Classification of change detection algorithms for object-based applications

Andrea Cavallaro and Touradj Ebrahimi
Swiss Federal Institute of Technology, CH-1015 Lausanne, Switzerland
Tel: +41 21 693 4728; fax: +41 21 693 7600; e-mail: andrea.cavallaro@epfl.ch

ABSTRACT

Change detection is a temporal segmentation tool aiming at identifying changes in image sets or image sequences at two different times. Many change detection algorithms have been proposed over the past decade for the generation of video objects in a wide range of applications, ranging from interactive multimedia to remote surveillance. Most of these algorithms are tailored to the specific application at hand. Therefore there is a need of a general model for change detection which could support the choice of the optimal change detection strategy. We present a unified approach to change detection based on a 4-step model. Change detection algorithms proposed in the literature are analysed in terms of the four building blocks of the proposed model. This study is useful not only for reviewing the state of the art of change detection, but also for refining and improving change detection methods, by providing useful guidelines for the better use of the algorithms.

1 Introduction

Change detection is an important tool for many computer vision applications such as multimedia, video surveillance, remote sensing, interactive and immersive gaming. Efficient and accurate techniques are required to detect and label the changes in image sets or image sequences between different time instants.

The general requirements for change detection algorithms are accuracy in the detection of object contours (*spatial accuracy*), and temporal stability of the detection (*temporal coherence*). Moreover, *sensitivity* and *robustness* are desirable features in a change detection algorithm. Sensitivity refers to the ability of detecting changes of small magnitude. Robustness can be seen as the property of providing good accuracy under varying conditions, for example illumination changes. Other requirements may also be introduced depending on the application. For example, *a priori* constraints of spatial connectivity can be applied, if this property characterizes the target of the detection process.

If we denote the image under test as $f(x, y, n)$, where x and y are the spatial coordinates, the problem of change detection consists in finding, for each frame n, a binary map $c(x, y, n)$, defining the pixels in $f(x, y, n)$, which have changed with respect to the reference image $f(x, y, r)$. The binary mask $c(x, y, n)$, resulting from the change detection analysis, is defined as

$$c(x, y, n) = \begin{cases} 1 & \text{if a change occurred at } (x, y) \text{ at time } n, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Different strategies can be adopted to compute $c(x, y, n)$ as a function of $f(x, y, n)$ and $f(x, y, r)$. We propose a decomposition of the change detection problem into four major steps, namely *feature extraction*, *feature analysis*, *classification*, and *post-processing*. This general structure allows us to compare the different techniques proposed in the literature. The block diagram of the proposed scheme is shown in Figure 1.

The choices related to the change detection algorithm comprise the features to extract from $f(x, y, n)$ and $f(x, y, r)$, the distance metric to quantify changes, and the classification strategy to detect changes. These choices are discussed in the following sections.

2 A 4-step model for change detection

2.1 Feature extraction

Each input frame $f(x, y, n)$ is transformed into the most appropriate feature space for the specific application. The result of the feature extraction step is a sequence, $g(x, y, n)$, on which the change detection operation will be performed. The signal $g(x, y, n)$ is derived from one image only (i.e., it does not contain motion information). To simplify the notation in this section, we will omit the time variable n. The signal $g(x, y)$ may represent the *luminance* [1, 2, 3, 4, 5, 6], the *color components* [7, 9], or more complex features. More complex features are usually employed in applications where the illumination of the scene cannot be easily modeled. Level lines [10], edge maps [11], vector representation of image regions [12], and albedo images [13] are examples of features which are more robust than intensity images for change detection under varying illumination.

Let us examine the different features. A method based on *level line* image representation is presented in [10].

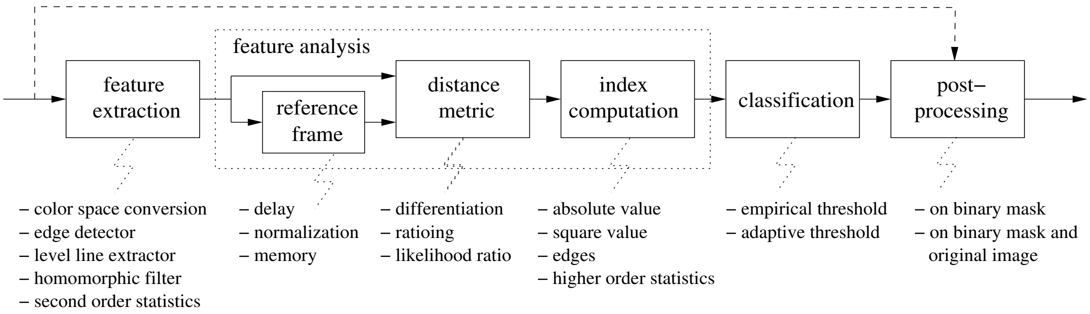

Figure 1: The four major steps of a change detection algorithm and different choices for their implementation

