sensors

MDPI

Article

Improving Video Segmentation by Fusing Depth Cues and the Visual Background Extractor (ViBe) Algorithm

Xiaoqin Zhou [1,2,3], Xiaofeng Liu [2,3,4,*], Aimin Jiang [2,3,4], Bin Yan [5] and Chenguang Yang [6]

[1] College of Computer and Information Engineering, Hohai University, Nanjing 211100, China; zhouxq@hhu.edu.cn
[2] Changzhou Key Laboratory of Robotics and Intelligent Technology, Changzhou 213022, China; jiangam@hhuc.edu.cn
[3] Jiangsu Key Laboratory of Special Robots, Hohai University, Changzhou 213022, China
[4] College of IoT Engineering, Hohai University, Changzhou 213022, China
[5] College of Electronics, Communication and Physics, Shandong University of Science and Technology, Qingdao 266590, China; yanbinhit@hotmail.com
[6] Zienkiewicz Centre for Computational Engineering, Swansea University, Swansea SA1 8EN, UK; cyang@theiet.org
* Correspondence: xfliu@hhu.edu.cn

Academic Editor: Joonki Paik
Received: 19 March 2017; Accepted: 18 May 2017; Published: 21 May 2017

Abstract: Depth-sensing technology has led to broad applications of inexpensive depth cameras that can capture human motion and scenes in three-dimensional space. Background subtraction algorithms can be improved by fusing color and depth cues, thereby allowing many issues encountered in classical color segmentation to be solved. In this paper, we propose a new fusion method that combines depth and color information for foreground segmentation based on an advanced color-based algorithm. First, a background model and a depth model are developed. Then, based on these models, we propose a new updating strategy that can eliminate ghosting and black shadows almost completely. Extensive experiments have been performed to compare the proposed algorithm with other, conventional RGB-D (Red-Green-Blue and Depth) algorithms. The experimental results suggest that our method extracts foregrounds with higher effectiveness and efficiency.

Keywords: object detection; background subtraction; video surveillance; Kinect sensor fusion

1. Introduction

In recent years, enormous amounts of data on human or animal behavior have been collected using 2D and 3D cameras, and automated methods for detecting and tracking individuals or animals have begun to play an important role in studies of experimental biology, behavioral science, and related disciplines. Tracking target is prone to being lost due to managing appearance changes, fast motion, and other factors. This results in the problem of detecting the tracking target again. Traditional detection methods (such as the detector in TLD (Tracking-Learning-Detection) [1]) treat every frame independently and perform full scanning of an input frame to localize all appearances that have been observed and learned in the past. It is inefficient and time consuming, especially for real-time tracking. However, foreground segmentation requires a detector to scan only the foreground region of an input frame using a scanning window, which greatly reduces the scanning time of the detector and also improves the classification accuracy. At present, a variety of foreground extraction methods have been proposed, such as the frame difference method [2], the background subtraction method [3], the optical flow method [4] and the block matching method [5]. The core of a background subtraction algorithm is the modeling of the background. Zones that show notable differences between the current frame and the background model are deemed to correspond to moving objects. Generally, background subtraction algorithms include the Average Background Model (AVG) algorithm, the Gaussian Mixture Model (GMM) algorithm [6], the Codebook algorithm [7] and the Visual Background Extractor (ViBe) algorithm [8–10]. The ViBe algorithm is a fast motion detection algorithm proposed by Olivier Barnich et al. [8]. It is characterized by a high processing efficiency and a good detection effect.

Most of the conventional methods mentioned above were designed for application to color images. However, depth is another interesting cue for segmentation that is less strongly affected by the adverse effects encountered in classical color segmentation, such as shadow and highlight regions. Depth cameras, such as the Microsoft Kinect and the ASUS Xtion Pro (ASUS, Taiwan), are able to record real-time depth video together with color video. Because of their beneficial depth imaging features and moderate price, such depth cameras are broadly applied in intelligent surveillance, medical diagnostics, and human–computer interaction applications [11–13]. The Kinect sensor is not sensitive to light conditions; it works well either in a bright room or in a pitch black one. This makes depth images more reliable and easier for a computer program to understand.

Most studies using the Kinect sensor have focused on human body detection and tracking [14–16]. The Histogram of Oriented Depths (HOD) detection algorithm, proposed in [14], can be used to match human body contour information in an image. In [15], a model was presented for detecting humans using a 2D head contour model and a 3D head surface model. In these studies, the computational complexity of the feature generation and matching process was relatively high.

Crabb et al. [17] and Schiller et al. [18] focused on combining the depth and color information obtained by low-resolution ToF (Time of Flight) cameras, but their methods are not well suited for video surveillance. For example, the method of Crabb et al. [17] requires the definition of a distance plane where no foreground object is located behind any part of the background. Fernandez-Sanchez et al. [19] proposed an adaptation of the Codebook background subtraction algorithm that focuses on different sensor channels. In [20], the method presented combines a mixture of Gaussian-based background subtraction algorithms with a new Bayesian network. The Bayesian network exploits the characteristics of the depth data using two dynamic models that estimate the spatial and depth evolution of the foreground/background regions. Camplani et al. [21] proposed a foreground/background segmentation method that combines two statistical classifiers using color and depth features. The combination of depth and color cues makes it possible to solve color segmentation issues such as shadows, reflections and camouflage. However, the computations required are too complicated. The fundamental idea of the combination strategy in [22] is that when depth measurement is reliable, the background subtraction from depth takes top priority. Otherwise, RGB (Red-Green-Blue) is used as an alternative. Noise is removed from the depth data using a noise model. They define the background as the stationary part of a scene. The Gaussian mixture model is observed for each pixel over a sequence of frames. These existing RGB-D (RGB and Depth) segmentation algorithms either suffer from ghosts, such as Depth-Extended Codebook (DECB) [19] and 4D version of Mixture of Gaussians (MOG4D) [18], or fail to achieve real-time performance [21]. A ghost is a set of interconnected points that is detected as a moving object but does not correspond to any real object (see Figure 1c). Ghosting greatly reduces the effectiveness of motion detection.

