EURASIP Journal on Image and Video Processing
a SpringerOpen Journal

RESEARCH Open Access

Background initialization and foreground segmentation for bootstrapping video sequences

Han-Hui Hsiao and Jin-Jang Leou*

Abstract

In this study, an effective background initialization and foreground segmentation approach for bootstrapping video sequences is proposed. First, a modified block representation approach is used to classify each block of the current video frame into one of four categories, namely, "background," "still object," "illumination change," and "moving object." Then, a new background updating scheme is developed, in which a side-match measure is used to determine whether the background is exposed. Finally, using the edge information, an improved noise removal and shadow suppression procedure with two morphological operations is adopted to enhance the final segmented foreground. Based on the experimental results obtained in this study, as compared with three comparison approaches, the proposed approach produces better background initialization and foreground segmentation results.

Keywords: Background initialization, Foreground segmentation, Side-match measure, Block representation, Shadow suppression

1. Introduction

The main purpose of foreground/background segmentation, a basic process of a computer vision application system, is to extract some interesting objects (the foreground) from the rest (the background) of each video frame in a video sequence [1]. Background subtraction is a popular foreground/background segmentation approach, which detects the foreground by thresholding the difference between the current video frame and the modeled background in a pixel-by-pixel manner [2]. The correctness of the modeled background is usually affected by three factors [3]: (1) illumination changes; (2) dynamic backgrounds: some "moving" objects, such as waving trees, fountains, and flickering monitors, are not interested for a vision-based surveillance system; and (3) shadows: foreground objects often cast shadows, which are different from the modeled background.

A background subtraction approach usually considers three main issues: background representation, background updating, and background initialization [1]. For the popular background subtraction approach called the Gaussian background model, Stauffer and Grimson [4] presented a pixel-wise background representation scheme using the mixture of Gaussians (MoG) and pixel-wise background updating to update the intensity mean and variance of each pixel in real-time. The MoG-based methods are effective for dynamic background scenes with multiple background variations, but they are sensitive to noise and illumination changes. Several existing MoG-based approaches are proposed to improve their performances by adaptation of some MoG parameters [5], such as the number of components [6,7], weights, mean, and variance [8-11], learning rate [8,9,12,13], and feature type [9,14-17], and by smoothing among spatially and temporally neighboring pixels using spatial and temporal dependencies [18]. In general, a training duration without foreground objects (non-bootstrapping) is required and some ghost (false positive) objects may be detected when some foreground objects change their motion status (static or moving) suddenly.

Recently, the background subtraction methods focused on background initialization for bootstrapping video sequences [19-24], in which a training duration without foreground objects is not available in some cluttered environments [3,19]. That is, background initialization for bootstrapping video sequences can be defined as follows: given a video sequence captured by a

* Correspondence: jjleou@cs.ccu.edu.tw
Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan

stationary camera, in which the background is occluded by some foreground objects in each frame of the video sequence, the aim is to estimate a background frame without foreground objects [22,24]. Background initialization for bootstrapping video sequences (or simply background initialization) is widely used in the intelligent video surveillance systems for monitoring crowded infrastructures, such as banks, subway, airports, and lobby.

Two simple background initialization techniques are the pixel-wise temporal mean and median filters over a large number of video frames [20,21]. For the pixel-wise temporal median filter, it is assumed that for each pixel within the estimation duration, the exposure of the background must be more than that of the foreground. Based on the block-wise strategy, Farin et al. [19] used a block similarity matrix to segment the input video frames into foreground and background regions, which contain the block-wise temporal differences between any video frame pair. Reddy et al. [22] proposed a block selection approach using the discrete cosine transform (DCT) among some neighboring blocks to estimate the unconstructed parts of the background. This approach is usually degraded by similar frequency content within a block candidate set and error propagation if some blocks in a video frame are erroneously estimated. Note that, to obtain the processing results, the whole video sequence should be available to Reddy et al.'s approach. Then, the DCT is replaced by the Hadamard transform to reduce the computation time for block selection [23]. In addition, a block selection refinement step using spatial continuity along block borders is added to prevent erroneous block selection. Most block-wise background initialization approaches need large memories and are computationally expensive. Furthermore, one free-background video frame is usually obtained as its output during the "learning" duration.

For the frame-wise strategy with temporal smoothing, the first video frame of a video sequence is usually treated as the initial background for background initialization. Most background initialization approaches maintain a modeled background by iterative updating with temporal smoothing between each input video frame and the modeled background. Liu and Chen [25] proposed a background modeling method, in which the background similarity using the mean and variance information is adopted to identify the background image. Moreover, Scott et al. [26] updated the mean and variance information by Kalman filter updating equations for maintaining the modeled background. Maddalena and Petrosino [27] automatically generated the background model without prior knowledge by using self-organizing artificial neural networks. Each color pixel is represented by $n \times n$ weight vectors to form a neural map. It is claimed that they can handle bootstrapping scenes containing dynamic backgrounds, gradual illumination changes, and shadows. Using the growing self-organizing map, Ghasemi and Safabakhsh [28] generated a codebook for detecting moving objects in the dynamic background scenes. The major advantage of the methods using variant self-organizing maps [27,28] is low computational complexity. Chiu et al. [29] proposed a pixel-wise color background modeling approach using probability theory and clustering. To estimate the modeled background completely, a suitable time duration is required, because each of the R, G, and B color components is iteratively updated by increasing/decreasing 1 in the range of 0–255. The main weakness for the background initialization and foreground segmentation approaches using the frame-wise strategy with temporal smoothing is that the "erroneous" parts in the modeled background are slowly updated. Furthermore, this type of approaches can work properly only when the video sequence contains fast "moving" foreground objects so that the background is exposed most of the time.

On the other hand, within some existing approaches [30-34], temporal smoothing is not adopted in background updating. Chein et al. [30] proposed a pixel-wise video segmentation approach with adaptive thresholding to determine each pixel as a moving or stationary one. Each pixel in the modeled background is then replaced by the corresponding pixel in the current video frame if the pixel is detected as a stationary one for some time duration. That is, this type of approaches might not work well in illumination-changing environments. Verdant et al. [31] proposed three analog-domain motion detection algorithms in video surveillance, namely, the scene-based adaptive algorithm, the recursive average with estimator algorithm, and the adaptive wrapping thresholding algorithm, in which background estimation and variance of each pixel are computed with nonlinear operations to perform adaptive local thresholding. Lin et al. [32] used a classifier to determine whether an image block belongs to the background for block-wise background updating. The classifier using two learning methods, namely, the support vector machine and column generation boost, is trained by some training data, which are manually labeled as foreground/background blocks before background initialization. In addition, some foreground prediction approaches may segment accuracy foreground without background modeling. For example, Tang et al. [33] proposed a foreground prediction algorithm, which estimates each pixel in the current video frame belonging to the foreground one. Given a segmentation result (an alpha matte) of the previous video frame as an opacity map, the opacity values [0–1] in an opacity map are propagated from the previous video frame to the current video frame using the foreground prediction algorithm. It was claimed that the foreground can be predicted accurately in sudden illumination changes. Zhao et al. [34]

proposed a learning-based background subtraction approach based on sparse representation and dictionary learning. They made two important assumptions, which enabled their approach to handle both sudden and gradual background changes.

In this study, an effective background initialization and foreground segmentation approach for bootstrapping video sequences is proposed, which contains a block-wise background initialization procedure and a pixel-wise foreground segmentation procedure. First, a modified block representation approach is used to classify each block of the current video frame into one of four categories. Then, a new background updating scheme is developed, in which a side-match measure is used to determine whether the background is exposed so that the modeled background can be well determined. Finally, using the edge information, an improved noise removal, and shadow suppression procedure with two morphological operations is adopted to enhance the final foreground segmentation results. The main contributions of the proposed approach include: (1) using motion estimation and correlation coefficient computation to perform block representation (classification); (2) developing four types of background updating for four types of block representation; (3) using side-match measure to perform background updating of "moving object" blocks; and (4) using a modified noise removal and shadow suppression procedure to improve final foreground segmentation results.

This article is organized as follows. In Section 2, the proposed background initialization and foreground segmentation approach is addressed. Experimental results are described in Section 3, followed by concluding remarks given in Section 4.

2. Proposed background initialization and foreground segmentation approach

Figure 1 shows the framework of the proposed video background initialization and foreground segmentation approach for bootstrapping video sequences, which contains four major processing steps, namely, block representation, background updating, initial segmented foreground, and noise removal and shadow suppression with two morphological operations. In Figure 1, the input includes the current (gray-level) video frame I^t and the previous (gray-level) video frame I^{t-1} of a bootstrapping video sequence, and the output includes the modeled background frame B^t and the segmented foreground frame F^t, where i denotes the frame number (index). Here, $I^t_{(x,y)}$, $I^{t-1}_{(x,y)}$, $B^t_{(x,y)}$, and $F^t_{(x,y)}$ denote pixels (x,y) in I^t, I^{t-1}, B^t, and F^t, respectively. Each video frame is $W \times H$ (pixels) in size, and each video frame is partitioned into non-overlapping and equal-sized blocks of size $N \times N$ (pixels). Let (i,j) be the block index, where $i = 0,1,2,\ldots,(W/N) - 1$ and $j = 0,1,2,\ldots,(H/N) - 1$. Here, $\mathbf{b}^t_{(i,j)} = \{I^t_{(iN+a,jN+b)}$: $a,\ b = 0,\ 1,\ 2,\ldots,N - 1\}$, $\mathbf{b}^{t-1}_{(i,j)} = \{I^{t-1}_{(iN+a,jN+b)}$: $a,\ b = 0,\ 1,\ 2,\ldots,N - 1\}$, and $\widetilde{b}^t_{(i,j)} = \left\{B^t_{(iN+a,jN+b)} : a,b = 0,1,2,\ldots,N-1\right\}$, denote blocks (i,j) in I^t, I^{t-1}, and B^t, respectively. In addition, let $\hat{B}^t$ denote the initial modeled background frame and $\hat{b}_{(i,j)}t = \left\{\hat{B}^t_{(iN+a,jN+b)} : a,b = 0,1,2,\ldots,N-1\right\}$, denote block (i,j) in $\hat{B}^t$.

