Moving object detection based on improved VIBE and graph cut optimization

Jianfang Dou*, Jianxun Li

Department of Automation, Shanghai Jiao Tong University, and Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

A R T I C L E   I N F O

Article history:
Received 30 November 2012
Accepted 28 April 2013

Keywords:
Background model
Moving object detection
VIBE
Mean shift
Feature clustering

A B S T R A C T

Extracting foreground moving objects from video sequences is an important task and also a hot topic in computer vision and image processing. Segmentation results can be used in many object-based video applications such as object-based video coding, content-based video retrieval, intelligent video surveillance and video-based human–computer interaction. In this paper, we present a novel moving object detection method based on improved VIBE and graph cut method from monocular video sequences. Firstly, perform moving object detection for the current frame based on improved VIBE method to extract the background and foreground information; then obtain the clusters of foreground and background respectively using mean shift clustering on the background and foreground information; Third, initialize the S/T Network with corresponding image pixels as nodes (except S/T node); calculate the data and smoothness term of graph; finally, use max flow/minimum cut to segmentation S/T network to extract the motion objects. Experimental results on indoor and outdoor videos demonstrate the efficiency of our proposed method.

© 2013 Elsevier GmbH. All rights reserved.

1. Introduction

Moving objects detection for a static camera has been extensively studied for many years [1,2]. Moving objects detection plays a very important role in many vision applications with the purpose of subtracting interesting target area and locating the moving objects from image sequences. It is widely used in vision systems such as traffic control, video surveillance of unattended outdoor environments, video surveillance of objects, activity recognition, object tracking and behavior understanding. Accurate moving object detection is essential for the robustness of intelligent video-surveillance systems.

Background subtraction and temporal differencing are two popular approaches to segment moving objects in an image sequence under a stationary camera. Background subtraction detects moving objects in an image by evaluating the difference of pixel features of the current scene image against the reference background image, such as the Gaussian mixture model (GMM) [3]. GMM is a widely used approach because of its self learning capacity and its robustness to variations in lighting. However, it still has some shortcomings. One problem is that it does not explicitly model the spatial dependencies of neighboring background pixels' colors. Therefore, some false positive pixels will be produced in highly dynamic scenes where dynamic texture does not repeat exactly. Temporal differencing such as W4 [4] use three parameters to model each pixel of background: the maximum of pixel gray, minimum, and the mean of two maximum differences of two successive frames in a period time. However, it is very effective to accommodate environmental changes and generally can only recover partial edge shapes of moving objects. For non-stationary cameras, optical flow method can be used, which assign to every pixel a 2-D velocity vector over a sequence of images. Moving objects are then detected based on the characteristics of the velocity vectors. Optical flow methods are computationally intensive, and can only detect partial edge shapes of moving objects. To improve the accuracy of the background model and to deal with highly dynamic scenes, the spatial information is exploited. Li [5] use spatial information at feature level, such as color and gradient. This can improve the accuracy of the background model and is most suitable for the stationary background. Olivier Barnich and Marc Van Droogenbroeck proposed Vibe algorithm [6,7]. The method adopt neighboring pixels to create the background model, by comparing the background model and the current input pixel values to detect targets, the method also gives three steps to update the field to adapt to changes in the environment. It uses randomly selected old samples to approximate the color distribution of the background. As described by the author, the advantage of the randomization is that it avoids replacement of the oldest samples. However, when the background have similar color with foreground
(i.e., camouflage problem), it will result in missing the foreground point and thus cause lack of accurate contour for post motion analysis such as recognition. And it also but slowed to eliminate ghost region. When the foreground object through these areas, the detection accuracy will drop; at the same time, goes up the high rate of false detection algorithm because of the ghosting region.

