sensors

MDPI

Article

Moving Object Detection Using Scanning Camera on a High-Precision Intelligent Holder

Shuoyang Chen [1], Tingfa Xu [1,2,*], Daqun Li [1], Jizhou Zhang [1] and Shenwang Jiang [1]

[1] School of Optoelectronics, Image Engineering & Video Technology Lab, Beijing Institute of Technology, Beijing 100081, China; 2120140517@bit.edu.cn (S.C.); ldq_1221@163.com (D.L.); xiaomianzhou@126.com (J.Z.); jiangwenj02@163.com (S.J.)

[2] Key Laboratory of Photoelectronic Imaging Technology and System, Ministry of Education of China, Beijing 100081, China

* Correspondence: ciom_xtf1@bit.edu.cn; Tel.: +86-10-6891-2567

Academic Editor: Vittorio M. N. Passaro
Received: 3 August 2016; Accepted: 20 September 2016; Published: 21 October 2016

Abstract: During the process of moving object detection in an intelligent visual surveillance system, a scenario with complex background is sure to appear. The traditional methods, such as "frame difference" and "optical flow", may not able to deal with the problem very well. In such scenarios, we use a modified algorithm to do the background modeling work. In this paper, we use edge detection to get an edge difference image just to enhance the ability of resistance illumination variation. Then we use a "multi-block temporal-analyzing LBP (Local Binary Pattern)" algorithm to do the segmentation. In the end, a connected component is used to locate the object. We also produce a hardware platform, the core of which consists of the DSP (Digital Signal Processor) and FPGA (Field Programmable Gate Array) platforms and the high-precision intelligent holder.

Keywords: moving object detection; background modeling; intelligent visual surveillance

1. Introduction

Moving object detection is an important part of computer vision filed. All of the identification and tracking parts afterwards depend on a precise and robust detection result, which makes the moving object detection step a really important portion [1–5]. Moving object detection can be used in many scenarios, such as intelligent video surveillance, motion analysis, and human–machine interface applications, therefore deep research of this topic is meaningful [6–10].

Researchers usually employ some sensors to gain information for the processing system [11–13]. In this paper, a scanning moving object detection system is used with a scanning high definition camera as its video image sensor. High-definition camera is widely used in both indoor and outdoor video surveillance systems. We can get far more than one field of view since we made our video image sensor scan around with the support of the holder mentioned before. This means we can gain richer scenes information than just using a high-definition camera. In this paper, the main work we have done is detecting one or more moving objects from the video sequence gained by a high definition camera rotating around using an intelligent holder. Scenarios including tracking and positioning of fire, personnel and vehicles are all applicable.

A surveillance system can be defined as a technological tool that assists humans by providing an extended perception and reasoning capability about situations of interest that occur in the monitored environments [14]. Human perception and reasoning are constrained by the capabilities and limits of human senses and mind to simultaneously collect, process, and store limited data.

Visual surveillance research focuses on intelligent visual surveillance (IVS) in a wide area. Research trends in IVS can be divided largely into image interpretation and wide area surveillance control techniques. The goal of image interpretation is to extract high-level information of a visual event from a dynamic scene [15]. Image interpretation often includes motion detection, object recognition, tracking, and behavior understanding. In this paper, the function of our moving object detecting system (belonging to IVS) is to detect the moving object from a complex outdoor background.

We put a high-definition camera on a High-precision Intelligent Holder. The positional accuracy of the presetting point achieves 0.1°. Reliable repetitive precision is available for multiple times of access. It is specially designed for continuous operation. The worm and gear driving device and stepper motor ensure the long service life and zero-deflection operation. Minimum speed 0.01°/s and maximum speed 25°/s satisfy different requirements of long-range and short-range observation. The data echo of focal length of lens and holder angle is supported, which is convenient for the back end to carry out intelligent analysis. A stabilized gyro connector is provided at the holder, which, together with a gyro instrument, will ensure the stable image in the on-vehicle, on-ship and on-plane conditions. When the holder rotates around in a given rule, the camera can gain a complete image of these areas.

There are some traditionally and widely used algorithms including frame difference method, traditional background subtraction method, optical flow method and statistical learning method [16]. The previous two methods only work on a stationary background. However, our research is on a dynamic background video, which is gained by a camera, meaning those two methods cannot be used directly. In this paper, as we build a camera scanning system, an aspect we thought about is the real-time performance of this system, which means that the computing complexity of our method should not be too high. Now, with reference to the basic method above, we can find that both optical flow [17–19] and statistical learning method [20–22] have a large amount of calculation. The edge detection method does not cost much computation time. In addition, the LBP (Local Binary Pattern) algorithm we use has less computational complexity and does not have complex iterative process. Even combined with the edge detection process, our method could still meet the real-time requirement. In addition, real-time surveillance systems, especially in the outdoor condition, have a high background complexity; that is, when the object detection method proceeds, we should consider the background complexity. Considering the illumination variation of the complex background, we will use the edge detection and the LBP algorithm together to dilute the influence. As mentioned above, moving object detection is a key part of a computer vision system and provides initial information about the object; thus, we can extract as much information as possible. When obtaining the edge differencing image, we cannot get as much information as we need. To avoid this, we perform background modeling and background subtraction to obtain more data from the moving object that needs to be extracted from the video. After that, we bring the results together and generate the final result. To achieve the real-time requirement, we also did some optimization work on the program. Though our method does not have a very high frame frequency, the processing speed at 19–35 fps can still achieve the real-time requirement.

In view of the four methods above, which are based on a stationary background, the video sequence we get from our video image sensor has a dynamic background. The proposed method uses LBP operator and edge detection is processed to do the background modeling and moving object detection work. From the above analysis, we can find that the exiting algorithms have drawbacks on the scenario with illumination variations, dynamic background and video noise. Our method can solve these three problems well by using the LBP operator and the speed of this detection process will be improved.

In this paper, we first use edge detection to get an edge difference image. After that, we conduct the multi-block temporal-analyzing LBP method and combine edge detection to do the moving object detection work. To elaborate these, we will discuss the related work in Section 2. The moving object detection work is shown in Section 3 (including edge detection, background modeling and object locating. The experiment and analysis are presented in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

There is an evaluation of background subtraction provided by Brutzer et al. [23]. In this section, we introduce some recent algorithms to do background modeling and moving object detection, which is the most related work to ours.

2.1. Intelligent Visual Surveillance

Intelligent visual surveillance improves conventional passive surveillance systems through automated object recognition and tracking, scene interpretation and indexing/retrieval of visual events. Visual surveillance techniques have initiated a wide variety of applications in access control, person specific identification, anomaly detection and alarming in academic community as well as industry and government [24]. Large research projects on visual surveillance have driven realization of practical visual surveillance systems. Successful visual surveillance systems such as the Visual Surveillance and Monitoring (VSAM) [25], the Annotated Digital Video for Intelligent Surveillance and Optimized Retrieval (ADVISOR) [26], and the Smart Surveillance System of IBM [27] have been developed by combining computer vision, system engineering, and communication techniques. Wang et al. [28] show an intelligent multi-camera video surveillance which emphasizes the connection and integration of different modules. Song et al. [29] present a comprehensive survey of recent research results to address the problems of intra-camera tracking, topological structure learning, target appearance modeling, and global activity understanding in sparse camera networks.

2.2. Moving Object Detection

Moving object detection and background modeling work always have to face these five major challenges: illumination changes, dynamic background, camouflage, bootstrapping and video noise. Most of the background modeling work starts around these challenges. In recent years, researchers explored some work on these problems. Cheng et al. [30] proposes a novel hybrid background subtraction technique to solve the bootstrapping problem. Huang et al. [31] have presented a novel motion detection approach based on radial basis function artificial neural networks to accurately detect moving objects not only in dynamic scenes but also in static scenes. Chen et al. [32] have presented an approach that is applicable not only in high bit-rate video streams but also in low bit-rate video streams. Cao et al. [33] have presented a unified framework for addressing the difficulties of dynamic background and irregular object movement, especially the one caused by irregular object movement. Zhang et al. [34] have proposed a new strategy to detect camouflaged moving objects which can identify the truly camouflaged areas. The approaches above all solved one or two problems from the aforementioned challenges. In this paper, the approach MB-TALBP (Multi-Block Temporal-Analyzing Local Binary Pattern) we present can solve three of the five challenges (which are illumination changes, dynamic background and video noise) together.

