Pattern Recognition Letters 31 (2010) 2286–2294

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

Mixed-state causal modeling for statistical KL-based motion texture tracking

Tomás Crivelli [a,*], Bruno Cernuschi-Frias [a,b], Patrick Bouthemy [c], Jian-Feng Yao [d]

[a] LIPSIRN, University of Buenos Aires, Paseo Colón 850, 1063 Bs. As., Argentina
[b] CONICET, Argentina
[c] INRIA Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France
[d] IRMAR, University of Rennes I, Campus de Beaulieu, 35042 Rennes Cedex, France

ARTICLE INFO

Article history:
Received 20 May 2009
Available online 11 July 2010
Communicated by Y.J. Zhang

Keywords:
Mixed-state Markov models
Motion textures
Visual tracking
Kullback–Leibler divergence

ABSTRACT

We are interested in the modeling and tracking of dynamic or motion textures, which refer to dynamic contents that can be classified as a texture with motion (fire, smoke, crowd of people). Experimentally we observe that they depict motion maps with values of a mixed type: a discrete value at zero (absence of motion) and continuous non-null motion values. We thus introduce a temporal mixed-state Markov model for the characterization of motion textures from which a set of 13 parameters is extracted as the descriptive feature of the dynamic content. Then, a motion texture tracking strategy is proposed using the conditional Kullback–Leibler (KL) divergence between mixed-state probability densities, which allows us to estimate the position using a statistical matching approach.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

In the context of visual motion analysis, *dynamic* or *motion textures* (Nelson and Polana, 1992; Doretto et al., 2003; Bouthemy et al., 2006; Crivelli et al., 2006) are dynamic video contents that exhibit some type of stationarity or regularity, both in the spatial and temporal dimension, and have an indeterminate spatio-temporal extent. Mostly, they refer to dynamic video contents displayed by natural scene elements such as rivers, sea-waves, smoke, moving foliage, fire, etc. They also encompass any dynamic visual information that, from the observer point of view, can be classified as a texture with motion (Fig. 1). For example, consider a walking person. This *activity* can be analyzed as attached to an articulated motion; however a group of persons or a crowd walking, observed from a wide angle may show a repetitive motion pattern, more adequate to be considered as a motion texture.

In this work we are interested in the modeling and tracking of motion textures. Critical vision-based surveillance applications such as fire or smoke detection or tracking an agitated crowd of people need for a compact representation of this type of dynamic phenomena. Here, we focus on the temporal modeling of the apparent motion maps depicted by motion textures and study the ability of the proposed model to be used as a powerful representation for tracking applications.

As pointed out in (Bouthemy et al., 2006), we observe that such motion maps exhibit values of a mixed type: a discrete component at zero (null motion) and continuous motion values (Fig. 1). Motion observations are then modeled using *mixed-state random variables*.

In this context, we introduce a *mixed-state Markov chain* (*MS-MC*) model for the statistical characterization of motion textures from which a set of 13 parameters is extracted as the descriptive feature of the dynamic content. Then a motion texture tracking strategy is proposed based on the computation of the conditional Kullback–Leibler (KL) divergence between mixed-state probability densities. This solves the problem of matching and thus it allows us to estimate the displacement of a motion texture.

1.1. Related work

Although 2D spatial textures have been vastly analyzed in the computer vision literature, temporal or dynamic textures have attracted an increasing interest in the last few years. A first distinction between different approaches, resides on the type of image features extracted from the image sequence. Doretto et al. (2003) have proposed the use of ARMA models directly applied to image intensities with convincing results on dynamic texture synthesis. This modeling approach has resulted in several variations and improvements (Vidal and Ravichandran, 2005; Chan and Vasconcelos, 2009).

Other approaches are based on extracting motion features from the image sequence instead of considering pixel-wise intensity values (Bouthemy et al., 2006; Crivelli et al., 2006; Fazekas et al., 2009). Particularly, normal flow is a very efficient and natural way of locally characterizing a dynamic texture (Fazekas and Chetverikov, 2005).

More related to the theoretical aspect of this paper, we can also mention other models that exploit the interaction between discrete and continuous values in computer vision problems. Starting

* Corresponding author. Tel.: +54 11 4343 0891x278; fax: +54 11 4804 4585.
E-mail addresses: tcrivell@irisa.fr (T. Crivelli), bcf@ieee.org (B. Cernuschi-Frias), Patrick.Bouthemy@inria.fr (P. Bouthemy), jian-feng.yao@univ-rennes1.fr (J.-F. Yao).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.06.016

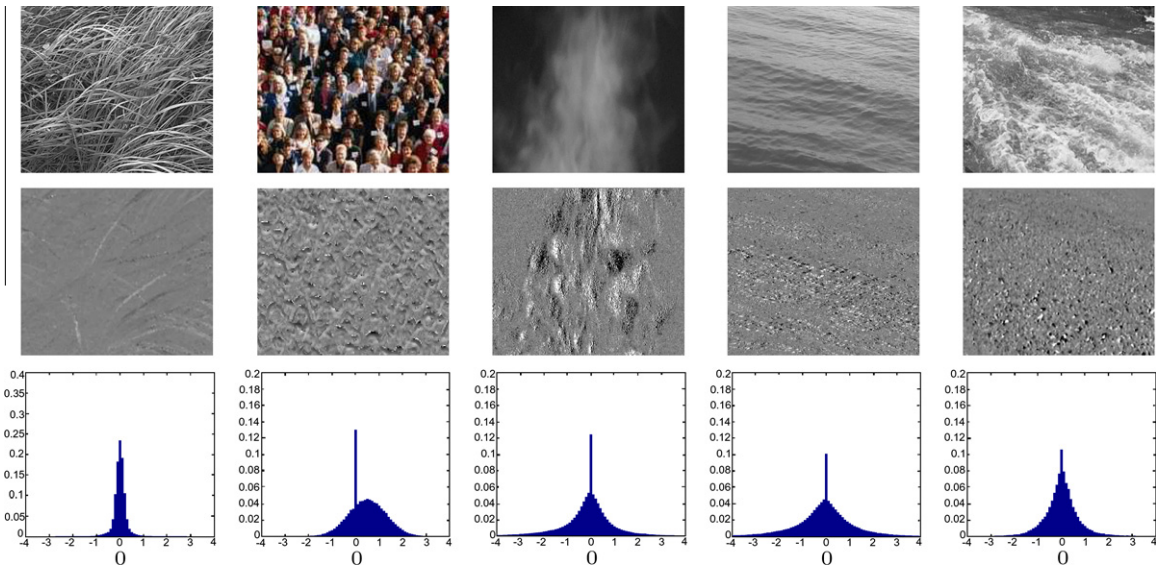


Fig. 1. Top row: sample images from videos of dynamic textures of different kind (grass, crowd, steam, water and river). Middle row: scalar motion map based on normal flow computation and obtained using two consecutive frames of the sequence, which we call a motion texture. Bottom row: motion histograms from a motion texture. Motion values display two components: a discrete value at zero and a continuous distribution for the rest.

with the seminal paper of Geman and Geman (1984) with the introduction of a *line process* for modeling edges between homogeneous image regions to be restored from different types of degradations. Then, it is worth mentioning previous works on fuzzy pixel classification as Salzenstein and Collet (2006). These authors introduce a class of fuzzy Markov models where each state variable, or classification variable, $x_i \in [0,1]$ represents a classification rate with hard ($x_i = 0$ or $x_i = 1$) and soft ($x_i \in (0,1)$) classification states following a continuous distribution. Finally we have the recently formulated mixed-state Markov random fields model that has been applied with promising results to modeling, segmentation and classification of motion textures (Bouthemy et al., 2006; Crivelli et al., 2006).

