Fast Dynamic Texture Detection

V. Javier Traver[1,2], Majid Mirmehdi[3], Xianghua Xie[4], and Raúl Montoliu[2,5]

[1]DLSI, [2]iNIT, [5]DICC, Univ. Jaume I, Castellón, Spain
{vtraver,montoliu}@uji.es
[3]Dept. Comp. Science, Univ. of Bristol, Bristol, UK
majid@cs.bris.ac.uk
[4]Dept. Comp. Science, Univ. of Wales Swansea, Swansea, UK
x.xie@swansea.ac.uk

Abstract. Dynamic textures can be considered to be spatio-temporally varying visual patterns in image sequences with certain temporal regularity. We propose a novel and efficient approach to explore the violation of the brightness constancy assumption, as an indication of presence of dynamic texture, using simple optical flow techniques. We assume that dynamic texture regions are those that have poor spatio-temporal optical flow coherence. Further, we propose a second approach that uses robust global parametric motion estimators that effectively and efficiently detect motion outliers, and which we exploit as powerful cues to localize dynamic textures. Experimental and comparative studies on a range of synthetic and real-world dynamic texture sequences show the feasibility of the proposed approaches, with results which are competitive to or better than recent state-of-art approaches and significantly faster.

Keywords: Dynamic texture; optical flow; brightness constancy assumption; global parametric motion.

1 Introduction

Dynamic textures (DTs) can be considered to be spatio-temporally varying visual patterns in image sequences, such as fire, waterfalls, crops in wind, shaking leaves, and many other instances of moving, structured or unstructured patterns. There has been a multitude of research works in DT segmentation, detection, recognition and synthesis, for example [1,2,3]. DT analysis has been found useful in a number of real-world applications, such as particle measurements [4], smoke detection in surveillance scenarios [5], and facial expression recognition [6].

DTs not only exhibit complex appearance but also commonly lack distinctive local features, for example due to transparency and challenging spatio-temporal variability. Linear models, particularly auto-regressive (AR) techniques, have been widely used in modeling DTs [7,8,9]. Most of the proposed AR models are first-order, which may prevent oscillations and higher-order temporal dependencies be correctly captured, but higher-order AR models such as [8] can improve the accuracy of synthesized sequences by capturing complex patterns over multiple frames. A DT constancy constraint was introduced in [10] as an analogy to the brightness constancy assumption for DTs. Very recently, layered models [11] have been proposed to deal with multiple different DTs

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 680–693, 2010.

in a single sequence. However, in general, AR models have been found mainly successful on synthetic DT sequences, in which the spatial extent of the DTs remains largely constant. For some DTs, such as smoke and fire, it is difficult for AR techniques to generalize. Furthermore, these methods rely on costly learning procedures and complicated mathematical and conceptual schemes [12].

Fractals [13], local binary patterns (LBP) [6], and spatio-temporal multiresolution histograms [14] have also been applied to DT analysis. These are illustrative examples of attempts to properly model the spatio-temporal nature of DTs. Campisi et al. [13] extend to the temporal dimension the self-similarity model which is known to be present in natural images. Zhao and Pietikäinen [6] use LBP in three planes (XY, YT, XT), allowing them to consider the spatio-temporal domain. Lu et al. [14] use histograms of velocity and acceleration at four levels of an image pyramid, applying spatio-temporal Gaussian filters to perform spatial and temporal multiresolution analysis.

Recently, Markov random fields have been used to model DT spatio-temporal dynamics, e.g. [15,16]. For example, in [16], the authors adopted two three-state hidden Markov models to model DT and non-DT moving objects. However, their focus was on one particular type of object, i.e. swaying leaves. Several other authors have also proposed specific models for specific DTs, e.g. steam [17], smoke [5,18], fire [19], or general fluids [20]. For example, steam is considered to blur image details and a supervised method using wavelets and other local features are applied in [17] for DT classification. Similarly, smoke smoothes image edges, a scenario which can be detected by monitoring the abrupt change of some local energy measure [5] just after the appearance of DT. Flickering and turbulence phenomena of smoke are exploited in [18]. Although some of these methods such as [21,18] perform in real-time, they are clearly not applicable to other kinds of DTs. An added restriction is that these techniques often assume the background scene to be known in advance.

Amongst many other motion cues [22,23], optical flow has been particularly useful in DT analysis. For example, in [24,1], optical flow is shown to be effective in discriminating different types of DTs. Regarding the optical flow, it is common to rely on the brightness constancy assumption (BCA) which states that a change of brightness at an image location has motion as its only cause. However, this has been found insufficient in dealing with DTs in real-world sequences [25], and alternative flow models to BCA, such as gradient constancy, color constancy and brightness conservation, were recently explored in [25], where an approach based on level-sets was formulated to segment image regions obeying either the BCA or some of these alternative flow models.

In contrast, a different, simpler route is explored in this paper. Since DT pixels do not follow the BCA, the dynamic texture can be located by detecting those pixels at which the estimated optical flow is not correct. In order to detect these optical flow "failures", two alternative approaches, as instances of a proposed general scheme for DT detection, are investigated (Sect. 2). One method is based on the fact that optical flow in DTs will exhibit changes in a local spatio-temporal neighborhood (Sect. 2.1). The other approach is based on using the motion outliers as detected by a robust global parametric 2D motion estimator (Sect. 2.2). In comparison to the state-of-the-art (Sect. 3), we obtain similarly accurate, if not better, results with methods which are both conceptually simpler and substantially faster (Sect. 4).