There are also some classic algorithms that play an important role in the moving object detection field. Stauffer et al. [35] have presented a background mixture model by a mixture of Gaussians and using an on-line approximation to update the model. The existing algorithm of Gaussian mixture background modeling [36–38] performs well in some cases. There are some new algorithms or improved algorithms that are based on the principles of Gaussian mixture algorithm. However, the Gaussian mixture algorithms also have some drawbacks, such as a relatively large amount of calculation, lack pace, and sensitive to illumination variation. The codebook algorithm [39–41], which has good results, is also sensitive to illumination variation. Wang et al. [42,43] have proposed moving target detection algorithm based on the adaptive statistical efficiency, the algorithm is called SACON (SAmple CONsensus). Calculating an estimated background model through each pixel sample consistency is simple, but it still has very strong performance. Comparing to the current algorithms, it can be seen that the algorithm has strong prospects. ViBe algorithm [44,45], which has fast speed and small amount of computation, is very good, but has a certain degree of noise robustness problem.

Background modeling method based on color information [46–48], referred to as color algorithms, that decompose the difference of pixels' Chromaticity and Brightness into differences in illumination has strong robustness, better results and faster speed, and, thus, can meet the basic requirements of real-time. Elgammal et al. [49,50] use kernel density estimation to establish a non-parametric background model. For each pixel, in order to estimate the potential of the probability density function, the observed intensity value is retained. Then, the probability of the new intensity value is calculated by the function.

2.3. Dynamic Background Modeling

In many video analysis applications, detecting moving objects from a video sequence captured by video sensor is one of the basic works. A common approach for this work is background modeling and subtraction, which first builds a suitable background model, and then the moving object will be extracted and labeled as foreground by the approach named background subtraction. This work of detection of moving objects is difficult to accomplish with the purpose of very high accuracy. The performance of background subtraction depends mainly on the background modeling technique used.

Pixel-based methods [51,52] assume that the time series of observations is independent of each pixel, which is unreasonable and restricts their use in dynamic background. In contrast, there are also some region based methods [53,54]. Similar to the textured-based method used in [55,56], it modeled the background with a group of histograms based on local binary patterns (LBP). This method to some extent can avoid labeling some moving background pixels as foreground since it extracts region texture features, which means it can be applied to the dynamic background scenario.

LBP algorithm has good performance during sudden changes in light conditions, LBP texture feature for maintaining robust background and foreground can be well positioned location and outline proposed moving object detection algorithm for robust to changes in illumination. Heikkila et al. [55,57] have done much research on the LBP algorithm including the region based method and the representation of LBP feature on a circular region. Since the LBP algorithm has some advantages, researchers in this filed have developed many impressive embranchments of it. Zhao et al. [58] have proposed an method called VLBP (Volume Local Binary Patterns) which has two drawbacks: (1) it can only be used offline because of the requirement that the whole video sequence must be acquired before computing; and (2) it considers too many pixels lead to a time-consuming counting process. Zhang et al. [56] have proposed an algorithm named STLBP (Spatio-Temporal Local Binary Patterns).

3. Moving Object Detection

In this section, we present the proposed procedure for moving object detection. To achieve better detection than previous standalone approaches, our proposed scheme uses both background subtraction and edge detection methods simultaneously. Our method is able to extract abundant information from the object both inside and along the edge by using two detection methods (see Figure 1).

Two consecutive frames are used to generate an edge difference image. The concrete steps are: first, do the edge detection on each of the images (which can generate two edge detected images), then do the frame difference with those two images that will generate a binary edge difference image. Our background modeling method (MB-TALBP) is used to generate background image and do the background subtraction with the current frame. After the step of background subtraction, we can also gain a binary image of the foreground. We let both of these two binary images do logic "AND" operation to generate a final difference image, which is the one we use to locate the object.

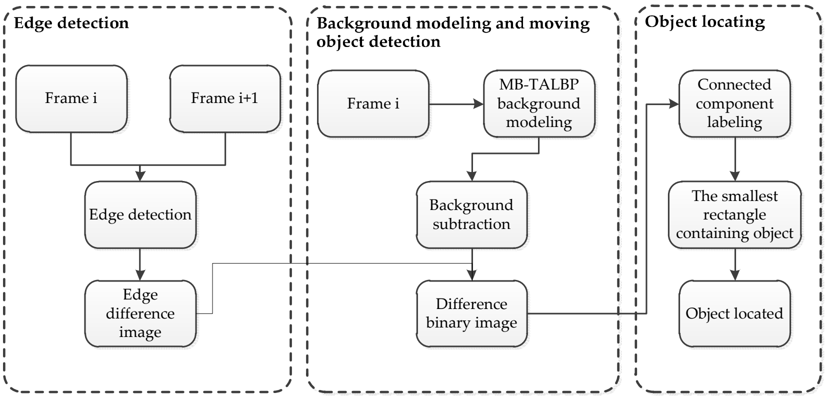


Figure 1. The flow chart of the proposed scheme.

3.1. Moving Edge Detection

Moving edge is the edge of moving objects. Difference image can be regarded as time gradient, while edge image is space gradient. Moving edge can be defined by the logic AND operation of difference image and the edge image [59]. The advantage of frame difference method is its small calculation, and the disadvantage is that it is sensitive to noise, for which another procedure we propose, background subtraction, can provide a partial solution. If the objects do not move but the brightness of the background changes, the results of frame difference methods may not be accurate enough. In order to avoid the weakness (sensitive to the illumination) and enhance its advantage (small calculation), we use the edge detection method to do the same part of work, which can overcome the disadvantage of frame difference method.

In this paper, we use canny operator to do the edge detection. Overall, the edge detection of the image must meet two steps: first, effectively suppress noise, by Gaussian operator image smoothing; and second, try to determine the exact position of the edge. Canny operator edge detection can be divided into four steps: (1) Gaussian smoothing function for the purpose of smoothing to remove noise; (2) first difference convolution template, to achieve edge enhancement; (3) non-maxima suppression (NMS), intended to preserve the maximum gradient direction (The third one is the most important step in this process); and (4) bilinear threshold, to get the edge points in order to determine whether the second step of its 80 neighborhood is obtained, and then make the connection.

3.2. Background Modeling

In this paper, we use LBP operator to do the background modeling. There are several advantages of this operator. First, when come into those complex scenarios, such as swaying trees, undulating lake, and flashing monitor, during these situations, in the background there is a new object that enters or older objects are removed, there is a good adaptability. Second, comparing to other common algorithm, such as GMM, this method cause less computational time. Third, this operator also has a good inhibition for noise.

Although the LBP operator has a good performance on background modeling, there are still some work that cannot be accomplished appropriately. We use an improved LBP operator, which combined with "multi-scale block LBP" [60] and "Spatio-temporal LBP" [56], do the modeling work so that the motion and texture information on space and time is on the binding. The computation is done based on average values of block sub regions, instead of individual pixels. In this way, the method presents several advantages: (1) it not only contains the connection between microstructures but also macrostructures of the image being processed, and then provides a more complete image representation than the basic LBP operator; and (2) this method can be computed very efficiently using integral images. Furthermore, in order to reflect the uniform appearance of LBP, we redefine the uniform patterns via statistical analysis.

While using LBP operator, we confirm the size of our operator, which, in the basic method, is 3 × 3. In this paper, we use lager operator to get some more information from the macrostructures, whose size is 9 × 9. This operator is composed of nine 3 × 3 pixel blocks. In each sub-region, average sum of image intensity is computed. These average sums are then compared with the gray value of the center block. After that, the LBP method is used to do the background modeling work.

