

Dynamic Texture Detection Based on Motion Analysis

Sándor Fazekas · Tomer Amiaz · Dmitry Chetverikov · Nahum Kiryati

Received: 31 October 2007 / Accepted: 7 October 2008 / Published online: 15 November 2008
© Springer Science+Business Media, LLC 2008

Abstract Motion estimation is usually based on the brightness constancy assumption. This assumption holds well for rigid objects with a Lambertian surface, but it is less appropriate for fluid and gaseous materials. For these materials an alternative assumption is required. This work examines three possible alternatives: gradient constancy, color constancy and brightness conservation (under this assumption the brightness of an object can diffuse to its neighborhood). Brightness conservation and color constancy are found to be adequate models. We propose a method for detecting regions of dynamic texture in image sequences. Accurate segmentation into regions of static and dynamic texture is achieved using a level set scheme. The level set function separates each image into regions that obey brightness constancy and regions that obey the alternative assumption. We show that the method can be simplified to obtain a less robust but fast algorithm, capable of real-time performance. Experimental results demonstrate accurate segmentation by the full level set scheme, as well as by the simplified method. The experiments included challenging image sequences, in which color or geometry cues by themselves would be insufficient.

Keywords Dynamic texture · Motion analysis · Optical flow · Level set · Real-time processing · Brightness conservation

S. Fazekas · D. Chetverikov
Computer and Automation Research Institute, Budapest, Hungary

T. Amiaz (✉) · N. Kiryati
School of Electrical Engineering, Tel Aviv University, Tel Aviv
69978, Israel
e-mail: tomrevam@yahoo.com

1 Introduction

Visual patterns such as fire and smoke, flowing water, or foliage blown by the wind are common in real-world scenes. These so called *dynamic textures*, which are time-varying with certain spatial and temporal stationarity, are easily observed by the human eye but are difficult to discern using computer vision methods (Nelson and Polana 1992; Doretto et al. 2003a). The importance of analysing dynamic textures lies in their relevance to a wide set of complex video processing tasks and scenarios, ranging from motion detection and recognition to content based multimedia indexing.

Recognition and classification of dynamic textures usually rely on both frame texture and motion (Nelson and Polana 1992; Szummer and Picard 1996; Fazekas and Chetverikov 2007). However, the spatial extent of dynamic textures (e.g. smoke, shallow water) can vary over time and they can also be partially transparent. Due to these problems, separating dynamic textures from a complex (cluttered, textured, moving) background is in many cases difficult, and single frame features in particular can be misleading. Motion analysis, on the other hand, as will be demonstrated in this paper, can be successfully employed to detect and accurately segment dynamic textures even in challenging cases.

Segmentation is one of the classical problems of computer vision. When an image sequence is available it is a well-known practice to use motion features—either alone or combined with other image cues (Murray and Buxton 1987; Shi and Malik 1998). Recently, Cremers and Soatto (2004) have suggested a variational level set method for motion based segmentation. Schoenemann and Cremers (2006) used graph cuts for the same purpose. Doretto et al. (2003b) developed a method for separating regions of different dynamics based on statistical characteristics.

Geometric, model-based, statistical and motion based techniques are used for dynamic texture analysis (Chetverikov and Péteri 2005). Motion is inferred from optical flow methods, which are usually based on the *brightness constancy assumption*. Under this assumption, the brightness of an object is constant from frame to frame. This holds well for rigid objects with a Lambertian surface, but fails for fluids and gaseous materials, that are common in dynamic textures. Therefore, optical flow based on brightness constancy cannot fully describe such visual patterns.

Several alternatives have been suggested in the past for cases in which brightness constancy does not hold: Gradient constancy provides a good solution for illumination changes and automatic adaptation of the camera to light (Uras et al. 1988; Schnörr 1984; Papenberg et al. 2006). Color constancy can adapt to multiplicative changes in illumination (Ohta 1989; Golland and Bruckstein 1997; Mileva et al. 2007). Finally, brightness conservation (also known as extended optical flow constraint) assumes the brightness of an image point can propagate to its neighborhood in the next frame, being *carried* by the flow (Song and Leahy 1991; Béréziat et al. 2000; Corpetti et al. 2000; Cuzol and Mémin 2005) (the idea was first proposed to deal with non-linear motion Schunck 1984).

Dynamic textures are usually defined by extending the concept of self-similarity—well-established for static textures—to the spatiotemporal domain. This definition encompasses two different types of textures: Weak and strong dynamic textures. *Weak* dynamic textures are the trivial case of moving textures. For such dynamic textures, there exists a local moving coordinate system in which the texture becomes fixed. Relying on the brightness constancy assumption, the appropriate motion compensation can be easily computed using standard optical flow algorithms (Horn and Schunck 1981; Lucas and Kanade 1981). However, a *strong* dynamic texture, possessing intrinsic dynamics, cannot be captured by this approach because of self-occlusion, material diffusion, and other physical processes not obeying the brightness constancy assumption.

The objective of this paper is to find a motion model which captures the intrinsic dynamics of dynamic textures, and use that model to segment video sequences between static/weak dynamic regions and strong dynamic texture regions. This is achieved by comparing the accuracy of four models on strong dynamic textures: brightness constancy, gradient constancy, color constancy and brightness conservation (this is essentially a model selection method Burnham and Anderson 1998). Brightness conservation and color constancy prove to be the best models of the four for strong dynamic textures.

Based on the above observations, we propose a segmentation scheme for detecting *strong* dynamic textures. The method is formulated as a general variational scheme with

different possible implementations. We present two major variants: (i) a level set method, (ii) a real-time thresholding technique. The first variant is the most elaborate, the most accurate, and the most robust. The second method is simple and fast: While being fairly accurate, it can run in real-time on typical computers available at the time of writing.

We test the proposed segmentation schemes on more than 30 video sequences (Fazekas et al. 2007) (most of them taken from the DynTex database Péteri et al. 2006) showing flowing water, steam, smoke, and fire—all in a natural context. Many of these sequences were recorded with a moving camera. The experimental results show the adequacy of our approach for detecting dynamic textures in challenging situations, in which methods based on geometric, color, or low level motion information would fail.