Algorithm 1. DT detection for pixel (i, j) at time t

Input: $\mathbf{v}$: past value,
$\mathbf{x}$: new value,
λ: weight of past and new information, and
$\mathcal{DT}$: DT probability at previous time $t-1$
Output: updated $\mathbf{v}$ and
$\mathcal{DT}(i,j)$: updated DT probability at current time t
1: $\varepsilon \Leftarrow$ evidenceOfDT($\mathbf{v}$, $\mathbf{x}$)
2: $\mathbf{v} \Leftarrow$ updateVisualCue($\mathbf{v}$, $\mathbf{x}$, ε)
3: $\mathcal{DT}(i,j) \Leftarrow$ temporalSmooth($\mathcal{DT}(i,j)$, ε, λ)
4: $\mathcal{DT}(i,j) \Leftarrow$ spatialSmooth($\mathcal{DT}$, i, j)

2 Proposed Methods

A general framework to detect dynamic textures in video sequences is proposed. It has as its core procedure the approach shown in Algorithm 1, which encapsulates a simple idea and admits a number of reasonable variants. For each pixel, information is kept on past visual data $\mathbf{v}$, and new visual cues $\mathbf{x}$ is computed at a given time step t. With $\mathbf{v}$ and $\mathbf{x}$, spatio-temporal evidence ε of DT is gained, and used to update the DT likelihood map $\mathcal{DT}$ at the corresponding pixel (i, j). The evidence ε is also used to update $\mathbf{v}$ with the current value $\mathbf{x}$. Then, in order to disregard short-time noisy detections and get more stable DT regions over time, the $\mathcal{DT}$ is temporally filtered. In particular, we use:

$$\mathcal{DT}(i,j) \Leftarrow \lambda \cdot \mathcal{DT}(i,j) + (1-\lambda) \cdot \varepsilon, \tag{1}$$

where the value for $\lambda \in (0, 1)$ can be chosen as a tradeoff between stability and reactivity of the detection. As a last step, the dynamic texture map $\mathcal{DT}$ is spatially smoothed with a $k \times k$ Gaussian kernel (where $k = 25$ was determined empirically). This spatial filtering is intended to remove small regions and provide smoother and more compact DT regions. The map is finally thresholded at 0.5 (which is also established empirically) to get a binary DT mask.

From this general description, a number of specific methods can be instantiated by defining the various elements of the algorithm: the visual information used for $\mathbf{v}$ and $\mathbf{x}$, how the evidence for DT is predicted, and how the updating of $\mathbf{v}$ is done (if at all). Two possible approaches, with their associated merits and shortcomings, are presented in this paper.

2.1 Optical-Flow-Based DT Detection (OFDT)

A characteristic of DT is that its visual appearance changes over time, and hence, detecting temporal changes is a reasonable approach to detect DT. In our first approach, based on optic flow, and referred to as OFDT, the evidenceOfDT($\mathbf{v}$, $\mathbf{x}$) function can be defined as

$$H(\theta_s - \mathcal{S}(\mathbf{v}, \mathbf{x})), \tag{2}$$

where $\mathcal{S}(\cdot, \cdot) \in [0, 1]$ is a similarity measure, θ_s is a similarity threshold, and $H(x)$ is the Heaviside (step) function (i.e. $H(x) = 1$ for $x > 0$, and $H(x) = 0$ for $x < 0$).

Smooth approximations of the step function can also be defined. Therefore, the evidence for DT is high when similarity between past and current visual data is low, which means a change is detected.

One visual cue which has often been used to characterize and recognize DTs is the optical flow, which can also be used to detect DT. Here, the values used for $\mathbf{v}$ and $\mathbf{x}$ are the two components of the flow vector, i.e. $\mathbf{v} = (v_1, v_2) = (v_x, v_y)$, and $\mathbf{x} = (x_1, x_2) = (v'_x, v'_y)$. One of the many possible similarity measures is

$$\mathcal{S}_{\mathsf{OFDT}}(\mathbf{v}, \mathbf{x}) = \frac{1}{2} \sum_{i=1}^{2} \exp\left(-\gamma_i \cdot \delta_i^2\right), \tag{3}$$

where $\delta_i = v_i - x_i$ is the difference in each component, and γ_i weights the squared difference δ_i^2 proportionally to a measure of the local variance of the corresponding flow component. The greater the local variance, the more importance is given to the difference. One way to set γ_i is given below. Other measures besides (3) were explored, in all cases seeking their normalization in $[0, 1]$ to set θ_s more easily.

The function used to update $\mathbf{v}$, updateVisualCue($\mathbf{v}$, $\mathbf{x}$, ε), is simply its assignment to the current value $\mathbf{x}$ when $\varepsilon = 1$. Other sensible definitions are possible, such as a weighted sum between past, $\mathbf{v}$, and new data, $\mathbf{x}$.

The key observations behind the OFDT method are: (1) the BCA does not hold on DT; and (2) the flow computed assuming BCA exhibits a weak temporal and spatial coherence in DT locations. Therefore, DT can be detected by detecting optical flow "failures". The weak temporal coherence is captured by the temporal change detection, while the lack of spatial coherence is captured by the local measure of flow variance. Both procedures, temporal change detection and spatial variance, are coupled through the similarity measure $\mathcal{S}$. Let (μ_i, σ_i) denote the mean and standard deviation of the local optical flow component $i \in \{1, 2\}$. Then, γ_i is set as the relative standard deviation, $\gamma_i = \frac{\sigma_i}{|\mu_i|}$. The use of the *relative* standard deviation here reflects the idea that the importance of the variance of the optical flow depends on its magnitude. For example, this allows us to capture subtle DT in regions of small flow that would be undetected otherwise. The values (μ_i, σ_i) are computed on 5×5 windows and, to make these local computations faster, the concept of integral images [26] is used.