We use the following equation to do the comparing work between the center pixel and the pixels from its neighborhood, which is the basic equation of a LBP modeling method.

$$L\left(x_c, y_c\right) = \sum_{p=0}^{7} s(g_p - g_c)2^p \tag{1}$$

where g_c corresponds to the grey value of the central pixel (x_c, y_c) and g_p to the grey values of the eight neighboring pixels. p is the number of neighboring pixels, which is 8 here. The function $s\left(x\right)$ is defined as follows:

$$s\left(x\right) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{2}$$

The result of computing this equation is counted as the LBP feature of one pixel or block, which is the center part of the operator. LBP features are robust to monotonic gray value changes and very fast to compute, which are very important for background modeling and subtraction. As mentioned above, achieving "real-time" requirement is of great importance to our system.

The form of the LBP operators are shown in Figure 2; the 9 × 9 sized blocks consist of many 3 × 3 blocks. The 3 × 3 blocks are the base unit of all the different sized operators and the size of the operators can be 3 × 3, 9 × 9, and even 27 × 27. On the basis of our research, the operators that are larger than 9 × 9 are not very helpful to the background modeling work. For these reasons, we only use 3 × 3 and 9 × 9 sized operators to do the background modeling work. After using this expended LBP operator and obtaining the coded feature of all the blocks of the processing image, we can integrate the entire feature into a histogram that describes the background feature of this scenario. The comparison between two images is processed by histogram intersection [61]. The equation below shows how the histogram intersection works.

$$\cap\left(a, b\right) = \sum_{n=0}^{N-1} \min\left(a_n, b_n\right) \tag{3}$$

where "a" and "b" are the histograms generated by 3 × 3 and 9 × 9 operators, respectively. N is the number of histogram dimensions. This measure has an intuitive motivation in that it calculates the common part of two histograms. Its advantage is that it explicitly neglects features that only occur in one of the histograms. The complexity is linear for the number of histogram dimensions. It is also possible to use other measures but this histogram intersection method is a more practical and efficient one.

After computing the single frame data of the gained video, the connection between two continuous frames are embodied by the method that we named as "temporal-analysis LBP". The equations of this algorithm are as follows:

$$L_{P,R}^{t}\left(x_{t,c}, y_{t,c}\right) = \sum_{p=0}^{P-1} s\left(g_{t,p} - g_{t,c}\right) 2^P \tag{4}$$

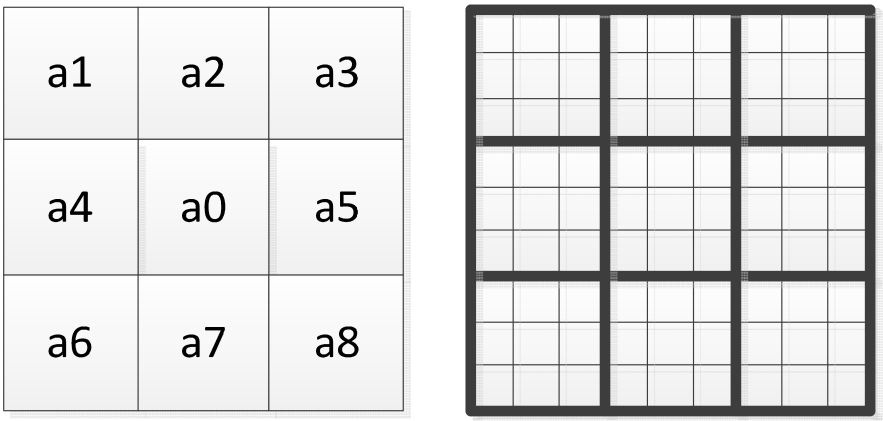
$$L_{P,R}^{t-1}\left(x_{t-1,c}, y_{t-1,c}\right) = \sum_{p=0}^{P-1} s\left(g_{t-1,p} - g_{t-1,c}\right) 2^P \tag{5}$$

$(x_{t,c}, y_{t,c})$ and $(x_{t-1,c}, y_{t-1,c})$ are the central pixel of current and previous frames. $L_{P,R}^{t}\left(x_{t,c}, y_{t,c}\right)$ and $L_{P,R}^{t-1}\left(x_{t-1,c}, y_{t-1,c}\right)$ are the spatial and temporal local binary patterns of pixel located in $(x_{t,c}, y_{t,c})$ and $(x_{t-1,c}, y_{t-1,c})$. R is the label of the size of the LBP blocks (which is "3, 9" here), and is not involved in computations. The former extracts the spatial texture features and the latter extracts the motion information of the neighboring two frames. We can compute two histograms H_t and H_{t-1} over this region. The equation to compute those two histograms are shown below.

$$H_t = \omega H_{t-1,i} + (1-\omega)\, H_{t,i}, \qquad i = 0, \ldots, 2^P - 1 \tag{6}$$

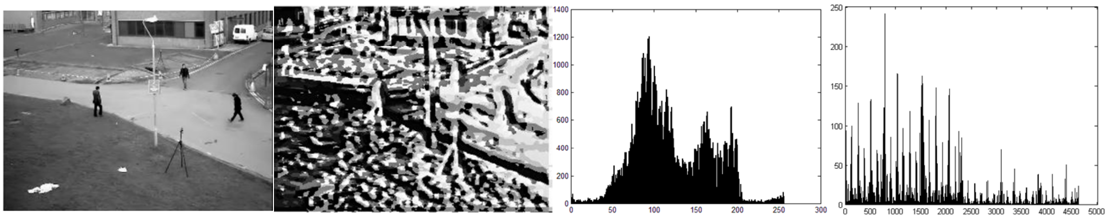
The equation above shows the summing operation between two bins corresponding to each histogram. The parameter "ω" of this temporal-analysis method are determined by the influence the previous frame exerts on the current frame, which we defined elementarily as 0.5.

Figure 3 show the image processed by LBP algorithm, which embodied rich texture information.



(**a**) (**b**)

Figure 2. (**a**) The basic 3 × 3 operator of the LBP algorithm; and (**b**) multi-scale LBP (Local Binary Pattern) based on blocks of 9 × 9 size. Computation is done based on average values of block sub regions, instead of individual pixels. In each sub-region, average sum of image intensity is computed. These average sums are then compared with the gray value of the center block.



(**a**) (**b**) (**c**) (**d**)

Figure 3. (**a**) The gray scale image of the original image; (**b**) the texture image of the background; (**c**) the histogram of the original image; and (**d**) the LBP histogram of the background image, which shows extensive information of the texture.

3.3. Object Locating

After the background modeling work has been done, we can easily extract the moving object from the frames. Although the object has been extracted, we still need to label the object by a rectangle for visual display. In this paper, we use connected component labeling to do the object locating work.

This process is to label the connected components in binary images, delete those connected components whose area are too small (false positives and noise), and get rounding rectangle of the object. In this section, our method is the following steps:

1 Progressive scan the images: Put each line in a continuous white pixels form a sequence called a "group", and note down its starting point, end point and the line number.

2 Except for the first row of all the rows: If it has no overlap area with the previous row, then give this row a new tab; if it has only one overlap area with previous row, then give it the same tab with the above one; if it has two or more overlap areas with the above area, then give it the smallest number of those areas and note down all these groups into an equivalent pair, indicating that they belong to the same class.

3 Convert the equivalent pair to equivalent sequences: Give each sequence the same reference numeral, since it is equivalent. Starting from Step 1, we give each equivalent sequence a tab.

4 Traverse the groups from the beginning one: Find each equivalent sequence, and give it a new tab.

5 Fill the label of each group in the marked image.

Once these steps have been done, we can get every eligible labeled area, meaning we can finally locate the object extracted from the image.