Preliminary results appeared in Amiaz et al. (2007). The current paper expands on the previous paper by examining the gradient constant and color constant alternatives to brightness conservation, replacing the first-order brightness conservation regularization previously used by second-order div-curl regularization, formulating a real time version, and adding results on more sequences.

2 Background

2.1 Motion Based Segmentation

Motion can be a useful cue for segmentation and has been used either alone (Cremers and Soatto 2004; Paragios and Deriche 2005) or to enhance other visual cues (Zheng and Blostein 1995; Galun et al. 2005). Motion based segmentation can proceed in two different ways: In the first, motion is estimated and then a segmentation algorithm is applied on the result (Wang and Adelson 1994). In the second, the motion and the segmentation are computed simultaneously (Cremers and Soatto 2004; Paragios and Deriche 2005). Usually, motion estimation relies on optical flow algorithms (Horn 1986).

For simultaneous segmentation and motion estimation, Cremers and Soatto (2004) describe a variational scheme based on a level set method. They use an evolving indicator function ϕ , which classifies every point according to whether its value is positive or negative. This approach was pioneered by Osher and Sethian (1988) (a similar technique was suggested by Dervieux and Thomasset 1979).

A simple variational level set segmentation scheme for grayscale images is formulated by Chan and Vese (2001), Vese and Chan (2002). It uses the Lagrangian

$$L_{CV}(I^+, I^-, \phi) = (I - I^+)^2 H(\phi) + (I - I^-)^2 H(-\phi) + \nu |\nabla H(\phi)|, \quad (1)$$

where I denotes the brightness of the image to be segmented, I^+ and I^- are the average brightness values of the regions corresponding to the positive and negative values of ϕ , H is the Heaviside function, $|\nabla H(\phi)|$ is the norm of the gradient of $H(\phi)$, and ν is a parameter.

In a similar way, Cremers and Soatto (2004) partition a video sequence into regions characterized by different types of affine motion. The Lagrangian associated with this segmentation is

$$L_{CS}(p_1, p_2, \phi) = \frac{p_1^T p_1}{|p_1|^2} H(\phi) + \frac{p_2^T p_2}{|p_2|^2} H(-\phi) + \nu |\nabla H(\phi)|, \quad (2)$$

where $p_i^T p_i / |p_i|^2$ measures the quality of the fit characterized by the affine motion vector p_i ($i = 1, 2$).

Extending the work presented in Cremers and Soatto (2004), Schoenemann and Cremers (2006) develop a graph cut method to partition frames in a video stream based on the locally characteristic affine motion. Graph cut methods (Shi and Malik 2000; Boykov and Jolly 2001) are based on creating a graph representation of the image with each pixel being a node. Termination nodes are added to represent the segmentation classes. In the method of Schoenemann and Cremers (2006), the links between pixel nodes encode smoothness constraints, while the links between pixel and termination nodes encode the fitness of the apparent motion compared to the parametric affine motion models corresponding to the termination nodes. An optimal graph cut is computed and the motion models are updated to those best fitting the segmented regions. This is repeated until a stable solution is reached.

Doretto et al. (2003b) proposed a method for segmenting a sequence of images into regions characterized by constant spatiotemporal statistics. They assume that the regions with different spatio-temporal statistics have boundaries that either do not change or change slowly. The local spatiotemporal dynamics in each region is modeled as a stationary Markov process. The model parameters are inferred using statistics based on simple linear-Gaussian models and a variational optimization framework. The method was successfully tested on videos combining visual patterns such as fire, smoke and water.

2.2 Dynamic Textures

The extension of texture analysis from the purely spatial domain to the spatiotemporal domain is motivated by the well-known importance of motion sensing in biological cognitive processes. For example, studies of visual perception in humans (Bruce et al. 1996) revealed that motion is used directly in recognizing aspects of the environment. Also, insects are essentially blind to anything that is standing still

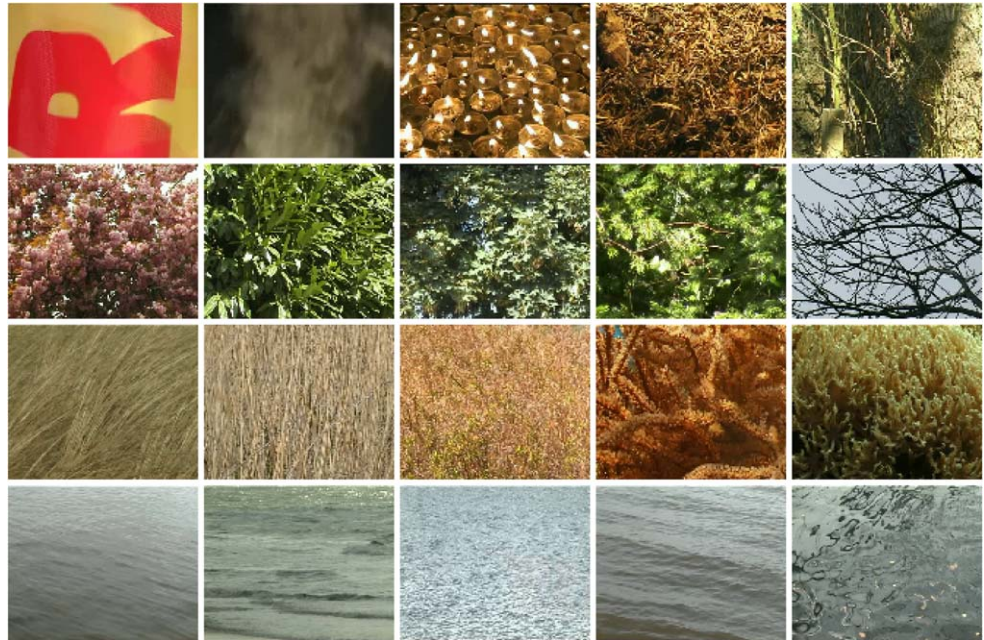
and the camouflage strategies of some animals are effective only as long as they are not moving.

Many different visual patterns can be classified as dynamic texture (see Fig. 1). These include physical processes such as surface waves, fire and smoke, flags, trees, bushes waving in the wind, underwater vegetation moved by currents, or the collective motion of different distinct elements such as a group of animals (e.g. ants or birds), a walking crowd, or cars in heavy traffic. All these examples exhibit spatiotemporal regularity with an indeterminate spatial and/or temporal extent (Nelson and Polana 1992; Soatto et al. 2001; Doretto et al. 2003a). Such visual patterns are encountered in several applications including motion detection and recognition in video surveillance as well as multimedia database queries. Dynamic textures can be used for several tasks, e.g. automated annotation, complex background modeling and video indexing, to mention just a few.