Recently, the graph-cut algorithm has been used to detect video moving objects though the energy minimization technique under the framework of MAP-MRF (Maximum a posterior-Markov random field). Most graph-cuts [8,9] algorithms focus on the iterative process and the priori information of moving objects in order to improve the detection accuracy. Stereo-based segmentation [10] seems to achieve the most robust results by fusing color, contrast and stereo matching information. But it requires special stereo input. However, graph cut requires labeling of the source and sink seeds by a human operator. Through our improved VIBE method, this can get initial foreground region and then provide the seeds of foreground and background.

In this paper, in order to overcome the shortcoming of origin VIBE and provide accurate silhouette with good spatial and temporal consistency, we present a novel moving object detection method based on improved VIBE and graph cut method from monocular video sequences. Firstly, perform moving object detection for the current frame based on improved VIBE method to extract the background and foreground information; Then obtain the clusters of foreground and background respectively using mean shift clustering on the background and foreground information; Third, initialize the S/T Network with corresponding image pixels as nodes (except S/T node); Calculate the data and smoothness term of graph; Finally, use max flow/minimum cut to segmentation S/T network to extract the motion objects. Experimental results on indoor and outdoor videos demonstrate the efficiency of our proposed method.

The paper is organized as follows: Section 2 reviews the previous work related to layer extraction. Section 3 addresses how to extract layer descriptions from a short video clip. Section 4 deals with the use of the occlusion order constraint, three-state pixel graph, and a multiframe graph cuts algorithm for obtaining layer segmentation in the presence of occlusion. Finally, in Section 5, we demonstrate several results obtained by our method.

2. Review of VIBE algorithm

Visual Background Extractor (ViBe) as a universal background subtraction method is proposed by Olivier. The algorithm adopts neighboring pixels to establish the background model by comparing the background model with the current pixel value. The implementation of the algorithm is subdivided into three steps [7]:

The first step is to initialize the single frame image from each pixel in the background model. Since there is no temporal information in a single frame, it is supposed that neighboring pixels share a similar temporal distribution. This means that the value of a pixel and its neighbor pixel values in spatial domain has a similar distribution. The right size range of the neighborhood is chosen to make sure that the background model includes a sufficient number of different samples, while keeping in mind that as the neighborhood scope increases, the correlation between pixel values at different locations decreases. It is assumed that \( T = 0 \) denotes the first frame; \( B^0(x, y) \), \( N_C(x, y) \), \( P^0(x, y) \) is the pixel background model value, spatial neighborhood value, pixel value, then:

\[
B^0(x, y) = \{P^0_m(x^m, y^m)| (x^m, y^m) \in N_C(x, y)\}
\]

where \((x^m, y^m)\) in \( N_C(x, y) \) is selected such as probability. \( m = 1, 2, \ldots, N \) is the number of samples.

Secondly there is a simple decision process to determine whether an input pixel belongs to the background or not, then to update the background pixel model. They denote by the pixel value \( P(x, y) \) in a given Euclidean color space taken by the pixel located at \((x, y)\) in the image, and each background model is modeled by a collection of 20 background samples. When \( T = t \), the background model of \((x, y)\) is \( B^t(x, y) \), the pixel value is \( P(x, y) \), the formula for determining the input frame image for foreground objects segmentation is as follows:

\[
P^t(x, y) = \begin{cases} 
\text{foreground} & |B^t(x, y) - P(x, y)| > R \\
\text{background} & |B^t(x, y) - P(x, y)| < R 
\end{cases}
\]

where superscript \( r \) is randomly chosen; \( R \) is the fixed shold (which in color space is a spherical radius). When \( P(x, y) \) is larger than or equal to a given background candidate \#min (the minimal cardinality), we think it is the corresponding background pixel, otherwise is foreground. In ViBe algorithm, \#min is set to 2. It means that it is sufficient to find at least two samples close enough to classify the pixel in the background.

Because the existence of a moving object in the first frame will introduce an artifact commonly called a ghost (regions of connected foreground points that do not correspond to any real object). Although using the subsequent frames can update the background to remove the ghost, the process is a little slow. Because of the existence of ghost, the accuracy of detection will be decreased. Otherwise, the detection result of ViBe algorithm is sensitive to illumination changes, dynamic scenes, cluttered background.