2.1. Initial modeled background processing

As the illustrated example shown in Figure 2, a sequence of initial modeled background frames $\hat{B}^t$ ($t = 1,2,\ldots$) will be obtained in the initial modeled background processing procedure. At the beginning ($t = 1$), each block $\hat{b}_{(i,j)}1$ of size $N \times N$ (pixels) in $\hat{B}^1$ is set to "undefined" (labeled in black), as shown in Figure 2l. Then, the initial modeled background frame $\hat{B}^t$ ($t = 2,3,\ldots,19$) is obtained based on the "updated" modeled background frame B^{t-1} (see Section 2.3) and the block motion representation frame $\hat{R}^t$. Each block of size $N \times N$ (pixels) in $\hat{R}^t$ is determined as either a "static" block (labeled in blue) or a "moving" block (labeled in red) by motion estimation (see Section 2.2) between two consecutive (gray-level) video frames I^{t-1} and I^t of the bootstrapping video sequence, as shown in Figure 2g–k. For one "undefined" block $\hat{b}_{(i,j)}t-1$ in $\hat{B}^{t-1}$, if its corresponding block in $\hat{R}^t$ is

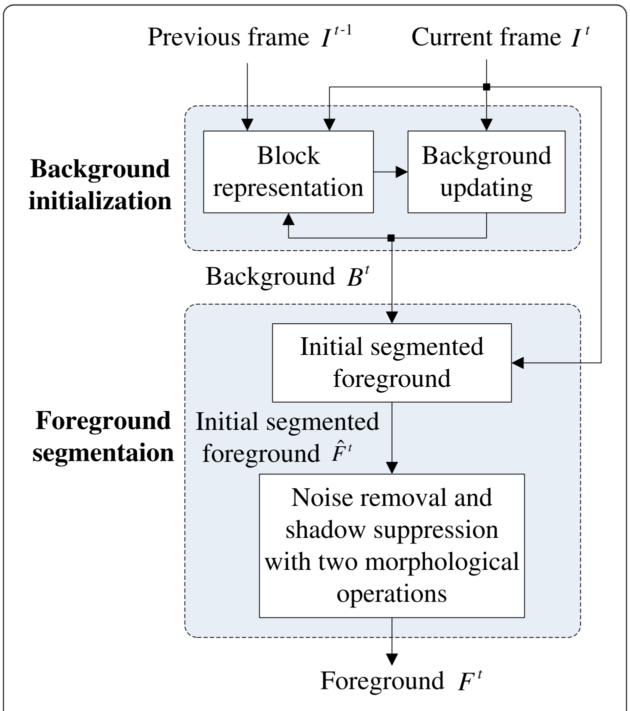
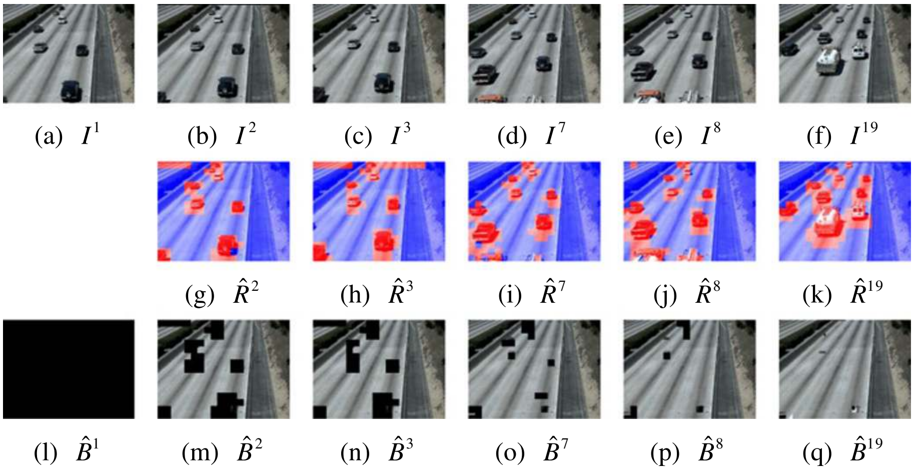

Previous frame I^{t-1}
Current frame I^t
Background initialization
Block representation
Background updating
Background B^t
Initial segmented foreground
Foreground segmentaion
Initial segmented foreground $\hat{F}^t$
Noise removal and shadow suppression with two morphological operations
Foreground F^t

Figure 1 The framework of the proposed video background initialization and foreground segmentation approach.


(a) I^1 (b) I^2 (c) I^3 (d) I^7 (e) I^8 (f) I^{19}

(g) $\hat{R}^2$ (h) $\hat{R}^3$ (i) $\hat{R}^7$ (j) $\hat{R}^8$ (k) $\hat{R}^{19}$

(l) $\hat{B}^1$ (m) $\hat{B}^2$ (n) $\hat{B}^3$ (o) $\hat{B}^7$ (p) $\hat{B}^8$ (q) $\hat{B}^{19}$

Figure 2 An illustrated example of initial modeled background processing: (a)-(f) the original video frames; (g)-(k) the block motion representation frames; (l)-(q) the initial modeled background frames.

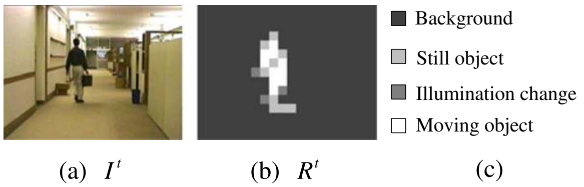
determined as a "static" block, i.e., its motion vector is (0,0), the "static" block $\hat{b}_{(i,j)}t$ in $\hat{B}^t$ is duplicated from the corresponding block $\mathbf{b}^t_{(i,j)}$ in I^t. Then, each "static" block $\hat{b}_{(i,j)}t$ in $\hat{B}^t$ will perform the background updating procedure (see Section 2.3) to obtain $\widetilde{b}^t_{(i,j)}$ in B^t Otherwise, the "undefined" block $\hat{b}_{(i,j)}t-1$ in $\hat{B}^{t-1}$ will remain as the "undefined" block $\hat{b}_{(i,j)}t$ in $\hat{B}^t$. That is, each "undefined" block $\hat{b}_{(i,j)}t$ will not participate the background updating procedure until $\hat{b}_{(i,j)}t$ is determined as a "static" block. As shown in Figure 2, each block in $\hat{R}^t$ is determined by motion estimation between two consecutive (gray-level) video frames, I^{t-1} and I^t, of the bootstrapping video sequence. Each block in the initial modeled background frame $\hat{B}^2$ is based on B^1 and $\hat{R}^2$, in which some blocks in $\hat{B}^2$ are still "undefined" (labeled in black). The initial modeled background frame $\hat{B}^3$ is obtained based on B^2 and $\hat{R}^3$. $\hat{R}^4$, $\hat{R}^5$,. . ., and $\hat{R}^{19}$ in the illustrated example can similarly be obtained. Note that, in the illustrated example shown in Figure 2, each initial modeled background frame $\hat{B}^t$ (t = 1,2,. . .,18) contains at least one "undefined" block.

Finally, as shown in Figure 2q, the initial modeled background frame $\hat{B}^{19}$ contains no "undefined" block. Here, for the illustrated example shown in Figure 2, the performance index T_1(=19) is defined as the frame index for initial modeled background processing. Afterwards, the initial modeled background frame $\hat{B}^t$ (t = 20,21,. . .) is duplicated from the "updated" modeled background frame B^{t-1}, i.e., $\hat{B}^t = B^{t-1}$ (t = 20,21,. . .) [35].

2.2. Block representation

As the illustrated example shown in Figure 3, in the proposed block representation approach, each block of the current video frame I^t is classified into one of the four categories, namely, "background," "still object," "illumination change," and "moving object." In Figure 3b, each block of the block representation frame R^t for I^t is labeled in four different gray levels. The block representation frame R^t is obtained based on the two consecutive video frames, I^t and I^{t-1}, and the initial modeled background frame $\hat{B}^t$ by the proposed block representation approach (as shown in Figure 4), in which motion estimation and correlation coefficient computation are used to perform block representation (classification).

Motion estimation is performed between the two consecutive video frames, I^t and I^{t-1} using a block matching algorithm so that each block in I^t is determined as either "static" or "moving." In this study, the sum of absolute differences (SAD) is used as the cost function for block matching between block $\mathbf{b}^t_{(i,j)}$ in I^t and the corresponding block in I^{t-1} and the search range for motion estimation is set to ±N/2 [35,36]. For a block in I^t, if the minimum SAD, $D_{mv(u,v)}$, for motion vector (u,v), is smaller than



(a) I^t (b) R^t (c)

Figure 3 An illustrated example of block representation: (a) the current video frame; (b) the block representation frame; (c) templates for different blocks in (b).

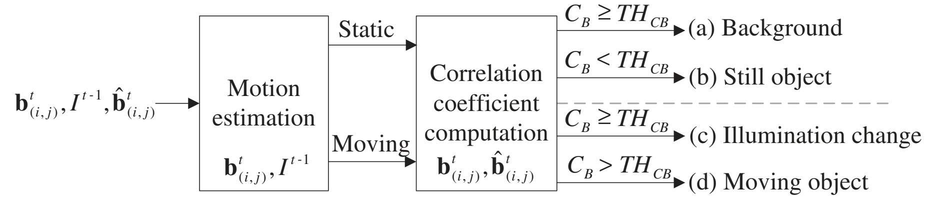

$\mathbf{b}^t_{(i,j)}, I^{t-1}, \hat{\mathbf{b}}^t_{(i,j)}$
Motion estimation
$\mathbf{b}^t_{(i,j)}, I^{t-1}$
Static
Moving
Correlation coefficient computation
$\mathbf{b}^t_{(i,j)}, \hat{\mathbf{b}}^t_{(i,j)}$
$C_B \geq TH_{CB}$
(a) Background
$C_B < TH_{CB}$
(b) Still object
$C_B \geq TH_{CB}$
(c) Illumination change
$C_B > TH_{CB}$
(d) Moving object

Figure 4 The flowchart of the proposed block representation approach.

90% of the SAD for the null-vector (0,0), $D_{mv(0,0)}$, the block is determined as a "moving" block; otherwise, it is determined as a "static" block [19,35].

On the other hand, the correlation coefficient $C_B(i, j)$ between block $\mathbf{b}^t_{(i,j)}$ in I^t and block $\hat{b}_{(i,j)}t$ in the initial modeled background frame $\hat{B}^t$ is computed as

$$C_B(i,j) = \frac{\sum \left| \mathbf{b}^t_{(i,j)} - \mu_{\mathbf{b}^t_{(i,j)}} | \times | \hat{b}^t_{(i,j)} - \mu_{\hat{b}^t_{(i,j)}} \right|}{\sqrt{\sum |\mathbf{b}^t_{(i,j)} - \mu_{\mathbf{b}^t_{(i,j)}}|^2} \times \sqrt{\sum |\hat{b}^t_{(i,j)} - \mu_{\hat{b}^t_{(i,j)}}|^2}} \tag{1}$$

where $\mu_{\mathbf{b}}$ is the mean of the pixel values in block $\mathbf{b}$. As shown in Figure 4, based on $C_B(i,j)$ and the threshold $\mathrm{TH_{CB}}$ a "static" block can be further classified into either a "background" block (if $C_B(i,j) \geq \mathrm{TH_{CB}}$) or a "still object" block (otherwise), whereas a "moving" block can be further classified into either an "illumination change" block (if $C_B(i,j) \geq \mathrm{TH_{CB}}$) or a "moving object" block (otherwise). Afterwards, four different block representations are obtained.

2.3. Background updating

By background updating, each block $\hat{b}_{(i,j)}t$ in the initial modeled background frame $\hat{B}^t$ can be updated to obtain the corresponding block $\tilde{b}^t_{(i,j)}$ in the modeled background frame B^t as follows. Both the "background" and "illumination change" blocks are updated by temporal smoothing, i.e., block $\tilde{b}^t_{(i,j)}$ in B^t is updated as the linearly weighted sum of block $\hat{b}_{(i,j)}t$ in $\hat{B}^t$ and block $\mathbf{b}^t_{(i,j)}$ in I^t. On the other hand, both the "still object" and "moving object" blocks are updated by block replacement.

(a) *Background*: the modeled background block $\tilde{b}^t_{(i,j)}$ in B^t is updated by

$$\tilde{b}^t_{(i,j)} = \alpha \cdot \hat{b}_{(i,j)}t + (1-\alpha) \cdot \mathbf{b}^t_{(i,j)} \tag{2}$$

where α, the updating weight, is empirically set to 0.9 in this study.

(b) *Still object*: the modeled background block $\tilde{b}^t_{(i,j)}$ in B^t is updated by

$$\begin{cases} \tilde{\mathbf{b}}^t_{(i,j)} = \mathbf{b}^t_{(i,j)}, & \text{if } \mathrm{Count}_{(i,j)} \geq \mathrm{TH_{still}}, \\ \tilde{\mathbf{b}}^t_{(i,j)} = \hat{\mathbf{b}}^t_{(i,j)}, & \text{otherwise}, \end{cases} \tag{3}$$

where $\mathrm{Count}_{(i,j)}$ is the number of times that $\mathbf{b}^t_{(i,j)}$ in I^t is successively determined as a "still object" block previously, and $\mathrm{TH_{still}}$ is a threshold for the time duration (in terms of the number of frames) that a "still object" block will learn to be a "background" block. That is, if an object (or a block $\mathrm{b}^t_{(i,j)}$ in I^t) does not "move" for a sufficient time duration, it will become some part of the background. As the illustrated example shown in Figure 5, the marked block $\mathrm{b}^{33}_{(11,13)}$ in I^{33} is detected as a "still object" block (in R^{33}) for a sufficient time duration ($\mathrm{TH_{still}}$ = 20). Then, its corresponding block $\tilde{b}^{33}_{(11,13)}$ in B^{33} will be updated (replaced) by $\mathrm{b}^{33}_{(11,13)}$ in I^{33}.