Here, we follow the same type of approach than Bouthemy et al. (2006) and Crivelli et al. (2006) but we extend it to temporal modeling, and consequently, we introduce causal mixed-state Markov models. We will exploit them for motion texture tracking. Indeed, efforts have been devoted mainly to modeling, classification and segmentation of dynamic textures, but the particular problem of dynamic texture tracking is still an open issue of great relevance.

2. Motion texture characterization by local motion measurements

We now define the motion measurement process that characterizes a motion texture. The so-called *Normal flow* is the component of the velocity vector at a point in the direction of the intensity gradient. It has been reported as effective for describing dynamic textures (Fazekas and Chetverikov, 2005) and in general, complex dynamic video contents (Fablet et al., 2002) as it gives a good compromise between quality of estimation and simplicity of calculation. It is derived from the optical-flow constraint (Horn and Schunck, 1981), and in vector form it is defined as

$$\boldsymbol{V}_i^{(n)}(t) = -\frac{\frac{\partial I_i(t)}{\partial t}}{\|\nabla I_i(t)\|}\hat{\mathbf{n}} \quad \text{with } \hat{\mathbf{n}} = \frac{\nabla I_i(t)}{\|\nabla I_i(t)\|}, \tag{1}$$

where $\nabla I_i(t)$ is the spatial intensity gradient at location i of the image and $\frac{\partial I_i(t)}{\partial t}$ is approximated by $I_i(t) - I_i(t-1)$. Here, we adopt the normal-flow-based motion measurement introduced in (Crivelli et al., 2006). First, we compute (1) for each image point, using two consecutive frames of the sequence. Then, in order to smooth out noisy measurements and enforce reliability, we apply a weighted vectorial average of $\boldsymbol{V}_i^{(n)}(t)$ over a window W of 3×3 points to obtain

$$\overline{\boldsymbol{V}}_i^{(n)}(t) = \frac{\sum_{j\in W} \boldsymbol{V}_j^{(n)}(t)\|\nabla I_j(t)\|^2}{\max\left(\sum_{j\in W}\|\nabla I_j(t)\|^2, \eta\right)}. \tag{2}$$

The magnitude of the gradient determines the relative weight of each point given that regions with large spatial intensity variation are more reliable for extracting motion information. Given that $\|\nabla I_j(t)\|$ can be very small, η is a regularization term fixed to $\eta = 10^{-4}$.

$\overline{\boldsymbol{V}}_i^{(n)}(t)$ is then a smoothed normal flow vector. Next, it is projected over the intensity gradient direction giving rise to the scalar motion observation

$$x_{i,t} = \overline{\boldsymbol{V}}_i^{(n)}(t) \cdot \hat{\mathbf{n}} \in (-\infty, +\infty). \tag{3}$$

Keeping the scalar component instead of the complete vectorial normal flow, reduces the dimension of the data to be modeled and still provides a meaningful quantity for dynamic content characterization. In Fig. 1 we see some examples of the scalar normal flow motion maps obtained between two consecutive frames of a motion texture video sequence.

2.1. Statistical properties of motion measurements

Experiments obtained from computing the proposed motion quantities for several different dynamic textures have shown that, if we observe the corresponding scalar motion histograms (Fig. 1), we note that the statistical distribution of the motion measurements has two elements: a discrete component at the null value $x_{i,t} = 0$, and a continuous distribution for the rest of the motion values.

The observation of a null value appears repeatedly in the motion maps, also following a textured pattern as well as it occurs for the rest of the motion values (Fig. 2). This is a typical structural characteristic of the motion measurements extracted from motion textures. As such, discrete and continuous values are not

independently distributed in space and time, indeed displaying mixed-state texture patterns.

3. Modeling motion textures with mixed-state Markov chains

3.1. Mixed-state statistical framework

A *mixed-state random variable* is constructed as follows: with probability $\rho \in (0,1)$, set $x = 0$, and with probability $= 1-\rho$, x follows a continuous probability density in $\mathbb{R}$, say $p^c(x)$. It results in a mixture of a discrete and a continuous random variable which is described by a mixed-state probability density function

$$p(x) = \rho \mathbf{1}_0(x) + (1-\rho)\mathbf{1}_0^*(x)p^c(x), \tag{4}$$

where $\mathbf{1}_0(x)$ is the indicator function of the discrete value *0* and $\mathbf{1}_0^*(x) = 1 - \mathbf{1}_0(x)$ its complementary function. This mixed-state p.d.f. is given with respect to the reference measure $m(dx) = \nu_0(dx) + \lambda(dx)$ where ν_0 is the discrete measure for the value *0* and λ the Lebesgue measure on $\mathbb{R}$, so that $\int p(x)m(dx) = 1$. Such a measure has also been used in (Salzenstein and Collet, 2006) for simultaneous fuzzy-hard image segmentation and Crivelli et al. (2008) for motion detection and background reconstruction.

3.2. Mixed-state Markov chains

A motion texture computed at time t of the video sequence using Eq. (3) is then a field of mixed-state random variables, i.e. $\boldsymbol{x}_t = \{x_{i,t}\}_{i\in S}$ with $S = \{1,2,\ldots,N\}$ the set of image locations. We propose to model a sequence of motion textures $\boldsymbol{X} = \{\boldsymbol{x}_t\}_{t:0\ldots T}$ as a stationary Markov chain, i.e. $p(\boldsymbol{X}) = p(\boldsymbol{x}_0)\prod_{t=1}^{T} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. The chain is described by the transition kernels $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$. Here we study a causal temporal model, for which a first assumption is to consider spatial conditional independence within a motion texture for time t

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \prod_{i\in S} p(x_{i,t}|\boldsymbol{x}_{t-1}). \tag{5}$$

The second assumption is that, given $\boldsymbol{x}_{t-1}$, $x_{i,t}$ depends only on a local neighborhood $\mathcal{N}_{i,t-1}$ of 'past' random variables at time $t-1$ (Fig. 3), namely $\boldsymbol{x}_{\mathcal{N}_{i,t-1}}$:

$$p(x_{i,t}|\boldsymbol{x}_{t-1}) = p(x_{i,t}|\boldsymbol{x}_{\mathcal{N}_{i,t-1}}). \tag{6}$$

In our case, we will assume that the temporal neighborhood is a 9-point set which includes the previous (at $t-1$) center, diagonal, anti-diagonal, horizontal and vertical neighbors for a point at time t as depicted in Fig. 3. Considering this type of causal temporal neighborhoods against spatial and non-causal interaction (e.g. in the form of Markov random fields (Crivelli et al., 2006; Bouthemy et al., 2006)) simplifies enormously not only the theoretical formulation of the model but also practical aspects as parameter estimation. Moreover, from a physical point of view, the assumption of causal interaction (instead of spatial interaction) is still valid as one can consider that the motion textures between consecutive time instants are statistically equal. See Fablet et al. (2002) as another example on the use of temporal neighborhoods.