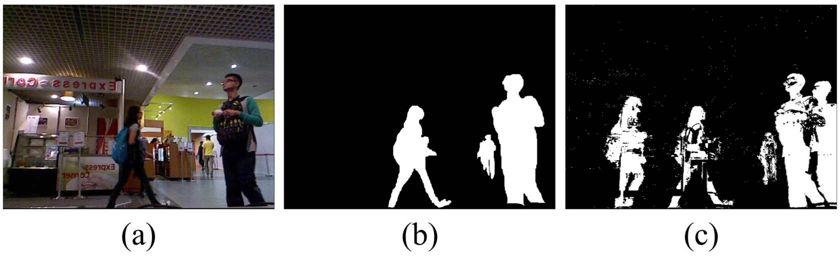

Figure 1. Ghosting in a foreground segmentation map generated by the Visual Background Extractor (ViBe) algorithm [8]: (**a**) color frame; (**b**) ground truth; (**c**) foreground extraction result.

In this paper, we propose an adaptive ViBe background subtraction algorithm that fuses the depth and color information obtained by the Kinect sensor to segment foreground regions. First, a background model and a depth model are established. Then, based on these models, we develop a new updating strategy that can efficiently eliminate ghosting and black shadows. The improved algorithm is evaluated using an RGB-D benchmark dataset [23] in addition to our own test RGB-D video and achieves good results that provide a perfect basis for subsequent feature extraction and behavior recognition.

The remainder of the paper is organized as follows. In Section 2, we briefly describe the original ViBe algorithm. Then, the improved algorithm is developed in Section 3. In Section 4, experimental results and discussions are presented. Finally, we conclude the paper in Section 5.

2. ViBe Background Subtraction Algorithm

In this section, we first review the basic ViBe algorithm. Then, we identify its disadvantages. This technique involves modeling the background based on a set of samples for each pixel. New frames are compared with the background model, pixel by pixel, to determine whether each pixel belongs to the background or the foreground.

2.1. Pixel Model

Background model construction begins from the first frame. Formally, let $v(x)$ denote the value in a given Euclidean color space associated with the pixel located at x in the image, and let v_i be the background sample value with index i. Each background pixel x is modeled based on a collection of N background sample values $M(x) \ = \{v_1, v_2, \cdots, v_N\}$.

2.2. Classification Process

If the Euclidean distance from a sample v_i in $M(x)$ to $v(x)$ is below a threshold R, then v_i is regarded as a neighbor of $v(x)$. We define the number of neighbors of the pixel located at x as $N_R(x) = \{\|v(x) - v_i\| < R, \forall v_i \in M(x)\}$. When $N_R(x)$ is greater than a threshold λ, x is a background pixel. Otherwise, it is a foreground pixel.

2.3. Updating the Background Model over Time

It is necessary to continuously update the background model with each new frame. This is a crucial step for achieving accurate results over time. When a pixel x is classified as background, the background model updating strategy is triggered. A sample is chosen randomly. Mathematically, the probability that a sample present in the model at time t will be preserved is given by $(N-1)/N$. Under the assumption of time continuity, for any later time $t + dt$, this probability is equal to:

$$P(t, t+dt) = (\frac{N-1}{N})^{(t+dt)-t} \tag{1}$$

which can be rewritten as:

$$P(t, t + dt) = e^{-ln(\frac{N}{N-1})^{dt}} \tag{2}$$

This expression shows that the expected remaining lifespan of any sample value in the model decays exponentially according to a random subsampling strategy. As in [8], we adopt a time subsampling factor of ϕ, meaning that a background pixel value has one chance in ϕ of being selected to update its pixel model.

Reduced pseudo-code for the ViBe construction phase is given in Algorithm 1.

The classical ViBe algorithm has the advantages of simple processing and outstanding performance. Its main drawback is the occurrence of ghosting. A moving object in the first frame often causes ghosting. To resolve this problem, we can take advantage of depth information. The Kinect sensor records the distance to any object that is placed in front of it. This feature can be utilized to determine whether a foreground pixel is a ghost.

Algorithm 1 Algorithm for ViBe construction

```
procedure ViBe(image, $N$, $R$, $\lambda$, $\phi$)
    for each pixel do
        while $matches < \lambda$ and $index < N$ do
            Calculate Euclidean distance between $v_x$ and $v_i$
            if $dist < R$ then
                $matches \leftarrow matches + 1$
            end if
            $index \leftarrow index + 1$
        end while
        if $matches \geqslant \lambda$ then
            Store that pixel $\in$ background
            Update current pixel background model with probability $1/\phi$
            Update neighboring pixel background model with probability $1/\phi$
        else
            Store that pixel $\in$ foreground
        end if
    end for
end procedure
```

3. Fusion: Depth-Extended ViBe (DEVB)

3.1. Depth Model

In general, the first frame in a video serves as the background. When the target appears in the first frame, most of detection methods get difficult to dissociate the foreground from the background quickly only by means of RGB images. However, it would become easy when exploiting depth cues. For a depth image, a pixel is brighter and its distance to camera is closer. Accordingly, the pixel values of a foreground are large, while those of a background are small. When a foreground shifts, the pixels corresponding to its initial position naturally get small. If those pixels are still recognized as the foreground, they would be a false foreground (a ghost).