(c) *Illumination change*: the modeled background block $\tilde{b}^t_{(i,j)}$ in B^t is similarly updated by Equation (2).

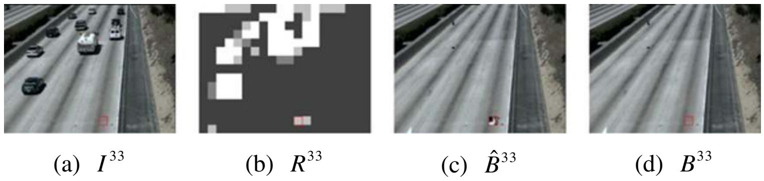

(a) I^{33} (b) R^{33} (c) $\hat{B}^{33}$ (d) B^{33}

Figure 5 An illustrated example of background updating for a "still object" block: (a) the original video frame; (b) the block representation frame; (c) the initial modeled background frame; (d) the modeled background frame.

(d) *Moving object*: the modeled background block $\widetilde{b}^t_{(i,j)}$ in B^t is updated by

$$\begin{cases} \widetilde{\mathbf{b}}^t_{(i,j)} = \mathbf{b}^t_{(i,j)}, & \text{if } \mathrm{SM}\left(\mathbf{b}^t_{(i,j)}\right) < \mathrm{SM}\left(\hat{\mathbf{b}}^t_{(i,j)}\right) \\ \widetilde{\mathbf{b}}^t_{(i,j)} = \hat{\mathbf{b}}^t_{(i,j)}, & \text{otherwise,} \end{cases} \tag{4}$$

where SM($\mathbf{b}^t_{(i,j)}$) and $\mathrm{SM}\left(\hat{b}^t_{(i,j)}\right)$ denote the side-match measures for block $\mathrm{b}^t_{(i,j)}$ from I^t embedded in $\hat{B}^t$ and that for block $\hat{b}_{(i,j)}t$ "embedded" in $\hat{B}^t$, respectively, as shown in Figure 6. The side-match measure (or the boundary match measure) [37,38] is widely used in various image/video error concealment algorithms due to its good trade-off in complexity and visual quality. SM($\mathbf{b}^t_{(i,j)}$) is defined as the sum of squared differences between the boundary of the embedded block $\mathrm{b}^t_{(i,j)}$ from I^t and the boundaries of the four neighboring blocks $\hat{b}_{(i-1,j)}t$, $\hat{b}_{(i+1,j)}t$, $\hat{b}_{(i,j-1)}t$, and $\hat{b}_{(i,j+1)}t$, in $\hat{B}^t$ (Figure 6a), i.e.,

$$\begin{aligned} \mathrm{SM}\left(\mathbf{b}^t_{(i,j)}\right) = & \sum_{b=0}^{N-1}\left(\hat{B}^t_{(iN-1,jN+b)} - I^t_{(iN,jN+b)}\right)^2 \\ & + \sum_{b=0}^{N-1}\left(\hat{B}^t_{(iN+N,jN+b)} - I^t_{(iN+N-1,jN+b)}\right)^2 \\ & \times \sum_{a=0}^{N-1}\left(\hat{B}^t_{(iN+a,jN-1)} - I^t_{(iN+a,jN)}\right)^2 \\ & + \sum_{a=0}^{N-1}\left(\hat{B}^t_{(iN+a,jN+N)} - I^t_{(iN+a,jN+N-1)}\right)^2 \end{aligned} \tag{5}$$

Similarly, $\mathrm{SM}\left(\hat{b}^t_{(i,j)}\right)$ is defined as the sum of squared differences between the boundary of block $\hat{b}_{(i,j)}t$ and the boundaries of its four neighboring blocks $\hat{b}_{(i-1,j)}t$, $\hat{b}_{(i+1,j)}t$, $\hat{b}_{(i,j-1)}t$, and $\hat{b}_{(i,j+1)}t$, in $\hat{B}^t$ (Figure 6b), i.e.,

$$\begin{aligned} \mathrm{SM}\left(\hat{\mathbf{b}}^t_{(i,j)}\right) = & \sum_{b=0}^{N-1}\left(\hat{B}^t_{(iN-1,jN+b)} - \hat{B}_{(iN,jN+b)}t\right)^2 \\ & + \sum_{b=0}^{N-1}\left(\hat{B}^t_{(iN+N,jN+b)} - \hat{B}_{(iN+N-1,jN+b)}t\right)^2 \\ & \times \sum_{a=0}^{N-1}\left(\hat{B}^t_{(iN+a,jN-1)} - \hat{B}_{(iN+a,jN)}t\right)^2 \\ & + \sum_{a=0}^{N-1}\left(\hat{B}^t_{(iN+a,jN+N)} - \hat{B}_{(iN+a,jN+N-1)}t\right)^2. \end{aligned} \tag{6}$$

Note that if a block in R^t is determined as a "moving object" block two times consecutively, the corresponding modeled background block $\widetilde{b}^t_{(i,j)}$ in B^t is updated by Equation (4). The side-match measure uses the camouflage of each "moving object" block to search the more suitable modeled background block so that we can speed up the background updating procedure. As the illustrated example shown in Figure 7, two marked blocks $\mathbf{b}_{(12,9)}$ and $\mathbf{b}_{(11,10)}$ in both I^{12} and I^{13} are detected as two "moving object" blocks in both R^{12} and R^{13} consecutively. Thus, their corresponding blocks $\widetilde{b}^{13}_{(12,9)}$ and $\widetilde{b}^{13}_{(11,10)}$ in B^{13} will be updated (replaced) by blocks $\mathbf{b}^{13}_{(12,9)}$ and $\mathbf{b}^{13}_{(11,10)}$ in I^{13}, respectively.

2.4. Initial segmented foreground

Based on the modeled background frame B^t performing background updating, as an illustrated example shown in

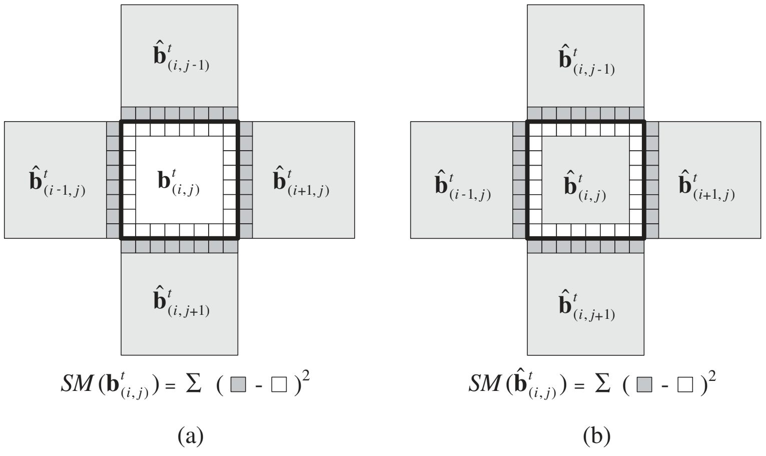
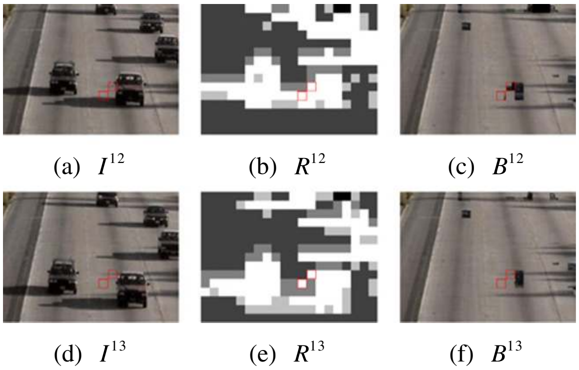

$\hat{\mathbf{b}}^t_{(i,j-1)}$
$\hat{\mathbf{b}}^t_{(i-1,j)}$
$\mathbf{b}^t_{(i,j)}$
$\hat{\mathbf{b}}^t_{(i+1,j)}$
$\hat{\mathbf{b}}^t_{(i,j+1)}$
$SM(\mathbf{b}^t_{(i,j)}) = \Sigma\ (▨ - □)^2$
(a)
$\hat{\mathbf{b}}^t_{(i,j-1)}$
$\hat{\mathbf{b}}^t_{(i-1,j)}$
$\hat{\mathbf{b}}^t_{(i,j)}$
$\hat{\mathbf{b}}^t_{(i+1,j)}$
$\hat{\mathbf{b}}^t_{(i,j+1)}$
$SM(\hat{\mathbf{b}}^t_{(i,j)}) = \Sigma\ (▨ - □)^2$
(b)

Figure 6 The side-match measures SM($\mathrm{b}^t_{(i,j)}$) and SM($\mathrm{b}^t_{(i,j)}$)of a "moving object" block in background updating.



(a) I^{12} (b) R^{12} (c) B^{12}

(d) I^{13} (e) R^{13} (f) B^{13}

Figure 7 An illustrated example of background updating for two "moving object" blocks: (a) and (d) are two original video frames; (b) and (e) are the corresponding block representation frames; (c) and (f) are the corresponding modeled background frames.

Figure 8, the initial (binary) segmented foreground frame $\hat{F}^t$ can be obtained as

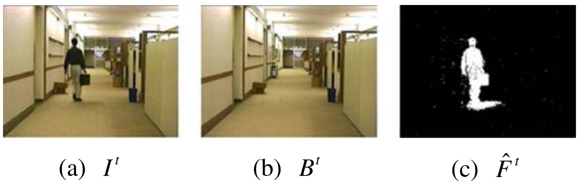
$$\hat{F}^t = \begin{cases} 1, & \text{if } I^t - B^t \geq TH_{\text{isf}}, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

where $\mathrm{TH}_{\mathrm{isf}}$ is a threshold, which is empirically set to 15 in this study.

2.5. Noise removal and shadow suppression with two morphological operations

As shown in Figure 8, $\hat{F}^t$ usually contains some fragmented (noisy) parts and shadows. To obtain the precise segmented foreground frame F^t, a noise removal and shadow suppression procedure is adopted, which combines the shadow suppression approach in [39] and the edge information extracted from I^t with $\hat{F}^t$ being the (binary) operation mask.