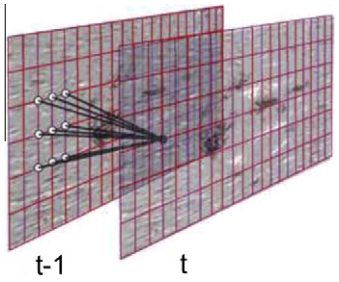


Fig. 3. Temporal neighborhood structure for the mixed-state Markov chain. At a given time instant t the motion values within $\boldsymbol{x}_t$ are considered conditionally independent given $\boldsymbol{x}_{t-1}$.

3.3. Gaussian mixed-state Markov chains (MS-MC)

Following the aforementioned assumptions, the mixed-state Markov chain motion texture model is defined by specifying the conditional densities $p(x_{i,t}|\boldsymbol{x}_{\mathcal{N}_{i,t-1}})$. They are chosen to be mixed-state conditional densities with Gaussian continuous part as suggested by Fig. 1, that is,

$$p(x_{i,t}|\boldsymbol{x}_{\mathcal{N}_{i,t-1}}) = \rho_{i,t}\mathbf{1}_0(x_{i,t}) + \rho_{i,t}^*\mathbf{1}_0^*(x_{i,t})\frac{1}{\sqrt{2\pi}\sigma_{i,t}} e^{-\frac{(x_{i,t}-m_{i,t})^2}{2\sigma_{i,t}^2}}, \tag{7}$$

where $\rho_{i,t} = P(x_{i,t} = 0|\boldsymbol{x}_{\mathcal{N}_{i,t-1}})$, $m_{i,t} \equiv m(\boldsymbol{x}_{\mathcal{N}_{i,t-1}})$ and $\sigma_{i,t}^2 \equiv \sigma^2(\boldsymbol{x}_{\mathcal{N}_{i,t-1}})$, are now functions of $\boldsymbol{x}_{\mathcal{N}_{i,t-1}}$. The Gaussian mixed-state Markov chain model is then described by these three parameters and how they depend on the neighbors. We make the following considerations:

- An interesting case for motion texture modeling is given when the mean $m_{i,t}$ is a weighted average of its past neighbors,

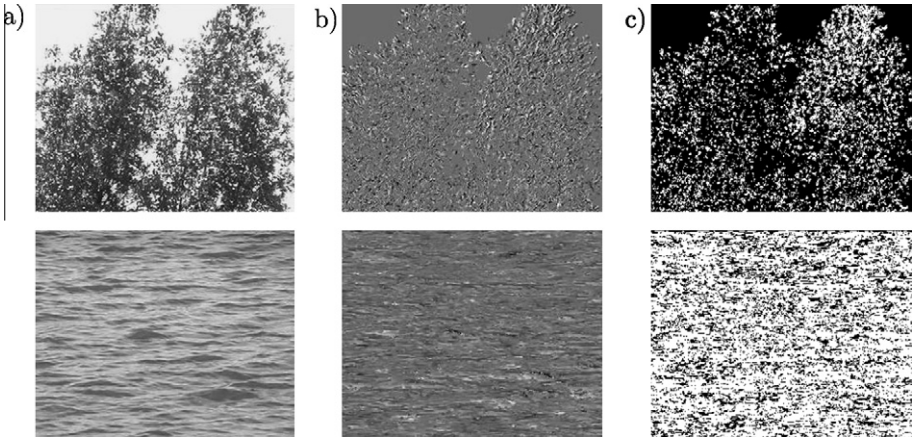


Fig. 2. (a) Sample images from motion textures. (b) The scalar motion values are spatially distributed forming a textured pattern. Here we mapped the motion measurements to the range of gray [0,255] where 128 corresponds to null motion. (c) The binary zero/non-zero map also is distributed following a textured pattern. White represents a motion value different from zero.

$$m_{i,t} = c + \sum_{j \in \mathcal{N}_{i,t-1}} h_j x_{j,t-1}, \tag{8}$$

and $\sigma^2_{i,t} = \sigma^2$ is a constant for every point. This enforces local correlation, captures important properties such as the orientation of the texture, and at the same time, keeps the model simple and with a limited number of parameters. A constant variance also assures that effectively the marginal density $p(x_i)$ has Gaussian continuous part in accordance with the motion histograms in Fig. 1.

- Regarding the conditional probability of the null value, $\rho_{i,t}$, one intuitively expects that when most of the neighbors are null (non-null), $\rho_{i,t}$ increases (decreases). This responds to a cooperative model (Besag, 1974). In like manner, larger motion values of the neighbors should make $\rho_{i,t}$ to decrease. Following the ideas of Crivelli et al. (2006) for purely (non-causal) spatial models, the conditional probability of the null value takes the form

$$\rho_{i,t} = \frac{1}{1 + \sqrt{2\pi}\sigma e^{f(\boldsymbol{x}_{\mathcal{N}_{i,t-1}})}}. \tag{9}$$

The function $f(\boldsymbol{x}_{\mathcal{N}_{i,t-1}})$ depends on the neighbors and controls the value of $\rho_{i,t}$. It is chosen in order to obtain the desired cooperative behavior as explained in the previous paragraph. It is defined as

$$f(\boldsymbol{x}_{\mathcal{N}_{i,t-1}}) = \alpha + \sum_{j \in \mathcal{N}_{i,t-1}} \beta_j \mathbf{1}^*_0(x_{j,t-1}) + \frac{m^2_{i,t}}{2\sigma^2} \tag{10}$$

where we have three terms. A constant term which determines a baseline for the conditional probability of the null value, independent of the neighbors. Then we have a term related to the discrete states which is an increasing function of the non-zero neighbors and finally a continuous term which is an increasing function of the continuous values of the neighbors (through Eq. (8)). This makes $\rho_{i,t}$ increase or decrease as expected.