The charm of the OFDT approach is that no alternative motion models are needed at all, and no complex procedures, such as those based on level sets, are really required. This clearly contrasts with the conceptual and computational complexity of previous recent approaches, e.g. [25], summarized in Sect. 3. Our approach shows that just conventional optical flow methods relying on the BCA can be used, e.g. the Lucas-Kanade and Horn-Schunck methods.

Other approaches for change detection not based on optical flow can be those that exploit appearance cues. For instance, RGB values can be considered for $\mathbf{v}$ and $\mathbf{x}$, and a color-based similarity measure can be defined for $\mathcal{S}$. Despite the simplicity of such an approach, tests with this appearance-based DT detection yields very good results, provided that the camera does not move; otherwise, changes in appearance may easily happen as a consequence of camera motion and non-uniform scene. The proposed OFDT method is more flexible than this and can deal with egomotion conditions, as demonstrated later in Sect. 4.

2.2 Motion Outliers-Based DT Detection (MODT)

In this section, we present a second approach to DT detection, referred to as MODT and also based on Algorithm 1, which exploits the global motion outliers in the image sequence. Global image motion can be estimated with parametric 2D techniques which, unlike optical flow methods, can deal better with larger image deformations and can be very robust to the presence of a large amount of motion outliers, i.e. image locations not following the motion of the main part of the image. Estimating the motion parameters $\boldsymbol{\mu}$ of a given motion model $\mathbf{f}(\mathbf{p};\boldsymbol{\mu})$ is stated in [27] as minimizing error measure

$$E(\boldsymbol{\mu}) = \sum_{\mathbf{p}} \rho(\mathrm{DFD}_{\boldsymbol{\mu}}(\mathbf{p})), \tag{4}$$

where

$$\mathrm{DFD}_{\boldsymbol{\mu}}(\mathbf{p}) = I(\mathbf{f}(\mathbf{p};\boldsymbol{\mu}), t+1) - I(\mathbf{p}, t) \tag{5}$$

is the displaced frame difference considering the geometric transformation corresponding to the motion model $\mathbf{f}(\mathbf{p};\boldsymbol{\mu})$ which maps location $\mathbf{p}$ to another position for a given motion parameter vector $\boldsymbol{\mu}$, ρ is an M-estimator (such as the Tukey's biweight), and $I(\mathbf{p}, t)$ is the gray-level value of image I at location $\mathbf{p} = (i, j)$ and time t.

To solve the M-estimator problem, iterative reweigthed least squared (IRLS) is used so that the problem is converted into a weighted least-squares problem, $E(\boldsymbol{\mu}) = \sum_{\mathbf{p}} w(\mathbf{p}) \cdot \mathrm{DFD}^2_{\boldsymbol{\mu}}(\mathbf{p})$. The weights $w(\mathbf{p})$ are defined as $w(\mathbf{p}) = \frac{\psi(r(\mathbf{p}))}{r(\mathbf{p})}$, with ψ being the influence function (the derivative of the ρ function), and $r(\mathbf{p}) = \mathrm{DFD}_{\boldsymbol{\mu}}(\mathbf{p})$ is the residual. The minimization of this error is performed using an incremental and multiresolution scheme that deals with larger motions and prevents falling into local minima. At each level of the multiresolution pyramid, the IRLS process is applied.

In our case, we are not interested in the recovery of the motion parameters (which such methods can estimate very accurately) but rather, in their ability to detect motion outliers. We considered the affine motion model, and used four levels in the pyramid. The key idea is that DT pixels can simply be identified with motion outliers. The weights $w(\mathbf{p}) = w(i, j) \in [0, 1]$ represent how well a pixel (i, j) supports the parametric motion model or not. Therefore, the visual cue used for $\mathbf{x}$ in Algorithm 1 is just $w(i, j)$ and the evidence function for DT is simply $1 - \mathbf{x}$. Note, in this instantiation of the proposed framework, the past visual information $\mathbf{v}$ is not used, since only the frame-to-frame motion outliers are considered, so no updating of $\mathbf{v}$ is required either.

2.3 Comparison of OFDT and MODT

OFDT and MODT identify DT locations by detecting "motion failures" either as spatio-temporally irregular optic flow or as outliers of global motion. In comparison to OFDT, MODT is even simpler, since only the weights $w(i, j)$, directly provided as a by-product of the general-purpose global parametric motion estimator, are used. MODT is as fast or faster than OFDT (depending on the particular way the optical flow is computed). Additionally, MODT is generally very effective given that the parametric 2D motion is estimated on a frame-to-frame basis and no analysis is done on the temporal change of outliers, as it is done on the temporal change of the optical flow in OFDT. For instance,

jerky camera motions are dealt with more robustly by MODT, since the frequent flow changes induced by egomotion can be misdetected by OFDT as DT. It is possible to explore, however, how the influence on past information, which is considered in OFDT within $\mathcal{S}$, could be removed by redefining $\mathcal{S}$ so that it considers only the spatial inhomogeneity of the optical flow, not its temporal change. This is important, since the temporal change might be due to a camera moving at non-constant speed rather than to genuine DT's dynamics. Future work will look into this possibility.