Let $\hat{F}_S t$ be the S (saturation) component of the original video frame (frame t) represented in the HSV color space and $\hat{F}_E t$ be the gradient image of I^t using the Sobel operator [40] with $\hat{F}^t$ being the (binary) operation mask. The segmented foreground frame $\bar{F}^t$ is defined as



(a) I^t (b) B^t (c) $\hat{F}^t$

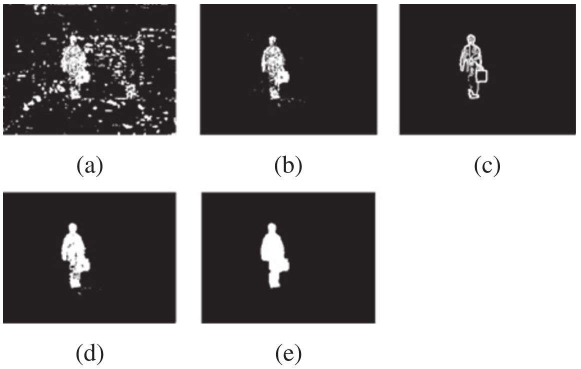
Figure 8 An illustrated example of initial segmented foreground: (a) the current video frame; (b) the modeled background frame; (c) the initial segmented foreground frame.

$$\bar{F}^t = \begin{cases} 1, & \text{if } \left(\hat{F}^t \cap \left(\hat{F}^t_S \geq \sigma_{\hat{F}} S^t\right)\right) \cup \left(\hat{F}^t_E \geq \mathrm{TH}_E\right), \\ 0, & \text{otherwise}, \end{cases} \tag{8}$$

where ∩ and ∪ denote the logical AND and OR operators, respectively, $\sigma_{\hat{F}} S^t$ is the standard deviation of $\hat{F}_S t$, and TH_E is a threshold. Here, TH_E is empirically set to 120 in this study. Figure 9 shows an illustrated example performing the noise removal and shadow suppression procedure. By applying the shadow suppression approach in [39], the "second" (binary) segmented foreground frame (shown in Figure 9b) is obtained based on $\hat{F}^t$ (shown in Figure 8c) and $\hat{F}_S t \geq \sigma_{\hat{F}} S^t$ (shown in Figure 9a). Based on the "second" (binary) segmented foreground frame (shown in Figure 9b), combining the gradient image $\hat{F}_E t$ of I^t (shown in Figure 9c) preserving the edge information in the initial (binary) segmented foreground frame $\hat{F}^t$, the segmented foreground frame $\bar{F}^t$ (shown in Figure 9d) is obtained by Equation (8). Finally, the final segmented foreground frame (shown in Figure 9e) is obtained as F^t with two morphological (erosion and dilation) operations [40].

3. Experimental results

In this study, experimental results are performed using Borland C++ on Intel Core 2 Quad CPU 2.4 GHz Microsoft Windows XP platform. Six bootstrapping video sequences, selected from three benchmark datasets, namely, ATON (http://cvrr.ucsd.edu/aton/shadow/index.html), PETS2006 (http://www.cvg.rdg.ac.uk/PETS2006/data.html), and BPI [24], are used in this study, which are listed and categorized in Table 1. In Table 1, the six bootstrapping video sequences are categorized as jiggled



(a) (b) (c)

(d) (e)

Figure 9 An illustrated example for noise removal and shadow suppression: (a) $\hat{F}_S t \geq \sigma_{\hat{F}} S^t$, (b) $\hat{F}^t \cap (\hat{F}_S t \geq \sigma_{\hat{F}} S^t)$, (c) $\hat{F}_E t \geq T_E$, (d) F^t, (e) F^t with two morphological (erosion and dilation) operations.

Table 1 The six bootstrapping video sequences and their categories

Video sequences	Benchmark and category	Video sequences	Benchmark and category
 "Highway-1"	ATON 15 fps Jiggled capture	 "S1-T1-C-4"	PETS2006 25 fps Shadow effect
 "Highway-2"	ATON 15 fps Jiggled capture	 "Vignal"	BPI 18 fps Heavy clutter
 "S1-T1-C-3"	PETS2006 25 fps Shadow effect	 "Granguardia"	BPI 18 fps Heavy clutter

capture, shadow effect, and heavy clutter. The video frames in Table 1 are 320 × 240 in size.

To evaluate the performance of the proposed approach, three comparison approaches, namely, MoG [4], Reddy background estimation (Reddy) [22], and self-organizing background subtraction (SOBS) [27], are implemented in this study. In MoG and Reddy, only the gray-level component of each video frame is employed, in SOBS, the H, S, and V components of each video frame are employed, and in the proposed approach, the gray-level video frames are used and additionally the S component is only used for shadow suppression. Note that, for the SOBS approach, each SOBS high-resolution video frame (3 W × 3H pixels) is downsampled to a video frame of the original resolution ($W \times H$ pixels) by local averaging.

3.1. Parameter setting

$\mathrm{TH}_{\mathrm{still}}$ in Equation (3) is a threshold for the time duration (in terms of the number of frames) that a "still object" block will learn to be a "background" block. If an object (or a block $\mathbf{b}^{t}_{(i,j)}$ in I^{t}) does not "move" for a sufficient time duration $\mathrm{TH}_{\mathrm{still}}$ it will be treated as some part of the background. If $\mathrm{TH}_{\mathrm{still}}$ is set to a small value, the modeled background frame B^{t} will easily be disturbed by moving objects in each bootstrapping video sequence. On the contrary, if $\mathrm{TH}_{\mathrm{still}}$ is set to a large value, the modeled background frame B^{t} might not be updated immediately. As the illustrated example shown in Figure 10, the modeled background frames B^{t} with $\mathrm{TH}_{\mathrm{still}} = 20$ and $\mathrm{TH}_{\mathrm{still}} = 40$ are illustrated, where performance index T_1 is defined as the frame index for initial modeled background processing (Section 2.1) and it is identically set to 21 for both $\mathrm{TH}_{\mathrm{still}} = 20$ and $\mathrm{TH}_{\mathrm{still}} = 40$. If performance index T_2 is defined as the frame index for constructing the free ("true") modeled background frame, for the illustrated example shown in Figure 10, $T_2 = 152$ for $\mathrm{TH}_{\mathrm{still}} = 20$, whereas $T_2 = 128$ for $\mathrm{TH}_{\mathrm{still}} = 40$. The two performance indexes (T_1 and T_2) for different thresholding values $\mathrm{TH}_{\mathrm{still}}$ of four bootstrapping video sequences, namely, "Highway-1," "Highway-2," "S1-T1-C-3," and "S1-T1-C-4," are illustrated in Figure 11.

Actually, $\mathrm{TH}_{\mathrm{still}}$ depends on the sizes of moving objects, the velocities of moving objects, and the frame rate (frames per second, fps) of each bootstrapping video sequence. Let A^{t} be the minimum bounding rectangle of a moving object in frame I^{t} and $A^{t\text{-}FR}$ be the minimum bounding rectangle of the moving object in frame $I^{t\text{-}FR}$ where FR (fps) is the frame rate of a bootstrapping video sequence. Note that the time difference between the two frames, $I^{t\text{-}FR}$ and I^{t}, is 1 s. Here, the moving object is roughly determined as "high-motion" if $A^{t\text{-}FR}$ and A^{t} do not contain any overlapping part. Otherwise, the moving object is roughly determined as "low-motion." In this study, if a bootstrapping video sequence contains "high-motion" moving object(s), then $(FR/2) \leq \mathrm{TH}_{\mathrm{still}} \leq FR$. Otherwise, $FR \leq \mathrm{TH}_{\mathrm{still}} \leq (FR + FR/2)$. The threshold values $\mathrm{TH}_{\mathrm{still}}$ for the six video sequences, namely, "Highway-1," "Highway-2," "S1-T1-C-3," "S1-T1-C-4," "Vignal," and "Granguardia," by the proposed approach are empirically set to 15, 15, 35, 35, 20, and 20, respectively.

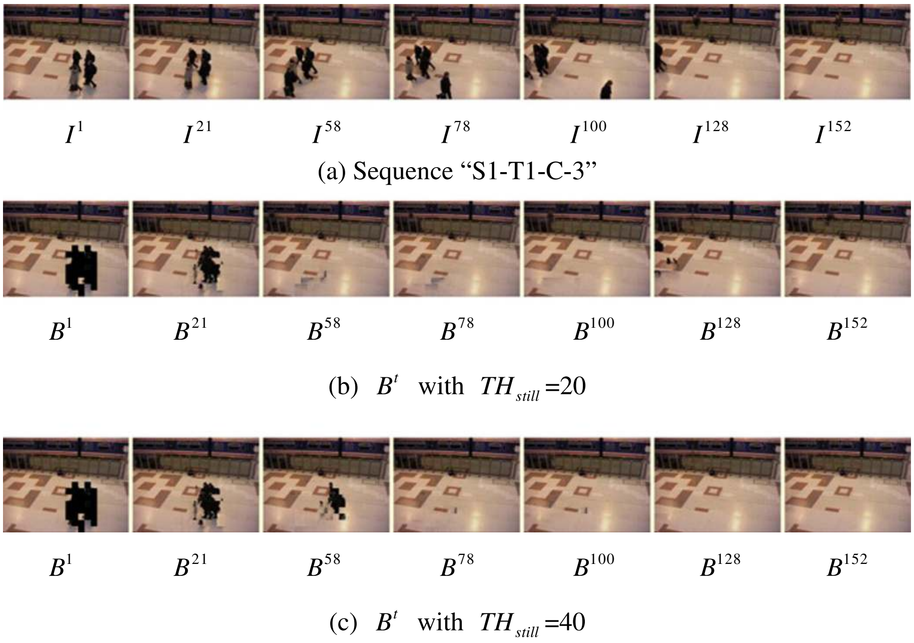


Figure 10 An illustrated example for frames I^1 I^{21}, I^{58}, I^{78}, I^{100}, I^{128}, and I^{152} of the bootstrapping video sequence "S1-T1-C-3" (a) and the corresponding modeled background frames B^t with T^1 = 21, TH_{still} = 20, (b) and TH_{still} = 40 (c).

3.2. Subjective comparisons

For background initialization, Figures 12, 13, 14, 15, 16, and 17 illustrate some frames of the six bootstrapping video sequences (a) and the corresponding modeled background frames B^t by Reddy (b), SOBS (c), and the proposed approach (d) with block size of 16 × 16. For the Reddy approach, given a video sequence of T video frames, each video frame is divided into non-overlapping blocks of size 16 × 16. Agglomerative clustering background estimation is applied in a block-by-block manner. Background areas are iteratively filled by selecting the most appropriate (smooth) candidate blocks. For the

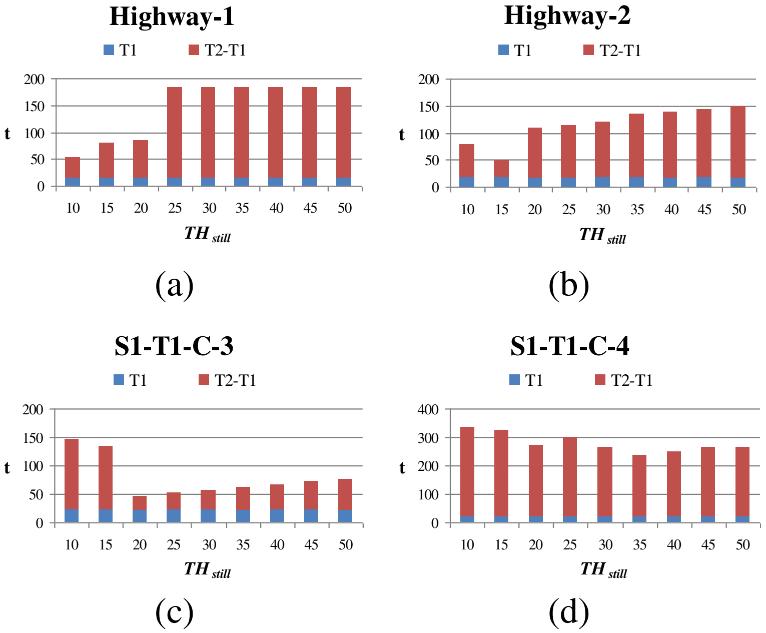


Figure 11 The performance indexes (T_1 and T_2) of four bootstrapping video sequences with different thresholding values TH_{still}: the performance indexes (T_1 and T_2) of "Highway-1" (a); "Highway-2" (b); "S1-T1-C-3" (c); and "S1-T1-C-4" (d).