- Finally from (8) and (9) we identify a set of 13 parameters (considering the five interacting directions described before) that define the Gaussian MS-MC model, which are $\varphi = \{\sigma^2, \alpha, \{\beta_j\}, c, \{h_j\}\}$. Note that we are assuming spatial homogeneity as for every point $x_{i,t}$ the same set of parameters applies.

Well known estimation techniques can be applied in order to estimate the parameters $\varphi = \{\sigma^2, \alpha, \{\beta_j\}, c, \{h_j\}\}$ that define the conditional distributions of Eq. (7). As we assume a spatially homogeneous and temporal stationary model, the parameters can be efficiently estimated from a single pair of consecutive motion textures when the number of points in S is sufficiently large, by maximizing the likelihood $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \varphi) = \prod_{i \in S} p(x_{i,t}|\boldsymbol{x}_{\mathcal{N}_{i,t}}, \varphi)$.

4. Matching of mixed-state models

In any tracking application, one needs to estimate the position of the tracked object at each instant. For doing this it is necessary to detect it and this usually involves searching the position in the image that best matches the object features.

For motion texture tracking, the object features will be the mixed-state model parameters and we propose to use the conditional Kullback–Leibler divergence (Cover and Thomas, 1991) between the mixed-state conditional distributions $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ in Eq. (5) as the matching cost. As a matter of fact, it is more formally correct to compute KL for the joint distribution $p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1})$, but this involves knowing $p(\boldsymbol{x}_{t-1})$ which is in fact, a correlated spatial field. A more complex (non-causal) model is required, which will be analyzed elsewhere. See Crivelli et al. (2006) for the case of purely spatial mixed-state Markov random fields that are plausible to be combined with the approach presented here. Moreover, this more complex formulation implies an increase in the number of motion texture parameters, which is not desirable, nor necessary as for the application at hand.

Given two sets of mixed-state model parameters, say φ_1 and φ_2, the conditional KL divergence from the density $p_1 = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \varphi_1)$ to the density $p_2 = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \varphi_2)$ is defined as (Cover and Thomas, 1991)

$$\mathrm{KL}(p_1|p_2) = E_{\varphi_1}\left[\log \frac{p_1}{p_2}\right] = \int \log \frac{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \varphi_1)}{p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \varphi_2)} p(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}|\varphi_1) dm, \tag{11}$$

which is independent of the measure m. This is not strictly a distance as it is not symmetric. Define then the symmetrized KL divergence as $d_{\mathrm{KL}}(p_1, p_2) = \frac{1}{2}[\mathrm{KL}(p_1|p_2) + \mathrm{KL}(p_2|p_1)]$. We now compute this quantity.

Eq. (7) can be more conveniently expressed in the form $p(x_{i,t}|\boldsymbol{x}_{\mathcal{N}_{i,t-1}}) = \exp Q_{i,t}(x_{i,t}, \boldsymbol{x}_{\mathcal{N}_{i,t-1}})/Z_i(\varphi)$ where $Z_i(\varphi)$ is a normalizing factor and:

$$Q_{i,t}(x_{i,t}, \boldsymbol{x}_{\mathcal{N}_{i,t-1}}) = -\frac{x^2_{i,t}}{2\sigma^2} + c\frac{x_{i,t}}{\sigma^2} + \sum_{j \in \mathcal{N}_{i,t-1}} \frac{h_j}{\sigma^2} x_{i,t} x_{j,t-1} + \alpha \mathbf{1}^*_0(x_{i,t}) + \sum_{j \in \mathcal{N}_{i,t-1}} \beta_j \mathbf{1}^*_0(x_{i,t}) \mathbf{1}^*_0(x_{j,t-1}) + \log \rho_{i,t}. \tag{12}$$

Then, $\log p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \varphi_k) = \sum_i Q^{(k)}_{i,t}(x_{i,t}, \boldsymbol{x}_{\mathcal{N}_{i,t-1}}) - \log Z_i(\varphi_k)$ for $k = 1,2$. Define $\Delta Q_{i,t}(\cdot) = Q^{(2)}_{i,t}(\cdot) - Q^{(1)}_{i,t}(\cdot)$ so that $\log \frac{p_1}{p_2} = \sum_i - \Delta Q_{i,t}(\cdot) + \log \frac{Z_i(\varphi_2)}{Z_i(\varphi_1)}$, and

$$d_{\mathrm{KL}}(p_1, p_2) = \frac{1}{2} \sum_i E_{\varphi_2}[\Delta Q_{i,t}(\cdot)] - E_{\varphi_1}[\Delta Q_{i,t}(\cdot)], \tag{13}$$

where $\log \frac{Z_i(\varphi_2)}{Z_i(\varphi_1)}$ cancels from both terms. Observing the expression of $Q^{(k)}_{i,t}(\cdot)$ in Eq. (12), we note that $d_{\mathrm{KL}}(p_1, p_2)$ can be computed by estimating the following expectations with respect to each model φ_k:

$$\begin{aligned} &E_{\varphi_k}[\mathbf{1}^*_0(x_{i,t})] = P_{\varphi_k}[x_{i,t} \neq 0], \quad E_{\varphi_k}[\mathbf{1}^*_0(x_{i,t})\mathbf{1}^*_0(x_{j,t-1})], \\ &E_{\varphi_k}[x_{i,t}], \quad E_{\varphi_k}[x^2_{i,t}], \quad E_{\varphi_k}[x_{i,t}x_{j,t-1}], \quad E_{\varphi_k}[\log \rho^{(1)}_{i,t}/\rho^{(2)}_{i,t}], \end{aligned} \tag{14}$$

where $\rho^{(k)}_{i,t}$ is as in (9), using the corresponding parameters φ_k. As we assume that we have a spatially homogeneous model, the latter expectations involved in Eq. (14) are equal for each site of the motion field, and thus, they can be efficiently estimated by simple averaging from the observed data and using the estimated parameters.

5. Application to motion texture tracking

Let us observe the example depicted in Fig. 5. We see a fire flame that moves towards the left due to the movement of the camera. The objective is to track and follow this motion texture using the proposed mixed-state model.