Currently the most popular methods of analysing dynamic textures (for a recent review see Chetverikov and Péteri 2005) are based on optical flow calculation (Nelson and Polana 1992; Boutheimy and Fablet 1998; Peh and Cheong 2002; Fablet and Boutheimy 2003; Lu et al. 2005; Péteri and Chetverikov 2005; Fazekas and Chetverikov 2005). Alternative approaches compute geometric properties in the spatiotemporal domain (Otsuka et al. 1998; Zhong and Scarlaffo 2002) or use local and global spatiotemporal filtering (Wildes and Bergen 2000) as well as different spatiotemporal transforms (Smith et al. 2002; Wu et al. 2001). There are also several model-based methods (Szummer and Picard 1996; Doretto et al. 2003a, 2004; Saisan et al. 2001; Fujita and Nayar 2003; Yuan et al. 2004), using estimated model parameters as features. The methods based on optical flow characterize the local dynamics of spatiotemporal textures through the computed velocity field describing the apparent motion of small image regions. In this approach, a dynamic texture can be viewed as a sequence of instantaneous motion patterns. When necessary, geometrical and color information can be also added to form a complete set of features for both motion and appearance based detection, segmentation and recognition.

Local optical flow calculation has an inherent difficulty in the fact that—independent of the imaging modality—a sensor “looking” at the world through a small hole cannot distinguish between different motion directions (Todorovic 1996). This so called *aperture problem* affects both human visual perception (Hildreth 1987) and computer vision algorithms (Horn 1986). Only the *normal flow* (orthogonal to edges and antiparallel to the local spatial image gradients) can be computed without ambiguity. The problem persists unless the motion estimation is extended to a larger region assuming correlated behavior of adjacent image elements.

Fig. 1 Examples of dynamic texture (DynTex Péteri et al. 2006)



Adding a smoothness constraint is the standard way to introduce correlated behavior (Horn and Schunck 1981), resulting in the removal of the aperture problem, the reduction of noise, but also the extension of motion over discontinuities. The alternative, local method, of dealing with the aperture problem, is to use only the normal flow, which is directly calculated from the three partial derivatives of the spatiotemporal intensity function. The normal flow, due to its locality, does not extend motion over discontinuities but tends to be very noise-prone.

Most of the work done so far in studying dynamic textures has used the normal flow, partly due to the influence of the successful pioneering work of Nelson and Polana (1992) and partly because its calculation is very easy and fast. Even though Nelson and Polana (1992) recognized already at the early state of their studies that the close relation of normal flow and spatial gradients and hence contours and shapes implies that the normal flow correlates with appearance features and thus it does not characterize the “pure dynamics”, no solution was proposed. Later, to overcome the problem, Fablet and Bouthemy (2003) used only the magnitude of the normal flow and recently Lu et al. (2005) as well as Fazekas and Chetverikov (2005, 2007) stressed the necessity to apply a complete flow calculation in extracting characteristics of dynamic textures.

2.3 Optical Flow Estimation

Among the numerous existing techniques used for motion estimation, the variational optical flow calculation methods are currently the most accurate. The Horn-Schunck algorithm (Horn and Schunck 1981) is the root of all these meth-

ods. The basic assumption used is that the brightness of objects is constant throughout the video sequence. With the correct displacement field (u, v) , this can be written as

$$I(x + u, y + v, t + 1) = I(x, y, t), \quad (3)$$

where I is the image brightness. The first-order Taylor approximation of the above *brightness constancy assumption* gives the so called *optical flow constraint*

$$I_t + uI_x + vI_y = 0, \quad (4)$$

where I_t, I_x, I_y are the partial temporal and spatial derivatives of I .

In order to overcome the aperture problem (Todorovic 1996; Horn 1986), Horn and Schunck impose a smoothness constraint on (u, v) , obtaining the Lagrangian

$$L_{HS}(u, v) = (I_t + uI_x + vI_y)^2 + \alpha(u_x^2 + u_y^2 + v_x^2 + v_y^2), \quad (5)$$

where α is a parameter and $u_x, u_y, v_x,$ and v_y are the flow derivatives. The optical flow is calculated by minimizing the functional

$$F_{HS}(u, v) = \int_{\Omega} L_{HS}(u, v) dx dy \quad (6)$$

using the calculus of variations (Horn 1986) (Ω is the image domain). The accuracy of the method can be enhanced by applying a multiscale coarse-to-fine scheme (Anandan 1989; Black and Anandan 1996), which makes the omission of the higher-order terms in the Taylor expansion (4) valid. Several other improvements were presented (see e.g. Brox et al. 2004), and recently segmentation has been used to enhance the optical flow itself (Amiaz and Kiryati 2006; Brox et al. 2006).

2.4 Brightness Constancy Alternatives

The brightness constancy assumption, as formulated in (3), states that for two consecutive frames of a video sequence—with a suitable displacement field (u, v) —the brightness of a point (x, y) in one frame is identical to the brightness of the point $(x + u, y + v)$ in the next frame. In other words, it states that by warping the image space in an appropriate way two consecutive images can be brought into point-by-point equality. There are many situations in which this assumption does not hold: For example, in case of occlusions, glinting surfaces (e.g. water surface), and complex physical processes such as smoke and fire, to name just a few. These situations are typical for dynamic textures.

The extent to which the brightness constancy assumption holds for an optical flow (u, v) computed with the Horn-Schunck method (Horn and Schunck 1981) (or more accurate methods, e.g. Brox et al. 2004 and Amiaz and Kiryati 2006) can be measured by calculating the optical flow residual

$$(I(x + u, y + v, t + 1) - I(x, y, t))^2, \quad (7)$$

where $I(x + u, y + v, t + 1)$ is computed with sub-pixel accuracy using bilinear interpolation.

