pseudocode for the initialization procedure can be found in Algorithm 1.

Algorithm 1 Scene Model Initialization Algorithm.

```
for $k = 1 \rightarrow samples$ do
    for $i = 1 \rightarrow rows$ do
        for $j = 1 \rightarrow columns$ do
            $M[i][j][k] \leftarrow I_k[i][j]$
            $j \leftarrow j + 1$
        end for
        $i \leftarrow i + 1$
    end for
    $k \leftarrow k + 1$
end for
```

A.3 Segmentation

Segmentation is performed by iterating over the spatial coordinates of an unsegmented video frame I_k and computing the background probability of each grayscale value $I_k[row][column$ with respect to the corresponding background model sample collection $M[row][column][samples]$ using a spherical cutoff kernel of radius R. The result of segmentation is a label image L_k, where each location within the label image is either *Foreground* or *Background*. The segmentation pseudocode is depicted in Algorithm 2.

Algorithm 2 Scene Model Frame Segmentation Algorithm.

```
for $k = samples + 1 \rightarrow frames$ do
  for $i = 1 \rightarrow rows$ do
    for $j = 1 \rightarrow columns$ do
      $Sum \leftarrow 0$
      for $l = 1 \rightarrow samples$ do
        if $I_k[i][j] - M[i][k][l] \leq R$ then
          $Sum \leftarrow Sum + 1$
        end if
        $l \leftarrow l + 1$
      end for
      if $Sum \geq T$ then
        $L_k[i][j] = Background$
      else
        $L_k[i][j] = Foreground$
      end if
      $j \leftarrow j + 1$
    end for
    $i \leftarrow i + 1$
  end for
  $k \leftarrow k + 1$
end for
```

A.4 Maintenance

Maintenance is performed by iterating over the label image L_k and integrating values from the current frame I_k into the model in the case where they have been labeled as $Background$. The outlying value within the background model $M[row][column][OutlierIndex]$ is replaced with the new background value $I_k[row][column$. The outlier is identified by iterating over the sample collection $M[row][column][samples]$ and computing the probability of each sample using a spherical cutoff kernel with radius V and then taking the index of the minimum value to be the $OutlierIndex$. The kernel radius V is estimated from the sample collection $M[row][column][samples]$ by computing the absolute median deviation between all of the possible sample pairs, excluding pairs of identical samples. Based on a random 1/16 chance, the new background value is propagated to a neighboring distribution. The probability of selecting each neighbor is set by assigning a weight to each neighbor based on its similarity to the current spatial location. Similarity between the sample collections is measured by computing the normalized cross correlation between 256 bin histograms of each model. The pseudocode of the model update policy is presented in Algorithm 3, the outlier identification instructions can be found in Algorithm 4 and the neighborhood similarity measurement is illustrated in Algorithm 5.

Algorithm 3 Scene Model Maintenance Algorithm.

```
for $i = 1 \rightarrow rows$ do
  for $j = 1 \rightarrow columns$ do
    if $L_k[i][j] = Background$ then
      $OutlierIndex \leftarrow LocateOutlier(M[i][j])$
      $M[i][j][OutlierIndex] \leftarrow I_k[i][j]$
      if $i > 1$ and $i < rows - 1$ and $j > 1$ and $j < columns - 1$ then
        if $UniformRandomInt(0, 15) == 0$ then
          $Weight[1] = MeasureSimilarity(M[i][j], M[i-1][j-1])$
          $Weight[2] = MeasureSimilarity(M[i][j], M[i-1][j])$
          $Weight[3] = MeasureSimilarity(M[i][j], M[i-1][j+1])$
          $Weight[4] = MeasureSimilarity(M[i][j], M[i][j+1])$
          $Weight[5] = MeasureSimilarity(M[i][j], M[i+1][j+1])$
          $Weight[6] = MeasureSimilarity(M[i][j], M[i+1][j])$
          $Weight[7] = MeasureSimilarity(M[i][j], M[i+1][j-1])$
          $Weight[8] = MeasureSimilarity(M[i][j], M[i][j-1])$
          $WeightSum \leftarrow 0$
          for $neighbor = 1 \rightarrow 8$ do
            $WeightSum \leftarrow WeightSum + Weight[neighbor]$
          end for
          $Temp = UniformRandomFloat(0, WeightSum)$
          $WeightSum \leftarrow 0$
          for $neighbor = 1 \rightarrow 8$ do
            if $WeightSum \leq Temp < WeightSum + Weight[neighbor]$
            then
              $NeighborSelection = neighbor$
            end if
            $WeightSum \leftarrow WeightSum + Weight[neighbor]$
          end for
          $(nrow, ncol) = DecodeCoordinates(NeighborSelection)$
          $OutlierIndex \leftarrow LocateOutlier(M[nrow][ncol])$
          $M[nrow][ncol][OutlierIndex] \leftarrow I_k[i][j]$
        end if
      end if
    end if
    $i \leftarrow i + 1$
  end for
  $k \leftarrow k + 1$
end for
```