We first consider that the initial position of the motion texture is given or set manually by defining a starting window W_o of a given size centered at initial location o. Then the motion fields $\boldsymbol{x}_0$ and $\boldsymbol{x}_1$ are obtained (Eq. (3)) for that window (for $t = 0$) using three consecutive frames and a motion texture model is estimated. We thus obtain a reference model φ_{ref} that characterizes the dynamic content we want to track. Let q_t be the position of the window at time t. Then we estimate q_t by applying the following rule:

$$\hat{q}_t = \underset{q_t \in \Lambda_{q_{t-1}}}{\operatorname{argmin}} \; d_{\mathrm{KL}}(W_o, W_{q_t}), \tag{15}$$

where we denote $d_{\mathrm{KL}}(W_o, W_{q_t})$ the KL divergence (Eq. (13)) between the reference motion texture model estimated in W_o and that

estimated in W_{q_t}. $\Lambda_{q_{t-1}}$ is a search area around the previously estimated position. At this point any search strategy can be used with a compromise between accuracy and speed, for obtaining the $\hat{q}_t$ that minimizes d_{KL}. We have chosen the diamond search algorithm (Zhu and Ma, 2000) among other standard search algorithms for block-matching, based on comparative results that will be presented in the next section. For each possible location q_t tested by the search algorithm we obtain the motion texture in W_{q_t}, estimate the corresponding parameters and compute $d_{\mathrm{KL}}(W_o, W_{q_t})$. For that we need to extract the expectations defined in (14) from W_o and the candidate W_{q_t}. Note that except for $E_{\varphi_k}\left[\log \rho_{i,t}^{(1)}/\rho_{i,t}^{(2)}\right]$ in (14), all the remaining expectations need to be computed only once for the reference model, at the initialization step.

The computed position $\hat{q}_t$ can be noisy around the true motion texture position. Thus, we apply a simple Kalman filter (Anderson and Moore, 1979) to the measurement process in order to enforce reliability and smoothness of the estimated paths. Here, we have considered a constant velocity state model. Finally, once a filtered position is obtained, the window is moved to the new position and the process starts again.

We have chosen a simple tracking algorithm by window matching while exploiting the motion texture mixed-state model that permits to accurately characterize this class of video contents. More sophisticated tracking approaches can be applied as well, also exploiting the motion features introduced here, which for example involve more complex filtering techniques (Pérez et al., 2002). We now report comparative experimental results of motion texture tracking.

5.1. Experimental results

In the experiments that follow, we have considered three methods for window matching:

I. Pixel-wise intensity matching by minimizing the Sum of Squared Differences (SSD).
II. Intensity histogram matching by minimization of a Bhattacharyya-coefficient-based distance

$$d(\upsilon_1, \upsilon_2) = \left(1 - \sum_{1}^{N} \sqrt{\upsilon_1(n)\upsilon_2(n)}\right)^{\frac{1}{2}}, \tag{16}$$

where $\upsilon_k(n)$ $k = 1,2$ are the N-bin intensity histograms to match, as in (Pérez et al., 2002).

III. Our method: mixed-state motion texture model matching by minimizing the Kullback–Leibler divergence (13).

Method I tends to be more suitable for rigid motion where the tracked objects keep a constant geometry while method II is much more robust to pose variations and moderate deformations.

For all the three methods, the diamond search strategy (Zhu and Ma, 2000) at each time instant was applied over a maximum displacement of 15 pixels in both vertical and horizontal directions.

5.1.1. Synthetic motion texture sequences

In order to assess quantitatively the performance of our method we have generated synthetic video sequences where the true motion texture trajectories are known. Three different simulated paths were considered as depicted in Fig. 4a–c. For each path we generated video sequences where a small window of a real motion texture is displaced along the trajectory and over a real background. We analyzed three situations (Fig. 4d–f). (1) A motion texture window of water over a dynamic background of steam. (2) A motion texture of water over a moving background of water generated by applying a global random motion to a static image. In this case the tracked window and the background share the same spatial intensity statistics but different motions. The last composition corresponds to (3) a motion texture window of a fountain with time-varying illumination over a motion texture of steam. The images are of 720×576 pixels and the motion texture window of 80×80 pixels.

In Table 1 we show the results of applying methods I (SSD), II (histogram) and III (our method: MS-MC) to the 9 different combinations of paths and compositions. We have computed (i) the Root Mean Square (RMS) error of the trajectory, that is, the root of the mean square distance between the true path and the estimated path, (ii) its maximum deviation from the ground truth and (iii)

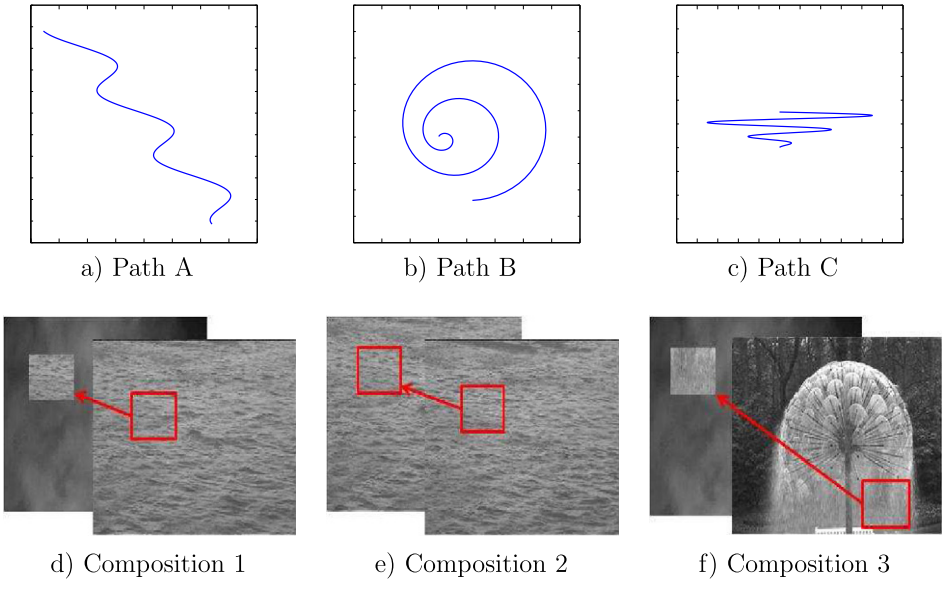


Fig. 4. (a–c) Synthetic paths generated for testing the motion texture tracking algorithm. (d) "Water on steam" composition. (e) "Water on water" composition where the background was generated by applying a global random motion to a static image. The tracked motion texture share the same spatial statistics of appearance with the background but different motion. (f) "Fountain on steam" composition. The brightness of the small window was varied along time.

Table 1
Tracking error for the synthetic motion texture sequences and for methods I (SSD), II (histogram) and our method III (MS-MC).

Path–composition	Method	RMSE (pixels)	Max error (pixels)	Target lost
A-1	SSD	392	647	$t = 9$
	Histogram	**2.25**	6	–
	MS-MC	**3.12**	**7**	–
B-1	SSD	402	482	$t = 11$
	Histogram	**3.38**	**10**	–
	MS-MC	**3.29**	**7**	–
C-1	SSD	311	479	$t = 12$
	Histogram	**19.8**	**60**	–
	MS-MC	**18.98**	**32**	–
A-2	SSD	318	563	$t = 12$
	Histogram	316	557	$t = 11$
	MS-MC	**7.17**	**11**	–
B-2	SSD	113	277	$t = 28$
	Histogram	128	267	$t = 13$
	MS-MC	**7.34**	**12**	–
C-2	SSD	181	474	$t = 23$
	Histogram	163	412	$t = 10$
	MS-MC	**21.33**	**29**	–
A-3	SSD	303	446	$t = 5$
	Histogram	126	228	$t = 37$
	MS-MC	**3.81**	**6**	–
B-3	SSD	306	465	$t = 5$
	Histogram	112	332	$t = 38$
	MS-MC	**3.65**	**7**	–
C-3	SSD	138	232	$t = 6$
	Histogram	106	333	$t = 34$
	MS-MC	**13.41**	**19**	–

we also indicate if the algorithm looses the track and if so, at which frame. We consider that the target is lost if for some t the estimated window falls completely outside the true motion texture window search area.