Additionally, it can be noticed that, in MODT, no similarity measure $\mathcal{S}$ has to be defined and no critical parameter has to be set (results are quite insensitive to the single parameter, λ). Since OFDT is a local method, it offers more flexibility, but in its current form its performance still depends on the particular choice of the similarity measure, the estimation of the flow vectors (which can be wrongly noisy in non-DT regions), and on having to properly set the parameters in the specific optic flow method used. This latter difficulty has also been experienced by other authors [4].

3 The Approach by Fazekas et al. [25]

The methods in Sect. 2 propose two approaches to exploit the spatio-temporal irregularity of optical flow in DT regions. In contrast, since DT does not follow the BCA, three alternative flow assumptions are explored in [25]: gradient constancy (GC), color constancy (CC) and brightness conservation (BC). These represent different attempts to account for illumination changes to capture the dynamics of DTs. Then, the DT detection problem is posed as that of minimizing the functional $F(u, v, \tilde{u}, \tilde{v}, C) = G(L_1; \Omega_1) + G(L_2; \Omega_2) + G(S; \Omega_1 \cup \Omega_2) + \nu|C|$, where $G(H; \Omega) = \int_\Omega H(u, v, \tilde{u}, \tilde{v})\, \mathrm{d}x\, \mathrm{d}y$; the first term integrates a Lagrangian $L_1(u, v)$ for the flow (u, v) following the BCA over a non-DT region Ω_1; the second term integrates a Lagrangian $L_2(\tilde{u}, \tilde{v})$ for the flow $(\tilde{u}, \tilde{v})$ obeying an alternative assumption (GC, CC, BC) over DT region Ω_2; the third term seeks the smoothness $S(u, v, \tilde{u}, \tilde{v})$ of both flows over the whole image; and the last term aims at penalizing long contours C separating regions Ω_1 and Ω_2, with ν being a scaling parameter. The unknown discontinuity set makes the direct minimization of the functional F hard. Thus, the authors reformulate the problem as a level-set functional

$$F_{LS}(u, v, \tilde{u}, \tilde{v}, \phi) = \int \left[T_1(u, v; \phi) + T_2(\tilde{u}, \tilde{v}; \phi) + T_3(u, v, \tilde{u}, \tilde{v}) + T_4(\phi)\right] \mathrm{d}x\, \mathrm{d}y, \tag{6}$$

where $T_1 = E_1(u, v) \cdot H(\phi)$, $T_2 = (\gamma E_2(\tilde{u}, \tilde{v}) + \rho)H(-\phi)$, $T_3 = S(u, v, \tilde{u}, \tilde{v})$, and $T_4 = \nu|\nabla H(\phi)|$. E_1 and E_2 are energy functions for the BCA and some of the alternative assumptions (GC, CC, BC) respectively, ϕ is an indicator function the sign of which distinguishes Ω_1 from Ω_2, $H(\cdot)$ is the Heaviside function, and S measures the flow smoothness. Parameter γ weights one kind of flow against the other; ρ is required to prevent Ω_2 to become the whole image, which may happen since the alternative flow assumptions are more general than the BCA; ν is used to adjust the smoothness of the contour C; parameters α, $\tilde{\alpha}$, $\tilde{\beta}$ are used in the smoothness terms for BC and CC; parameter λ is used in the definition of the energy terms E_2; and the Heaviside function uses a parameter ϕ_0. About a dozen parameters have to be set.

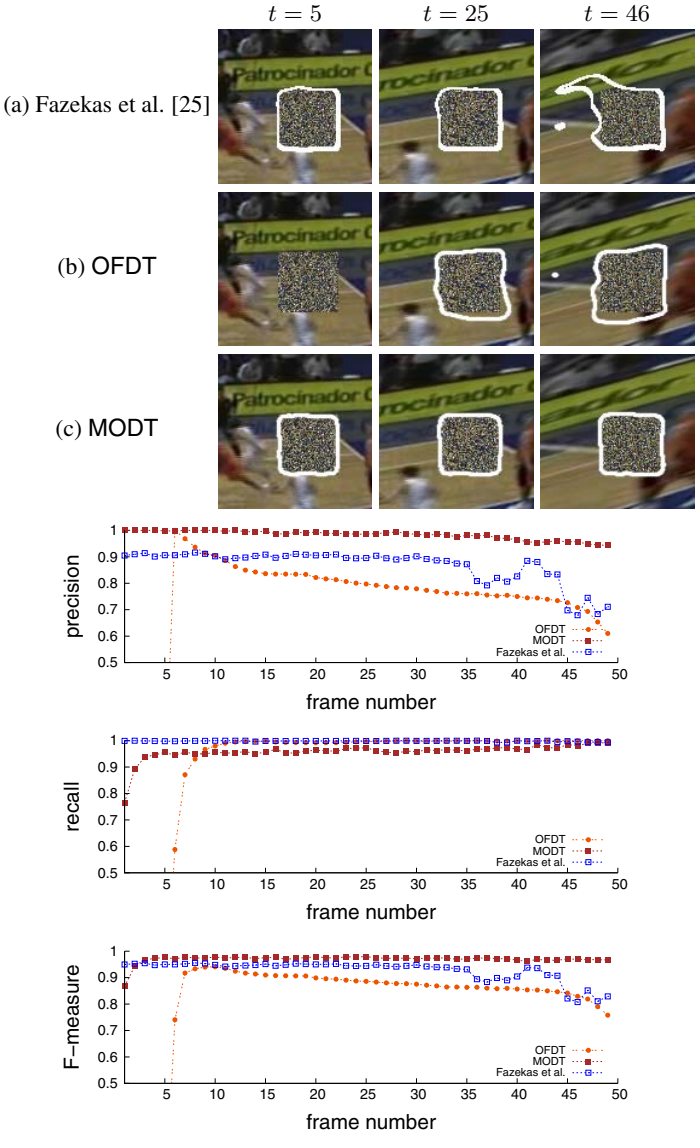
In addition to this sophisticated level-set approach, Fazekas et al. [25] considered a simpler non-level-set based approximation to DT, consisting of using the residual of the optical flow (v_x, v_y), defined as $r(v_x, v_y) = (I(x + v_x, y + v_y, t + 1) - I(x, y, t))^2$, and then setting a threshold for it. This method is reported to be very sensitive to the choice of the threshold for the residual, and fails under several conditions such as significant egomotion. They suggest yet another idea consisting of comparing this residual $r = r(v_x, v_y)$ with the no-flow residual $r_0 = r(0, 0)$, so that DT is flagged when $r_0 - r < \theta_r$, the threshold θ_r being obtained from a simple parametric classifier. It is assumed that DT and non-DT regions are linearly separable in this residual difference space. However, it is possible that one of the regions dominates the other, causing the parametric thresholding to be no longer effective. This idea has been recently revisited [28] by including a spatiotemporal median filter of optic flow residual maps. An adaptive way to set the threshold for these maps is also suggested, which relies on the DT occupying a "significant" part of the data in the first n frames. However, no analysis is provided on how effective this adaptive threshold is.