Calculating the above residual for several dynamic textures available in the DynTex database (Péteri et al. 2006), we found that in certain image regions it is large and comparable to the null flow residual

$$(I(x, y, t + 1) - I(x, y, t))^2, \quad (8)$$

even for the most accurate optical flow calculation methods (see Brox et al. 2004; Amiaz and Kiryati 2006) tested. The large residual is not a consequence of numerical errors or inaccuracy of the flow, but is due to the fact that the classical brightness constancy assumption simply does not hold in certain conditions.

Many alternatives to brightness constancy have been proposed. Most were introduced to compensate for changes in illumination. When the change in illumination is additive, the intuitive choice is to assume constancy of the image gradients. Papenberg et al. (2006) have studied the inclusion of gradient constancy in variational optical flow. They suggest many options: gradient, gradient magnitude, Laplacian, Hessian, and Hessian magnitude. Of these, we choose to use the Laplacian as it proves to be very accurate in their work, and is also rotationally invariant. The resulting constancy assumption is:

$$\Delta I(x + u, y + v, t + 1) = \Delta I(x, y, t). \quad (9)$$

The equivalent to the optical flow constraint in this case is:

$$I_{xxt} + I_{yyt} + u(I_{xxx} + I_{yyx}) + v(I_{xxy} + I_{yyy}) = 0. \quad (10)$$

Color constancy has also been suggested to compensate for illumination changes. Its main advantage is that it holds even when the changes in illumination are not additive, but on the other hand the spatial resolution of the color components in most cameras is lower than that of the intensity component. The basic question in using color, is the choice of color coordinates. Mileva et al. (2007) formulated a non-linear variational optical flow estimation algorithm based on spherical coordinates in the RGB volume, with excellent results. The color constancy assumption is:

$$I^r(x + u, y + v, t + 1) = I^r(x, y, t), \quad (11)$$

$$I^\phi(x + u, y + v, t + 1) = I^\phi(x, y, t), \quad (12)$$

$$I^\theta(x + u, y + v, t + 1) = I^\theta(x, y, t), \quad (13)$$

where r, ϕ , and θ mark the spherical coordinates components. Linearizing these assumption we obtain:

$$I_t^r + uI_x^r + vI_y^r = 0, \quad (14)$$

$$I_t^\phi + uI_x^\phi + vI_y^\phi = 0, \quad (15)$$

$$I_t^\theta + uI_x^\theta + vI_y^\theta = 0. \quad (16)$$

A different approach is to relax the constancy assumption into a conservation assumption, meaning that the flow “carries” the brightness as a “physical quantity” and thus brightness changes are encoded in the divergence of the flow. Under certain conditions, such as for smoke (excluding the source), this is a physically correct model, where image brightness is related to material density (for example in cardiac images Song and Leahy 1991 or meteorological images Béréziat et al. 2000; Corpetti et al. 2000; Cuzol et al. 2007). However, the model can be justified even when the brightness is not directly related to a physical density. A glinting water surface is an example of such a process: Waves result in contraction and expansion of reflection spots, which also obey (when not saturated) the brightness conservation assumption.

Considering an arbitrary region Ω of an image I , brightness conservation can be defined as the equilibrium of the total brightness change on Ω and the brightness moving in and out through its boundary. With mathematical notations usual in physics, this can be written as

$$\int_{\Omega} \partial_t I \, dA + \int_{\partial\Omega} I \vec{f} \cdot \vec{n} \, dL = 0, \quad (17)$$

where $\partial_t I$ is the time derivative of I , $\vec{f} = (u, v)$ is the flow, $\partial\Omega$ denotes the boundary of Ω , \vec{n} is the external normal of $\partial\Omega$, dA is a volume element, and dL is a surface element.

The above equation is the integral form of the so called *continuity equation* used in physics for describing conservation of material quantities (e.g. mass or charge). Through

mathematical transformations and the divergence theorem, one can derive an equivalent differential equation

$$\partial_t I + \vec{f} \cdot \nabla I + I \operatorname{div}(\vec{f}) = 0. \tag{18}$$

Writing this in the form of (4) we obtain

$$I_t + uI_x + vI_y + Iu_x + Iv_y = 0, \tag{19}$$

where u_x and v_y denote partial derivatives of the flow. It can be observed that this is the first order Taylor approximation of

$$I(x + u, y + v, t + 1) = I(x, y, t)(1 - u_x - v_y), \tag{20}$$

which we call the *brightness conservation assumption*.

Let us compare this result to the brightness constancy assumption (3). It can be observed that (20) allows not only translational motion of objects with constant intensity, but also brightness diffusion expressed in the divergence of the flow ($u_x + v_y$). In other words, a flow satisfying equation (20) can describe more than a warp of the image space: It can also account for *changes* in brightness, and hence it can provide a more general motion model than optical flow based on brightness constancy.

The proper smoothness term for brightness conservation has been studied extensively (Corpetti et al. 2000; Cuzol et al. 2007). The standard first order smoothness term used in the Horn-Schunck Lagrangian (5) does not achieve numerical stability. A second order regularizer known as the div-curl regularizer is used in this work. Adding to the continuity equation (19), obtains

$$\begin{aligned} L_{BC}(u, v, \xi, \zeta) = & (I_t + uI_x + vI_y + Iu_x + Iv_y)^2 \\ & + \alpha((u_x + v_y - \xi)^2 + (u_y - v_x - \zeta)^2) \\ & + \beta(\xi_x^2 + \xi_y^2 + \zeta_x^2 + \zeta_y^2), \end{aligned} \tag{21}$$

where ξ and ζ are scalar fields added to provide the smoothness at the div and curl values of the optical flow field. We found that a non-robust regularizer of the scalar fields was sufficient in this case. The Lagrangian is integrated to obtain the functional

$$F_{BC}(u, v) = \int L_{BC}(u, v) \, dx dy, \tag{22}$$

which is minimized using variational calculus.

3 Dynamic Texture Segmentation

The proposed dynamic texture segmentation scheme extends the currently available motion based segmentation methods (Sect. 2.1) to dynamic textures (Sect. 2.2), based on the usual optical flow estimation techniques (Sect. 2.3) and the alternatives to brightness constancy (Sect. 2.4). Video

frames are partitioned into two regions satisfying different motion models: a static or *weak* dynamic texture region Ω_1 obeying the brightness constancy assumption (3) and a *strong* dynamic texture region Ω_2 obeying one of the alternative assumptions (9), (11–13), and (20).

