Computer Vision and Image Understanding 000 (2016) 1–9

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Adaptive maintenance scheme for codebook-based dynamic background subtraction

Zhi Zeng*, Jianyuan Jia, Zhaofei Zhu, Dalin Yu

School of Mechano-electronics Engineering, Xidian University, Taibai Road, Xi'an 710071, China

ARTICLE INFO

Article history:
Received 18 December 2015
Revised 18 August 2016
Accepted 19 August 2016
Available online xxx

Keywords:
Background subtraction
Arbitrary cylindrical color model
Codebook model
Adaptive scheme

ABSTRACT

We propose a novel adaptive maintenance scheme for the codebook-based background subtraction algorithm. With this technique, the accuracy and efficiency of the model are significantly improved. In the proposed method, we develop an equal-qualification updating strategy to replace the maximum-negative-run-length-based filtering strategy. Further, we substitute the cache-based foreground learning process with a random updating scheme. These modifications not only preserve the accuracy of the codebook model but also significantly reduce the number of parameters used in the maintenance scheme. In the modified framework, parameters that are scenario-sensitive are identified through extensive experiments and analysis. Then, adaptive methods are proposed for them. The proposed method ensures the best performance of the system across a variety of complex scenarios. In our experiments, comparisons are provided to confirm that the performance of the codebook model is significantly improved owing to the adaptive technique. The overall performance of the proposed method is evaluated against more than 20 state-of-the-art methods using several modern datasets. It is demonstrated that, despite using only color information, the proposed method outperforms the majority of the solutions by a significant margin.

1. Introduction

Background subtraction is a fundamental step in the majority of computer vision applications including intelligent visual surveillance (Unzueta et al., 2012), intelligent visual observation of animals and insects (Mashak and Hosseini, 2012), optical motion capture (Guerra-filho, 2005), human-machine interaction (Senior et al., 2010), and content-based video coding (Paul et al., 2013). The purpose of a background subtraction algorithm is to distinguish moving objects from the scene.

Background subtraction is a challenging task because a real background in nature is not static. For example, a real background may consist of objects with repetitive motion, such as waving trees or grasses. Such a background has multiple appearances (Stauffer and Grimson, 1999). In this example, if we monitor a pixel of the charge-coupled device (CCD) sensor, we may observe several clusters of samples locate at different regions in the red, green, and blue (RGB) color space. Therefore, a basic background modeling technique such as running average (Lee and Hedley, 2002) or single Gaussian model (Wren et al., 1997) cannot model such a background effectively. To solve this problem, the Gaussian mixture model (GMM) (Stauffer and Grimson, 1999) was proposed where a mixture of Gaussian probability density functions is used for modeling each pixel. This method has been widely used and improved in terms of accuracy and robustness (Lee, 2005; Porikli and Tuzel, 2005; Zivkovic and Van Der Heijden, 2006). Many other nonparametric models have been developed to address this challenge, including Kernel density estimation (KDE) (Elgammal et al., 2002), the codebook model (Kim et al., 2005), and consensus-based model (Barnich and Van Droogenbroeck, 2011).

A real background may also contain surfaces with complex illumination variations such as shadows, highlights, and rippling water. For such a background, both direct and diffuse light contribute to the background illumination and the intensity of illumination may change quickly and drastically (Kim et al., 2005). To address this challenge, background features that tolerate complex illumination variations have been developed, including the gradient (Javed et al., 2002), edge (Jain et al., 2007), cylindrical color model (Kim et al., 2005), and local binary pattern (Heikklä et al., 2006).

Another challenge is that the dynamics of a background are not the same in different applications or even different regions of the same scene. This leads to inconsistent performance of a segmentation algorithm across different scenes or across different regions of the same scene (St-Charles et al., 2015a). Therefore, the majority of the methods can provide acceptable results on a specific sequence when adjusted accordingly. However, such adjustments lack flexibility and cannot address backgrounds consisting of regions of different dynamics effectively. To address this problem, several attempts have been made to adaptively adjust parameters

* Corresponding author.
E-mail address: zengzhi_2012@hotmail.com (Z. Zeng).

http://dx.doi.org/10.1016/j.cviu.2016.08.009
1077-3142/

according to the dynamics of the background (St-Charles et al., 2015a; Hofmann et al., 2012).

Even after many attempts, existing background segmentation techniques continue to have difficulty with these problems (Sobral and Vacavant, 2014; Bouwmans, 2014). A statistical report from a recent benchmark dataset indicates that dynamic background subtraction remains one of the most challenging tasks (Goyette et al., 2014). Therefore, how to develop an accurate, robust, and concise background model and its maintenance scheme is an open problem and attracts considerable attention. For a summary of related works, one can refer to several surveys (Sobral and Vacavant, 2014; Bouwmans, 2014; Piccardi, 2004; Radke et al., 2005; Nascimento and Marques, 2006; Brutzer et al., 2011).