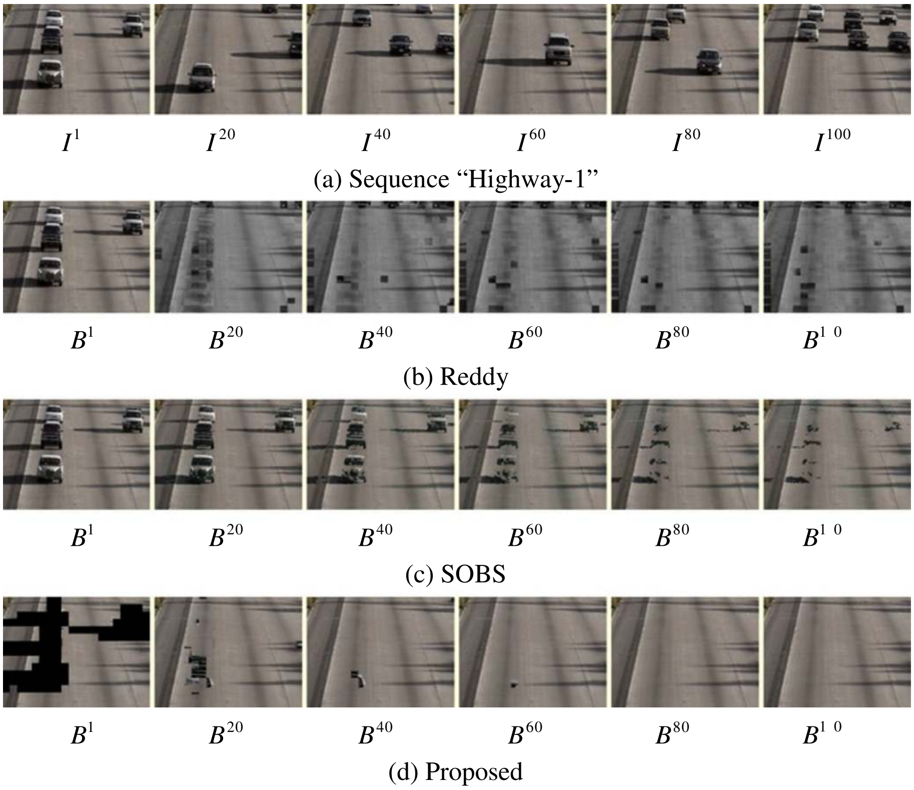
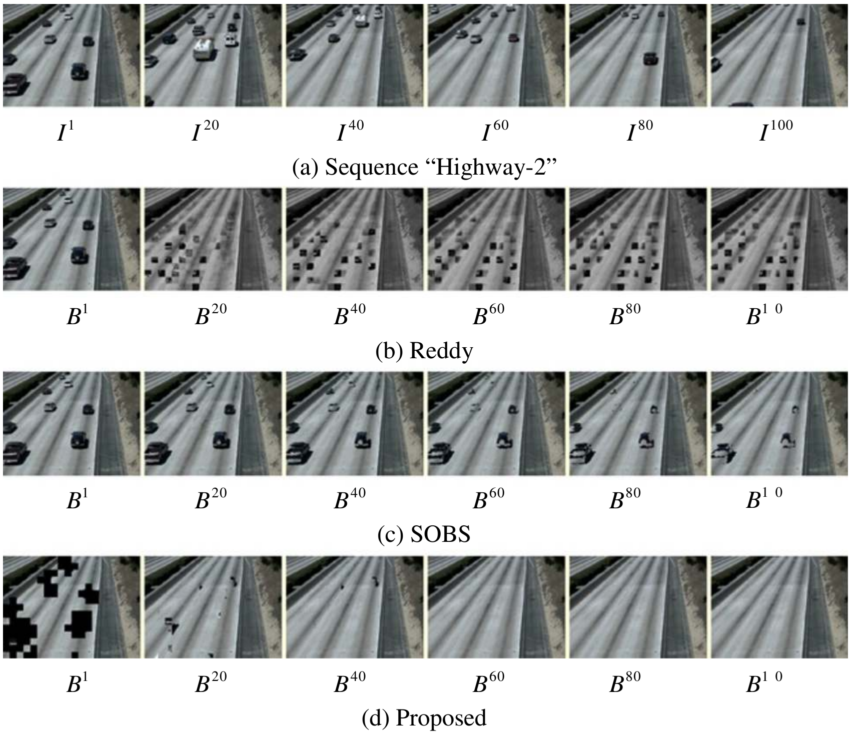


Figure 12 Some background initialization results B^t of frames I^1, I^{20}, I^{40}, I^{60}, I^{80}, and I^{100} of the bootstrapping video sequence "Highway-1" (a) by Reddy (b), SOBS (c), and the proposed approach (d) with block size 16 × 16 and $\text{TH}_{\text{still}} = 15$.

Reddy approach, a bootstrapping video sequence with a large number of video frames is required to obtain the free ("true") modeled background frame, due to some blocks in each video frame might be erroneously estimated based on the corresponding candidate block set in the frequency domain. For the SOBS approach, the modeled background frame $B^l = I^l$. Then, each subsequent modeled background frame B^t is obtained by pixel-wise background updating. The SOBS approach can obtain the modeled background frame of a bootstrapping video sequence with suitable parameter values [27]. However, as a pixel-based approach, to obtain the free ("true") modeled background frame, it needs a long time duration to eliminate the foreground objects in B^l Based on our experimented results, the performance indexes T^2 for the six video sequences by the SOBS approach are 220 for "Highway-1," 306 for "Highway-2," 305 for "S1-T1-C-3," 335 for "S1-T1-C-4," >260 for "Vignal," and >450 for "Granguardia," respectively. For each bootstrapping video sequence, the proposed approach can obtain the free ("true") modeled background frame "completely" after T^2 The performance indexes T^2 for the six video sequences, namely, "Highway-1," "Highway-2," "S1-T1-C-3," "S1-T1-C-4," "Vignal," and "Granguardia," by the proposed approach are 80, 48, 62, 236, 205, and 414, respectively, which are indeed less than the corresponding values by the SOBS approach.

For foreground segmentation, Figures 18, 19, 20, 21, 22, and 23 illustrate some segmented foreground frames F^t by MoG (a), Reddy (b), SOBS (c), and the proposed approach (d). For Reddy, SOBS, and the proposed approach, the segmented foreground frames are obtained by background subtraction of the corresponding bootstrapping video sequences shown in Figures 12, 13, 14, 15, 16, and 17, whereas, for MoG, the segmented foreground frames are obtained by the pixel-wise MoG method in [4]. For SOBS, the contents of red rectangles in the segmented foreground frames indicate the ghost objects. As shown in Figures 18, 19, 20, 21, 22, and 23, the segmented foreground frames of the MoG approach are usually good for bootstrapping video sequences containing some dynamic background, but the segmented foreground frames of the MoG approach may obtain fragmented (noisy) foreground objects for bootstrapping video sequences containing some low-motion moving objects and some noisy background due to jiggled capture. The SOBS approach may obtain good segmented foreground objects without shadow. However, each modeled background frame B^t of the SOBS



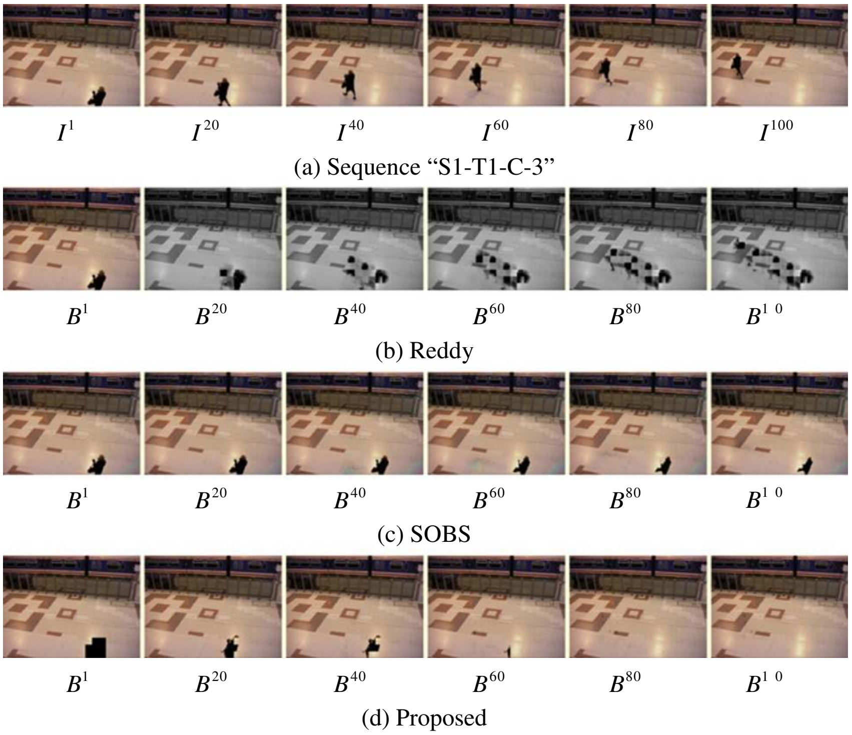
(a) Sequence “Highway-2”

(b) Reddy

(c) SOBS

(d) Proposed

Figure 13 **Some background initialization results B^t of frames I^1, I^{20}, I^{40}, I^{60}, I^{80}, and I^{100} of the bootstrapping video sequence “Highway-2” (a) by Reddy (b), SOBS (c), and the proposed approach (d) with block size 16 × 16 and $\text{TH}_{\text{still}} = 15$.**



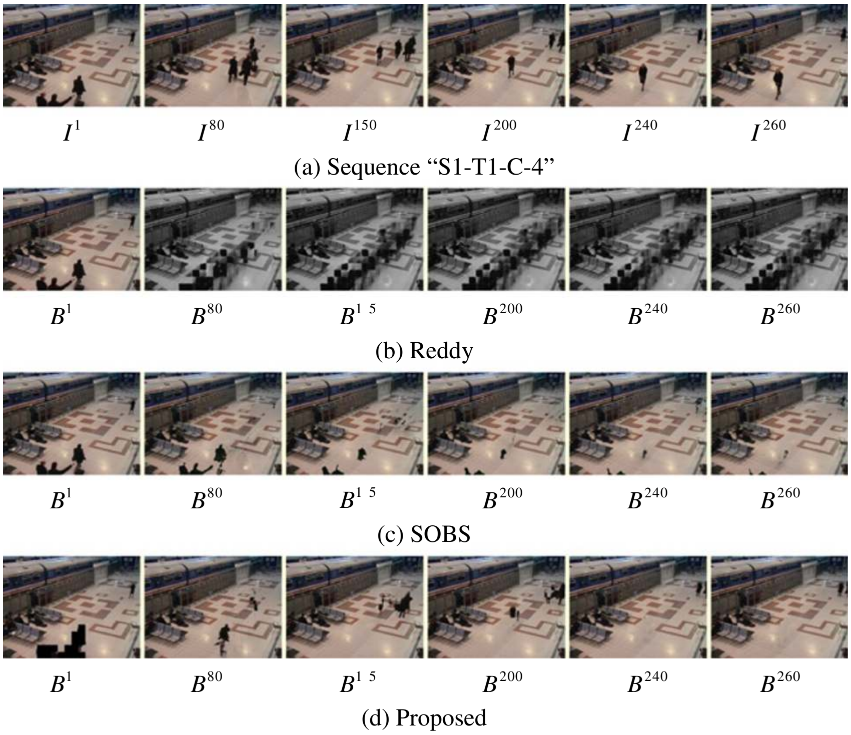
(a) Sequence “S1-T1-C-3”

(b) Reddy

(c) SOBS

(d) Proposed

Figure 14 **Some background initialization results B^t of frames I^1, I^{20}, I^{40}, I^{60}, I^{80}, and I^{100} of the bootstrapping video sequence “S1-T1-C-3” (a) by Reddy (b), SOBS (c), and the proposed approach (d) with block size 16 × 16 and $\text{TH}_{\text{still}} = 35$.**