Denoting by \mathcal{C} the contour separating the two regions, the segmentation problem can be formulated as minimizing the Dynamic Texture Segmentation functional

$$\begin{aligned} F_{DTS}(u, v, \tilde{u}, \tilde{v}, \mathcal{C}) = & \int_{\Omega_1} L_1(u, v) \, dx dy \\ & + \int_{\Omega_2} L_2(\tilde{u}, \tilde{v}) \, dx dy \\ & + \int S(u, v, \tilde{u}, \tilde{v}) \, dx dy + \nu|\mathcal{C}|, \end{aligned} \tag{23}$$

where (u, v) is a brightness constancy optical flow field, and (\tilde{u}, \tilde{v}) is a flow field based on an alternative assumption. The first term of the functional represents the quality of brightness constancy modeling in Ω_1 , the second term the quality of the alternative model in Ω_2 , the third term enforces the smoothness of (u, v) and (\tilde{u}, \tilde{v}) over the whole image domain, and the last term ensures that the length of the separating contour \mathcal{C} is small (ν being a scaling parameter).

The first two terms of F_{DTS} represent the adequacy of the two motion models. One option would be to directly impose the brightness constancy assumption (3) and the alternative assumptions (9), (11–13), and (20). This would result in non-linear Euler-Lagrange equations. Linear equations are obtained when minimizing the energies

$$E(u, v) = R^2(u, v), \tag{24}$$

$$\tilde{E}_{GC}(\tilde{u}, \tilde{v}) = \lambda R^2(\tilde{u}, \tilde{v}) + \tilde{R}_{GC}^2(\tilde{u}, \tilde{v}), \tag{25}$$

$$\tilde{E}_{CC}(\tilde{u}, \tilde{v}) = \lambda \tilde{R}_{CCr}^2(\tilde{u}, \tilde{v}) + \tilde{R}_{CC\phi}^2(\tilde{u}, \tilde{v}) + \tilde{R}_{CC\theta}^2(\tilde{u}, \tilde{v}), \tag{26}$$

$$\tilde{E}_{BC}(\tilde{u}, \tilde{v}) = \tilde{R}_{BC}^2(\tilde{u}, \tilde{v}) \tag{27}$$

where λ is a weight between brightness constancy and gradient or color constancy, and R and \tilde{R}_* are the residuals obtained from the optical flow constraint and the linear versions of the alternative assumptions, respectively:

$$R(u, v) = I_t + uI_x + vI_y, \tag{28}$$

$$\begin{aligned} \tilde{R}_{GC}(\tilde{u}, \tilde{v}) = & I_{xxt} + I_{yyt} + u(I_{xxx} + I_{yyy}) \\ & + v(I_{xxy} + I_{yyx}), \end{aligned} \tag{29}$$

$$\tilde{R}_{CCr}(\tilde{u}, \tilde{v}) = I_t^r + uI_x^r + vI_y^r, \tag{30}$$

$$\tilde{R}_{CC\phi}(\tilde{u}, \tilde{v}) = I_t^\phi + uI_x^\phi + vI_y^\phi, \tag{31}$$

$$\tilde{R}_{CC\theta}(\tilde{u}, \tilde{v}) = I_t^\theta + uI_x^\theta + vI_y^\theta, \tag{32}$$

$$\tilde{R}_{BC}(\tilde{u}, \tilde{v}) = I_t + \tilde{u}I_x + \tilde{v}I_y + I\tilde{u}_x + I\tilde{v}_y. \tag{33}$$

The third term of F_{DTS} is used to enforce smoothness by penalizing large optical flow gradients. Our choice for the smoothness term is

$$S_{CC}(u, v, \tilde{u}, \tilde{v}) = S_{GC}(u, v, \tilde{u}, \tilde{v}) = \alpha(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \tilde{\alpha}(\tilde{u}_x^2 + \tilde{u}_y^2 + \tilde{v}_x^2 + \tilde{v}_y^2), \quad (34)$$

$$S_{BC}(u, v, \tilde{u}, \tilde{v}, \xi, \zeta) = \alpha(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \tilde{\alpha}((u_x + v_y - \xi)^2 + (u_y - v_x - \zeta)^2) + \tilde{\beta}(\xi_x^2 + \xi_y^2 + \zeta_x^2 + \zeta_y^2), \quad (35)$$

which sums the smoothness terms of (5) and (21). The parameters α , $\tilde{\alpha}$, and $\tilde{\beta}$ specify the relative weight of the classic optical flow gradients, the conservation flow gradients, and the smoothness of the auxiliary scalar fields, respectively.

The functional described above is similar to the Mumford-Shah functional (Mumford and Shah 1989). Its direct minimization is difficult due to the unknown discontinuity set in the integration domain. One way to deal with the unspecified discontinuity in (23) is the level set approach. In this method the disjoint regions are separated by the sign of an indicator function ϕ . In our case the two regions are

$$\Omega_1 = \{(x, y) \mid \phi(x, y) > 0\} \quad \text{and} \quad (36)$$

$$\Omega_2 = \{(x, y) \mid \phi(x, y) < 0\}, \quad (37)$$

representing the weak and strong dynamic texture regions. The length of the separating contour is calculated as

$$|\mathcal{C}| = \int |\nabla H(\phi)| \, dx \, dy, \quad (38)$$

where H denotes the Heaviside function.

Following the general framework outlined above, we define the Level Set Dynamic Texture Segmentation functional

$$F_{LSDTS}(u, v, \tilde{u}, \tilde{v}, \phi) = \int [E(u, v)H(\phi) + (\gamma \tilde{E}_*(\tilde{u}, \tilde{v}) + \rho)H(-\phi) + S_*(u, v, \tilde{u}, \tilde{v}) + \nu |\nabla H(\phi)|] \, dx \, dy. \quad (39)$$

γ is a weight parameter between the two types of flow. The parameter ρ penalizes the size of region Ω_2 , which is necessary because the alternative assumptions are more general than brightness constancy. Otherwise, the minimization of F_{LSDTS} would tend to suppress the weak dynamic texture region Ω_1 in favor of Ω_2 . The smoothness of the separating boundary can be adjusted with ν . The asterisk (*) denotes the specific alternative assumption: GC —gradient

constancy, CC —color constancy, and BC —brightness conservation. Simultaneous motion detection and segmentation is achieved by minimizing the objective functional.