Algorithm 4 Scene Model Outlier Identification Function.

```
$ListIndex = 1$
for $i = 1 \rightarrow samples$ do
    for $j = 1 \rightarrow samples$ do
        if $i \neq j$ then
            $List[ListIndex] \leftarrow |M[row][column][i] - M[row][column][i]|$
            $ListIndex \leftarrow ListIndex + 1$
        end if
        $j \leftarrow j + 1$
    end for
    $i \leftarrow i + 1$
end for
$Sort(List)$
if $samples$ is odd then
    $V = List(Floor(ListIndex/2))$
else
    $V = (List(ListIndex/2) + List((ListIndex/2) + 1))/2$
end if
$SampleProbability \leftarrow zeros(1, samples)$
for $i = 1 \rightarrow samples$ do
    for $j = 1 \rightarrow samples$ do
        if $|M[row][column][i] - M[row][column][j]| < V$ then
            $SampleProbability[i] \leftarrow SampleProbability[i] + 1$
        end if
        $j \leftarrow j + 1$
    end for
    $i \leftarrow i + 1$
end for
$MinProb \leftarrow samples$
$OutlierIndex \leftarrow null$
for $i = 1 \rightarrow samples$ do
    if $SampleProbability[i] < MinProb$ then
        $MinProb \leftarrow SampleProbability[i]$
        $OutlierIndex \leftarrow i$
    end if
    $i \leftarrow i + 1$
end for
return $OutlierIndex$
```

Algorithm 5 Scene Model Similarity Measurement Function.

```
$Hist_a, Hist_b \leftarrow zeros(1, 256)$
for $i = 1 \rightarrow samples$ do
    $Bin \leftarrow M[row][column][sample]$
    $Hist_a[Bin] \leftarrow Hist_a[Bin] + 1$
    $Hist_b[Bin] \leftarrow Hist_b[Bin] + 1$
end for
$Numerator, Denominator \leftarrow 0$
for $i = 1 \rightarrow samples$ do
    $Numerator \leftarrow Numerator + Hist_a[i] \cdot Hist_b[i]$
    $Denominator \leftarrow Numerator + Hist_b[i] \cdot Hist_b[i]$
end for
if $Denominator == 0$ then
    return $0$
else
    return $Numerator/Denominator$
end if
```

Algorithm 6 Scene Model Spatial Coordinate Decode Function.

```
if $NeighborSelection == 1$ then
    $nrow = i - 1$
    $nrow = j - 1$
else if $NeighborSelection == 2$ then
    $nrow = i - 1$
    $nrow = j$
else if $NeighborSelection == 3$ then
    $nrow = i - 1$
    $nrow = j + 1$
else if $NeighborSelection == 4$ then
    $nrow = i$
    $nrow = j + 1$
else if $NeighborSelection == 5$ then
    $nrow = i + 1$
    $nrow = j + 1$
else if $NeighborSelection == 6$ then
    $nrow = i + 1$
    $nrow = j$
else if $NeighborSelection == 7$ then
    $nrow = i + 1$
    $nrow = j - 1$
else
    $nrow = i$
    $nrow = j - 1$
end if
return $(nrow, ncol)$
```