To eliminate ghosting, the ViBe algorithm is improved by enhancing the matching conditions when a pixel is classified as foreground. A depth model $M_D(x)$ is added. Initially, the pixel values of the first depth frame are saved to $M_D(x)$. This depth model also has an updating strategy similar to that for the background model. When the updating strategy is triggered, the depth value $M_D(x)$ is replaced with that corresponding to the current pixel. If the following condition is satisfied, this pixel will be considered a ghost pixel:

$$v(x, t_0) - v(x, t) > \tau \tag{3}$$

where $v(x, t_0)$ is the value of the pixel located at x in the depth model $M_D(x)$ at time t_0, $v(x, t)$ is the value of the pixel located at x in the current depth image at time t. The threshold τ is an empirical value, which can be tuned to obtain the best result. The value of τ is recommended to lie within the range of $[1, 3]$.

3.2. Fusion Algorithm for Color and Depth Images

A color image conforms to an individual's visual habits and provides detailed information such as color and texture. The most intuitive fusion strategy is to add the depth information as a fourth channel to the ViBe algorithm for color images. The channel combination $f = (f_r, f_g, f_b, d)$ is formed from the three color channels in RGB space and the depth value d. In this way, we intend to utilize depth information as a measure of reliability during segmentation. A depth image quantifies the distance from an object to the camera. The higher the value of a pixel is, the more reliable its measurement is in the depth image. Thus, we use the inverse of the depth image to allow it to be used in the same way as a variance image.

The inverted depth values are normalized between zero and one, and the normalized uncertainty is denoted by $\sigma(x)$. In a region where the depth uncertainty is high, the depth measurement is considered unreliable. For example, where there are holes in the depth image, the fusion result will depend on the color image. We weight the normalized depth $d\hat{(x)}$ with the uncertainty $\sigma(x)$, resulting in w_d values that range between zero and one depending on $\sigma(x)$. Meanwhile, the color value $c\hat{(x)}$ is multiplied by $\sigma(x)$ and added to itself to obtain the weighted color value, defined as w_c. The combined normalized image is denoted by $I\hat{(x)}$, as shown in Equation (4).

$$\begin{cases} w_d = (1 - \sigma(x))d\hat{(x)} \\ w_c = (1 + \sigma(x))c\hat{(x)} \\ I\hat{(x)} = \frac{1}{2}(w_d + w_c) \end{cases} \tag{4}$$

The original algorithm can eliminate ghosts and black shadows in subsequent frames, but the process is relatively slow. We propose a fusion strategy that incorporates an additional depth model to effectively remove ghosting and black shadows. Reduced pseudo-code for the DEVB construction phase is given in Algorithm 2.

Algorithm 2 Algorithm for DEVB construction

```
procedure DEVB($image_{fusion}$,$image_{depth}$,$N$, $R$, $\lambda$, $\phi$)
    for $each\ image_{fusion}\ pixel$ do
        while $matches < \lambda$ and $index < N$ do
            Calculate Euclidean distance between $v_x$ and $v_i$
            if $dist < R$ then
                $matches \leftarrow matches + 1$
            end if
            $index \leftarrow index + 1$
        end while
        if $matches \geqslant \lambda$ then
            Store current pixel $\in$ background
            Update current pixel background model in $image_{fusion}$ with probability $1/\phi$
            Update neighboring pixel background model in $image_{fusion}$ with probability $1/\phi$
            Update current pixel $model_{depth}$ using $image_{depth}$
        else
            Store current pixel $\in$ foreground
            Find ghosts, calculate Euclidean distances between pixels at the same position in
    $image_{depth}$ and $model_{depth}$
            if $dist > \tau$ then
                Store that pixel $\in$ background
                Update current pixel background model in $image_{fusion}$ with probability $1/\phi$
                Update neighboring pixel background model in $image_{fusion}$ with probability $1/\phi$
            end if
        end if
    end for
end procedure
```

Section 4 presents experiments performed without post-processing and the results obtained using three RGB-D algorithms (MOG4D [18], DECB [19] and DEVB) as well as the color-based ViBe algorithm.

4. Experiments and Results

The program development environment consisted of VC++2010, OpenCV SDK2.4.3 and OpenNI1.5.2.7. The PC was equipped with a Core Duo 2 CPU E7500 with 2.00 GB of RAM. The video frame rate was 30 fps, and the size of both the color and depth images was 640 × 480. We compared the results of our method with those obtained using two state-of-the-art RGB-D fusion-based background subtraction algorithms, namely, MOG4D [18] and DECB [19], as well as the ViBe algorithm on the color images [8] (ViBe) and the ViBe based only on depth (ViBe1D). To evaluate these algorithms objectively through a quantitative analysis, we required a benchmark that would provide information on both color and depth images. The chosen benchmark sequences are publicly available at [23]. However, the depth image sequences provided by this source could not be utilized directly. The depth images are in the 16-bit png format, with the first three bits swapped with the last. We needed to swap them back after reading each image to obtain values for each pixel corresponding to the distance from the Kinect sensor to the object in mm. The sequences $child\_no1$, new_ex_occ4, $walking\_occ1$ and $new\_ex\_no\_occ$ from [23] and our own $pigeon$ were chosen for testing.