For Composition 1 and every path (A-1,B-1,C-1), histogram and MS-MC methods are able to track the motion texture with similar performance as shown by the RMS values without loosing the target. The intensity spatial statistics of the tracked content and the background are different enough for the histogram method to perform well. However for the SSD method, the deformable dynamic content makes the algorithm fail early in the sequence. As said before, in Composition 2 the spatial statistics of the motion texture window and the background are equal, but they have a different motion distribution. SSD and histogram fail completely loosing the target at the beginning of the sequence for all paths (A-2,B-2,C-2). Our method captures the dynamics of the motion texture and thus distinguishes the tracked content from the background at each time instant, by means of modeling the spatio-temporal distribution of motion. Finally for Composition 3, we show that our algorithm is robust to illumination changes, basically due to the proposed motion measurements (3) which are invariant to these variations. The other methods fail as soon as the brightness difference between the learned content at the initial frame and the motion texture window at the current frame becomes large.

5.1.2. Real motion texture sequences

In Fig. 5 we display the results for the real sequence 'Fire Flame' for different frames where the motion is given by a panning camera. We see that the motion texture is far from being localized in space and its extent is wider than the size of the tracked window, which was set to 50×50 pixels. The method I (SSD method with corresponding tracking results in green in Fig. 5a and b) is not able to track the flame and tends to match the ascending smoke until it breaks down. Method II (histogram method, blue track) is more coherent with the expected trajectory but it does not keep the target correctly located at every frame. Our mixed-state causal model method III (red track), performs very well, specially considering that there is another motion texture of smoke that could disturb

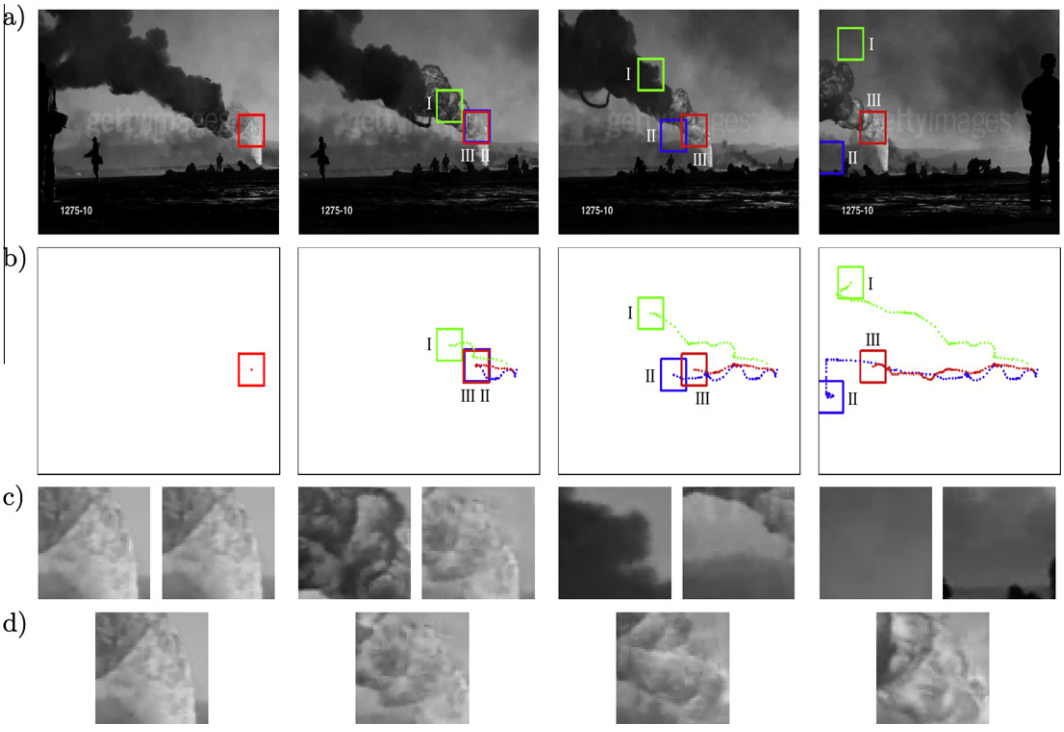


Fig. 5. (a) Results of tracking the fire flame with methods I (green), II (blue) and our method III (red) for frames 1, 23, 53 and 116. (b) Estimated tracks for each method. (c) Content of the tracked window for method I (left), method II (right). (d) Content of the tracked window for our method (III) at the four instants. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Fig. 6. (a) Results of tracking fire (antorch) with methods I (green), II (blue) and our method III (red) for frames 1, 10, 28 and 49. (b) Estimated tracks for each method. (c) Content of the tracked window for method I (left), method II (right) and (d) for our method III at the four instants. Note the large occlusion in the last frame. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

the matching process. In Fig. 5c and d we display the content of the tracked window for the different methods.

We observe the results for the 'Antorch' sequence in Fig. 6. Again it corresponds to fire, but note that it is very different than before. Moreover the flame is partially occluded in several frames. The rapid variations of the motion texture (in size and location) may produce some perturbation on the smoothness of the estimated trajectory, compared to what one would expect. Method II (blue track in Fig. 6a and b) fails completely, loosing the target just after the start of the sequence. Method I (green track) performs