4. Experiment and Analysis

In order to evaluate our method and compare it with other algorithms, we conduct this experiment using some video sequences. In this section, we first introduce our experiment setup in Section 4.1, including the detailed introduction of the hardware platform we used and the "high-precision intelligent holder" that we mentioned in the title of this paper. We then show the result of software simulation using those result images in Section 4.2, which is for a visual comparison. Finally, the quantitative analyses of our experiments are shown in Section 4.3 by utilizing several tables.

4.1. Experimental Setup

To prove the ability of our method in background subtraction step, we mainly focus on the performance of detection methods on situations lacking training opportunities, including situations with illumination change and a dynamic background. In the experiments, we first use the video sequences to test the ability of each method in dealing with situations, such as those mentioned above. We then use the sequences to verify the capability of each method in coping with those situations. First, we select five test video sequences, each moving objects in its initial frames and the continuous flow of moving objects throughout the sequence.

The hardware we put in use is combined with a camera, a platform with FPGA and DSP, a high-precision intelligent holder. All of our hardware platform components interact with each other and accomplish the detection work together. We show each of the portions in Figure 4 with some material images that will make a visual introduction of the whole system. The camera is treated as a main video sensor.

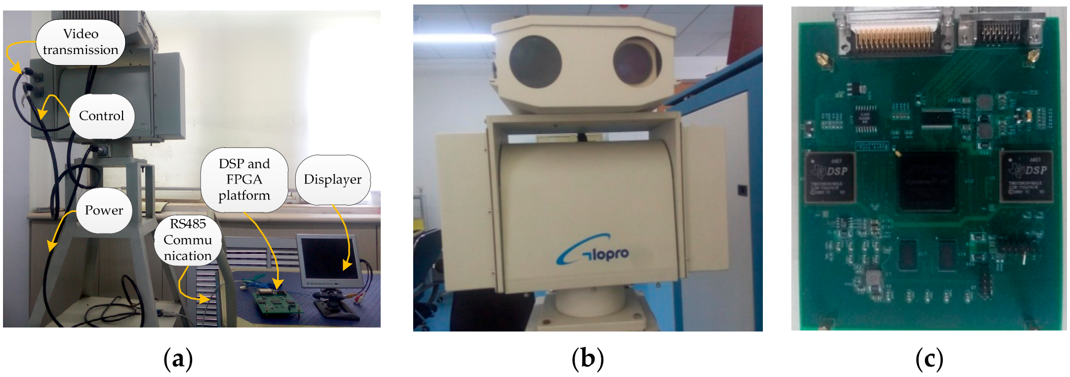


Figure 4. (**a**) Our moving object detecting system and the description of each component; (**b**) the high-precision intelligent holder; and (**c**) the platform of FPGA (Field Programmable Gate Array) and DSP (Digital Signal Processor).

4.1.1. Methods for Comparison

In this section, we introduce the methods we used to compare with our algorithm below:

1 GMM (Gaussian Mixture Modeling) is a background modeling based approach, which is based on each pixel in the time domain to build the distribution model of each pixel sequentially to achieve the background modeling purposes [36–38]. Gaussian mixture background model is a weighted finite number of Gaussian functions which can describe the state of the pixel multimodal, suitable for light gradient, swaying trees and other complex background accurately modeled. Through continuous improvement of many researchers, the method has become the most common background extraction method.

2 ViBe (Visual Background extractor) is a high-efficiency algorithm. This algorithm adopts a new thought to detect targets using random principles in the object detection work. The basic idea is, for each pixel, random sampling radius R within the scope of the model as a background pixel, and the default is 20 sampling points. Compared to some other detection algorithms, ViBe has a small amount of calculation, small footprint, fast processing speed, good detection effect, faster speed and the ablation area of Ghost stable and reliable characteristics respond noise.

3 GMG (an algorithm for finding the Global Minimum with a Guarantee) combines the static background image and each pixel Bayesian estimation division. It uses very little information before (the default is 120 before) the image background modeling. It uses probability prospects estimation algorithm (using a Bayesian estimation to identify prospects). This is an adaptive estimation, the new observed objects have more weight than the old object, which means the results adapt to light changes. Some morphological operations such as opening operation, closing operation, etc. are used to remove unwanted noise.

4 KDE (Kernel Density Estimation) is a well-known moving object detection algorithm. By employing a few frames of the method data, the algorithm can do background modeling with a fast extraction of moving targets in subsequent frames. However, the noise is large and some small moving objects are easily lost. The model is based on estimating the intensity density directly from sample history values. The main feature of the model is that it represents a very recent model of the scene and adapts to changes quickly. A framework was presented to combine a short-term and a long-term model to achieve more robust detection results [49,50].

5 LBAdaptiveSOM (Local Background Adaptive Self-Organizing Modeling) [62] is a self-organizing method for modeling background by learning motion patterns and so allowing foreground/background separation for scenes from stationary cameras. The method is strongly required in video surveillance systems. This method learns background motion trajectories in a self-organizing manner, which makes the neural network structure much simpler. The approach is suitable to be adopted in a layered framework, where operating at region-level, it can improve detection results allowing to more efficiently tackle the camouflage problem and to distinguish moving objects from those that were initially moving and have stopped.

4.1.2. Test Video Sequences

We choose some video sequences from the BMC (Background Models Challenge) dataset including dynamic background, illumination variation, introduction or removal of background objects, noise-corrupted, etc. To be specific, we select five typical crowded scenarios from the BMC (Background Models Challenge) [63] (including a campus, a road, a railway station, a night scenario and a parking lot). For each selected indoor sequence coming from the BMC dataset, manually-labeled ground-truth references are provided. We choose five video sequences of this benchmark dataset to show the visual comparisons in Section 4.2 (each video show one typical frame); after that, we will show the quantitative comparisons in Section 4.3 (experimental data are from the whole sequences). All of the test data are provided with manually-labeled ground-truth references.

4.2. Visual Comparisons

All of the videos we show in this section are from the BMC dataset. Based on the consideration of clarity and representation, we choose five video sequences in the dataset to show the results of these comparative experiments. The experiment results of all five shown video sequences and the rest will be presented in line chart form in the analyzing portion in Section 4.3.

According to the order of appearance, they are: "Wandering students", "Rabbit in night", "Beware of the trains", "One rainy hour" and "Big trucks".

Some representative experimental results are shown in Figures 5–9, the result of KDE algorithm have quite low metrics, because of its poor ability to resist noise disturbance that other methods do not have. As a kind of classic algorithm, GMM algorithm gives some stable but ordinary results. ViBe algorithm, which has fast calculation speed, has similar results with GMM. GMG can generate a good result in some scenarios but in other situations the result can be hard to accept. Through a visual judgment of those images we show in this section, our method always gets the best results.

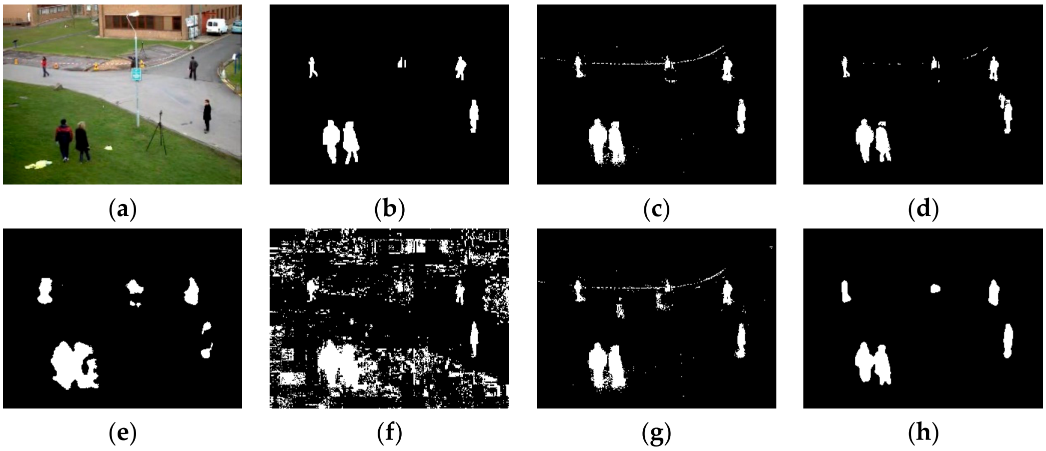


Figure 5. (**a**) Frame 555 of input images; (**b**) the ground truth; (**c**) GMM (Gaussian Mixture Modeling); (**d**) ViBe (Visual Background extractor); (**e**) GMG (an algorithm for finding the Global Minimum with a Guarantee); (**f**) KDE (Kernel Density Estimation); (**g**) LBAdaptiveSOM (Local Background Adaptive Self-Organizing Modeling); and (**h**) our method.