Many metrics can be used to assess the output of a background subtraction algorithm given a series of ground truths for several frames in each sequence. Various relative metrics can be calculated based on the numbers of true and false positives and negatives (TP, FP, TN, and FN). These metrics are most widely used in computer vision to assess the performance of a binary classifier, as in [24].

PWC is the percentage of wrong classifications in the entire image. This measure represents a trade-off between the abilities of an algorithm to detect foreground and background pixels. In general, a lower value of this estimator indicates better performance.

$$PWC = \frac{FN + FP}{TP + TN + FP + FN} \times 100 \quad (5)$$

The proposed approach relies on several parameters originating from the ViBe algorithm: N, R, λ, and ϕ in [8]. Considering our aim of evaluating the overall performance of the algorithms, we chose a unique set of parameters that yielded sufficiently good results on the complete dataset. According to [8], and in our experience, the use of a radius $R = 20$ and a time subsampling factor $\phi = 16$ leads to excellent results in every situation. To determine optimal values for λ, we computed the evolution of the PWC of DEVB on the *new_ex_no_occ* sequence for λ ranging from 1 to 20. The other parameters were fixed to $N = 20, R = 20$, and $\phi = 16$. Based on joint consideration of the value of λ in [8] and in Figure 2, we set the optimal value of λ to $\lambda = 2$. As λ rises, the computational cost increases. Once the value of 2 has been selected for λ, we study the influence of the parameter N on the performance of DEVB. Figure 3 shows the percentages obtained using the *new_ex_no_occ* sequence for N ranging from 2 to 50. We observe that higher values of N provide better performance. However, they tend to saturate for values higher than 20. We select N at the beginning of the basin, that is, $N = 20$. To ensure a fair comparison of the performances of the various algorithms, all algorithms were applied without morphological filtering.

The first sequence, *child_no*1, shows a child and two adults playing in a living room. The main difficulties in this sequence are light reflections and subjects that sometimes remain still or move only slowly. Figure 4 shows the segmentations produced by the four methods as well as the original color and depth frames and the hand-generated segmentations (ground truths). Ghosts appear in some frames for MOG4D, DECB, ViBe and ViBe1D, greatly reducing the effectiveness of foreground detection. The ViBe algorithm yields worse results than the other algorithms in frame 160 because of the reflection in the color image. In general, the DEVB algorithm achieves improvement over ViBe by virtue of the additional depth model, which allows the ghosts and black shadows to be effectively removed.

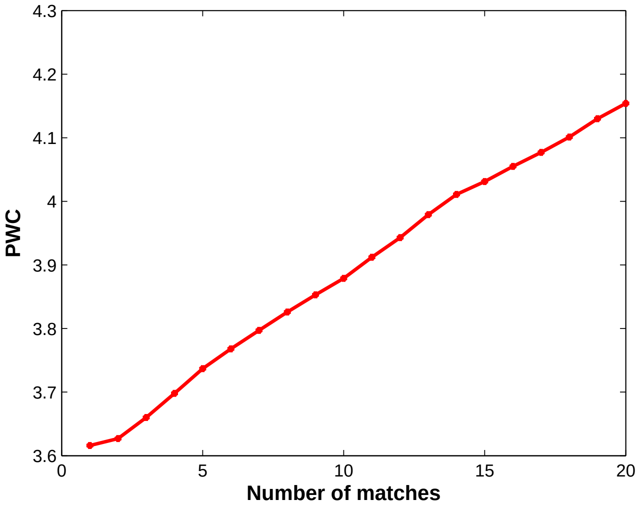

Figure 2. Percentage of wrong classifications (PWCs) for λ ranging from 1 to 20. The other parameters of depth-extended ViBe (DEVB) were set to $N = 20, R = 20$, and $\phi = 16$.

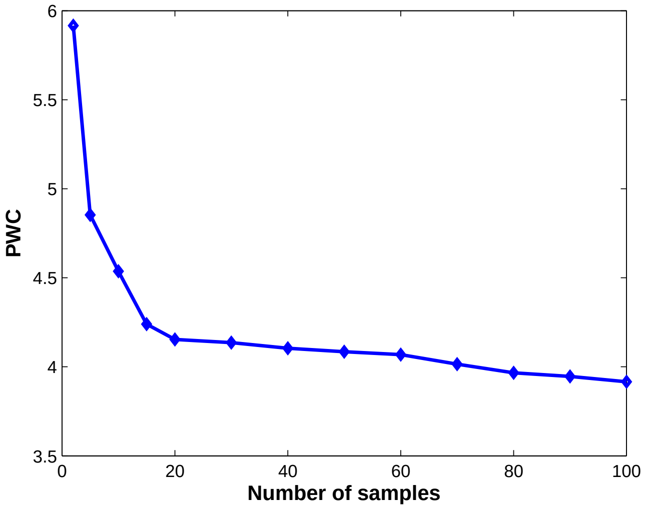

Figure 3. PWCs given the number of samples collected in a background model.

Table 1 shows the quantitative *PWC* results obtained by the four approaches on the evaluation frames from the *child_no*1 sequence. All RGB-D approaches achieve improvements with respect to ViBe, obtaining lower *PWC* values. A lower *PWC* value indicates better performance. The proposed DEVB algorithm achieves the lowest average error rate of 7.023% in Table 1, which indicates that our method performs better than the other algorithms.

