Image and Vision Computing 37 (2015) 31–47

Contents lists available at ScienceDirect

Image and Vision Computing

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

Editor's Choice Article

Local color transformation analysis for sudden illumination change detection[☆]

Francisco Javier López-Rubio, Ezequiel López-Rubio [*]

Department of Computer Languages and Computer Science, University of Málaga, Bulevar Louis Pasteur, 35, 29071 Málaga, Spain

ARTICLE INFO

Article history:
Received 1 October 2013
Received in revised form 26 May 2014
Accepted 30 March 2015
Available online 3 April 2015

Keywords:
Background modeling
Foreground detection
Illumination invariance
Color transformation

ABSTRACT

Sudden illumination changes are a fundamental problem in background modeling applications. Most strategies to solve it are based on determining the particular form of the color transformation which the pixels undergo when an illumination change occurs. Here we present an approach which does not assume any specific form of the color transformation. It is based on a quantitative assessment of the smoothness of the local color transformation from one frame to the background model. In addition to this, an assessment of the obtained illumination states of the pixels is carried out with the help of fuzzy logic. Experimental results are presented, which demonstrate the performance of our approach in a range of situations.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The segmentation of the foreground objects in a scene is a fundamental task which lays at the foundation of many computer vision systems. Most approaches to this task create a model of the background that is updated progressively as time passes. Consequently sudden illumination changes are challenging problems, since the appearance of the background no longer matches the past observations. This issue has been studied for many years [6,9,10,20] due to its relevance to applications. Practical computer vision systems are placed in environments where the illumination conditions vary through time. For example, indoor scenes frequently exhibit light switching, while passing clouds affect outdoor cameras.

There are different ways to address this problem. On one side there are pixel-level algorithms that work by analyzing the scene on a pixel by pixel basis, so that an independent decision is made for each pixel. On the other hand, there are other approaches which work at a higher level, namely block-level (where the decision for one pixel depends on the information from several nearby pixels) or frame-level (where all the pixels of the frame might be taken into account to decide whether a pixel belongs to the foreground). It is often the case that methods use multiple levels simultaneously to analyze the video. A representative case of this approach is the Wallflower system 28, which uses the three levels already mentioned. One of the downsides attributed to this method is that it uses a non flexible criterion in real situations because it selects a set of background scene models representing different situations and each frame has to be assigned to the model which produces the fewest foreground pixels. This implies that each possible situation must be predicted in advance, and that a representative background model of each situation must be found. Hence, the approach is less adequate for scenes where unpredictable events occur which affect the background, such as a left object or a parked car which starts moving.

There are other methods using region-level analysis as in the case of [22]. This approach analyzes the image at block and pixel levels. The main idea is that each pixel belongs to several overlapping blocks, so that it is determined whether it belongs to the background or not depending on how it has been classified in each of these blocks.

A more recent algorithm is [14], which uses pixel-level and image-level elements. The proposal consists in using a two-layer architecture based on a Gaussian Mixture Model to represent the background. Then the result is optimized using a Markov random field decision framework.

On the other hand, there are many algorithms that work at the pixel level. For example [27] is based on applying a homomorphic filter, while [12] performed an analysis with stereo vision and employs a disparity model created offline to mitigate the consumption of extra CPU time required for this type of processing. Also, [8] uses discriminative texture features to capture background statistics, by means of texture operators named local binary patterns (LBP). Finally, [6] presents an adaptive algorithm that uses multiple feature subspaces and Principal Component Analysis to capture and learn different lighting conditions.

Our aim here is to develop an illumination change detection system which works along an existing foreground detection algorithm. A

[☆] Editor's Choice Articles are invited and handled by a select rotating 12 member Editorial Board committee. This paper has been recommended for acceptance by Dr. Bodo Rosenhahn.

[*] Corresponding author.

E-mail addresses: xavierprof@hotmail.com (F.J. López-Rubio), ezeqlr@lcc.uma.es (E. López-Rubio).

URL: http://www.lcc.uma.es/%7Eezeqlr/index-en.html (E. López-Rubio).

http://dx.doi.org/10.1016/j.imavis.2015.03.001
0262-8856/© 2015 Elsevier B.V. All rights reserved.

previous example of an add-on illumination change detection algorithm can be found in [32]. Unlike our proposal, they assume that the order of pixel values is preserved in local neighborhoods when illumination changes occur, and they root their proposal on the analysis of physical properties, namely the radiance. In contrast to this, we are not interested in the particular form of the color transformation, but in its smoothness properties. This way, we consider that any color transformation which is not smooth can not correspond to a lighting change, i.e. it is due to a foreground object. Hence, we are able to detect variations in illumination irrespective of the particular features of the color transformation at hand. Moreover, the procedure is largely independent from the baseline background model, so it can be used to improve a number of well known methods which are not specifically designed to work under illumination changes.

This paper is organized as follows. Section 2 presents the illumination change detection method. Section 3 presents some experimental results to demonstrate the ability of our approach to manage complex scenes. The main features and properties of our proposal are discussed in Section 4. Finally, Section 5 is devoted to conclusions.