4 Results

Short synthetic videos have been generated so that ground-truth dynamic texture maps are available, which allows a quantitative comparison. Additionally, qualitative comparisons are performed with video sequences of real DTs. Our DT detection results are depicted with the contours of the connected components in the DT map $\mathcal{DT}$. To distinguish exterior from interior contours, they are depicted in white and black, respectively. Thus, a black contour within a white one means a region that is *not* DT.

The outlier identification for the MODT method uses a robust parametric 2D motion estimation algorithm [27]. Our proposed methods are compared to the level-set-based system by Fazekas et al. [25] using an implementation provided by its authors. No comparison is performed with their fast approximations since they are less accurate than their level-set procedure, and we are interested in using their most accurate results. We have focussed our comparative analysis to learning-free, motion-based methods and in particular to studying whether alternative flow models to the BCA are actually required. Therefore, we make no comparison with non-motion based or learning-based methods, such as AR or Markov-based models, which are outside the scope of our work here.

For MODT, the parameter $\lambda = 0.5$ is set for all tests. For OFDT, the Lucas-Kanade method for optical flow computation was applied, on 5×5-sized windows, after Gaussian averaging and downsampling. The similarity threshold was $\theta_s = 0.95$, and the temporal parameter was $\lambda = 0.9$ for the synthetic sequences and $\lambda = 0.7$ for the real ones. For Fazekas et al. [25], the parameters were in the values set in their software and their paper [25]. For the synthetic sequences, only the size of the images was changed accordingly, and the number of levels in the Gaussian image pyramid was set to 3 instead of 4, as required for the smaller image sizes used (128×128). For the real sequences, the results were not recomputed, but directly taken from the web page [29] associated with [25]. In this case, interior and exterior contours are both in red.



(d) Precision, recall and F-measure

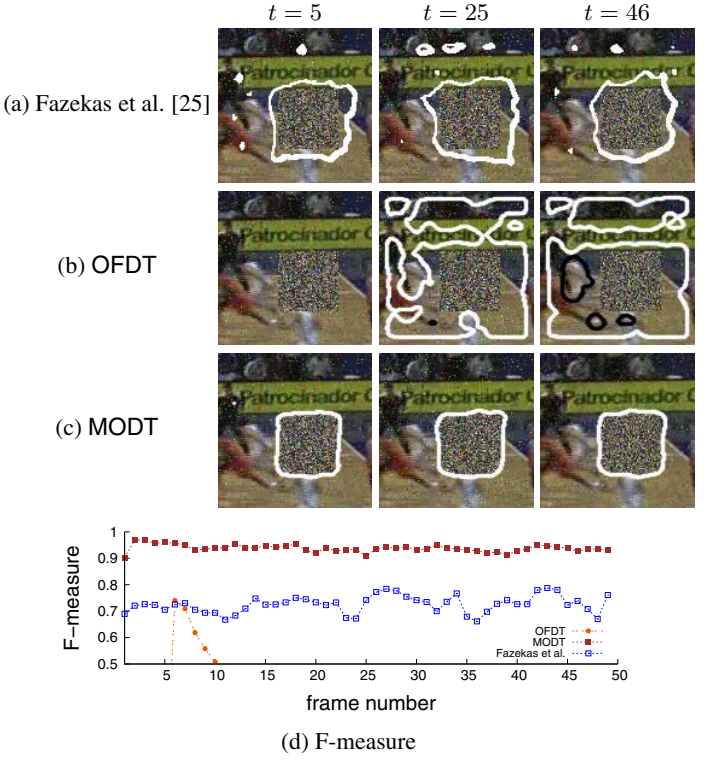
Fig. 1. Exp. 1: background moving affinely, random static DT

4.1 Synthetic Sequences

Quantitative assessment based on ground-truth DT masks, uses the number of true positives p (DT correctly classified as such), false positives $\bar{p}$ (non-DT misclassified as DT) and false negatives $\bar{n}$ (DT misclassified as non-DT). From these, precision π and recall ρ are computed, respectively, as $\pi = \frac{p}{p+\bar{p}}$ and $\rho = \frac{p}{p+\bar{n}}$. The F-measure, $F = \frac{2 \cdot \pi \cdot \rho}{\pi + \rho}$,

combines π and ρ to summarize the performance as a single value. All these measures range in $[0, 1]$, and the higher, the better.