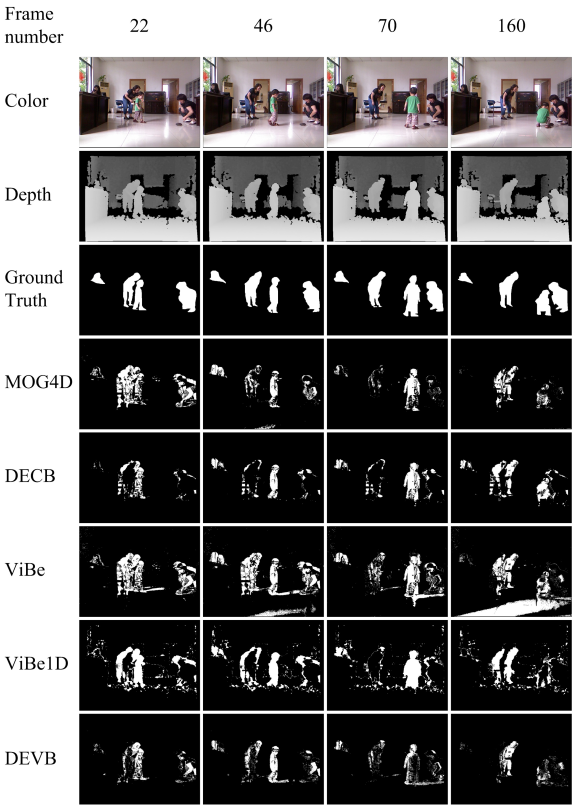

Figure 4. Comparison of background/foreground segmentation images generated by various background subtraction techniques for four frames taken from the *child_no*1 sequence without morphological filtering. The segmented images produced by our method are the closest to the ground-truth references. MOG4D: 4D version of Mixture of Gaussians. DECB: Depth-Extended Codebook. ViBe: visual background extractor. ViBe1D: ViBe based only on depth. DEVB: Depth-Extended ViBe.

Table 1. Segmentation evaluation for the *child_no*1 sequence. The table shows the *PWC* results for the various approaches on four different evaluation frames and the mean values for this sequence.

Approach	Frame 22	Frame 46	Frame 70	Frame 160	Mean
DEVB	**6.114**	**6.035**	8.826	**7.118**	**7.023**
ViBe	7.808	8.932	**7.879**	16.455	10.269
ViBe1D	11.183	10.686	9.458	10.848	10.544
MOG4D	7.519	6.559	8.003	8.13	7.553
DECB	8.271	8.072	8.672	8.334	8.337

The second sequence, *new_ex_occ*4, shows two individuals walking in front of a coffee shop. The main difficulties presented by this sequence are flickering lights and areas where depth information cannot be obtained by the active infrared sensor. Figure 5 shows the segmentations produced by the four approaches. Our DEVB algorithm achieves good results, whereas ghosting greatly reduces the effectiveness of the other algorithms in foreground extraction.

Table 2 shows the quantitative *PWC* results obtained by the four approaches on the evaluation frames from the *new_ex_occ*4 sequence. The proposed DEVB algorithm achieves the lowest *PWC* of 3.968, which indicates that our method performs better than the other algorithms. ViBe yields the worst result of $PWC = 12.872$ on this sequence.

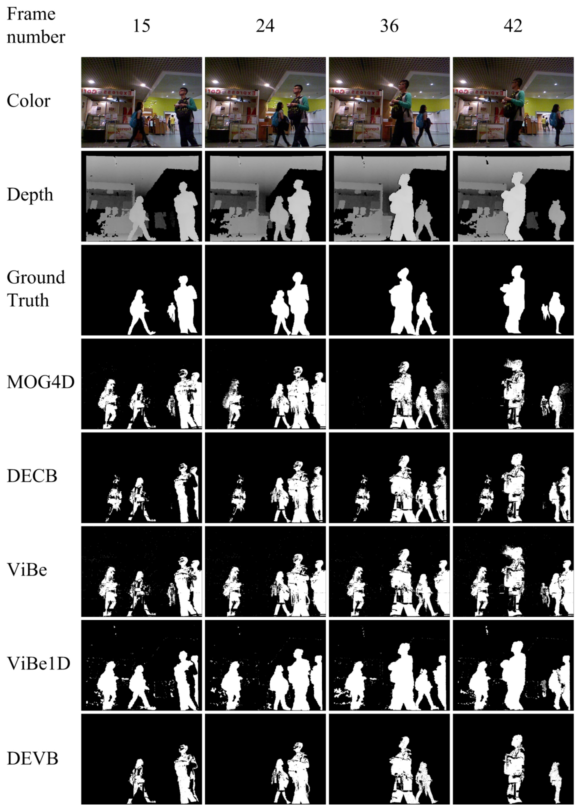

Figure 5. Results obtained in the test sequence *new_ex_occ*4.

Table 2. Segmentation evaluation for the *new_ex_occ*4 sequence. The table shows the *PWC* results for the various approaches on four different evaluation frames as well as the mean values for this sequence.

Approach	Frame 15	Frame 24	Frame 36	Frame 42	Mean
DEVB	**5.129**	**2.997**	**3.261**	**4.485**	**3.968**
ViBe	12.294	12.394	12.684	14.116	12.872
ViBe1D	12.096	12.554	12.461	11.999	12.278
MOG4D	11.685	11.157	4.845	5.189	8.219
DECB	9.753	8.522	8.706	9.909	9.223

The third sequence, *walking_occ*1, shows a few people walking in and out of the camera field. In addition, there are flickering lights on the ceiling and sudden illumination changes. Figure 6 shows the segmentations produced by the four approaches. DECB and ViBe1D are less affected by the sudden illumination changes. In addition to ghosting, ViBe1D results in black shadows in frames 35 and 62. The reason for the generation of black shadows is that a new moving object reaches the previous position of an old target. Because the pixel values are similar, the foreground is misclassified as background.