A good background subtraction method has to be adapted to changes of the background caused by different lighting conditions but also to those caused by the addition, removal, or displacement of some of its parts, camera jitter.

The third step is to randomly update the background model with each new frame. Because of the strong statistical correlation between a pixel and its neighbor pixel, when a pixel is detected as the background pixel, its neighbor pixel is highly possible to be considered as the background pixel with high possibility. Consequently, it allows using the background pixels to update the background model of neighboring pixels. Through the random updating policy we can merge the foreground objects which halt suddenly or stay long into the background model.

3. Our proposed algorithm

The block diagram of our algorithm is demonstrated by Fig. 1. The steps of our proposed approach – foreground segmentation for moving object in this paper can be briefly summarized as follows:
Fig. 2. Illustration of improved foreground object detection.

(1) Firstly, perform moving object detection for the current frame based on improved VIBE method to extract the background \( P^O(p,t) \) and foreground information \( P^F(p,t) \).

(2) For any pixel \( p \) in the current frame, if \( p \) is foreground pixel, then add it to \( P^F \), \( P^F \) is the set of foreground pixel; if \( p \) is background pixel, then add it to \( P^O \), \( P^O \) is the set of background pixel;

(3) Then obtain the clusters of foreground and background \( (R^F_i) \) \( (i = 1, 2, \ldots , m) \) and \( (R^O_i) \) \( (j = 1, 2, \ldots , n) \) respectively using mean shift clustering on the pixel of \( P^F \) and \( P^O \);

(4) Initialize the S/T Network model with corresponding image pixels as nodes (except S/T node);

(5) Calculate Tlink and Nlink to build likelihood energy function according to Eqs. (16)–(19) and construct Graph Cut model;

(6) Use max flow/minimum cut to segmentation S/T network to get binary label of each node.

(7) If one node is labeled S, then its corresponding image pixel is foreground; other wise is background, then get the current foreground object mask.

We will give a detail description of our proposed algorithm GC_STVIBE in the following section.

3.1. Improved VIBE

3.1.1. Foreground detection

For the second step of classify the current pixel as foreground or background, the origin VIBE is based on pixel level. Based on our experiment, we found that if background have similar color as foreground, the foreground cannot detected correctly, that is to say, the boundary of the detection is inaccurate and is not good for the post processing such as perform object tracking and recognition. In order to improved this limitation, we detect the foreground at region level. Fig. 2 illustration of improved foreground detection.

Denote \( O \) is the current frame pixel at \( (x,y) \), \( R_0 \) is \( 3 \times 3 \) neighbor of \( O \); \( C_i(i = 1, 2, \ldots , N) \) is the corresponding position of background model, \( R(i = 1, 2, \ldots , N) \) is \( 3 \times 3 \) neighbor of \( C_i \). \( N \) is the number of samples.

\[
SSDIO(R_i, R_0) = \sum_{n=1}^{m} \sum_{m=1}^{m} |R_i(x + n, y + m) - R(x + n, y + m)|^2
\]

where \( i = 1, 2, \ldots , N \). \( cnt \) is used to count the number of samples which meet \( SSDIO(R_i, R_0) < \text{thr} \).

We redefine the formula of foreground object segmentation as below:

\[
cnt = \begin{cases} \text{cnt} = \text{cnt} + 1 & \text{SSDIO}(R_i, R_0) < \text{thr} \\ \text{cnt} & \text{SSDIO}(R_i, R_0) \geq \text{thr} \end{cases}
\]

\[
P^F(x,y) = \begin{cases} \text{foreground} & \text{cnt} < 2 \\ \text{background} & \text{cnt} \geq 2 \end{cases}
\]

3.1.2. Background model update

In the third step of origin VIBE algorithm, the updating of the background model is also based on pixel level instead of region level, this does not consider the neighbor statistics efficiently and may cause discontinuity for the background pixel, thus increase the missing foreground. In our improved method, we update the background model at region level, as \( 3 \times 3 \) neighborhood of pixel. This is shown in Fig. 3.