This approach exploits the fact that a global illumination variation changes the number, but not the the geometry of the level lines. A level line λ is the boundary of a level set. A level set S_λ consists of all image pixel positions with a luminance value at least equal to λ:

$$S_\lambda = \{(x,y) : f(x,y) \geq \lambda\}. \tag{2}$$

The image containing the level lines can be expressed, starting from the input $f(x,y)$, as a *level line map*, $g(x,y)$. A level line map can be defined as

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) = \lambda_i \quad \text{with} \quad i = 1,\ldots,N, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where N is the number of level lines considered for creating the map.

Based on the same principle as level lines, algorithms based on *edges* avoid illumination variations to be interpreted as structural changes [11]. The use of edges provides good spatial accuracy. In addition, since edge maps are bi-level images, they are convenient from computation and storage viewpoints.

Another illumination invariant feature for change detection is the *albedo image*. The albedo image represents the reflectance component of an image, which is independent from the illumination and contains mainly physical object information. If we model the intensity $f(x,y)$ as the product of reflection $r(x,y)$ and illumination $i(x,y)$, $f(x,y) = r(x,y)i(x,y)$, it is possible to extract the reflectance component (albedo) by applying a homomorphic filter [13] to the input intensities. First, the illumination and the reflectance component are separated by applying a log-filter, $\log f(x,y) = \log r(x,y) + \log i(x,y)$, which transforms the multiplicative relation between $r(x,y)$ and $i(x,y)$ into an additive one. Logarithmically transformed images basically have two spectral components: a low-frequency peak associated mainly with illumination, and a mid- and high-frequency plateau associated mainly with reflectance. To extract the reflection component, the high frequencies are eliminated through low-pass filtering $\mathcal{LP}$,

$$g(x,y) = \exp\left[\log f(x,y)\right] - \mathcal{LP}\left[\log f(x,y)\right], \tag{4}$$

thus providing the illumination invariant feature on which the change detection analysis can be applied.

A different approach to extracting features $g(x,y)$ for change detection involves modelling the intensity distribution of the signal $f(x,y)$. The model is generally characterised by region based statistics. A second order model is usually employed to describe the local intensity distribution in a region of an image, for example the variance and the mean of the region [14], a quadratic function modeling the region [15], or its partial derivatives [2]. These methods based on second-order statistics have been employed in early change detection techniques. They are based on modelling the unknown signal. They are not used in modern change detection and they are reported here only for sake of completeness.

2.2 Feature analysis

In order to detect significant changes, features in the current frame $g(x,y,n)$ are compared to those in a reference frame $g(x,y,r)$. The reference frame may be the *previous frame* in the sequence, or an image representing the *background* of the scene. The background frame can be fixed, or it can be updated periodically. In the former case, it is usually a frame taken from the sequence (e.g., the first frame [5, 7]). In the latter case, temporal integration of the previous frames [16] is used.

We represent the general transformation $\mathcal{T}$ for comparing the features as a composition of a distance operator $\mathcal{T}_d$ and a function $\mathcal{T}_1$, so that $t(x,y,n) = t_1(t_d(g(x,y,n), g(x,y,r)))$. The distance operator provides a pixel level feature distance and can be implemented as pixel-wise difference, image ratioing, vector difference, or a difference metric based on second order statistics. *Pixel-wise difference* (image differentiation) can be expressed as $t_d(x,y,n) = g(x,y,n) - g(x,y,r)$. This operator is applied to intensity images representing luminance [16] or color [7], and to binary maps representing edges [11] or level lines [10]. *Vector difference* is a distance metric that operates on vector features. The vectors may consist of color features or region features [12]. The *image ratio* represents the ratio between the value of intensity of the same pixel in different time instants:

$$t_d(x,y) = \frac{g(x,y,n)}{g(x,y,r)} \tag{5}$$

This computation can be done at pixel or at region level and is robust to illumination variations. Difference metrics based on *second order statistics* can also be used. These techniques compare the intensities in a region surrounding each pixel in $g(x,y,n)$ and $g(x,y,r)$ by means of some characteristic functions. A difference metric based on second order statistics in a region W is the *likelihood ratio* [14]

$$t_d(x,y,n) = L = \frac{1}{\sigma_n^2\sigma_r^2}\left[\frac{\sigma_n^2+\sigma_r^2}{2} + \left(\frac{\mu_n-\mu_r}{2}\right)^2\right]^2 \tag{6}$$

where (σ_n^2, μ_n) and (σ_r^2, μ_r), are the variance and the mean in W for current and reference images, respectively.