Table 3 shows the quantitative *PWC* results obtained by the four approaches on the evaluation frames from the *walking_occ*1 sequence. The DEVB algorithm achieves $PWC = 8.847$ in frame 8, whereas the *PWC* values obtained by the other algorithms are higher by more than a factor of three. Moreover, despite being affected by illumination changes in the RGB space, DEVB achieves an average *PWC* of 8.721 (Table 3), indicating that our method is fairly robust to difficult situations.

The fourth sequence, *new_ex_no_occ*, shows a lady walking in front of a coffee shop. The scenario is similar to the *new_ex_occ*4 sequence discussed above. Figure 7 shows the segmentations produced by the four approaches. A large amount of noise is generated by the ViBe1D algorithm because of the holes in the original depth image.

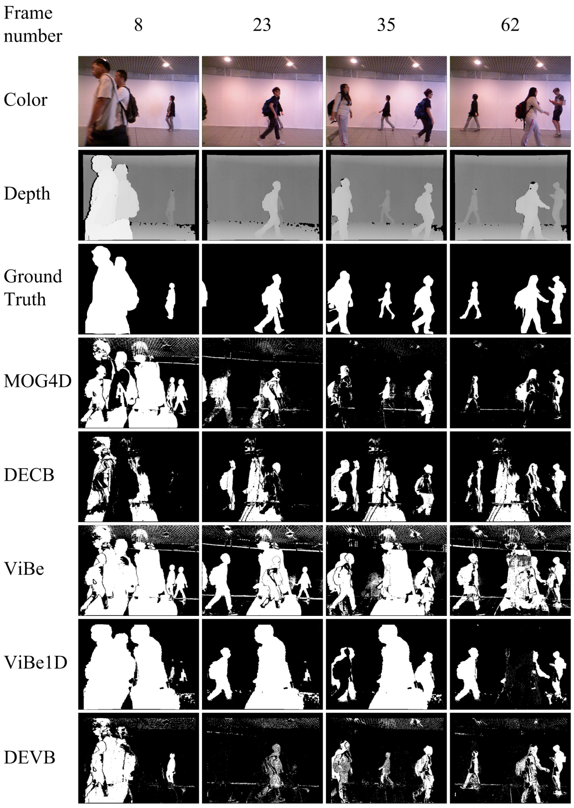

Figure 6. Results obtained in the test sequence *walking_occ*1.

Table 3. Segmentation evaluation for the *walking_occ*1 sequence. The table shows the *PWC* results for the various approaches on four different evaluation frames and the mean values for this sequence.

Approach	Frame 8	Frame 23	Frame 35	Frame 62	Mean
DEVB	**8.847**	**7.728**	**10.17**	**8.137**	**8.721**
ViBe	33.287	30.628	30.221	24.078	29.554
ViBe1D	29.944	26.806	31.726	14.332	25.702
MOG4D	38.186	11.195	10.798	9.185	17.341
DECB	30.262	18.616	22.371	21.162	23.103

Table 4 shows the quantitative *PWC* results obtained by the four approaches on the evaluation frames from the *new_ex_no_occ* sequence. The proposed DEVB algorithm achieves $PWC = 2.437$. This value is the lowest in Table 4, which indicates that our method performs better than the other algorithms. ViBe1D obtains the worst result in this sequence, with $PWC = 12.389$.

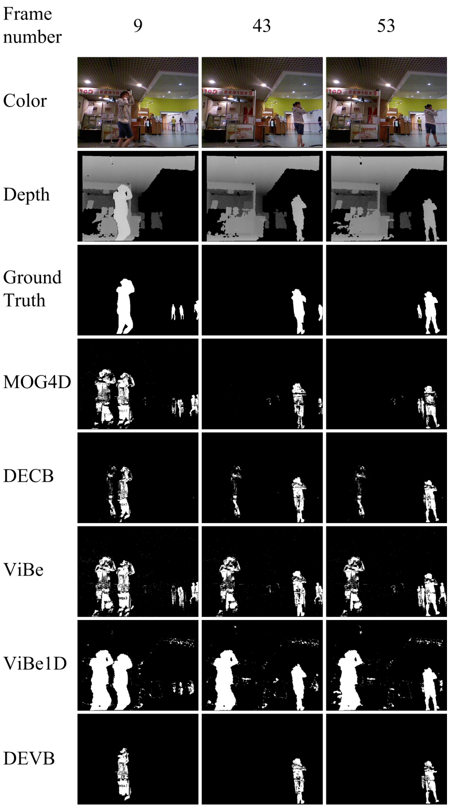

Figure 7. Results obtained in the test sequence *new_ex_no_occ*.

Table 4. Segmentation evaluation for the *new_ex_no_occ* sequence. The table shows the *PWC* results for the various approaches on three different evaluation frames as well as the mean values for this sequence.