Suppose current pixel \( f_{g}(x,y) \) for the detected foreground mask, we will update the corresponding background model pixel if these condition below meets:

\[
\text{sum} = \sum_{n=1}^{m} \sum_{m=1}^{m} f_{g}(x + n, y + m)
\]

\[
\text{Update(Model}(x,y)) = \begin{cases} 1 & \text{if } \text{sum} = 0 \\ 0 & \text{else} \end{cases}
\]

3.2. Mean shift clustering

3.2.1. Mean shift for mixed feature spaces

An appealing technique to extract the clusters is the mean shift (MS) algorithm, which does not require to fix the (maximum) number of clusters. On the other hand the kernel bandwidth and shape for each dimension has to be chosen or estimated. Mean shift is an iterative gradient ascent method used to locate the density modes of a cloud of points, i.e. the local maxima of its density [13]. In [13], two applications of the feature-space analysis technique are discussed based on the MS procedure: discontinuity preserving filtering and the segmentation of gray-level or color images. Here the theory is briefly reminded [11–13]. Given the set of points \( \{X(i)\}_{i=1,2,...,M} \) in the \( d \)-dimensional space \( R^d \), the non-parametric density estimation at each point \( x \) is given by:

\[
f_{H,d}(X) = \frac{1}{n(2\pi)^{d/2}||H||^{1/2}} \sum_{i=1}^{M} k(||H^{-1/2}(X - X(i))||^2)
\]

where \( k(x) = \int_{-\infty}^{\infty} c_{k,d} k(||x||^2) dx = 1 \), \( c_{k,d} \) is normalization constant and \( c_{k,d} > 0 \) is a kernel profile, which is a monotonically decreasing function and \( H \) the bandwidth matrix. Introducing the notation

\[
g(X) = -k'(X)
\]
leads to the density gradient:

$$\nabla f_{H_g}(X) = H^{-1} f_{H_g}(X) m_{H_g}(X)$$  \hspace{1cm} (10)

where $m_{H_g}(X)$ is the “mean shift” vector,

$$m_{H_g}(X) = \frac{\sum_{i=1}^{M} X_i g(|H^{-1/2}(X - X_i)|^2) - X}{\sum_{i=1}^{M} g(|H^{-1/2}(X - X_i)|^2)}$$  \hspace{1cm} (11)

Using exactly this displacement vector at each step guarantees convergence to the local maximum of the density. With a $d$-variate Gaussian kernel, Eq. (11) becomes

$$m_{H_g}(X) = \frac{\sum_{i=1}^{M} X_i \exp(-1/2D^2(X, X_i; H)) - X}{\sum_{i=1}^{M} \exp(-1/2D^2(X, X_i; H))}$$  \hspace{1cm} (12)

where

$$D^2(X, X_i; H) = (X - X_i)^T H^{-1} (X - X_i)$$  \hspace{1cm} (13)

is the Mahalanobis distance from $X$ to $X_i$.

Assume now that the $d$-dimensional space can be decomposed as the Cartesian product of $S$ (2 in our case) independent spaces associated to different types of information (e.g. position, color), also called feature spaces or domains, with dimensions $d_1, s = 1, 2, \ldots, S$ (where $\sum_{s=1}^{S} d_s = d$). Because the different types of information are independent, the bandwidth matrix $H$ becomes $H = \text{diag}[H_1, H_2, \ldots, H_S]$ and thus the mean shift vector can be rewritten as

$$m_{H_g}(X) = \frac{\sum_{i=1}^{M} X_i \prod_{s=1}^{S} \exp(-1/2D^2(X_s, X_i^s; H_s)) - X}{\sum_{i=1}^{M} \prod_{s=1}^{S} \exp(-1/2D^2(X_s, X_i^s; H_s))}$$  \hspace{1cm} (14)

where $X_i^T = (X_i^{(s)}_1, X_i^{(s)}_2, \ldots, X_i^{(s)}_{d_s})$ and $X^T = (X_1^T, X_2^T, \ldots, X_M^T)$.