After the distance operator T_d, the images $t_d(x,y,n)$ can be further transformed to derive the activity index which is used for change detection. In some cases, the result $t_d(x,y,n)$ of the distance operator is taken directly as activity index, and the function T_i is therefore the identity function. Examples of this transformation T_i are absolute value, square value, second or fourth order moments, and edges. They can be applied to different spatial supports. If the features are image intensities or binary masks, that is, edges or level lines, the *absolute* or *square value* is applied after the distance operator *image difference*. Then $t(x,y,n) = \|t_d(x,y,n)\|_p$ with $p = \{1,2\}$, representing the absolute or square value, respectively. *Moments* on rectangular windows W are computed when features are intensity images (luminance or color components). In this case, the index assumes the following form

$$t(x,y,n) = \frac{1}{N^2}\sum_{(i,j)\in W_{(x,y)}} \left(t_d(i,j,n)-\mu\right)^s \tag{7}$$

where $t_d(x,y,n)$ is either a difference image or a ratio image, s is the order of the moment, and μ the mean. Different combination of $t_d(x,y,n)$ and s have been used to compute the activity intex. In [1], $t_d(x,y,n)$ is the result of image differencing on luminance intensities and $s = 4$, that is $t(x,y,n)$ is a fourth order moment. In [2, 12], $t_d(x,y,n)$ is the result of image rationing on luminance values and $s = 2$. Thus $t(x,y,n)$ is second order moment (shading model). This index of activity can be attributed to the entire region [2] (overlapped windowing) or to the central pixel [12] (non-overlapped windowing). This second solution provides a better spatial resolution.

Finally, an *edge detector* can be used as T_i after the distance metric to provide robustness to illumination variations [7].

2.3 Classification

The indicator of the level of activity $t(x,y,n)$ computed in the previous section has to be classified in one of the two classes: *changed* or *unchanged*. To obtain this classification, $t(x,y,n)$ is binarized by thresholding. The threshold can be set empirically [2, 12, 14] or computed adaptively [16, 1, 3, 4, 8, 6, 13, 9]. In the former case, the threshold is fixed for all pixels in frame and all the frames in the sequence. The value is usually determined experimentally based on a large database. In the latter case, the threshold is adapted according to some rules. To compute an *adaptive* and *local* threshold, a region based statistical analysis can be used if the probability density function of the camera noise is known. The statistical analysis is based on modeling the intensity distribution of noise [16, 1, 3, 4, 6]. Instead of thresholding the difference image, this approach compares the statistical behavior of a small neighborhood at each pixel position in the difference image to a model of the noise that could affect the difference image. The comparison is based on a significance test.

A different way of computing an *adaptive* and *local* threshold is presented in [8], where the spatial contextual information of each pixel is taken into account. The threshold for the test is adapted to the label constellation $c(x,y,n)$ of a 3-by-3 window W. When scanning the image, only four neighboring labels in the causal part of the neighborhood W are available. For this reason, the value of the other four is approximated with the labels of the previous classification $c(x,y,n-1)$. If $K(x,y,n)$ denotes the number of pixels detected as changed in the neighborhood W at time n, the new test becomes

$$t(x,y,n) > \tau + \theta_1\left[4 - K(x,y,n)\right] \tag{8}$$

with $0 \leq K(x,y,n) \leq 8$, and θ_1 is a positive potential which determines the range of $t(x,y,n)$. This spatially adaptive threshold allows the creation of compact and smoothly shaped regions of change, and reduces scattered errors caused by noise. An extension of this method is presented in [13]. Besides contextual spatial information, temporal information is integrated in the computation of the adaptive threshold. In this case, the label $c(x,y,n-1)$ of pixel (x,y) in the previous frame is also considered. The spatio-temporally adaptive threshold is given by

$$t(x,y,n) > \tau + \theta_1\left[4 - K(x,y,n)\right] + \theta_2\left[0.5 - c(x,y,n-1)\right] \tag{9}$$

with $c(x,y,n-1) \in \{0,1\}$, and θ_1 and θ_1 are positive potentials.

2.4 Post-processing

The result of the classification step, $c(x,y,n)$, is affected by different kinds of noise. Examples are artifacts inherent to the specific change detection algorithms, shadows, uncovered background, and camera noise (which may be not completely eliminated by the detection algorithm). All these sources of noise are responsible for false alarms in the change detection mask $c(x,y,n)$.