Approach	Frame 9	Frame 43	Frame 53	Mean
DEVB	**4.102**	**1.549**	**1.66**	**2.437**
ViBe	8.188	5.832	6.097	6.706
ViBe1D	14.009	11.249	11.91	12.389
MOG4D	7.977	1.719	1.842	3.846
DECB	5.264	2.222	2.225	3.237

The fifth sequence, *pigeon*, is from our own test RGB-D video. The pigeon has a wide range of activities in the scene as shown in Figure 8. It can jump onto a high platform and walk beside a water bottle and a feeder, which serve as the background. Because of the fast and abrupt pose changes of the pigeon, most of the tracking algorithms fail to track in most of the frames. We focus on detection technology and on utilizing the corresponding depth images, which extracts foregrounds more effectively and more efficiently. In addition to ghosting, ViBe1D results in much noise because of the holes in the original depth image. DECB suffers from ghosting in all test frames.

Table 5 shows the quantitative *PWC* results obtained by the four approaches on the evaluation frames from the *pigeon* sequence. The proposed DEVB algorithm achieves $PWC = 0.778$. This value is the smallest in Table 5, which indicates that our method performs better than the other algorithms. ViBe1D obtains the worst result on this sequence, with $PWC = 2.245$.

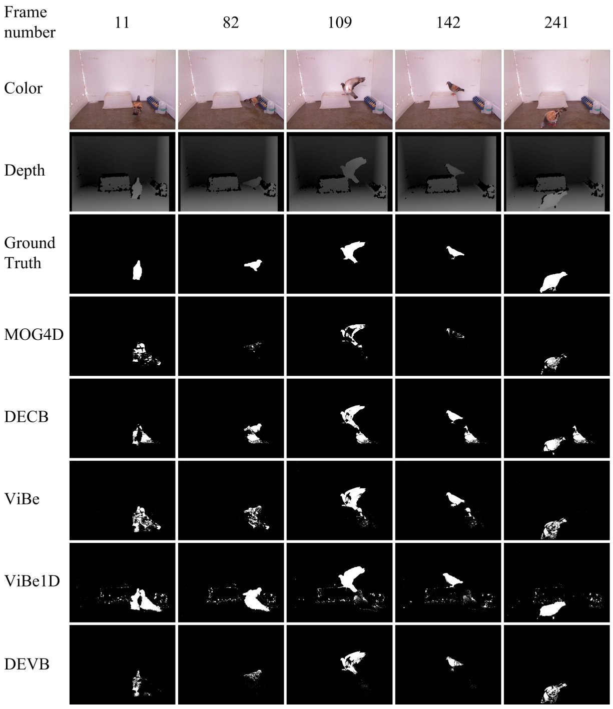

Figure 8. Results obtained in the test sequence *pigeon*.

Table 5. Segmentation evaluation for the *pigeon* sequence. The table shows the *PWC* results for the various approaches on five different evaluation frames as well as the mean values for this sequence.

Approach	Frame 11	Frame 82	Frame 109	Frame 142	Frame 241	Mean
DEVB	**0.972**	**0.626**	**0.897**	**0.204**	**1.191**	**0.778**
ViBe	1.837	1.126	1.415	0.603	1.268	1.250
ViBe1D	3.559	3.025	2.156	1.224	1.261	2.245
MOG4D	1.360	0.794	1.222	0.630	1.444	1.090
DECB	1.742	1.157	1.563	1.218	2.029	1.542

Finally, Figure 9 shows the average *PWC* value obtained by each approach on each sequence. According to this figure, DEVB yields the best results on every sequence. The *walking_occ*1 sequence is particularly complicated because of the high pedestrian flow; consequently, for each algorithm, the error rate is considerably increased.

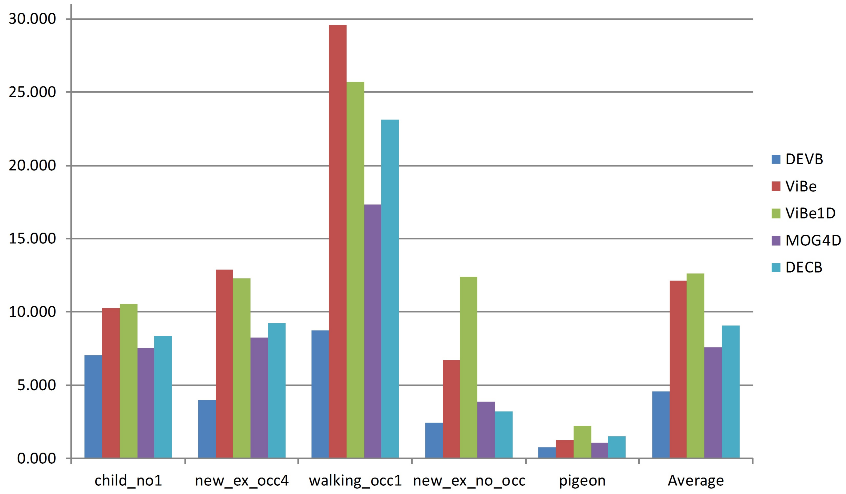

Figure 9. Average *PWC* values for each of the four sequences and for the entire dataset.

5. Conclusions

In this paper, we present an efficient moving object detection algorithm that fuses depth and color information. To incorporate the features of depth images, a depth model is designed in addition to a background model. The inspection mechanism for the classification of foreground pixels is further considered. Finally, we propose a new updating strategy based on the developed background and depth models, which can eliminate ghosting and black shadows almost completely. Experimental results indicate that our method is able to extract foregrounds efficiently, providing an excellent basis for the subsequent motion analysis of a scene. Our proposed method could serve as a convenient research tool for the detection of moving objects captured by the Kinect sensor.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (61471157); the Fundamental Research Funds for the Central Universities of China (2011B11114, 2012B07314, 2015B38214); the Natural Science Foundation of Jiangsu Province, China (BK20141157, BK20141159); Shandong Provincial Natural Science Foundation (ZR2014JL044); and the Open Foundation Programs of Changzhou Key Laboratory of Robotics and Intelligent Technology (CZSR2014003).