F_{LSDTS} is minimized by deriving, discretizing, and solving the Euler-Lagrange equations for u , v , \tilde{u} , \tilde{v} , and ϕ . The equations for u and v are

$$I_x R(u, v)H(\phi) - \alpha(u_{xx} + u_{yy}) = 0, \quad (40)$$

$$I_y R(u, v)H(\phi) - \alpha(v_{xx} + v_{yy}) = 0. \quad (41)$$

For \tilde{u} and \tilde{v} , we have in the gradient constancy case:

$$\gamma[\lambda I_x R(\tilde{u}, \tilde{v}) + (I_{xxx} + I_{yyy})\tilde{R}_{GC}(\tilde{u}, \tilde{v})]H(-\phi) - \tilde{\alpha}(\tilde{u}_{xx} + \tilde{u}_{yy}) = 0, \quad (42)$$

$$\gamma[\lambda I_y R(\tilde{u}, \tilde{v}) + (I_{xxy} + I_{yyx})\tilde{R}_{GC}(\tilde{u}, \tilde{v})]H(-\phi) - \tilde{\alpha}(\tilde{v}_{xx} + \tilde{v}_{yy}) = 0. \quad (43)$$

In the color constancy case:

$$\gamma[\lambda I_x^r \tilde{R}_{CCr}(\tilde{u}, \tilde{v}) + I_x^\phi \tilde{R}_{CC\phi}(\tilde{u}, \tilde{v}) + I_x^\theta \tilde{R}_{CC\theta}(\tilde{u}, \tilde{v})]H(-\phi) - \tilde{\alpha}(\tilde{u}_{xx} + \tilde{u}_{yy}) = 0, \quad (44)$$

$$\gamma[\lambda I_y^r \tilde{R}_{CCr}(\tilde{u}, \tilde{v}) + I_y^\phi \tilde{R}_{CC\phi}(\tilde{u}, \tilde{v}) + I_y^\theta \tilde{R}_{CC\theta}(\tilde{u}, \tilde{v})]H(-\phi) - \tilde{\alpha}(\tilde{v}_{xx} + \tilde{v}_{yy}) = 0. \quad (45)$$

Finally, for the brightness conservation case:

$$\gamma I \tilde{R}_{BCx}(\tilde{u}, \tilde{v})H(-\phi) + \tilde{\alpha}(\tilde{u}_{xx} + \tilde{u}_{yy} - \xi_x - \zeta_y) = 0, \quad (46)$$

$$-\gamma I \tilde{R}_{BCy}(\tilde{u}, \tilde{v})H(-\phi) - \tilde{\alpha}(\tilde{v}_{xx} + \tilde{v}_{yy} + \zeta_x - \xi_y) = 0, \quad (47)$$

$$\tilde{\alpha}(\tilde{u}_x + \tilde{v}_y - \xi) + \tilde{\beta}(\xi_{xx} + \xi_{yy}) = 0, \quad (48)$$

$$\tilde{\alpha}(\tilde{u}_y - \tilde{v}_x - \zeta) + \tilde{\beta}(\zeta_{xx} + \zeta_{yy}) = 0. \quad (49)$$

Here, \tilde{R}_{BCx} and \tilde{R}_{BCy} denote partial derivatives of \tilde{R}_{BC} with respect to x and y . Finally, the indicator function ϕ must satisfy

$$\delta(\phi) \left[E(u, v) - \gamma E_*(\tilde{u}, \tilde{v}) - \rho - \nu \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] = 0, \quad (50)$$

where δ is the Dirac function—the derivative of the Heaviside function.

We discretize (50) according to the method presented in Vese and Chan (2002) (see details in the Appendix). Equations (40) and (41) are discretized and solved as described in Horn and Schunck (1981) (and the discretization and solution of (42–45) is a trivial extension of this method). Below, we describe the details of discretizing equations (46) and (47) following the same method.

After discretization (using central difference), we obtain

$$\begin{aligned}
 &-\gamma I(I_{tx} + I_x(2\tilde{u}_x + \tilde{v}_y) + \tilde{u}I_{xx} + \tilde{v}_x I_y \\
 &\quad + \tilde{v}I_{xy} + I(\tilde{u}_x - 2\tilde{u} + \tilde{v}_{xy}))H(-\phi) \\
 &-\tilde{\alpha}(\tilde{u}_x + \tilde{u}_y - 4\tilde{u} - \xi_x - \zeta_y) = 0, \tag{51}
 \end{aligned}$$

$$\begin{aligned}
 &-\gamma I(I_{ty} + I_y(2\tilde{v}_y + \tilde{u}_x) + \tilde{v}I_{yy} + \tilde{u}_y I_x \\
 &\quad + \tilde{u}I_{xy} + I(\tilde{v}_y - 2\tilde{v} + \tilde{u}_{xy}))H(-\phi) \\
 &-\tilde{\alpha}(\tilde{v}_x + \tilde{v}_y - 4\tilde{v} + \zeta_x - \xi_y) = 0. \tag{52}
 \end{aligned}$$

Here, we used the notations

$$\tilde{u}_x = \tilde{u}(x - 1, y) + \tilde{u}(x + 1, y), \tag{53}$$

$$\tilde{u}_y = \tilde{u}(x, y - 1) + \tilde{u}(x, y + 1). \tag{54}$$

The definitions of \tilde{v}_x and \tilde{v}_y are analogous.