The mean shift filtering is obtained by successive computations of Eq. (12) or Eq. (14) and translation of the kernel according to the mean shift vector. This procedure converges to the local mode of the density [13].

For color image segmentation, the feature vector adopted in mean shift clustering is the concatenation of the two spatial domain coordinates and the three color values in a given color space. Mean shift procedure is applied in the joint spatial-color domains. The distance between feature vectors is measured by the Euclidean distance. CIE $L^\ast u^\ast v^\ast$ color space is usually preferred to the RGB color space since Euclidean distance defined in the $L^\ast u^\ast v^\ast$ space better approximates human perception of color distance. The segmentation is actually a merging process performed on a region that is produced by the MS filtering.

Mean shift algorithm is described as the following:

1) Choose the radius $h$ of the search window.
2) Choose the initial location of the window.
3) Compute the mean shift vector and translate the search window by that amount.
4) Repeat till convergence.

### 3.2.2. Bandwidth selection

The partition of the feature space is obtained by grouping together all the data points whose associated mean shift procedures converged to the same mode. The quality of the results highly depends on the choice of the bandwidth matrix $H$. In [12], Comaniciu proposes to find the best bandwidths within a range of $B$ predefined matrices ($H^b, b = 1, 2, \ldots, B$). Mean Shift partitioning is first run at each scale (for $b$ varying from 1 to $B$). For each data point $X_i$, an analysis of the sequence of clusters to which the point is associated is performed. The scale for which the cluster is the most stable is selected, along with associated bandwidth, for data point $X_i$. Therefore, the algorithm can be decomposed in two steps. The first one is called bandwidth evaluation at the partition level. It consists in finding a parametric representation of each cluster in order to do the comparisons. The second step called evaluation at the data level is the analysis of cluster sequences at each data point.

### 3.3. Unsupervised learning based on mean shift clustering

Extract the background and foreground information of current time $T$ using improved VIBE method, then foreground information is labeled as white while the background is labeled as black. We use mean shift to cluster on the background and foreground information of the current frame and obtain foreground clusters $\{R^i_b\}_{i = 1, 2, \ldots, m}$, background clusters $\{R^j_b\}_{j = 1, 2, \ldots, n}$, $m$ is the number of foreground clusters and $n$ is the number of

![Fig. 4](image1.png)

*Fig. 4.* An example of a graph $G$ for a 1D image. Nodes $p$, $q$, $r$, and $o$ correspond to the pixels in the image. After computing a minimum cut $C$, the nodes are partitioned into supporting pixels $p$, $q$ (source) and unsupported pixels $r$, $o$ (sink). The weights of the links are listed in the table on the right.

![Fig. 5](image2.png)

*Fig. 5.* Curtain (Frame 1847) comparative foreground segmentation methods. (a) Input image; (b) ground-truth; (c) ACML; (d) GMM; (e) VIBE; (f) GC-STVIBE.
background clusters. \( F \) denotes as foreground, \( B \) for background. The feature vector in mean shift clustering is \( [x, y, R, G, B] \).