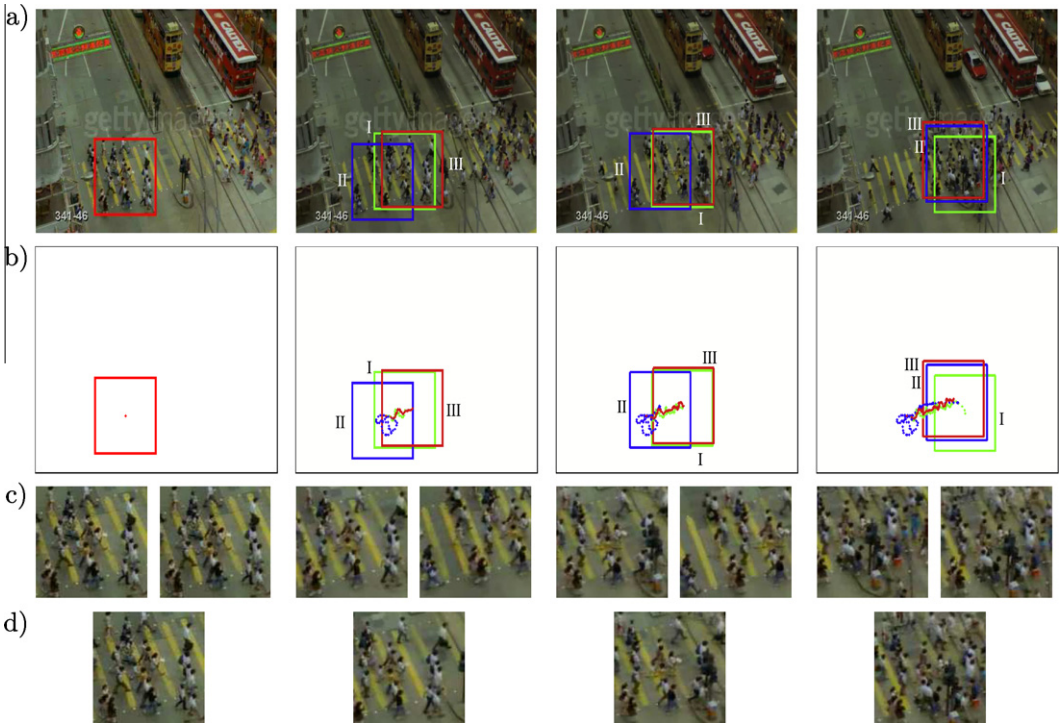


Fig. 7. (a) Results of tracking a crowd with methods I (green), II (blue) and our method III (red) for frames 1, 36, 66 and 96. (b) Estimated tracks for each method. (c) Content of the tracked window for method I (left), method II (right) and (d) for our method III at the four instants. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Parameters estimated in the tracked window for the three tested sequences at four different time instants. t_0 Corresponds to the reference model learned in the initialization step (C: center, H: horizontal, V: vertical, D: diagonal, AD: anti-diagonal).

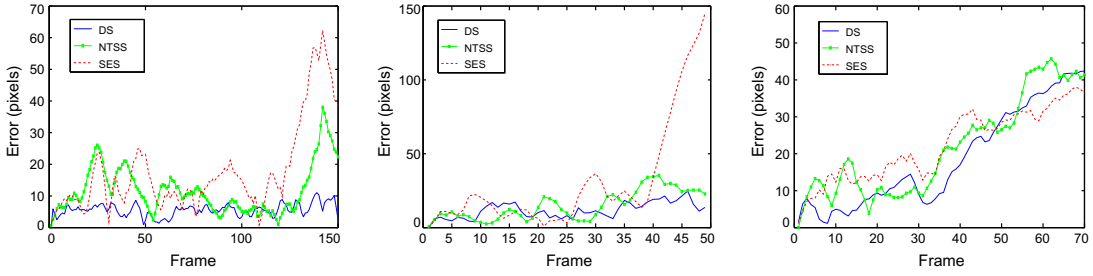
	$(2\sigma^2)^{-1}$	α	c	β_C	h_C	β_H	h_H	β_V	h_V	β_D	h_D	β_{AD}	h_{AD}
Fire flame													
$t_0 = 1$	0.21	−3.28	−0.55	0.58	−0.12	0.63	−0.01	0.29	−0.07	0.34	0.02	0.24	0.02
$t_1 = 23$	0.20	−1.65	−0.03	0.29	−0.06	0.31	0.02	0.16	−0.08	0.19	0.04	0.22	0.04
$t_2 = 53$	0.17	−1.65	−0.23	0.44	−0.03	0.36	0.08	0.19	−0.08	0.07	0.06	0.19	0.07
$t_3 = 116$	0.24	−1.41	0.11	0.48	−0.03	0.41	−0.01	0.18	−0.01	0.12	0.01	0.13	0.06
Antorch													
$t_0 = 1$	0.01	−2.12	1.36	0.92	−0.01	1.18	−0.01	0.46	0.00	0.31	−0.03	−0.19	−0.03
$t_1 = 10$	0.10	−2.23	1.61	0.71	−0.07	0.63	−0.04	0.21	0.01	0.16	−0.01	−0.12	−0.01
$t_2 = 28$	0.06	−2.16	0.47	0.49	−0.01	0.55	−0.03	0.35	0.01	0.05	−0.04	−0.14	−0.03
$t_3 = 48$	0.01	−0.22	−9.12	−0.32	0.32	−0.48	0.23	−0.01	0.08	−0.33	0.70	−0.36	0.20
Crowd													
$t_0 = 1$	0.87	−3.44	−0.11	0.08	0.06	0.41	0.06	0.22	0.01	0.61	0.08	0.61	0.03
$t_1 = 36$	1.85	−2.30	−0.03	0.30	0.07	0.46	0.15	0.35	0.09	0.52	0.14	0.42	0.08
$t_2 = 66$	1.63	−2.04	−0.02	0.29	0.03	0.50	0.21	0.42	0.07	0.45	0.14	0.34	0.10
$t_3 = 96$	1.96	−1.65	0.03	0.32	0.03	0.37	0.18	0.21	0.10	0.40	0.15	0.31	0.08

better but still does not keep the target correctly located. With our mixed-state causal model (red track), the flame is tracked satisfactorily even in the presence of great variations of shape and intensity as seen in Fig. 6d. Note also the large occlusions in the last depicted frame.

Finally, we have processed a challenging sequence (Fig. 7) of a motion texture that corresponds to a crowd of people crossing a street. Such a human motion, viewed from a long distance, can be considered as a repetitive motion pattern. Note that this motion texture is more sparse than the previous ones, showing many null motion values between persons. This is explicitly modeled within the mixed-state framework and exploited as a particular characteristic of the dynamic content. The different individuals enter and exit the texture, making it very difficult to track the group in a compact way. However, our method (III, red track in Fig. 7a and b) performs satisfactory, even when the tracked group blends with the one that goes in the opposite direction. The estimated trajectory shows some expected variations due to the complexity of the scene, but it is globally correct. Method I (green track) also performs satisfactory but it shows some deviations from the expected trajectory, specially in the last depicted frame. Method II (blue track) behaves erratically at the beginning as seen in the second and third displayed frames, probably due to the fact that it is invariant to the spatial distribution of intensity and thus the two approaching persons from behind are incorrectly included in the tracked window, despite their distance from the rest.

In Table 2 we display the parameters estimated for the tracked window at each of the four time instants depicted in Figs. 5–7. The values should not be compared in the sense of the Euclidean distance, but by means of the KL divergence. Nevertheless, one can observe that for each tested sequence the parameters show coherent values for different instants. Note the coherency in the sign and the order of magnitude. For the antorch sequence, the occlusion in Fig. 6d at t_3 is the cause of a noticeable difference in the parameters with respect to the reference model.