As a well-known cluster model, the codebook-based background segmentation approach offers many advantages over traditional methods, such as its ability to work with illumination changes and process multi-appearance backgrounds; it also incurs less memory cost. The classical codebook model (Kim et al., 2005) has several distinguishing features including the cylindrical color model, codebook structure, and maximum negative run length-controlled filter process. During the past decade, many works have been dedicated to the improvement of the codebook model (Doshi and Trivedi, 2006; Guo et al., 2011; Zaharescu and Jamieson, 2011; Hu et al., 2012; Guo et al., 2013; Zeng and Jia, 2014). The classical codebook model uses a cylindrical color model that is developed based on the observation that under lightening variations, pixel values are almost totally distributed in an elongated shape along the axis going toward the origin of the RGB color space. This color model uses cylinders whose axes point to the origin of the RGB space to model the distribution of the pixel value. The geometrical parameters of this color model are the center, color distortion, and intensity bounds. A low-pass filter is used to update the center and the color distortion. The bound is updated using the maximum criteria. Incorrectly learned backgrounds are deleted using a maximum-run-length technique. This technique is developed under the assumption that an appearance that disappears for a long time should not be used for background modeling. Finally, new backgrounds are learned using a cache mechanism. When a foreground model stays in the cache sufficiently long, it is moved from the cache to the background model.

The codebook model involves many parameters, especially when multi-feature and multi-scale methods are used (Guo et al., 2011; Zaharescu and Jamieson; 2011). A common approach to tune these parameters is to optimize them according to an estimation function with respect to several training sequences. However, because some of these parameters are scenario sensitive, one must choose training sequences according to the application carefully (Kim et al., 2005; Guo et al., 2011). Further, such a video-level parameter-tuning scheme cannot address a background consisting of regions with highly different dynamics. Therefore, a robust and accurate pixel-level adaptive parameter-adjusting approach for the maintenance scheme is desired. Unfortunately, to our knowledge, there remains a deficiency of effective adaptive mechanisms to adjust the maintenance parameters for the codebook model owing to the complexity of its maintenance scheme.

To solve this problem, in this paper, a novel adaptive maintenance scheme is proposed for the codebook model in three steps. First, an equal-qualification updating strategy is developed to replace the maximum-run-length strategy. Second, a random updating scheme is used to replace the cache mechanism. Finally, parameters that are scenario sensitive are identified through extensive experiments and analysis. Then, adaptive methods are proposed for them. Experimental results indicate that these modifications not only preserve the accuracy of the codebook model but also significantly reduce the amount of scenario sensitive parameters used in the maintenance scheme.

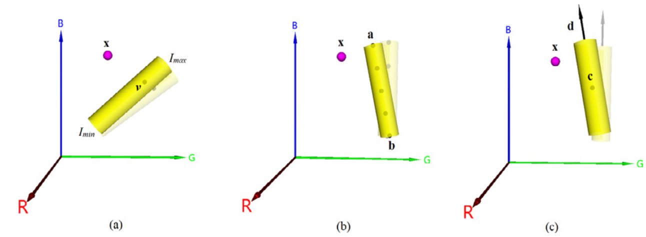

Fig. 1. Cylindrical color model, the arbitrary-cylindrical color model, and the modified arbitrary-cylindrical color model. Yellow cylinders are new color models and transparent cylinders are previous color models. Pink balls are new incoming samples. (a) Classical cylindrical color model. This color model is controlled by its center, illumination bounds, and the color distortion bound. Illumination bounds are updated according to the maximum criterion. Other parameters are updated through a low-pass filter. (b) Arbitrary cylindrical color model. This color model is controlled by its two ending points and the color distortion bound. It is updated by solving a linear regression problem. Approximations of previous samples are drawn in gray dots along the axis of the cylinder. (c) Modified arbitrary-cylindrical color model. This color model is controlled by its center, direction, and two bounds. All parameters are updated through a low-pass filter. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

In the proposed model, two sets of parameters are scenario sensitive: the geometry of the new added code word and the learning probability. The geometry of a new added code is set to be the weighted average of code words in the codebook. The learning probability is determined according to the contrast between the misclassified foreground and background color model and the spacial-temporal consistency of the segmentation results. Experiments demonstrate that this adaptive scheme is effective in addressing camouflage-moving objects under dynamic backgrounds.

In our experiments, we compare the proposed method with the classical codebook model. It is verified that the performance of the codebook model is significantly improved using the proposed adaptive maintenance scheme. Moreover, the proposed method ensures the best performance of the system across a variety of complex scenarios without manual tuning. The overall performance of the proposed method is also evaluated against more than 20 state-of-the-art methods on a modern benchmarks dataset (Goyette et al., 2014). For a more comprehensive comparison, we also evaluate the proposed method against six states-of-the-art methods on eight dynamic scenes collected from three other datasets (Li et al., Nov. 2004; Bloisi et al., 2011; Toyama et al., 1999). Experimental results confirm that despite using only color information, the proposed method outperforms the majority of the solutions. Please note that our program is available on line at https://github.com/zengzhi2015/AAC.

The codebook model and its modifications are introduced in Section 2. The proposed adaptive scheme is presented in Section 3. Implementation details are explained in Section 4. In Section 5, experiments are conducted to evaluate the proposed method. Finally, the paper is concluded in Section 6.

2. Codebook model and its modifications

2.1. Classical codebook model

In the classical codebook model, each pixel is modelled by a codebook containing one or more code words. Each code word is represented by an RGB vector $\mathbf{v}_i = \{R_i, G_i, B_i\}$ and a 6-tuple $\mathbf{aux}_i = \{I_{\min}, I_{\max}, f_i, \lambda_i, p_i, q_i\}$, where $I_{\min}$ and $I_{\max}$ are the minimum and the maximum brightness, respectively. f_i is the frequency with which the code word has occurred. λ_i is the maximum-negative-run-length, meaning the longest time interval during which this code word has not been accessed. p_i and q_i are the first and last access times of the code word, respectively. The updating process of a code word is illustrated in Fig. 1(a). The working