Influence of background motion (Experiment 1). A background image is moved affinely, while a rectangular-shaped dynamic texture is built fully randomly (the value for each pixel in each frame is independently drawn from a uniform distribution). Results at three frames of the 50-frame sequence are shown in Fig. 1. It can be seen that while all approaches behave well most of the time, at the end of the sequence, the Fazekas et al. [25] approach and OFDT have many false positives. OFDT does not detect the DT at the beginning since, in (1), $\lambda = 0.9$ weights past information much more than the new, and the DT likelihood map is initialized to 0. The evolution of π, ρ and F over time is given in Fig. 1(d). Generally, MODT is more precise than Fazekas et al. [25] and OFDT. Both of these latter methods, and particularly [25], have higher recall, at the expense of more false detections. The overall behavior, as captured by F, is best for MODT. This example illustrates how the global approach of MODT is more robust against egomotion conditions than the optical flow alternatives OFDT and Fazekas et al. [25]. As a quantitative guide, average and standard deviations of π, ρ and F over the sequence are collected in Table 1 for this and the following experiments.



(d) F-measure

Fig. 2. Exp. 2: background with 5% random noise, random static DT

Table 1. Average (μ) and standard deviation (σ) for precision (π), recall (ρ) and F-measure (F) for all the synthetic experiments

		π		ρ		F	
Exp.	Method	μ_π	σ_π	μ_ρ	σ_ρ	μ_F	σ_F
1	[25]	0.87	0.06	1.00	0.00	0.93	0.04
	OFDT	0.70	0.27	0.86	0.33	0.77	0.29
	MODT	0.96	0.14	0.94	0.14	0.97	0.02
2	[25]	0.57	0.04	1.00	0.00	0.73	0.03
	OFDT	0.24	0.14	0.87	0.33	0.37	0.16
	MODT	0.87	0.13	0.97	0.14	0.94	0.01
3	[25]	0.70	0.04	1.00	0.00	0.82	0.02
	OFDT	0.60	0.24	0.86	0.33	0.70	0.26
	MODT	0.73	0.11	0.95	0.14	0.84	0.01

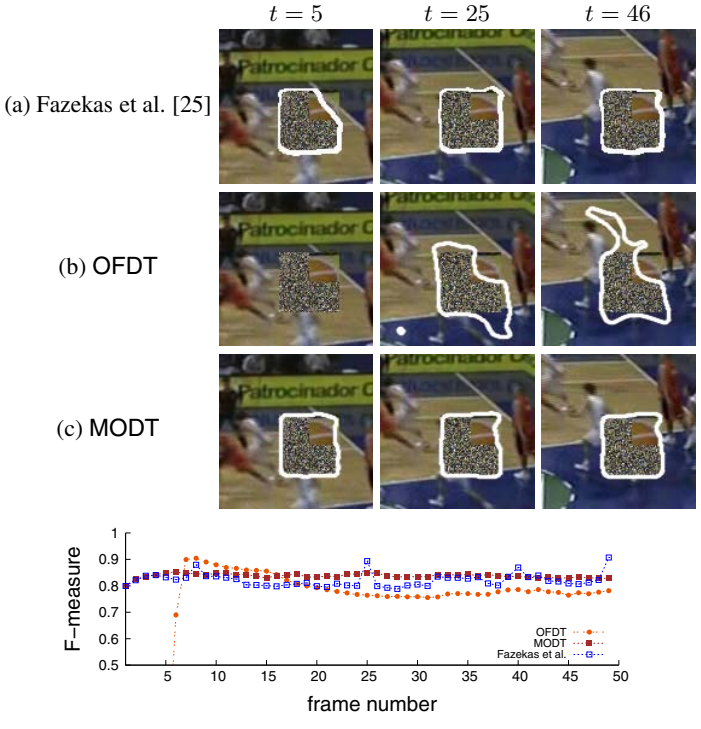
Influence of image noise (Experiment 2). Random noise (5%) is added to each frame of a sequence with a static background, and a DT region is generated randomly, as before. As shown in Fig. 2, the local methods (OFDT and Fazekas et al. [25]) exhibit many false positives, while the misclassifications are much fewer in MODT. This better performance by MODT is quantitatively reflected in the F-measure (Fig. 2(d) and Table 1). OFDT results are poorer than Fazekas et al. [25], possibly because Fazekas et al. [25] compute the flow with the Horn-Schunk (HS) method which, unlike Lucas-Kanade (LK), enforces global smoothness of the flow. However, in agreement with [4], we found it harder to select the right parameters for HS than for LK.

Influence of independent motion (Experiment 3). For this experiment, the background translates at a constant speed. The values of three-quarters of a rectangle (in an L-shaped form) is set randomly and the remaining top-right quarter has constant contents. This constant region is equivalent to an independently moving object. As illustrated in Fig. 3, MODT fails to detect the true extent of the DT region, as it misdetects the constant region as DT. The reason is that, given the global parametric motion of the background, both the true DT and the constant region are indistinguishably treated as motion outliers. While Fazekas et al. [25] works slightly better in this case, its behavior is unstable and surprisingly poor since, given its local nature and its elaborate design, it should deal with this situation more successfully. In fact, Fig. 3(b) shows the results of the (also local) OFDT approach, which exhibits quite promising results. The F-measure for the three methods is compared in Fig. 3(d) and Table 1.