### 3.4. Graph Cut

In this paper, we use terminology and notations similar to [9]. For example, Fig. 4 shows a typical weighted graph with four nodes. This graph \( G = (V, \mathcal{E}) \) is defined by a set of nodes \( V \) (image pixels) and a set of undirected edges \( \mathcal{E} \) which connect these nodes as shown in Fig. 4. In this graph, there are two distinct nodes (or terminals) \( s \) and \( t \), called the source and sink, respectively. The edges connected to the source or sink are called \( f \)-links, such as \((s,p)\) and \((p,t)\). The edges connected to two neighboring pixel nodes are called \( n \)-links, which are bidirectional, such as \((p,q)\) and \((q,p)\). The weights of these \( n \)-links in both directions may not be equal. A cut \( C \) is a subset of edges which separates the nodes into two parts; one part belongs to the source and the other belongs to the sink. The cost of a cut is the summation of the weights of its edges. The minimum cut of the graph will generate an optimal segmentation in the image. In the application, the given source and sink to the graph cuts. A cut with \( s/t \) on a graph is to set two disjoint subsets \( S \) and \( T \) such that the source \( s \) is in \( S \) and the sink \( t \) is in \( T \). The minimum cut problem on a graph is to find a cut that has the minimum cost among all cuts. One of the fundamental results in combinatorial optimization is that the minimum \( s/t \) cut can be solved by finding the maximum flow from the source \( s \) to the sink \( t \). Generally speaking, maximum flow is the maximum amount of water that can be sent from the source \( s \) to the sink \( t \) with the graph edge is the pipe and edge weight is the capacity of the pipe. The experiment [14] shows the maximum flow is equal to the minimum cut. Most of the algorithms are push-label or augmenting paths to find the minimum cut. Augmenting paths always push flow along the unsaturated paths from the source to the sink until the maximum flow in graph \( G \) is reached. Push-label method uses quite a different approach. They do not valid flow during the operation, there are active nodes that have a positive weight that allow the flow access, the algorithm use a labeling of nodes giving a low bound estimate on the distance to the sink on unsaturated path. We will use the first method to do our approach.

Given a labeling system \( f \), each pixel in the image will be assigned one label. The graph cuts algorithm seek energy minimum in the computer vision. This problem can be naturally formulated in the terms of energy minimum. In this framework, it seek the label \( f \) that minimum the energy as:

\[
E = E_{\text{data}}(f) + E_{\text{smooth}}(f) = \sum_{p \in P} D(f_p, p) + \sum_{(p,q) \in N} V(p, q) \cdot T(f_p \neq f_q) \tag{15}
\]

where is \( E_{\text{data}} \) a data error term, \( E_{\text{smooth}} \) is a piecewise smoothness term, \( P \) is the set of pixels in the image, \( D(f_p, p) \) is a data penalty function when it is assigned a label \( f_p, V(p,q) \) is a smoothness penalty function between two neighboring pixels \( p \) and \( q \). \( N \) is a four-neighbor system, \( f_p \) is the label of a pixel, and \( T(\cdot) \) is 1 if its argument is true and 0 otherwise. In this bi-partitioning problem, the label \( f_p \) is either 0 or 1. If \( f_p = 1 \), the pixel \( p \) is supporting the seed region, otherwise, this pixel is not supporting the region.
3.5. Constructing network flow graph

3.5.1. Initialize the S/T network model

In the S/T Network model, S node denotes object, T node denotes background. They are not exist in fact and do not corresponding any pixel in the image. For other nodes, one of each node express the corresponding pixel in the image and is one to one mapping.

3.5.2. Calculate the weight of Tlink

Firstly, for each node $p$ (except S/T node), calculates the minimum distance and the background and foreground clusters according to Eqs. (16) and (17).

$$d_f^p = \min \| C(p) - R^f_i \|$$

$$d_b^p = \min \| C(p) - R^b_j \|$$

where $d_f^p$ is the minimum distance between node $p$ and foreground clusters, $d_b^p$ is the minimum distance between node $p$ and background clusters. Calculate $s$-tlink and $t$-tlink of each node according to Eq. (18).

$$V_{p}(s-tlink) = \frac{d_b^p}{d_b^p + d_f^p} \cdot \lambda, V_{p}(t-tlink) = 0 \quad \forall p \in F$$

$$V_{p}(t-tlink) = \frac{d_f^p}{d_b^p + d_f^p} \cdot \lambda, V_{p}(s-tlink) = 0 \quad \forall p \in B$$

where $V_{p}(s-tlink)$ is the weight of arc between node $p$ and node $S$, $V_{p}(t-tlink)$ is the weight of arc between node $p$ and node $T$. $\lambda$ is the weight coefficients.