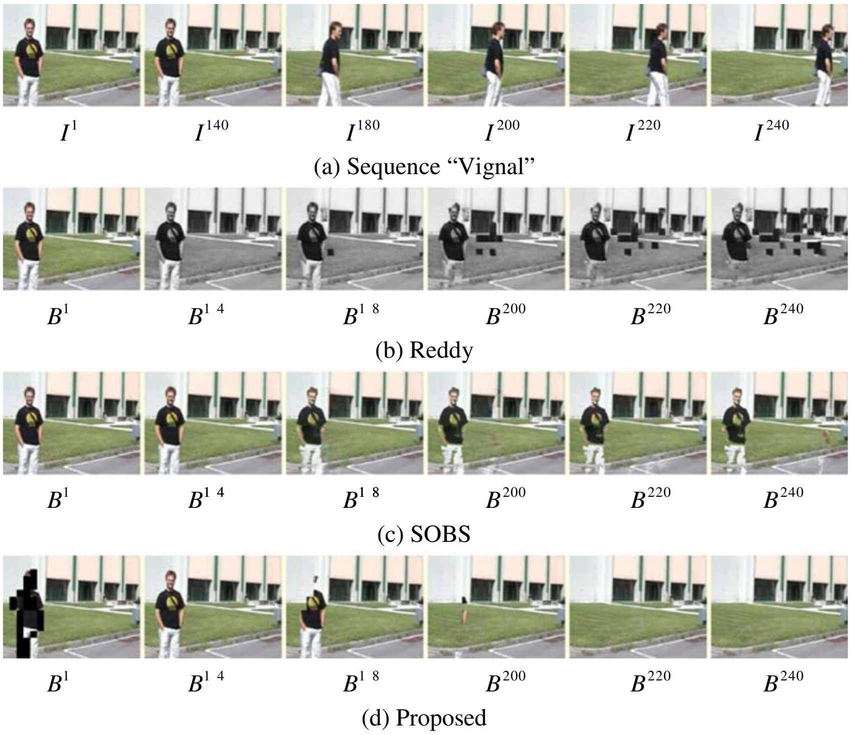
(a) Sequence “S1-T1-C-4”

(b) Reddy

(c) SOBS

(d) Proposed

Figure 15 Some background initialization results B^t of frames I^1, I^{80}, I^{150}, I^{200}, I^{240}, and I^{260} of the bootstrapping video sequence “S1-T1-C-4” (a) by Reddy (b), SOBS (c), and the proposed approach (d) with block size 16 × 16 and TH_{still} = 35.



(a) Sequence “Vignal”

(b) Reddy

(c) SOBS

(d) Proposed

Figure 16 Some background initialization results B^t of frames I^1, I^{140}, I^{180}, I^{200}, I^{220}, and I^{240} of the bootstrapping video sequence “Vignal” (a) by Reddy (b), SOBS (c), and the proposed approach (d) with block size 16 × 16 and TH_{still} = 20.

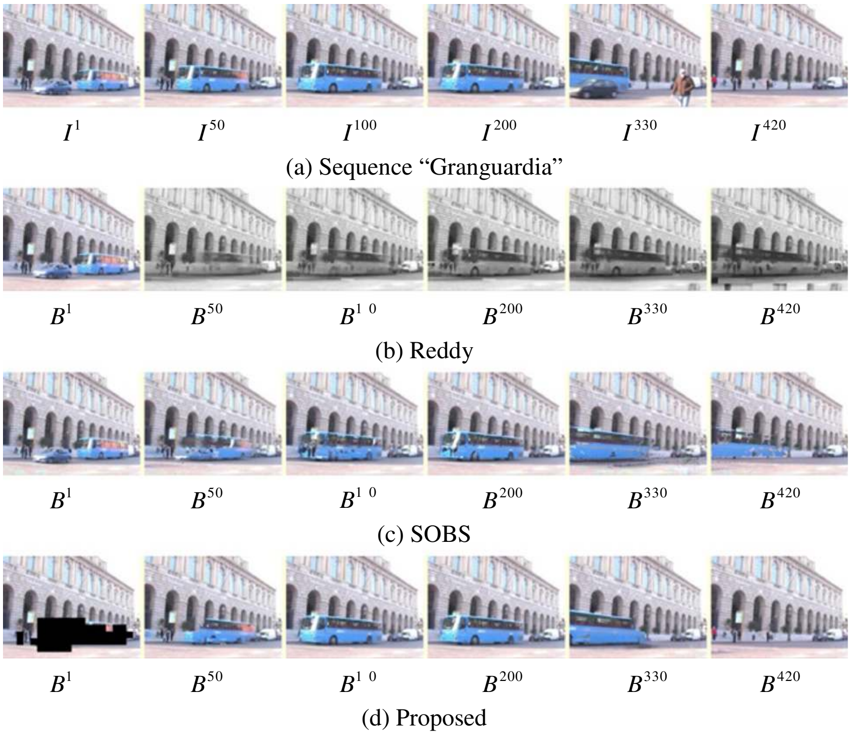


Figure 17 **Some background initialization results B^{t} of frames I^{1}, I^{50}, I^{100}, I^{200}, I^{330}, and I^{420} of the bootstrapping video sequence “Granguardia” (a) by Reddy (b), SOBS (c), and the proposed approach (d) with block size 16 × 16 and $\text{TH}_{\text{still}} = 20$.**

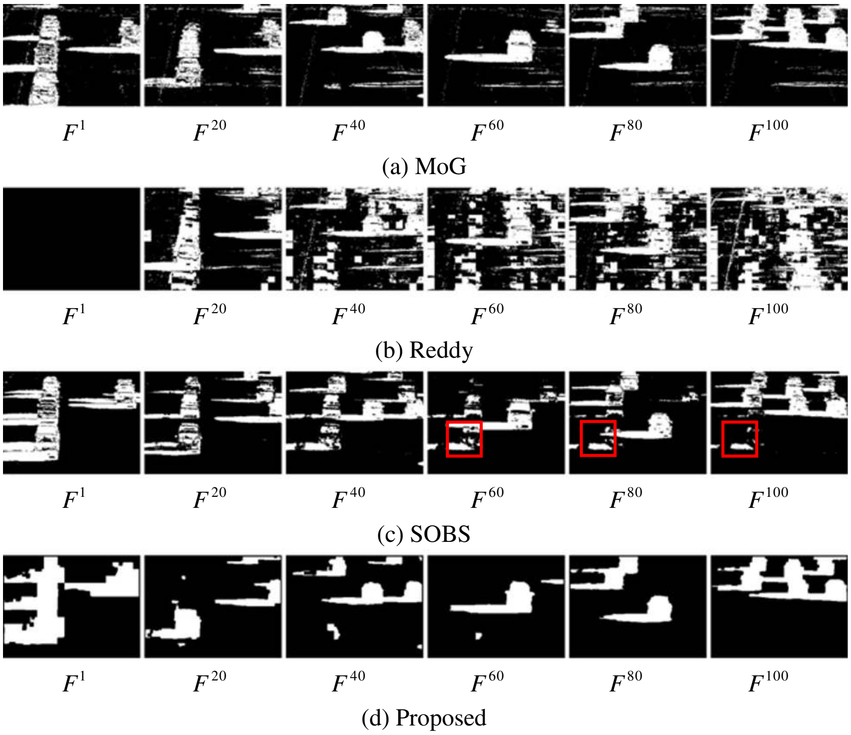


Figure 18 **Some foreground segmentation results F^{t} of frames I^{1}, I^{20}, I^{40}, I^{60}, I^{80}, and I^{100} of the bootstrapping video sequence “Highway-1” by MoG (a), Reddy (b), SOBS (c), and the proposed approach (d) with $T_2 = 80$.**

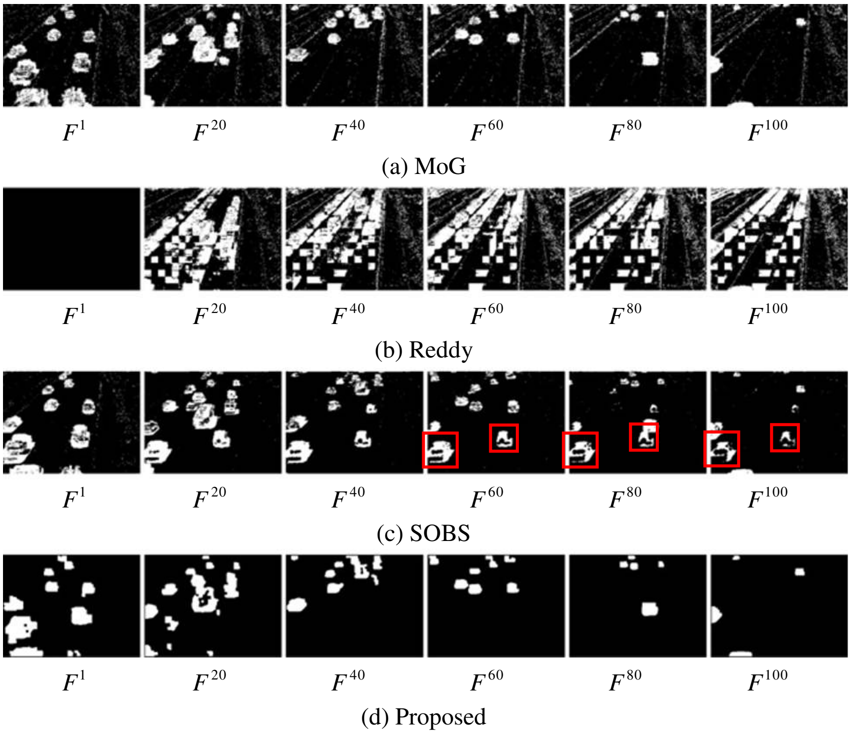


Figure 19 Some foreground segmentation results F^t of frames I^1, I^{20}, I^{40}, I^{60}, I^{80}, and I^{100} of the bootstrapping video sequence "Highway-2" by MoG (a), Reddy (b), SOBS (c), and the proposed approach (d) with T_2 = 48.

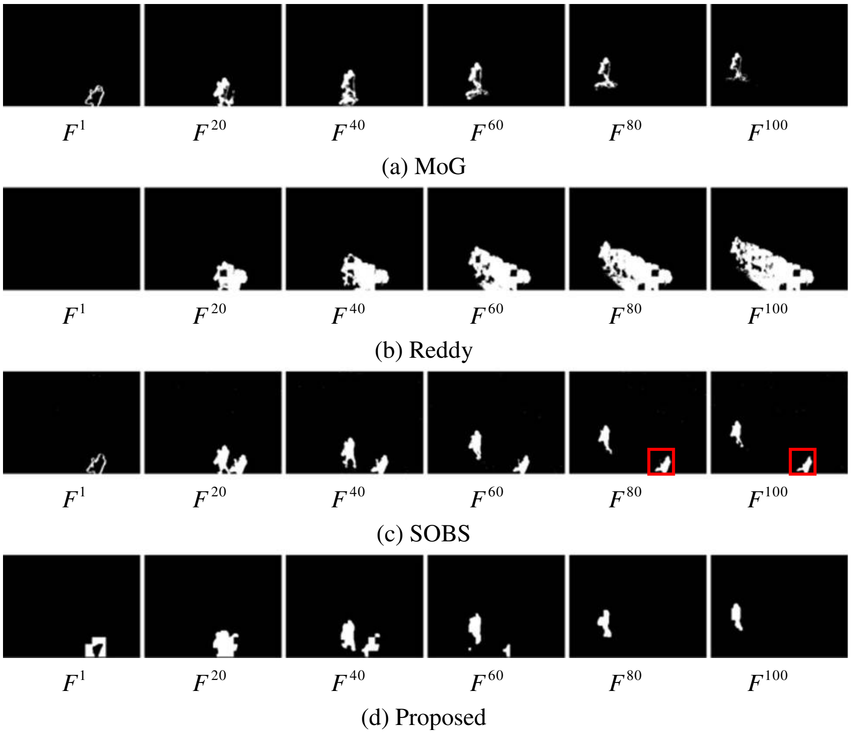


Figure 20 Some foreground segmentation results F^t of frames I^1, I^{20}, I^{40}, I^{60}, I^{80}, and I^{100} of the bootstrapping video sequence "S1-T1 -C-3" by MoG (a), Reddy (b), SOBS (c), and the proposed approach (d) with T_2 = 62.