2. The method

The illumination change management procedure that we present here has two parts. The first one classifies the pixels of the current frame according to its current state with respect to illumination changes (Subsection 2.1). The second part uses this information to decide which pixels must undergo a reset because an illumination change has rendered their background models outdated (Subsection 2.2).

2.1. Illumination state estimation

Here we must estimate the illumination state of each pixel of the current video frame. The illumination state of pixel i is formed by three fuzzy variables *Rough*, *Difference*, and *Baseline*; their values (membership degrees) will be noted α_i, β_i, $\gamma_i \in [0, 1]$, respectively. The interpretation of the variables is as follows:

- *Rough* indicates whether the transformation of the colors in the previous frame to the colors in the current frame is not smooth in the vicinity of pixel i. If α_i is high, then it is unlikely that an illumination change is happening, since illumination changes produce smooth changes in the colors of the background and the foreground objects.
- *Difference* indicates whether the color of pixel i in the current frame is very different from that stored in the background model for pixel i. If β_i is high, then either an illumination change is happening or a foreground object is present.
- *Baseline* indicates whether the baseline background model has detected a foreground object. Please note that $\gamma_i \in \{0, 1\}$ for background models that do not output a degree of confidence for the foreground detection.

Next we describe how to compute the fuzzy membership values α_i and β_i; please note that γ_i is simply the output of the baseline background model.

Let us center our attention in a small neighborhood W_i of pixel i. In our experiments we have considered square windows of size 5×5 pixels, which offer a good tradeoff between efficiency and accuracy. Now we define a vector field which represents the transformation of the colors of the neighborhood W_i in the background model to the colors of W_i in the current frame:

$$f_i : \mathbb{R}^3 \rightarrow \mathbb{R}^3 \tag{1}$$

where tristimulus color values are assumed, but any color space could be used. Now, our task is to detect those vector fields f_i which are not smooth. A simple and fast way of measuring the smoothness of f_i is based on the computation of the following quotients:

$$q_i(j,k) = \frac{a_i(j,k)}{b_i(j,k)} \tag{2}$$

$$a_i(j,k) = \left\| f_i\left(\mathbf{x}_j\right) - f_i(\mathbf{x}_k) \right\|^2 \tag{3}$$

$$b_i(j,k) = \left\| \mathbf{x}_j - \mathbf{x}_k \right\|^2 \tag{4}$$

where $\mathbf{x}_j$ and $\mathbf{x}_k$ are the colors in the background model of two pixels $j, k \in W_i$, and $f_i(\mathbf{x}_j)$ and $f_i(\mathbf{x}_k)$ are the colors of those pixels in the current frame. The vector field f_i is non smooth whenever q_i attains high values for some pairs $j, k \in W_i$. As seen in Fig. 1, a smooth transformation is one that maps similar colors in the background model to similar colors in the current frame, even if the colors change considerably from the background model to the current frame. That is, the distances $\|\mathbf{x}_j - f_i(\mathbf{x}_j)\|$ are irrelevant to the smoothness of f_i. This way we can manage the switching of lights of any color, for example yellow or blue lights. It must be pointed out that low values of q_i are not interesting because they are usually associated with homogeneous foreground objects passing in front of textured backgrounds, which are adequately managed by standard foreground detection algorithms.

In practice pixel noise can lead to large errors in the estimation of q_i, in particular if the denominator b_i contains noise. We alleviate this by considering a filtered quantity ϕ_i:

$$\phi_i(j,k) = \begin{cases} \dfrac{a_i(j,k)}{B_{low}} & \text{if } b_i(j,k) < B_{low} \\ 0 & \text{if } b_i(j,k) > B_{high} \\ q_i(j,k) & \text{otherwise} \end{cases} \tag{5}$$

where B_{low} and B_{high} are suitable lower and upper thresholds for the denominator value b_i, with $B_{low} < B_{high}$. Please note that B_{low} manages low lighting conditions, where pixel values have little precision. On the other hand, B_{high} ensures that highly textured backgrounds (which are associated to high values of a_i and b_i) are not mistaken as non

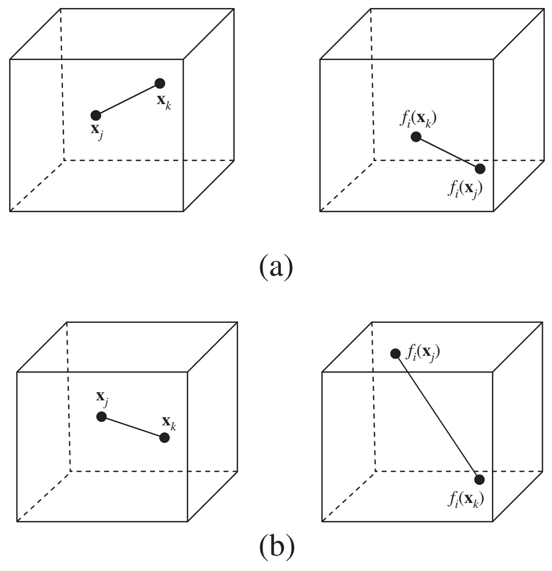

Fig. 1. Color transformations from the background model to the current frame: (a) smooth, (b) rough. The cubes represent the tristimulus color space.