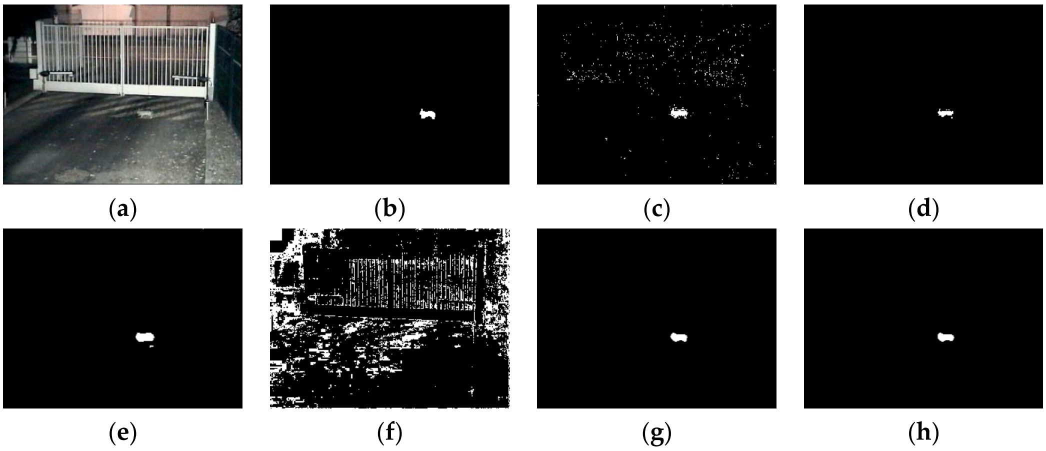
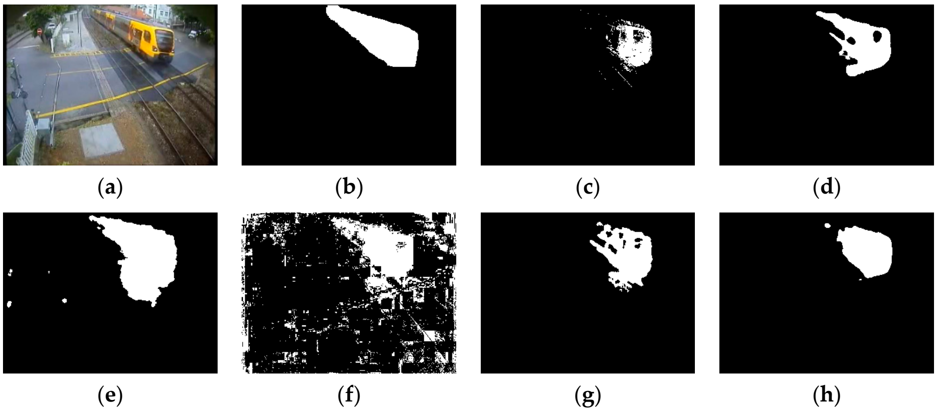


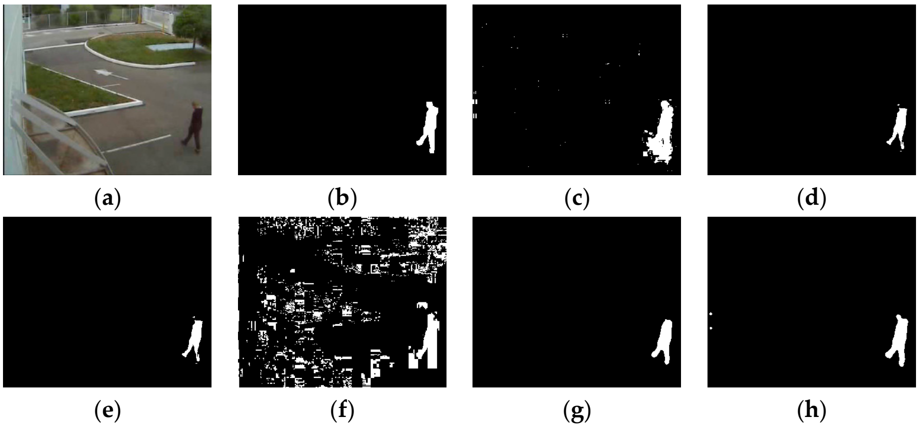
Figure 6. (**a**) Frame 1252 of input images; (**b**) ground truth; (**c**) GMM (Gaussian Mixture Modeling); (**d**) ViBe (Visual Background extractor); (**e**) GMG (an algorithm for finding the Global Minimum with a Guarantee); (**f**) KDE (Kernel Density Estimation); (**g**) LBAdaptiveSOM (Local Background Adaptive Self-Organizing Modeling) ; and (**h**) our method.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

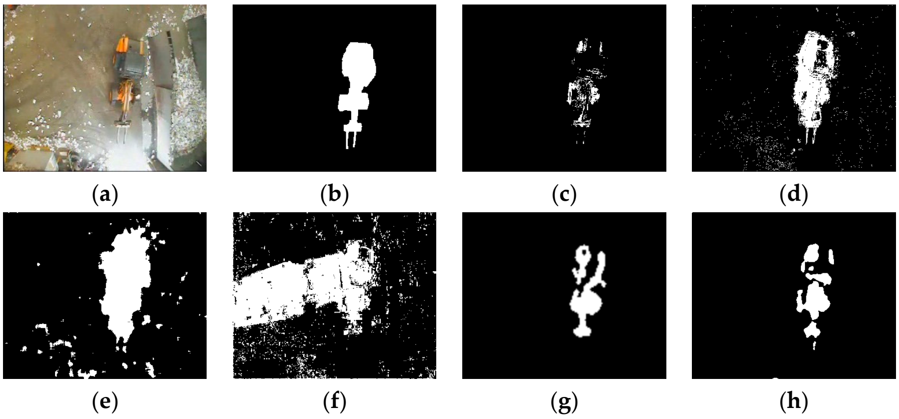
Figure 7. (**a**) is the frame 389 of input images; (**b**) ground truth; (**c**) GMM (Gaussian Mixture Modeling); (**d**) ViBe (Visual Background extractor); (**e**) GMG (an algorithm for finding the Global Minimum with a Guarantee); (**f**) KDE (Kernel Density Estimation); (**g**) LBAdaptiveSOM (Local Background Adaptive Self-Organizing Modeling); and (**h**) our method.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 8. (**a**) is the frame 15553 of input images; (**b**) ground truth; (**c**) GMM (Gaussian Mixture Modeling); (**d**) ViBe (Visual Background extractor); (**e**) GMG (an algorithm for finding the Global Minimum with a Guarantee); (**f**) KDE (Kernel Density Estimation); (**g**) LBAdaptiveSOM (Local Background Adaptive Self-Organizing Modeling); (**h**) our method.



(**a**) (**b**) (**c**) (**d**)

(**e**) (**f**) (**g**) (**h**)

Figure 9. (**a**) is the frame 485 of input images; (**b**) ground truth; (**c**) GMM (Gaussian Mixture Modeling); (**d**) ViBe (Visual Background extractor); (**e**) GMG (an algorithm for finding the Global Minimum with a Guarantee); (**f**) KDE (Kernel Density Estimation); (**g**) LBAdaptiveSOM (Local Background Adaptive Self-Organizing Modeling); (**h**) our method.