Even though not shown here, tests of OFDT with a variational optical flow computation method were conducted and, when applied to these synthetic sequences, yielded extremely good results. While this indicates that OFDT in its current form is still sensitive to how the optic flow is estimated, it also provides strong evidence of the validity and interest of the approach.

4.2 Real Sequences

In order to test the approaches with sequences of real dynamic textures (fire, water, steam, smoke, etc.), the DynTex database [30] has been used. Results for three arbitrarily selected frames of some of these sequences are shown in Figs. 4-6. Fig. 4 illustrates



(d) F-measure

Fig. 3. Exp. 3: translating background, L-shaped random DT and static constant region

the results for a waterfall sequence while the camera is moving horizontally. It can be seen that the proposed approaches, despite their simpler design, work as well as, if not better than, Fazekas et al. [25], at higher frame rate (see Table 2).

The results for a smoke sequence are shown in Fig. 5. While all approaches behave reasonably well, they all have problems with detecting dense (smoke) regions, an issue related with the aperture problem. However, similarly to the synthetic examples, Fazekas et al. [25] tends to miss more true DT and misdetects some non-DT in comparison to OFDT and MODT. At $t = 100$, OFDT misdetects some regions in the wall as DT (lower-right part of the image), possibly because of illumination changes, a situation which is dealt with more robustly by MODT and Fazekas et al. [25].

Finally, Fig. 6 illustrates an example where Fazekas et al. [25] offers results better than MODT but worse than OFDT. MODT does not fare as well due to the large degree of motion outliers or their uneven distribution. On the other hand, the duck is correctly left undetected as DT most of the time by all methods, but all of them have problems when it moves faster at the end of the sequence. The likely reason for this is that, as in the third synthetic example, MODT, as a global parametric motion estimation method, is unable to distinguish independently moving objects from true DT. Indeed, OFDT as a local method, in this sequence (Fig. 6(c)) and others, can deal with these kinds of scenarios as well or better than Fazekas et al. [25].

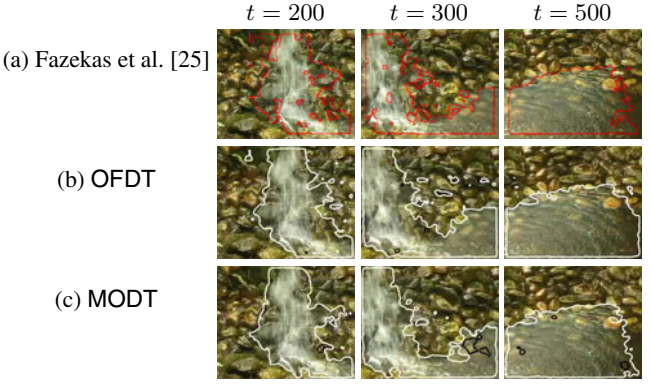


Fig. 4. Results with real sequence 6481i10.avi with the three methods

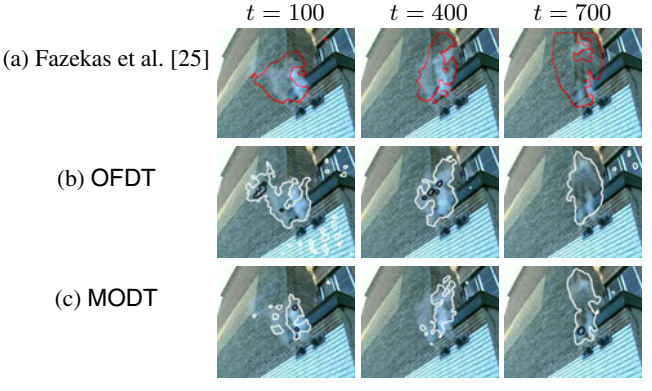


Fig. 5. Results with real sequence 648ea10.avi with the three methods

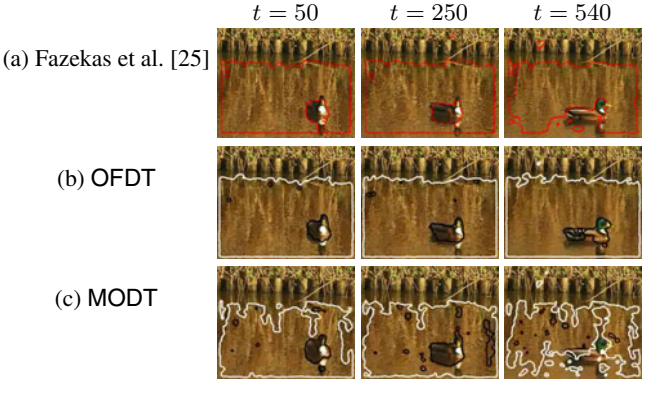


Fig. 6. Results with real sequence 644ce10.avi with the three methods

Table 2. Computation times for the three methods

Figure	DynTex sequence	# frames	Fazekas et al. [25]	OFDT	MODT
4	6481i10.avi	670	$>$ 5 hours	9.1 min.	4.3 min.
5	648ea10.avi	884	$>$ 7 hours	8.8 min.	5.3 min.
6	644ce10.avi	540	$>$ 4 hours	6.0 min.	3.8 min.
Average time per frame (s)			29	0.6	0.4

The algorithms were timed running on an Intel Pentium 1.7 GHz PC for sequences of 352×288-sized frames. Both OFDT and MODT perform significantly faster than Fazekas et al. [25] as shown in Table 2. These times are only approximate measurements of unoptimized implementations (including even I/O operations).