To reduce the probability of false alarms, various post-processing strategies have been proposed in the literature. These strategies are applied either to the binary image result of the classification only, or to both the binary images and the original frame. In the former case, the post-processing is based on four- or *eight-connected components analysis* or *morphological filtering* [1, 9]. The use of binary masks in the post-processing stage has the advantage of reducing the false alarm probability at low computational cost. However, the *a priori* topological assumptions (compactness and regular contours) on which they are based may not always be valid. For this reason, these techniques often result in blocky contours. To solve this problem, the original sequence may be used along with the classification result in the post-processing. Motion, color and edge information are typical examples of features that are analysed to improve the spatial accuracy of the change detection result [19].

3 Conclusions

In the framework of semantic video object extraction based on motion, a unified approach to change detection has been proposed. This approach is based on a decomposition of the problem into four major steps: feature extraction, feature analysis, classification, and post-processing. The different choices and their implications in the design of change detection algorithms have been discussed. This study is useful not only for reviewing the state of the art of change detection, but also for refining and improving change detection methods, by providing useful guidelines for the better use of the algorithms. Our current work is the integration of the presented analytical comparison with a quantitative comparison based on objective evaluation metrics.

References

[1] A. Neri, S. Colonnese, G. Russo, and P. Talone, "Automatic moving object and background separation," *Signal Processing*, vol. 66, no. 2, pp. 219–232, 1998.

[2] K. Skifstad and R. Jain, "Illumination independent change detection for real world image sequences," *Computer Vision, Graphics, and Image Processing*, vol 46, pp. 387–399, june 1989.

[3] M. Hotter, R. Mester, and F. Muller, "Detection and description of moving objects by stochastic modelling and analysis of complex scenes," *Signal Processing: Image Communication*, vol. 8, pp. 281–293, 1996.

[4] T. Aach, A. Kaup, and R. Mester, "Statistical model-based change detection in moving video," *Signal Processing*, vol 31, pp. 165–180, 1993.

[5] S.C.Liu, C.W.Fu, and S.Chang, "Statistical change detection with moments under time-varying illumination," *IEEE Transactions on Image Processing*, vol. 7, no. 9, pp. 1258–1268, 1998.

[6] R. Mech and M. Wollborn, "A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera," *Signal Processing*, vol. 66, no. 2, pp. 203–217, 1998.

[7] A. Cavallaro and T. Ebrahimi, "Change detection based on color edges," in *Proc. of IEEE International Symposium on Circuits and Systems*, Sydney, Australia, 2001.

[8] T. Aach and A. Kaup, "Bayesian algorithms for change detection in image sequences using Markov random fields," *Signal Processing: Image Communication*, vol 7, no. 2, pp. 147–160, 1995.

[9] R. Costantini et al."Countering illumination variations in a video surveillance environment," in *Proc. of SPIE, Electronic Imaging Conference*, San Jose, USA, 2001.

[10] D. Aubert, "Passengers queue measurement," in *Proc. of 10th International Conference on Image Analysis and Processing*, Venice, Italy, pp. 1132–1135, 1999.

[11] A. Makarov, J. Vesin, and F. Reymond, "Intrusion detection robust to slow and abrupt lighting changes," in *Proc. of SPIE Electronic Imaging - Real Time Imaging*, San Jose, USA, pp. 44–54, 1996.

[12] E. Durucan and T. Ebrahimi, "Robust and illumination invariant change detection based on linear dependence for surveillance applications," in *Proc. of X EUSIPCO*, Tampere, Finland, pp. 1041–1044, 2000.

[13] D. Toth, T. Aach, and V. Metzler, "Bayesian spatio-temporal motion detection under varying illumination illumination-invariant change detection," in *Proc. of X EUSIPCO*, Tampere, Finland, pp. 3–7, 2000.

[14] R. Jain and H. Nagel, "On the analysis of of accumulative difference pictures from image sequences of real world scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 1, pp. 206–214, 1979.

[15] Y. Hsu, H. Nagel, and G.Rekers, "New likelihood test methods for change detection in image sequences," *Computer Vision, Graphics and Image Processing*, vol 26, pp. 73–106, 1984.

[16] A. Cavallaro and T. Ebrahimi, "Video object extraction based on adaptive background and statistical change detection," in *Proc. of SPIE, Visual Communications and Image Processing*, San Jose, California, USA, pp. 465–475, 2001.

[17] R. Lillestrand, "Techniques for change detection," *IEEE Transactions Computer*, vol. 21, no. 7, pp. 654–659, 1972.

[18] M. Ulstad, "An algorithm for estimating small scale differences between two digital images," *Pattern Recognition*, vol. 5, pp. 323–333, 1973.

[19] T. Horprasert, D. Harwood, and L. Davis, "A statistical approach for real-time robust background subtraction and shadow detection," in *Proc. of Int. Conf. on Computer Vision*, pp. 436–439, 1999.