Author Contributions: Xiaoqin Zhou and Xiaofeng Liu devised the research, Xiaoqin Zhou, Xiaofeng Liu and Aimin Jiang designed methods and experiments; All authors wrote the paper. All authors have read and approved the final manuscript.

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

References

1. Kalal, Z.; Mikolajczyk, K.; Matas, J. Tracking-Learning-Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1409–1422.
2. Migliore, D.A.; Matteucci, M.; Naccari, M. A Revaluation of Frame Difference in Fast and Robust Motion Detection. In Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks, New York, NY, USA, 2006; pp. 215–218.
3. Elhabian, S.Y.; Elsayed, K.M.; Ahmed, S.H. Moving Object Detection in Spatial Domain using Background Removal Techniques - State-of-Art. *Recent Pat. Comput. Sci.* **2008**, *1*, 32–54.
4. Barron, J.L.; Fleet, D.J.; Beauchemin, S.S. Performance of optical flow techniques. *Int. J. Comput. Vis.* **1994**, *12*, 43–77.
5. Barjatya, A. Block matching algorithms for motion estimation. *IEEE Trans. Evolut. Comput.* **2004**, *8*, 225–239.
6. Amamra, A.; Mouats, T.; Aouf, N. GPU based GMM segmentation of kinect data. In Proceedings of the International Symposium Elmar, Zadar, Croatia, 10–12 September 2014; pp. 1–4.

7. Murgia, J.; Meurie, C.; Ruichek, Y. An Improved Colorimetric Invariants and RGB-Depth-Based Codebook Model for Background Subtraction Using Kinect. In *Human-Inspired Computing and Its Applications*; Springer: Berlin, Germany, 2014; pp. 380–392.
8. Barnich, O.; Van Droogenbroeck, M. ViBe: A universal background subtraction algorithm for video sequences. *IEEE Trans. Image Process.* **2011**, *20*, 1709–1724.
9. Yin, C.; Luo, Y.; Zhang, D. A novel background subtraction for intelligent surveillance in wireless network. In Proceedings of the Wireless Communications and Networking Conference, Istanbul, Turkey, 6–9 April 2014; pp. 3017–3021.
10. Kryjak, T.; Gorgon, M. Real-time implementation of the ViBe foreground object segmentation algorithm. In Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, Kraków, Poland, 8–11 September 2013; pp. 591–596.
11. Han, J.; Shao, L.; Xu, D.; Shotton, J. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE Trans. Cybern.* **2013**, *43*, 1318–1334.
12. Moeslund, T.B.; Hilton, A.; Krüger, V. A survey of advances in vision-based human motion capture and analysis. *Comput. Vis. Image Underst.* **2006**, *104*, 90–126.
13. Ji, X.; Liu, H. Advances in view-invariant human motion analysis: A review. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2010**, *40*, 13–24.
14. Spinello, L.; Kai, O.A. People detection in RGB-D data. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots & Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 3838–3843.
15. Xia, L.; Chen, C.C.; Aggarwal, J.K. Human detection using depth information by Kinect. *Appl. Phys. Lett.* **2011**, *85*, 5418–5420.
16. Ming, M.; Yang, F.; She, Q.; Yao, S.; Luo, Z. Human motion detection based on the depth image of Kinect. *Chin. J. Sci. Instrum.* **2015**, *36*, 386–393.
17. Crabb, R.; Tracey, C.; Puranik, A.; Davis, J. Real-time foreground segmentation via range and color imaging. In Proceedings of the CVPRW '08, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA , 23–28 June 2008; pp. 1–5.
18. Schiller, I.; Koch, R. Improved Video Segmentation by Adaptive Combination of Depth Keying and Mixture-of-Gaussians. In *Image Analysis: 17th Scandinavian Conference, SCIA 2011, Ystad, Sweden, May 2011. Proceedings*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 59–68.
19. Fernandez-Sanchez, E.J.; Javier, D.; Eduardo, R. Background Subtraction Based on Color and Depth Using Active Sensors. *Sensors* **2013**, *13*, 8895–8915.
20. Del Blanco, C.R.; Mantecón, T.; Camplani, M.; Jaureguizar, F.; Salgado, L.; García, N. Foreground Segmentation in Depth Imagery Using Depth and Spatial Dynamic Models for Video Surveillance Applications. *Sensors* **2014**, *14*, 1961–1987.
21. Camplani, M.; Salgado, L. Background foreground segmentation with RGB-D Kinect data: An efficient combination of classifiers. *J. Vis. Commun. Image Represent.* **2014**, *25*, 122–136.
22. Nguyen, V.T.; Vu, H.; Tran, T.H., An Efficient Combination of RGB and Depth for Background Subtraction. In *Some Current Advanced Researches on Information and Computer Science in Vietnam: Post-proceedings of the First NAFOSTED Conference on Information and Computer Science*; Springer: Cham, Germany, 2015; pp. 49–63.
23. Song, S.; Xiao, J. Tracking Revisited Using RGBD Camera: Unified Benchmark and Baselines. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, NSW, Australia, 1–8 December 2013; pp. 233–240.
24. Goyette, N.; Jodoin, P.M.; Porikli, F.; Konrad, J.; Ishwar, P. Changedetection. net: A New Change Detection Benchmark Dataset. In Proceedings of the 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, USA, 16–21 June 2012; pp. 1–8.