Some of the videos selected have a dynamic background, such as "Wandering students" and "One rainy hour"; and some of them have illumination variations, such as "Rabbit in night". When processing a video with dynamic background, some of the algorithms do not perform well. As we can see, in the first sequence, GMM, ViBe, and KDE all regard the dynamic part of background as a detected object and the same situation also shows in other video sequences. Illumination variations also produce some influence of those methods. Similar to the results of the second sequence ("Rabbit in the night"), GMM, GMG, and KDE cannot generate the full object.

4.3. Quantitative Comparisons

To assess the detection results objectively, four indexes (i.e., Recall, Precision, F-measure and Similarity) are put to use by us to evaluate the performance of these techniques at the pixel level. Among these four, TP is the number of true positives, TN represents the number of true negatives, FP means the number of false positives, and FN represents the number of false negatives. The four indexes are defined as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{8}$$

$$\text{F-means} = \frac{\left(\alpha^2 + 1\right)\text{PR}}{\alpha^2\left(\text{P} + \text{R}\right)} \tag{9}$$

$$\text{Similarity} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}} \tag{10}$$

We can discover something that the two indexes "Recall" (R) and "Precision" (P) a contradict each other. When this happens, we use the latter two indexes to achieve an overall consideration. When $\alpha = 1$, F-means $= \text{F} - 1 = \frac{2\text{PR}}{\text{P} + \text{R}}$.

In this section, we mainly do a quantitative comparison of six different moving object detection methods with our method. All the video sequences are the same as the ones used in the previous section for visual comparisons.

We made several tables (Tables 1–5) to provide a more intuitive data form. Each table represents a given video sequence from the dataset.

Table 1. Comparison of indexes of "Wandering students".

Method	Precision	Recall	F-1	Similarity
ViBe	0.7452	0.6134	0.6729	0.4966
GMM	0.7538	0.6085	0.6734	0.4801
GMG	0.8801	0.4521	0.5973	0.4007
LBAdaptiveSOM	0.9426	0.7259	0.8202	0.6847
KDE	0.5112	0.5438	0.4681	0.3274
MB-TALBP	0.9521	0.7372	0.8279	0.7142

Table 2. Comparison of indexes of "Rabbit in the night".

Method	Precision	Recall	F-1	Similarity
ViBe	0.7678	0.8887	0.6130	0.4419
GMM	0.7121	0.8431	0.7721	0.6542
GMG	0.8546	0.4781	0.6159	0.4788
LBAdaptiveSOM	0.9216	0.7249	0.8115	0.6630
KDE	0.3325	0.1716	0.2263	0.1156
MB-TALBP	0.9332	0.7470	0.8298	0.7049

Table 3. Comparison of indexes of "Beware of the trains".

Method	Precision	Recall	F-1	Similarity
ViBe	0.6949	0.5100	0.6286	0.5270
GMM	0.7051	0.5349	0.6083	0.4412
GMG	0.8325	0.6591	0.7357	0.5869
LBAdaptiveSOM	0.8547	0.7026	0.7712	0.5940
KDE	0.5465	0.4692	0.5049	0.3323
MB-TALBP	0.8456	0.6914	0.7807	0.6912

Table 4. Comparison of indexes of "One rainy hour".

Method	Precision	Recall	F-1	Similarity
ViBe	0.8288	0.7600	0.7929	0.6494
GMM	0.8546	0.6527	0.7401	0.6095
GMG	0.8167	0.7533	0.7837	0.6681
LBAdaptiveSOM	0.9254	0.7441	0.8232	0.7354
KDE	0.4267	0.3491	0.3840	0.1729
MB-TALBP	0.9400	0.7529	0.8361	0.7456

Table 5. Comparison of indexes of "Big trucks".

Method	Precision	Recall	F-1	Similarity
ViBe	0.9024	0.7195	0.8006	0.6819
GMM	0.5726	0.4117	0.4764	0.2202
GMG	0.7649	0.4674	0.5802	0.3519
LBAdaptiveSOM	0.7752	0.6429	0.7029	0.5461
KDE	0.1438	0.4194	0.2142	0.0826
MB-TALBP	0.8053	0.6521	0.7206	0.5583

We employ some tables to show the comparison of the indexes mentioned in this section. The tables show the metrics for one typical frame in each video sequence. Clearly, the results of KDE algorithm have quite low metrics, because of its poor ability to resist noise disturbance; and GMM, ViBe, GMG, LBAdaptiveSOM and our method (we call it MB-TALBP) produce much better results in this respect, and MB-TALBP ranks first in the index "Precision" for all videos. Compared to all of the indicators (i.e., F-1 and Similarity), our method performs the best.

The first video has a dynamic background. As shown in Table 1, our method has good performance in this scenario. The other methods in this experiment do not perform very well because these algorithms regard the dynamic background as foreground. This experiment shows the advantage we have in the dynamic background situation.

This video, named "RABBIT IN THE NIGHT", has an illumination change. From the table, we can see that our method has a large advantage compare to the other methods. That is to say, our method can deal with the situation of illumination change very well, which is a significant advantage.

The two video sequences above both have strong video noise. From the result of the experiment, our method has a good result, better than the other methods. These two experiment results show that the proposed method has good immunity to video noise.

The fifth video in this experiment has a frequently changing background, which is a challenge for every algorithm. Most algorithms get poor result in this condition, as shown in Table 5. ViBe has a fast speed of background modeling, which is a great advantage when dealing with this situation. Although the experiment result of our method is not as good as ViBe in this condition, our method still performs better than the other methods. In addition, according to the experiment data, the gap between our method and ViBe algorithm is not too large. The proposed method still gets second place in this experiment.

As shown in Tables 1–5, our method (MB-TALBP) has the best results in four of five experiments. In Tables 1 and 4, since there is an obvious dynamic part in the backgrounds of the test video sequences, GMM, GMG, KDE and ViBe methods perform poorly. The LBAdaptiveSOM method shows a better result compared to those four algorithms, and a little worse than our method. Table 2 shows the experiment results of the sequence named "rabbit in night", which has an illumination change scenario. As shown in Table 2, our method still performs the best in this complex scenario. In Table 3, the backgrounds of the test sequences changes quickly, and our method performs better than four methods and a little worse than ViBe. The fifth video sequence has the frequent introduction and removal of objects and the indexes in Table 5 show that our method has the second best result, which means our method, is stable when facing this complex situation. The fifth video sequence has a frequently changed background and most methods in this experiment have a low result. ViBe algorithm has a fast speed of background modeling and is really good at dealing with the situation in video 5, as seen in Table 5. Although the ViBe algorithm has a better result in one of the experiments, our method still has the advantage in the rest of the experiments, which means our method performs better than the ViBe algorithm.

Our method still has the second best results in the experiments in which our method is not the best one. That is to say, our method can adapt to dynamic backgrounds, illumination variations, noise-corruption and other complex scenarios very well. Figure 10 shows the experiment results of all the 10 videos.

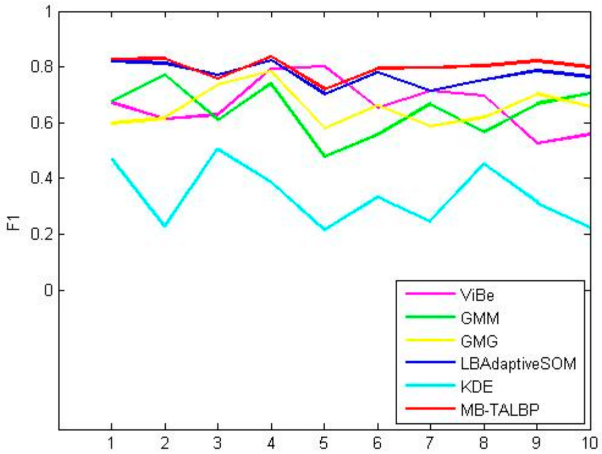


Figure 10. This image shows the line chart of the F-1 value of the experiment result of those methods we mentioned above.