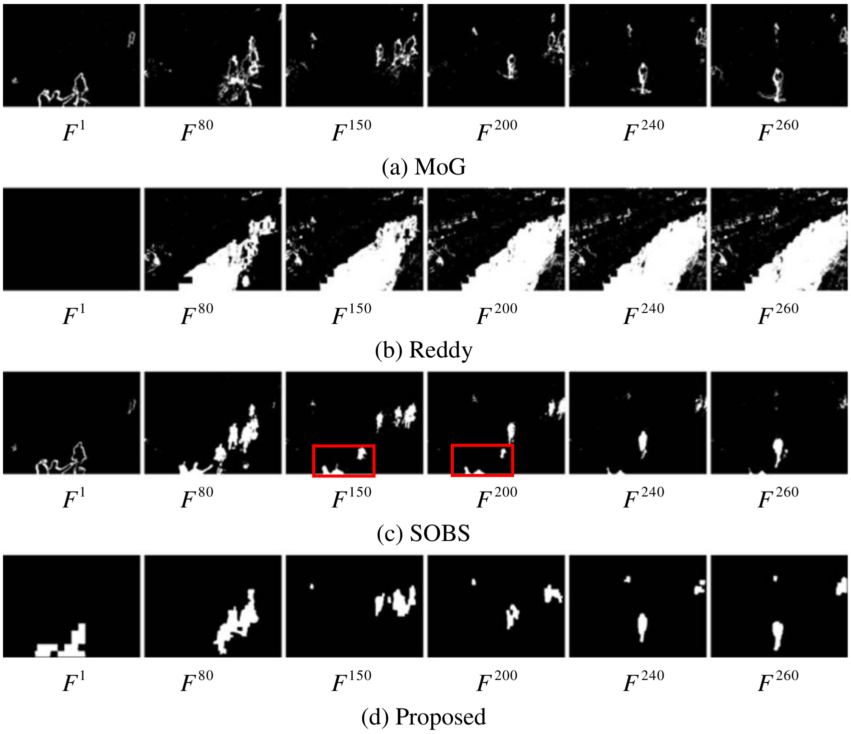


Figure 21 Some foreground segmentation results F^t of frames I^1, I^{80}, I^{150}, I^{200}, I^{240}, and I^{260} of the bootstrapping video sequence "S1-T1-C-4" by MoG (a), Reddy (b), SOBS (c), and the proposed approach (d) with T_2 = 236.

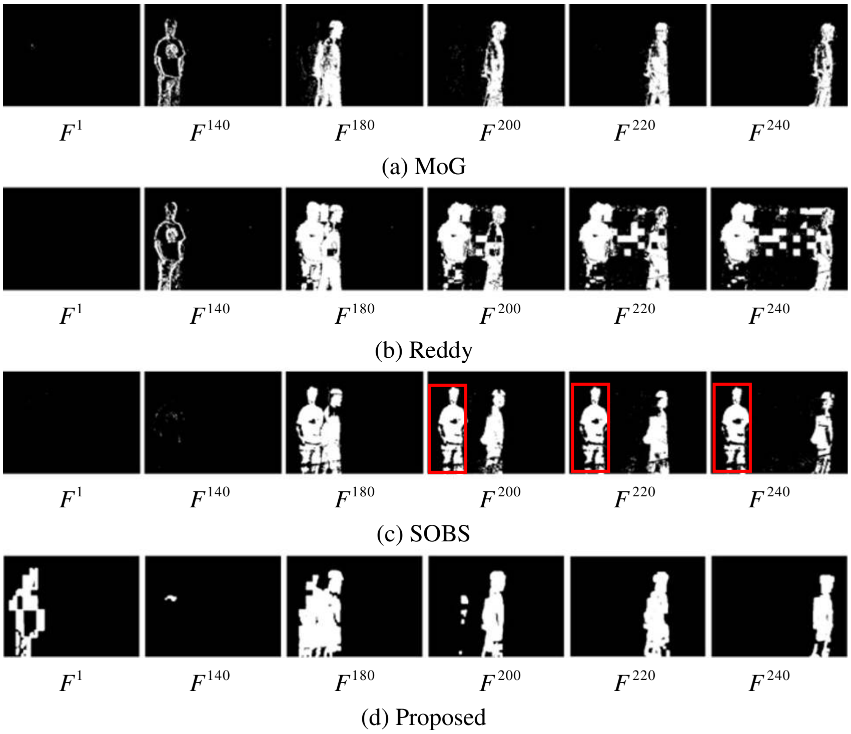
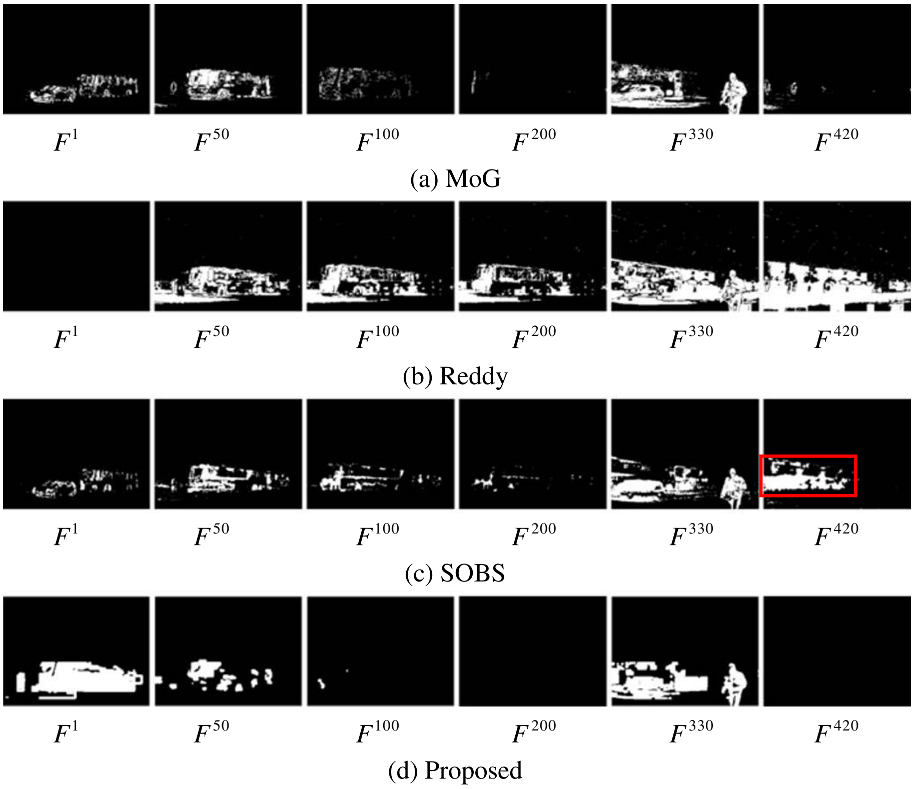


Figure 22 Some foreground segmentation results F^t of frames I^1, I^{140}, I^{180}, I^{200}, I^{220}, and I^{240} of the bootstrapping video sequence "Vignal" by MoG (a), Reddy (b), SOBS (c), and the proposed approach (d) with T_2 = 205.



(a) MoG

(b) Reddy

(c) SOBS

(d) Proposed

Figure 23 Some foreground segmentation results F^t of frames I^1, I^{50}, I^{100}, I^{200}, I^{330}, and I^{420} of the bootstrapping video sequence "Granguardia" by MoG (a), Reddy (b), SOBS (c), and the proposed approach (d) with T_2 = 414.

approach may preserve some foreground objects in I^l resulting in some ghost objects in the segmented foreground frame F^t. The proposed approach has good foreground segmentation results for bootstrapping video sequences, i.e., good segmented foreground objects (without shadow and ghost objects) can be obtained after T_2.

Table 2 lists the average processing times (s) of obtaining a segmented foreground frame for the six bootstrapping video sequences by MoG, Reddy, SOBS, and the proposed approach with block size 16 × 16. Note that the average processing times (s) of obtaining a segmented foreground frame for the six video sequences are evaluated by 100 bootstrapping video frames. The average frame processing times of each comparison approach (except Reddy) for different bootstrapping video sequences are similar. However, the average frame processing times of Reddy for different video sequences are not similar, which are influenced on the complexity (contents) of video sequences. Table 3 lists the average frame processing times (s) of the three processing steps, namely, block representation, background updating, and foreground segmentation, for the six bootstrapping video sequences by the proposed approach with block size 16 × 16. Note that foreground segmentation contains initial segmented foreground (processing time ≈ 0 second) and noise removal and shadow suppression with two morphological operations. The average frame processing times (0.488 and 0.124 seconds) for the two processing steps, namely, block representation and foreground segmentation, depend on the total number of blocks/pixels of a bootstrapping video frame, which are relatively stable. On the other hand, the average frame processing time of the processing step, namely, background updating, depends on temporal smoothing and block replacement for various block representations, which is relatively small. As the results listed in Table 3, motion estimation of block representation using a block matching algorithm constitutes the major part of the processing time of a bootstrapping video sequence by the proposed approach, which may be greatly reduced by parallel implementation.

Table 2 The average frame processing times (s) for the six bootstrapping video sequences by MoG, Reddy, SOBS, and the proposed approach with block size 16 × 16

	MoG	Reddy	SOBS	Proposed
Average	0.068 ± 0.004	0.403 ± 0.252	0.389 ± 0.025	0.615 ± 0.024

Table 3 The average frame processing times (s) of the three processing steps, namely, block representation, background updating, and foreground segmentation, for the six bootstrapping video sequences by the proposed approach with block size 16 × 16

	Background initialization		Foreground segmentation
	Block representation	Background updating	
Average	0.488 ± 0.0243	0.002 ± 0.0005	0.124 ± 0.0026

3.3. Objective comparisons

For foreground segmentation, to perform objective comparisons between the three comparison approaches (MoG, SOBS, and the proposed approach), the “baseline” category of “changedetection.net” video dataset [41] is employed. For MoG, SOBS, and the proposed approach, both the input video sequences and the processing results are processed in a frame-by-frame manner. On the other hand, for Reddy, to obtain the processing results, the whole video sequence should be available to Reddy. Therefore, Reddy is excluded in the following comparisons. Let *TP* be number of true positives, *TN* be number of true negatives, *FN* be number of false negatives, and *FP* be number of false positives. The four evaluation metrics, namely, FPR, FNR, PWC, and FM, are employed in this study, which are defined as [41]

1. false positive rate (FPR): $FP/(FP + TN)$,
2. false negative rate (FNR): $FN/(TN + FP)$,
3. percentage of wrong classifications (PWC): $100 \times (FN + FP)/(TP + FN + FP + TN)$,
4. f-measure (FM): $2 \times (PR \times RE)/(PR + RE)$.