Equations (51) and (52) of \tilde{u}, \tilde{v} can be represented in matrix form:

$$\begin{pmatrix} \tilde{A} & \tilde{C} \\ \tilde{C} & \tilde{B} \end{pmatrix} \begin{pmatrix} \tilde{u} \\ \tilde{v} \end{pmatrix} = \begin{pmatrix} \tilde{D} \\ \tilde{E} \end{pmatrix}, \tag{55}$$

where

$$\tilde{A} = \gamma I(2I - I_{xx})H(-\phi) + 4\tilde{\alpha}, \tag{56}$$

$$\tilde{B} = \gamma I(2I - I_{yy})H(-\phi) + 4\tilde{\alpha}, \tag{57}$$

$$\tilde{C} = -\gamma II_{xy}H(-\phi), \tag{58}$$

$$\begin{aligned}
 \tilde{D} = &\gamma I[I_{tx} + I_x(2\tilde{u}_x + \tilde{v}_y) + I_y\tilde{v}_x + I(\tilde{v}_{xy} + \tilde{u}_x)]H(-\phi) \\
 &+ \tilde{\alpha}(\tilde{u}_x + \tilde{u}_y - \xi_x - \zeta_y), \tag{59}
 \end{aligned}$$

$$\begin{aligned}
 \tilde{E} = &\gamma I[I_{ty} + I_y(2\tilde{v}_y + \tilde{u}_x) + I_x\tilde{u}_y + I(\tilde{u}_{xy} + \tilde{v}_y)]H(-\phi) \\
 &+ \tilde{\alpha}(\tilde{v}_x + \tilde{v}_y + \zeta_x - \xi_y). \tag{60}
 \end{aligned}$$

Iterations are carried out on $u, v, \tilde{u}, \tilde{v}, \xi, \zeta$ and ϕ in alternation. Because a continuous limit and a first order Taylor approximation are assumed in the flow field equations, (u, v) and (\tilde{u}, \tilde{v}) need to be small. This is achieved by using a coarse-to-fine scheme: The images are warped according to the flow calculated at a coarse scale and small corrections are added repeatedly at finer scales.

Both flow fields (u, v) and (\tilde{u}, \tilde{v}) , the scalar fields ξ, ζ and the indicator function ϕ are initialized to zero. Based on our experience, there is no significant dependence on the initialization of ϕ . To break the ambiguous case $\phi = 0$, during the calculations, random noise—close to the estimated level of numerical errors—can be added to ϕ . The results are not significantly affected by the random seed used.

It is expected that the regions Ω_1 and Ω_2 will change slowly with time. However, this expectation was not formulated within the functional (39). A post-processing step performed on the indicator function ϕ was the method chosen

to apply this assumption. For each frame, the binary mask $\phi < 0$ (i.e., strong dynamic texture) was summed with that of the previous frame, then a median filter was applied to the result.

4 Real-Time Segmentation

The method described in the previous section is accurate and robust, but not fast. In this section we consider the reduction of computational cost, with the goal of reaching real-time performance. The first step is to drop the dependence of the functional on the brightness conservation flow and compare the Horn-Schunck residual to a threshold. However, this means that wherever optical flow estimation is problematic (for any reason, such as fast motion), the algorithm will detect strong dynamic texture. When $\gamma = \tilde{\alpha} = \tilde{\beta} = 0$, the functional (39) loses its dependence on (\tilde{u}, \tilde{v}) and becomes

$$\begin{aligned}
 &F_{LSDTS'}(u, v, \phi) \\
 &= \int [(I_t + uI_x + vI_y)^2 H(\phi) + \rho H(-\phi) \\
 &\quad + \alpha(u_x^2 + u_y^2 + v_x^2 + v_y^2) + v|\nabla H(\phi)|] dx dy. \tag{61}
 \end{aligned}$$

The Euler-Lagrange equations of (u, v) remain the same, while the new descent equation of ϕ is

$$\delta(\phi) \left[(I_t + uI_x + vI_y)^2 - \rho - v \operatorname{div} \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \right] = 0. \tag{62}$$

This simplification significantly lowers the computational load, as the iterations on \tilde{u} and \tilde{v} are avoided.

Pushing further the simplifications formulated in (61), an even faster algorithm can be obtained: Instead of simultaneous motion estimation and segmentation, we calculate the Horn-Schunck flow first (based on equations (40) and (41) with fixed $\phi = +\infty$) and then compute the optical flow residual defined in (7). Thresholding this residual, a rough segmentation is achieved. The same post-processing method is applied as for the level set method (Sect. 3).

This variant is fast enough for real-time computation. However, it requires the setting of a threshold value, which generally depends on the specific video sequence (but note that the experimental results shown in Fig. 6 have all been obtained with the same threshold). The choice of threshold for the algorithm can be performed either interactively by a human, or automatically by the algorithm. An automatic method, based on Otsu’s threshold selection technique (Otsu 1979), performed reasonably well in our tests, provided that a sufficient number of frames, either fixed or a sliding window, was used. This simple method had difficulties with extensive changes in the dynamics of the processed scenes (for

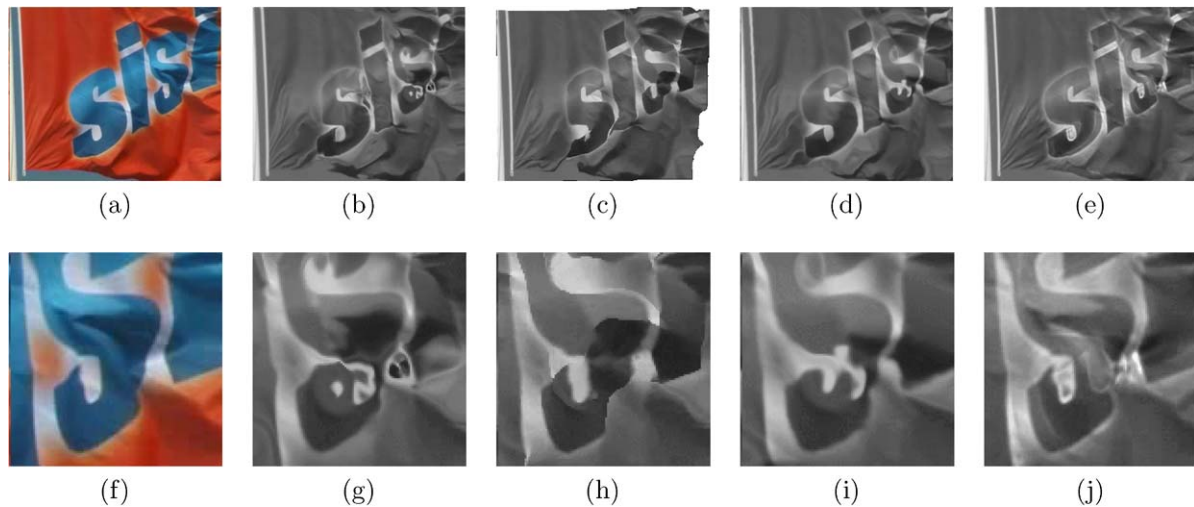


Fig. 2 Motion compensation with classical optical flow methods and the proposed brightness conservation flow: Original frame (a), Horn-Schunck flow (b), Brox et al. (2004) flow (c), color constancy based flow (d) and brightness conservation flow (e), details of the images are

provided in (f–j). Significant distortion of the warped image is evident in brightness constancy and gradient based methods and little distortion in the color constancy method, while the best reconstruction is achieved by brightness conservation

example significant changes in camera motion, or when people move into a scene previously dominated by moving foliage).