The quantitative comparisons have indicated that our method has a good advantage over the other methods in real scenes that often occur in reality and is could be of great use when applied to those situations that require rapid and accurate detection of moving objects. Our method has been evaluated against five methods in 10 different video sequences including dynamic background, illumination variation, introduction/removal of background objects and noise-corrupted scenes. Figure 7 shows that our method performs well in most experiments and comparisons to other approaches presented in this experiment have shown that our approach has an obvious advantage to four of the methods (GMM, ViBe, GMG and KDE). It has proven to be tolerant to the dynamic background, and introduction and removal of background objects. Furthermore, the method is capable of dealing with illumination variations.

The proposed method is implemented on a PC with an i5-4570 3.2-GHz processor and 4 Gb RAM. Our MATLAB algorithm (including showing the binary results) can achieve the processing speeds of 26 fps, 35 fps and so on in the experiment. The detailed data are as follows.

The data of execution time are shown in Table 6: the execution time of our method in these experiments has an average result of 39.92 ms/frame (which is 25.05 fps). The portion of background modeling work (the LBP algorithm) takes most of the computation time, which means the execution time of background modeling determines the whole execution time. From these data, we can see our method has achieved the real-time requirement.

Table 6. Execution time of MB-TALBP (Multi-Block Temporal-Analyzing Local Binary Pattern) in software experiments.

Video	1	2	3	4	5	6	7	8	9	10
Time (ms/frame)	42.3	41.7	43.0	27.6	40.6	40.8	35.3	43.4	29.8	33.7

The hardware platform we used has one EP3C120F780I7N FPGA chip and two TMS320C6416 DSP chips as its core control and processing units, respectively. In our research, the execution time of processing one frame on the hardware platform is 49–54 ms (which is 18–20 fps). The speed of data processing on the hardware platform can also achieve the real-time requirement.

5. Conclusions

In this paper, we proposed a combinational method for moving object detection. In our method, the multi block temporal-analyzing LBP algorithm is used to do the background modeling work. Additionally, the edge detection method is only applied to enhance the ability of resisting illumination variation. The connected component labeling method is used to locate the object from the segmentation result. Experimental results show the preponderance of the proposed method in terms of precision, which is an important role of moving object detection. Though our method can handle illumination variation, dynamic background and video noise well, there are still some situations that it cannot handle very well, such as the frequently changing of background. Further research should be proceed from this angle: dealing with scenarios that have frequently changing backgrounds. The speed of background modeling, the frequency of background updating and the simplification of computing the background model are the primary issues that should be considered in our future research. We believe that solving these problems will enhance our method.

Acknowledgments: This work was supported by the Major Science Instrument Program of the National Natural Science Foundation of China under Grant 61527802, the General Program of National Nature Science Foundation of China under Grants 61371132, and 61471043, and the International S&T Cooperation Program of China under Grants 2014DFR10960.

Author Contributions: Shuoyang Chen and Tinfa Xu designed the global structure and the experiments. Shenwang Jiang and Jizhou Zhang designed the embedded system. Shuoyang Chen performed both software and hardware experiments. Daqun Li analyzed the data. Shuoyang Chen wrote the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Chiranjeevi, P.; Sengupta, S. Moving object detection in the presence of dynamic backgrounds using intensity and textural features. *J. Electron. Imaging* **2011**, *20*, 043009. [CrossRef]
2. Hassanpour, H.; Sedighi, M.; Manashty, A.R. Video frame's background modeling: Reviewing the techniques. *J. Signal Inf. Process.* **2011**, *2*, 72. [CrossRef]
3. Jiménez-Hernández, H. Background subtraction approach based on independent component analysis. *Sensors* **2010**, *10*, 6092–6114. [CrossRef] [PubMed]
4. Jones, G.A.; Paragios, N.; Regazzoni, C.S. *Video-Based Surveillance Systems: Computer Vision and Distributed Processing*; Springer: New York, NY, USA, 2012.
5. Joshi, K.A.; Thakore, D.G. A survey on moving object detection and tracking in video surveillance system. *Int. J. Soft Comput. Eng.* **2012**, *2*, 44–48.

6. Bouwmans, T. Recent advanced statistical background modeling for foreground detection—A systematic survey. *Recent Patents Comput. Sci.* **2011**, *4*, 147–176. [CrossRef]
7. Tu, G.J.; Karstoft, H.; Pedersen, L.J.; Jørgensen, E. Illumination and reflectance estimation with its application in foreground detection. *Sensors* **2015**, *15*, 21407–21426. [PubMed]
8. Varcheie, P.D.Z.; Sills-Lavoie, M.; Bilodeau, G.-A. A multiscale region-based motion detection and background subtraction algorithm. *Sensors* **2010**, *10*, 1041–1061. [CrossRef] [PubMed]
9. Yagi, Y.; Makihara, Y.; Hua, C. Moving object detection device. US Patent US8958641 B2, 17 February 2015.
10. Zhou, X.; Yang, C.; Yu, W. Moving object detection by detecting contiguous outliers in the low-rank representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 597–610. [CrossRef] [PubMed]
11. Fernandez-Sanchez, E.J.; Diaz, J.; Ros, E. Background subtraction based on color and depth using active sensors. *Sensors* **2013**, *13*, 8895–8915. [CrossRef] [PubMed]
12. Kim, D.-S.; Kwon, J. Moving object detection on a vehicle mounted back-up camera. *Sensors* **2015**, *16*. [CrossRef] [PubMed]
13. Lee, J.; Park, M. An adaptive background subtraction method based on kernel density estimation. *Sensors* **2012**, *12*, 12279–12300. [CrossRef]
14. Regazzoni, C.S.; Ramesh, V.; Foresti, G.L. Special issue on video communications, processing, and understanding for third generation surveillance systems. *Proc. IEEE* **2001**, *89*, 1355–1367.
15. Kim, I.S.; Choi, H.S.; Yi, K.M.; Choi, J.Y.; Kong, S.G. Intelligent visual surveillance—A survey. *Int. J. Control Autom. Syst.* **2010**, *8*, 926–939. [CrossRef]
16. Mohamed, S.S.; Tahir, N.M.; Adnan, R. Background modelling and background subtraction performance for object detection. In Proceedings of the 2010 6th International Colloquium on Signal Processing and Its Applications (CSPA), Malacca, Malaysia, 21–23 May 2010; pp. 1–6.
17. Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1337–1342. [CrossRef]
18. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artif. Intell.* **1981**, *17*, 185–203. [CrossRef]
19. Talukder, A.; Matthies, L. Real-time detection of moving objects from moving vehicles using dense stereo and optical flow. In Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2004; Volume 7, pp. 3718–3725.
20. Horprasert, T.; Harwood, D.; Davis, L.S. A statistical approach for real-time robust background subtraction and shadow detection. In Proceedings of the IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; pp. 1–19.
21. Hu, W.; Xiao, X.; Fu, Z.; Xie, D.; Tan, T.; Maybank, S. A system for learning statistical motion patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1450–1464. [PubMed]
22. Li, L.; Huang, W.; Gu, I.Y.-H.; Tian, Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Trans. Image Process.* **2004**, *13*, 1459–1472. [CrossRef] [PubMed]
23. Brutzer, S.; Höferlin, B.; Heidemann, G. Evaluation of background subtraction techniques for video surveillance. In Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Colorado Springs, CO, USA, 20–25 June 2011; pp. 1937–1944.
24. Hu, W.; Tan, T.; Wang, L.; Maybank, S. A survey on visual surveillance of object motion and behaviors. *IEEE Trans. Syst. Man Cybern. C* **2004**, *34*, 334–352. [CrossRef]
25. Collins, R.T.; Lipton, A.J.; Kanade, T.; Fujiyoshi, H.; Duggins, D.; Tsin, Y.; Tolliver, D.; Enomoto, N.; Hasegawa, O.; Burt, P. A system for video surveillance and monitoring. *Robot. Inst.* **2000**, *5*, 329–337.
26. Siebel, N.T.; Maybank, S. The advisor visual surveillance system. In Proceedings of the ECCV 2004 Workshop Applications of Computer Vision (ACV), Prague, Czech Republic, 11–14 May 2004.
27. Shu, C.-F.; Hampapur, A.; Lu, M.; Brown, L.; Connell, J.; Senior, A.; Tian, Y. IBM smart surveillance system (s3): A open and extensible framework for event based surveillance. In Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance, Como, Italy, 15–16 September 2005; pp. 318–323.
28. Wang, X. Intelligent multi-camera video surveillance: A review. *Pattern Recognit. Lett.* **2013**, *34*, 3–19. [CrossRef]
29. Song, M.; Tao, D.; Maybank, S.J. Sparse camera network for visual surveillance—A comprehensive survey. **2013**, arXiv:1302.0446.
30. Cheng, F.-C.; Chen, B.-H.; Huang, S.-C. A hybrid background subtraction method with background and foreground candidates detection. *ACM Trans. Intell. Syst. Technol.* **2015**, *7*. [CrossRef]