Table 4 lists the objective performance comparisons by four evaluation metrics, FPR, FNR, PWC, and FM, for the four video sequences in the “baseline” category of “changedetection.net” video dataset by MoG, SOBS, and the proposed approach. Table 5 lists the objective performance comparisons by four evaluation metrics, FPR, FNR, PWC, and FM, for the four video sequences in the “baseline” category of “changedetection.net” video dataset by the proposed approach with different block sizes (8 × 8, 16 × 16, and 32 × 32). In Tables 4 and 5, the best evaluation metrics FPR, FNR, PWC, and FM are marked in bold font. Note that the smaller FPR and FNR values respond the better performances, whereas the larger PWC and FM values respond the better performances. Here, for a fair comparison, the proposed approach does not perform the two morphological operations in the noise removal and shadow suppression procedure. Based on the experimental results listed in Table 4, in general, the foreground segmentation results of the proposed approach are better than those of MoG and SOBS. On the other hand, based on the experimental results listed in Table 5, the foreground segmentation results of the proposed approach using three different block sizes (8 × 8, 16 ×16, and 32 × 32) are substantially similar. The average frame processing times of the proposed approach using three different block sizes (8 × 8, 16 × 16, and 32 × 32) are 0.256, 0.615, and 2.166 seconds, respectively. To reduce the average frame processing time of the proposed approach, block size 8 × 8 is recommended.

For background initialization, including the evaluation of foreground masks, we can also evaluate the performance of the estimated background. In this study, the PSNR value of the estimated background, with respective to one “free” background (the groundtruth), is employed. The “free” background (the groundtruth) is synthesized by the “static” parts in different frames of the whole bootstrapping video sequence. The average PSNR values of SOBS and the proposed approach for the “baseline” category of “changedetection.net” video dataset [41] are 26.46 and 28.96 dB, respectively.

Table 4 Objective performance comparisons by four evaluation metrics FPR, FNR, PWC, and FM for the four video sequences in the “baseline” category of “changedetection.net” video dataset by MoG, SOBS, and the proposed approach

	FPR	FNR	PWC	FM
MoG	0.0158	0.0169	3.0802	0.5998
SOBS	0.0577	0.0007	5.5604	0.5076
Proposed	0.0043	0.0174	2.0448	0.6997

Table 5 Objective performance comparisons by four evaluation metrics FPR, FNR, PWC, and FM for the four video sequences in the “baseline” category of “changedetection.net” video dataset by the proposed approach with different block sizes (8 × 8, 16 × 16, and 32 × 32)

	FPR	FNR	PWC	FM
8 × 8	0.0043	0.0174	2.0448	0.6997
16 × 16	0.0044	0.0178	2.0942	0.6992
32 × 32	0.0049	0.0189	2.2429	0.6730

4. Concluding remarks

In this study, an effective background initialization and foreground segmentation approach for bootstrapping video sequences is proposed, in which a modified block representation approach, a new background updating scheme, and an improved noise removal and shadow suppression procedure with two morphological operations are employed. Based on the experimental results obtained in this study, as compared with MoG [4], Reddy [22] and SOBS [27], the proposed approach has better background initialization and foreground segmentation results. In addition, bootstrapping video sequences with jiggled capture, shadow effect, and heavy clutter can be well handled by the proposed approach.

Competing interests

The authors declare that they have no competing interests.

Acknowledgment

This study was supported in part by the National Science Council, Taiwan, Republic of China under Grants NSC 99-2221-E-194-032-MY3 and NSC 101-2221-E-194-031.

Received: 4 February 2012 Accepted: 8 January 2013
Published: 28 February 2013

References

1. TB Moeslund, A Hilton, V Kruer, A survey of advances in vision-based human motion capture and analysis. Comput. Vis. Image Understand. **104**(2), 90–126 (2006)
2. M Piccardi, Background subtraction techniques: a review, in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, vol. 4 (The Hague, The Netherlands, 2004), pp. 3099–3104
3. K Toyama, J Krumm, B Brumitt, B Meyers, Wallflower: principles and practice of background maintenance, in *Proceedings of the 7th IEEE International Conference on Computer Vision*, vol. 1 (Kerkyra, Greece, 1999), pp. 255–261
4. C Stauffer, WEL Grimson, Adaptive background mixture models for real-time tracking, in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 2 (Ft. Collins, CO, USA, 1999), pp. 246–252
5. T Bouwmans, FE Baf, B Vachon, Background modeling using mixture of Gaussians for foreground detection—a survey. Recent Patents Comput. Sci. **1**(3), 219–237 (2008)
6. A Shimada, D Arita, R Taniguchi, Dynamic control of adaptive mixture-of-Gaussians background model, in *Proceedings of the IEEE International Conference on Video and Signal Based Surveillance* (Sydney, Australia, 2006), pp. 1–5
7. R Tan, H Huo, J Qian, T Fang, Traffic video segmentation using adaptive-k Gaussian mixture model, in *Proceedings of the International Workshop on Intelligent Computing in Pattern Analysis/Synthesis* (Xi'An, China, 2006), pp. 125–134
8. H Wang, D Suter, A re-evaluation of mixture-of-Gaussian background modeling, in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2 (Philadelphia, PA, USA, 2005), pp. 1017–1020
9. J Lindstrom, F Lindgren, K Ltrstrom, J Holst, U Holst, Background and foreground modeling using an online EM algorithm, in *Proceedings of the IEEE International Workshop on Visual Surveillance* (Graz, Austria, 2006), pp. 9–16
10. B Han, X Lin, Update the GMMs via adaptive Kalman filtering, in *Proc SPIE*, vol. 5960 (Beijing, China, 2005), pp. 1506–1515
11. Y Zhang, Z Liang, Z Hou, H Wang, M Tan, An adaptive mixture Gaussian background model with online background reconstruction and adjustable foreground mergence time for motion segmentation, in *Proceedings of the IEEE International Conference on Industrial Technology* (Istanbul, Turkey, 2005), pp. 23–27
12. B White, M Shah, Automatically tuning background subtraction parameters using particle swarm optimization, in *Proceedings of the IEEE International Conference on Multimedia and Expo* (Beijing, China, 2007), pp. 1826–1829
13. HH Lin, JH Chuang, TL Liu, Regularized background adaptation: a novel learning rate control scheme for Gaussian mixture modeling. IEEE Trans Image Process. **20**(3), 822–836 (2011)
14. V Jain, B Kimia, J Mundy, Background modeling based on subpixel edges, in *Proceedings of the IEEE International Conference on Image Processing*, vol. 6 (San Antonio, Texas, USA, 2007), pp. 321–324
15. YL Tian, M Lu, A Hampapur, Robust and efficient foreground analysis for real-time video surveillance, in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, vol. 1, 2005, pp. 1182–1187
16. P Tang, L Gao, Z Liu, Salient moving object detection using stochastic approach filtering, in *Proceedings of IEEE International Conference on Image and Graphics* (Chengdu, China, 2007), pp. 530–535
17. R Caseiro, JF Henriques, J Batista, Foreground segmentation via background modeling on Riemannian manifolds, in *Proceedings of International Conference on Pattern Recognition* (Istanbul, Turkey, 2010), pp. 3570–3574
18. K Quast, A Kaup, Auto GMM-SAMT: an automatic object tracking system for video surveillance in traffic scenarios. EURASIP J. Image Video Process. **2011**(814285), 1–14 (2011)
19. D Farin, PHN de With, W Effelsberg, Robust background estimation for complex video sequences, in *Proceedings of the International Conference on Image Processing* (Barcelona, Spain, 2003), pp. 145–148
20. M Massey, W Bender, Salient stills: process and practice. IBM Syst J. **35**(3–4), 557–573 (1996)
21. H Wang, D Suter, A novel robust statistical method for background initialization and visual surveillance, in *Proceedings of the 7th Asian Conf. on Computer Vision, Part I* (Hyderabad, India, 2006), pp. 328–337
22. V Reddy, C Sanderson, BC Lovell, A Bigdeli, An efficient background estimation algorithm for embedded smart cameras, in *Proceedings of the IEEE International Conference on Distributed Smart Cameras* (Como, Italy, 2009), pp. 1–7
23. D Baltieri, R Vezzani, R Cucchiara, Fast background initialization with recursive Hadamard transform, in *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance* (Boston, MA, USA, 2010), pp. 165–171
24. A Colombari, A Fusiello, Patch-based background initialization in heavily cluttered video. IEEE Trans. Image Process. **19**(4), 926–933 (2010)
25. H Liu, W Chen, An effective background reconstruction method for complicated traffic crossroads, in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics* (San Antonio, TX, USA, 2009), pp. 1376–1381
26. J Scott, MA Pusateri, D Cornish, Kalman filter based video background estimation, in *Proceedings of the IEEE Applied Imagery Pattern Recognition Workshop* (Washington DC, USA, 2009), pp. 1–7
27. L Maddalena, A Petrosino, A self-organizing approach to background subtraction for visual surveillance applications. IEEE Trans. Image Process. **17**(7), 1168–1177 (2008)
28. A Ghasemi, R Safabakhsh, Unsupervised foreground-background segmentation using growing self organizing map in noisy backgrounds, in *Proceedings of the IEEE International Conference on Computer Research and Development*, vol. 1 (Shanghai, China, 2011), pp. 334–338
29. CC Chiu, MY Ku, LW Liang, A robust object segmentation system using a probability-based background extraction algorithm. IEEE Trans. Circuits Syst. Video Technol. **20**(4), 518–528 (2010)
30. SY Chien, YW Huang, BY Hsieh, SY Ma, LG Chen, Fast video segmentation algorithm with shadow cancellation, global motion compensation, and adaptive threshold techniques. IEEE Trans. Multimed. **6**(5), 732–748 (2004)
31. A Verdant, P Villard, A Dupret, H Mathias, Three novell analog-domain algorithms for motion detection in video surveillance. EURASIP J. Image Video Process. **2011**(698914), 1–13 (2011)
32. HH Lin, TL Liu, JH Chuang, Learning a scene background model via classification. IEEE Trans. Signal Process. **57**(5), 1641–1654 (2009)
33. Z Tang, Z Miao, Y Wan, FF Jesse, Foreground prediction for bilayer segmentation of videos. Pattern Recognit. Lett. **32**(14), 1720–1734 (2011)
34. C Zhao, X Wang, WK Cham, Background subtraction via robust dictionary learning. EURASIP J. Image Video Process. **2011**(972961), 1–12 (2011)
35. HH Hsiao, JJ Leou, An effective foreground/background segmentation approach for bootstrapping video sequences, in *Proceedings of the 2011 IEEE International Conference on Acoustics, Speech, and Signal Processing* (Prague, Czech Republic, 2011), pp. 1177–1180
36. M Ghanbari, *Standard Codecs: Image Compression to Advanced Video Coding* (The Institution of Engineering and Technology, London, UK, 2003)

37. L Atzori, FGB de Natale, C Perra, A spatio-temporal concealment technique using boundary matching algorithm and mesh-based warping (BMA-MBW). IEEE Trans. Multimed. **3**(3), 326–338 (2001)
38. T Thaipanich, PH Wu, CC Kuo, Video error concealment with outer and inner boundary matching algorithms, in *Proc. SPIE*, vol. 6696 (San Diego, CA, USA, 2007), pp. 1–11
39. YP Guan, Spatio-temporal motion-based foreground segmentation and shadow suppression. IET Comput. Vis. **4**(1), 50–60 (2010)
40. RC Gonzalez, RE Woods, *Digital Image Processing*, 3rd edn. (Pearson Prentice Hall, Upper Saddle River, NJ, USA, 2008)
41. N Goyette, P Jodoin, F Porikli, J Konrad, P Ishwar, Changedetection.net: a new change detection benchmark dataset, in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition* (Sherbrooke, Canada, 2012), pp. 1–8

doi:10.1186/1687-5281-2013-12
Cite this article as: Hsiao and Leou: Background initialization and foreground segmentation for bootstrapping video sequences. *EURASIP Journal on Image and Video Processing* 2013 **2013**:12.