3.5.3. Calculate the weight of Nlink

Nlink is the arc for nodes $p$ and $q$ with the neighbor relationship. For the neighboring nodes $p$ and $q$, Nlink is calculated according to Eq. (19).

$$V_{p,q}(nlink) = \frac{1}{{(1 + \exp(||C(p) - C(q)||^2)}}$$

where $V_{p,q}(nlink)$ is weight of neighboring nodes $p$ and $q$. 
After build the Graph Cut model, we can get the result of foreground segmentation through max flow/minimum cut for S/T network.

4. Experiment and results

In this section, we experimentally test some result in the indoor and outdoor scenes with moving object; these situation will consider in our paper, the main goal is to prove our method is effective and result is relative accuracy.

The indoor and outdoor video frames are obtained from web [15]. We then compare our proposed GC_STViBe method with three existing algorithms on indoor and outdoor scenes. The visual examples and quantitative evaluations of the experiments are described in the following subsections.

4.1. Quantitative evaluations

Quantitative evaluation of proposed method and comparison with three existing methods were also performed in this study. The results were evaluated quantitatively from the comparison with the “ground truths” in terms of:

1. The Number of False Positive (FP): the number of background points that are mis-detected as foreground.
2. The Number of True Positives (TP): the number of foreground points that are detected as foreground points.

4.2. Experiment results

Curtain video is one of Office environments of indoor view. An office environment is usually composed of stationary background objects. The difficulties for foreground detection in these scenes can be caused by shadows, changes of illumination conditions, and camouflage foreground objects (i.e. the color of the foreground object is similar to that of the covered background). In some cases, background may consist of dynamic objects, such as waving curtains, running fans, and flickering screens. Water Surface is another type of environment of outdoor view. Changes for this type in the background are often caused by motion of water, or the changes in the weather.

The segmentation results for four example frames of Curtain video and Water Surface video are shown in Figs. 5–8 and Figs. 10–13. In each figure, the displayed images are: (a) a frame from the video, (b) ground truth, (b) the result of ACMLI, (c) the result of GMM, (d) the result of VIBE, (e) the result of the proposed method. Figs. 9 and 14 are the illustrations of TP for four example frames of Curtain and Water Surface videos.

<table>
<thead>
<tr>
<th>Method frames</th>
<th>ACMLI</th>
<th>GMM</th>
<th>VIBE</th>
<th>GC_STViBe</th>
</tr>
</thead>
<tbody>
<tr>
<td>1847</td>
<td>(518)</td>
<td>(718)</td>
<td>(569)</td>
<td>(883)</td>
</tr>
<tr>
<td>2266</td>
<td>(154)</td>
<td>(613)</td>
<td>(386)</td>
<td>(774)</td>
</tr>
<tr>
<td>2801</td>
<td>(323)</td>
<td>(818)</td>
<td>(517)</td>
<td>(851)</td>
</tr>
<tr>
<td>2893</td>
<td>(98)</td>
<td>(779)</td>
<td>(475)</td>
<td>(895)</td>
</tr>
</tbody>
</table>
The quantitative evaluation results over the examples displayed in Figs. 5–8 and Figs. 10–13 are shown in Tables 1 and 2 respectively. The quantitative evaluation agrees with the conclusions from the visual observation of the experimental results (Fig. 14).

5. Conclusion

In this paper, we present a novel moving object detection method based on improved VIBE and graph cut method from monocular video sequences. Firstly, perform moving object detection for the current frame based on improved VIBE method to extract the background and foreground information; then obtain the clusters of foreground and background respectively using mean shift clustering on the background and foreground information; third, initialize the S/T Network with corresponding image pixels as nodes (except S/T node); calculate the data and smoothness term of graph; finally, use max flow/minimum cut to segmentation S/T network to extract the motion objects. Experimental results show that the quantitative detection indicators of the proposed algorithm perform better in indoor and outdoor conditions.

Acknowledgments

This work was jointly supported by National Natural Science Foundation (61175008); 973 Project (2009CB824900, 2010CB734103); the Space Foundation of Supporting-Technology no. 2011–HT–SHJ002.

References