5 Conclusions

In comparison to recent DT methods, we proposed two simpler and much faster approaches. Unlike recently suggested in [25], we show that an alternative flow model to the BCA is *not* required. The key observation is that locations of DT can be detected as violations of the BCA either as a lack of spatio-temporal coherence of locally computed optical flow (method OFDT), or as motion outliers detected with some robust global parametric motion estimation (method MODT). Competitive results with both synthetic and real sequences were presented. Future work for improved DT detection is to explore the combination of the robustness of the global parametric method with the flexibility of the local optical flow approach in a joint framework.

Acknowledgements. V. J. Traver was funded by grant JC2008-00397 from the Spanish Ministerio de Ciencia e Innovación. We thank the Vista research team at Irisa/Inria Rennes for the use of Motion2D software, the authors of [25] for their code, and the Spanish research programme Consolider Ingenio-2010 for grant CSD2007-00018.

References

1. Chetverikov, D., Péteri, R.: A brief survey of dynamic texture description and recognition. In: Proc. Intl. Conf. Computer Recognition Systems, pp. 17–26 (2005)
2. Péteri, R., Chetverikov, D.: Dynamic texture recognition using normal flow and texture regularity. In: IbPRIA, pp. 223–230 (2005)
3. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. IJCV 51, 91–109 (2003)
4. Atcheson, B., Heidrich, W., Ihrke, I.: An evaluation of optical flow algorithms for background oriented schlieren imaging. Experiments in Fluids 46, 467–476 (2009)
5. Vezzani, R., Calderara, S., Piccinini, P., Cucchiara, R.: Smoke detection in video surveillance: The use of ViSOR. In: ACM IVR, pp. 289–297 (2008)
6. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE-PAMI 29, 915–928 (2007)
7. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: CVPR, vol. 2, pp. 58–63 (2001)

8. Hyndman, M., Jepson, A., Fleet, D.: Higher-order autoregressive models for dynamic textures. In: BMVC (2007)
9. Chan, A.B., Vasconcelos, N.: Variational layered dynamic textures. In: CVPR (2009)
10. Vidal, R., Ravichandran, A.: Optical flow estimation and segmentation of multiple moving dynamic textures. In: CVPR, pp. 516–521 (2005)
11. Chan, A., Vasconcelos, N.: Layered dynamic textures. IEEE-PAMI 31, 1862–1879 (2009)
12. Doretto, G., Cremers, D., Favaro, P., Soatto, S.: Dynamic texture segmentation. In: ICCV, vol. 2, pp. 1236–1242 (2003)
13. Campisi, P., Maiorana, E., Neri, A., Scarano, G.: Video texture modelling and synthesis using fractal processes. IET Image Processing 2, 1–17 (2008)
14. Lu, Z., Xie, W., Pei, J., Huang, J.: Dynamic texture recognition by spatiotemporal multiresolution histograms. In: IEEE Workshop. on Motion & Video Computing, vol. 2, pp. 241–246 (2005)
15. Ghanem, B., Ahuja, N.: Extracting a fluid dynamic texture and the background from video. In: CVPR (2008)
16. Toreyin, B., Cetin, A.: HMM based method for dynamic texture detection. In: IEEE 15th. Signal Processing and Communications Applications (2007)
17. Ferrari, R.J., Zhang, H., Kube, C.R.: Real-time detection of steam in video images. PR 40, 1148–1159 (2007)
18. Xiong, X., Caballero, R., Wang, H., Finn, A.M., Lelic, M.A., Peng, P.Y.: Video-based smoke detection: Possibilities, techniques, and challenges. In: IFPA (2007)
19. Toreyin, B., Cetin, A.: On-line detection of fire in video. In: CVPR (2007)
20. Corpetti, T., Memin, E., Pérez, P.: Dense estimation of fluid flows. IEEE-PAMI 24, 365–380 (2002)
21. Toreyin, B.U., Dedeoglu, Y., Cetin, A.E.: Wavelet based real-time smoke detection in video. In: EUSIPCO (2005)
22. Rahman, A., Murshed, M.: Real-time temporal texture characterisation using block based motion co-occurrence statistics. In: ICIP, pp. III: 1593–1596 (2004)
23. Bouthemy, P., Hardouin, C., Piriou, G., Yao, J.: Mixed-state auto-models and motion texture modeling. JMIV 25, 387–402 (2006)
24. Fazekas, S., Chetverikov, D.: Analysis and performance evaluation of optical flow features for dynamic texture recognition. Signal Processing: Image Comm. 22, 680–691 (2007)
25. Fazekas, S., Amiaz, T., Chetverikov, D., Kiryati, N.: Dynamic texture detection based on motion analysis. IJCV 82, 48–63 (2009)
26. Viola, P.A., Jones, M.J.: Robust real-time face detection. IJCV 57, 137–154 (2004)
27. Odobez, J., Bouthemy, P.: Robust multiresolution estimation of parametric motion models. Int. J. Visual Communication and Image Representation 6, 348–365 (1995)
28. Chetverikov, D., Fazekas, S., Haindl, M.: Dynamic texture as foreground and background. In: MVA (2010), doi:10.1007/s00138-010-0251-6 (Published online: February 21, 2010)
29. Fazekas, S., Amiaz, T., Chetverikov, D., Kiryati, N.: (Dynamic texture detection and segmenation), `http://vision.sztaki.hu/~fazekas/dtsegm`
30. Péteri, R., Huskies, M., Fazekas, S. (DynTex: a comprehensive database of dynamic textures), `http://www.cwi.nl/projects/dyntex`