31. Huang, S.-C.; Do, B.-H. Radial basis function based neural network for motion detection in dynamic scenes. *IEEE Trans. Cybern.* **2014**, *44*, 114–125. [CrossRef] [PubMed]
32. Chen, B.-H.; Huang, S.-C. An advanced moving object detection algorithm for automatic traffic monitoring in real-world limited bandwidth networks. *IEEE Trans. Multimedia* **2014**, *16*, 837–847. [CrossRef]
33. Cao, X.; Yang, L.; Guo, X. Total variation regularized rpca for irregularly moving object detection under dynamic background. *IEEE Trans. Cybern.* **2016**, *46*, 1014–1027. [CrossRef] [PubMed]
34. Zhang, X.; Zhu, C. Camouflage modeling for moving object detection. In Proceedings of the 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), Chengdu, China, 12–15 July 2015; pp. 249–253.
35. Stauffer, C.; Grimson, W.E.L. Adaptive background mixture models for real-time tracking. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, 23–25 June 1999; pp. 1585–1594.
36. Lee, D.-S.; Hull, J.J.; Erol, B. A bayesian framework for gaussian mixture background modeling. In Proceedings of the 2003 International Conference on Image Processing, ICIP 2003, Barcelona, Spain, 14–17 September 2003.
37. Reynolds, D.A.; Quatieri, T.F.; Dunn, R.B. Speaker verification using adapted gaussian mixture models. *Digit. Signal Proc.* **2000**, *10*, 19–41. [CrossRef]
38. Zivkovic, Z. Improved adaptive gaussian mixture model for background subtraction. In Proceedings of the 17th International Conference on Pattern Recognition, ICPR 2004, Cambridge, UK, 23–26 August 2004; pp. 28–31.
39. Guo, J.-M.; Hsia, C.-H.; Liu, Y.-F.; Shih, M.-H.; Chang, C.-H.; Wu, J.-Y. Fast background subtraction based on a multilayer codebook model for moving object detection. *IEEE Trans. Circuits Syst. Video Technol.* **2013**, *23*, 1809–1821. [CrossRef]
40. Kim, K.; Chalidabhongse, T.H.; Harwood, D.; Davis, L. Background modeling and subtraction by codebook construction. In Proceedings of the International Conference on Image Processing (ICIP), Singapore, 24–27 October 2004; pp. 3061–3064.
41. Kim, K.; Chalidabhongse, T.H.; Harwood, D.; Davis, L. Real-time foreground–background segmentation using codebook model. *Real Time Imaging* **2005**, *11*, 172–185. [CrossRef]
42. Wang, H.; Suter, D. Robust adaptive-scale parametric model estimation for computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1459–1474. [CrossRef] [PubMed]
43. Wang, H.; Suter, D.; Schindler, K.; Shen, C. Adaptive object tracking based on an effective appearance filter. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1661–1667. [CrossRef] [PubMed]
44. Barnich, O.; Van Droogenbroeck, M. Vibe: A powerful random technique to estimate the background in video sequences. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 945–948.
45. Barnich, O.; Van Droogenbroeck, M. Vibe: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.* **2011**, *20*, 1709–1724. [CrossRef] [PubMed]
46. Cucchiara, R.; Grana, C.; Piccardi, M.; Prati, A.; Sirotti, S. Improving shadow suppression in moving object detection with hsv color information. In Proceedings of the 2001 IEEE Intelligent Transportation Systems Proceedings, Oakland, CA, USA, 25–29 August 2001; pp. 334–339.
47. Doliotis, P.; Stefan, A.; McMurrough, C.; Eckhard, D.; Athitsos, V. Comparing gesture recognition accuracy using color and depth information. In Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments, Rhodes, Greece, 25–27 May 2011.
48. Liu, F.; Gleicher, M. Learning color and locality cues for moving object detection and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–26 June 2009; pp. 320–327.
49. Elgammal, A.; Duraiswami, R.; Harwood, D.; Davis, L.S. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. IEEE* **2002**, *90*, 1151–1163. [CrossRef]
50. Elgammal, A.; Harwood, D.; Davis, L. Non-parametric model for background subtraction. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 26 June–1 July 2000; pp. 751–767.

51. Harville, M. A framework for high-level feedback to adaptive, per-pixel, mixture-of-gaussian background models. In Proceedings of the European Conference on Computer Vision, Zurich, Swizerland, 6–12 September 2002; pp. 543–560.
52. Hofmann, M.; Tiefenbacher, P.; Rigoll, G. Background segmentation with feedback: The pixel-based adaptive segmenter. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 38–43.
53. Eng, H.-L.; Wang, J.; Kam, A.H.; Yau, W.-Y. Novel region-based modeling for human detection within highly dynamic aquatic environment. In Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, Washington, DC, USA, 27 June–2 July 2004.
54. Zhong, J.; Sclaroff, S. Segmenting foreground objects from a dynamic textured background via a robust kalman filter. In Proceedings of the 9th IEEE International Conference on Computer Vision, Nice, France, 14–17 October 2003; pp. 44–50.
55. Heikkila, M.; Pietikainen, M. A texture-based method for modeling the background and detecting moving objects. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 657–662. [CrossRef] [PubMed]
56. Zhang, S.; Yao, H.; Liu, S. Dynamic background modeling and subtraction using spatio-temporal local binary patterns. In Proceedings of the 15th IEEE International Conference on Image Processing, San Diego, CA, USA, 12–15 October 2008; pp. 1556–1559.
57. Heikkilä, M.; Pietikäinen, M.; Schmid, C. Description of interest regions with local binary patterns. *Pattern Recognit.* **2009**, *42*, 425–436. [CrossRef]
58. Zhao, G.; Pietikainen, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 915–928. [CrossRef] [PubMed]
59. Zhan, C.; Duan, X.; Xu, S.; Song, Z.; Luo, M. An improved moving object detection algorithm based on frame difference and edge detection. In Proceedings of the 4th International Conference on Image and Graphics, Jacksonville, FL, USA, 22–24 August 2007; pp. 519–523.
60. Liao, S.; Zhu, X.; Lei, Z.; Zhang, L.; Li, S.Z. Learning multi-scale block local binary patterns for face recognition. In Proceedings of the International Conference on Biometrics, Seoul, Korea, 27–29 August 2007; pp. 828–837.
61. Liao, S.; Zhao, G.; Kellokumpu, V.; Pietikäinen, M.; Li, S.Z. Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 1301–1306.
62. Maddalena, L.; Petrosino, A. A fuzzy spatial coherence-based approach to background/foreground separation for moving object detection. *Neural Comput. Appl.* **2010**, *19*, 179–186. [CrossRef]
63. Vacavant, A.; Chateau, T.; Wilhelm, A.; Lequièvre, L. A benchmark dataset for outdoor foreground/background extraction. In Proceedings of the Asian Conference on Computer Vision, Daejeon, Korea, 5–9 November 2012; pp. 291–300.