The performance of other search algorithms compared to diamond search (DS) (Zhu and Ma, 2000) was also tested on the three real sequences. We have considered three additional methods for block matching frequently used for implementing video coding standards which we refer as: Exhaustive Search (ES), New Three Step Search (NTSS) (Li et al., 1994) and Simple and Efficient Search (SES) (Lu and Liou, 1997). First, we assume that the ES method gives the lower trajectory estimation error as it tests every possible displacement at each frame. We thus compute the position error of each algorithm with respect to the estimate given by ES at each time instant and also the average number of search windows tested by each algorithm. Recall that we consider a search area of 31×31 pixels so ES always tests 961 possible windows. In



a) Fire flame b) Antorch c) Crowd

Sequence	ES	DS	NTSS	SES
Fire-flame	- (961)	5.54 (41.08)	11.75 (39.16)	17.71 (31.24)
Antorch	- (961)	11.73 (42.92)	14.93 (37.08)	30.63 (28.08)
Crowd	- (961)	18.63 (29.08)	21.76 (33.49)	22.16 (16.39)

d) RMS Error (average number of search windows)

Fig. 8. Position error in pixels of the search algorithms diamond search (DS), Simple and Efficient Search (SES) and New Three Step Search (NTSS) w.r.t. Exhaustive Search (ES) at each frame. Plots correspond to (a) fire flame, (b) antorch and (c) crowd. (d) Root Mean Square error of each algorithm w.r.t. ES and average number of search windows.

Fig. 8a–c we plot the position error for each method and each sequence and in Fig. 8d we give the values of the Root Mean Square (RMS) error and the number of search windows. DS gives the lowest RMS error for the three sequences. SES has the worst performance but takes less searches. NTSS takes a similar number of searches compared to DS but has a higher error. SES was discarded as the error is notably higher despite it is faster. Between NTSS and DS, DS was considered as the best choice for it is closer to ES.

6. Conclusions

We have proposed a new approach to dynamic texture modeling and tracking, based on a temporal statistical parametric model of the apparent motion extracted from video sequences. The mixed-state motion texture model has shown to be a powerful non-linear representation for describing complex dynamic content with only a few parameters.

We have developed a motion texture matching strategy by means of the computation of the Kullback–Leibler divergence between mixed-state densities. This allowed us to deal with the problem of motion texture tracking.

The results obtained so far are very encouraging, showing a good performance of the method when applied to complex scenes, not only related to natural motion textures (e.g. fire), but also to textured motion patterns as a crowd of people.

As for future work, we are investigating a more sophisticated way of tracking motion textures, that takes into account that their spatial extent can vary considerably with time. Thus it is necessary to deal with a deformable tracked window, that may change in size and shape.

Finally, the concept of a mixed-state random field opens a door to consider the introduction of several discrete states, in particular symbolic abstract labels. The theoretical results presented here on mixed-state Markov chains for numeric discrete states, can be extended to more general situations as simultaneous decision-estimation problems, without much effort as shown in (Crivelli et al., 2008).

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.patrec.2010.06.016.

References

Anderson, B., Moore, J., 1979. Optimal Filtering. Prentice-Hall Inc., Englewood Cliffs, NJ.

Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems. J. Roy. Statist. Soc. Ser. B 36, 192–236.

Bouthemy, P., Hardouin, C., Piriou, G., Yao, J., 2006. Mixed-state auto-models and motion texture modeling. J. Math. Imag. Vision 25, 387–402.

Chan, A., Vasconcelos, N., 2009. Layered dynamic textures. IEEE Trans. Pattern Anal. Machine Intell. 31, 1862–1879.

Cover, T., Thomas, J., 1991. Elements of Information Theory. John Wiley and Sons, Inc., New York.

Crivelli, T., Cernuschi-Frias, B., Bouthemy, P., Yao, J., 2006. Mixed-state Markov random fields for motion texture modeling and segmentation. In: Proc. 13th IEEE Internat. Conf. on Image Processing, ICIP'06, pp. 1857–1860.

Crivelli, T., Piriou, G., Bouthemy, P., Cernuschi, B., Yao, J., 2008. Simultaneous motion detection and background reconstruction with a mixed-state conditional Markov random field. In: Proc. 10th European Conf. on Computer Vision, ECCV'08. LNCS, vol. 5302, pp. 113–126.

Doretto, G., Chiuso, A., Wu, Y., Soatto, S., 2003. Dynamic textures. Internat. J. Comput. Vision 51 (2), 91–109.

Fablet, R., Bouthemy, P., Perez, P., 2002. Non-parametric motion characterization using causal probabilistic models for video indexing and retrieval. IEEE Trans. Image Process. 11 (4), 393–407.

Fazekas, S., Chetverikov, D., 2005. Normal versus complete flow in dynamic texture recognition: A comparative study. In: Texture 2005: 4th Internat. Workshop on Texture Analysis and Synthesis, ICCV'05, pp. 37–42.

Fazekas, S., Amiaz, T., Chetverikov, D., Kiryati, N., 2009. Dynamic texture detection based on motion analysis. Internat. J. Comput. Vision 82 (1), 48–63.

Geman, S., Geman, D., 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. IEEE Trans. Pattern Anal. Machine Intell. 6, 721–741.

Horn, B., Schunck, B., 1981. Determining optical flow. Artif. Intell. 17 (1–3), 185–203.

Li, R., Zeng, B., Liou, M., 1994. A new three-step search algorithm for block motion estimation. IEEE Trans. Circuits Systems Video Technol. 4 (4), 438–442.

Lu, J., Liou, M., 1997. A simple and efficient search algorithm for block-matching motion estimation. IEEE Trans. Circuits Systems Video Technol. 7 (2), 429–433.

Nelson, R.C., Polana, R., 1992. Qualitative recognition of motion using temporal texture. CVGIP: Image Understan. 56 (1), 78–89.

Pérez, P., Hue, C., Vermaak, J., Gangnet, M., 2002. Color-based probabilistic tracking. In: ECCV '02: Proc. 7th European Conf. on Computer Vision – Part I. Springer-Verlag, London, UK, pp. 661–675.

Salzenstein, F., Collet, C., 2006. Fuzzy Markov random fields versus chains for multispectral image segmentation. IEEE Trans. Pattern Anal. Machine Intell. 28 (11), 1753–1767.

Vidal, R., Ravichandran, A., 2005. Optical flow estimation and segmentation of multiple moving dynamic textures. In: IEEE Conf. on Computer Vision and Pattern Recognition, CVPR 2005, vol. 2, pp. 516–521.

Zhu, S., Ma, K.-K., 2000. A new diamond search algorithm for fast block-matching motion estimation. IEEE Trans. Image Process. 9 (2), 287–290.