We have also designed and implemented a more sophisticated method, which relied on setting the threshold based on pixels suspected to be dynamic texture. The first stage of this method, was to compare for each pixel the Horn-Schunck residual (7) with the null flow residual (8). The average m and standard deviation σ of the motion compensation residual was computed for the pixels in which the motion compensation residual was smaller than half the null flow residual. The threshold was set to $m + 2\sigma$. This threshold was more robust than a fixed threshold, because it followed the changes in the video dynamics. However, it failed when the analyzed frames were either dominated by a strong dynamic texture or there was no strong dynamic texture at all.

5 Experimental Results

5.1 Motion Compensation

We have tested the proposed brightness conservation flow on several video sequences taken from the DynTex database (Péteri et al. 2006). We present the result on the flapping flag (6483c10) sequence. We compare motion compensation results obtained with the color constancy flow (26), brightness conservation flow (21), the classical Horn-Schunck flow (5) and the high-accuracy optical flow algorithm of Brox et al. (2004). Figure 2 demonstrates that the brightness conservation flow can reproduce the original image almost perfectly, while the brightness constancy

flows produce visible errors originating from the fact that they cannot accommodate changes in brightness. The color constancy flow is better than the brightness based flows, but fails in the color-less white areas.

The flows are computed between the first two frames of each video sequence. Motion compensation is calculated by warping the second frame according to the optical flow in order to be comparable to the first one. Motion compensation with the brightness conservation is calculated using the formula

$$I(x, y, t) = \frac{I(x + u, y + v, t + 1)}{1 - u_x - v_y}. \quad (63)$$

The input images were pre-blurred with a Gaussian filter ($\sigma = 0.4$). Coarse-to-fine iterations were run over a Gaussian pyramid with scale factor 0.5. The brightness conservation flow was calculated with $\alpha = 10^{-3}$, $\beta = 10^{-2}$, 4 level pyramid and 150 iterations on each level. The Horn-Schunck flow was calculated with $\alpha = 5 \times 10^{-3}$, 4 level pyramid and 150 iterations. The color constancy flow was calculated with $\lambda = 0$, $\alpha = 5 \times 10^{-3}$, 3 level pyramid and 1000 iterations. The parameters used for the Brox et al. algorithm were those described in their paper (Brox et al. 2004).

5.2 Segmentation Results

We have tested the proposed segmentation methods on more than 30 video sequences (Fazekas et al. 2007). Table 1 describes the nine sequences we show in this paper. The videos used comprise of dynamic textures in a natural context (flowing water, smoke and fire), shot with a moving or stationary camera. In the tests, the segmentation is based

Table 1 Short description of the video sequences used for demonstration in this paper (see the results presented in Figs. 3, 4 and 6)

| | Code | Description | Camera |
|---|---------|--|---------|
| a | 6482910 | Wide stream with water rippling over underwater pebbles | Panning |
| b | 6483c10 | Narrow creek winding between larger pebbles | Panning |
| c | 6481i10 | Water falling over round pebbles and gathering into a pool | Panning |
| d | 73v195t | Fire, smoke, and steam (while cooking goulash) | Panning |
| e | 73v1985 | Fire and smoke (closeup of (d)) | Panning |
| f | 648ea10 | Steam (or smoke) coming from a ventilation system | Fixed |
| g | 64adg10 | Rain drops falling over a small lake | Panning |
| h | 647bc10 | Shipcanal and shoreline in Amsterdam | Panning |
| i | 644ce10 | Duck floating on water | Fixed |

on the motion only. The videos were processed in grayscale for the brightness conservation, gradient constancy and real-time algorithms and in color for the color constancy algorithm. The frame size was at resolution 360×288 and the pixel values were scaled to $[0, 1]$.

The parameters used in testing the brightness conservation segmentation method (Sect. 3) were the same for all videos: $\alpha = 5 \times 10^{-3}$, $\tilde{\alpha} = 0.5 \times 10^{-3}$, $\tilde{\beta} = 1$, $\gamma = 0.5$, $\rho = 10^{-4}$, and $\nu = 10^{-3}$. The parameters used for the color constancy case were the same, together with $\lambda = 0$. The Heaviside function was implemented as $H(\phi) = 0.5 + \arctan(\phi/\phi_0)/\pi$ with $\phi_0 = 10^{-3}$. The parameters were not fine tuned, setting their order of magnitude was enough to handle a large number of examples. The frames were pre-smoothed with a Gaussian kernel ($\sigma = 0.8$). A Gaussian pyramid with 4 levels and scaling factor 0.5 was built and 150 iterations were executed at each level. The indicator function ϕ was not calculated on the image border (12 pixel width). The post-processing used a median filter with a 7×7 pixels window.

Figures 3 and 4 present segmentation results in very good agreement with the actual position of flowing water, steam, smoke, and fire. In particular, notice the correct segmentation of the rock protruding from the water in (a), and the differentiation between the duck and the duck's reflection in (i). Only small regions were falsely detected as weak dynamic texture. These are typically regions where there is not too much activity and—because all these materials are transparent—they could not be distinguished from the background. There are also small non-dynamic areas (e.g. edges of the pebbles protruding from the water) being misdetected, probably due to over-smoothing of the level set function ϕ . Another source of errors is due to motion blur caused by the limitations of the video recording system (in high velocity motion of either the camera or objects in the scene). Tuning the parameters to each sequence could improve the accuracy of the segmentation, however the sensitivity is low, for example Fig. 5 demonstrates that the sensitivity of the algorithm to the value of ρ is very low. The processing time for

a single frame in this method was around one minute on a computer with a 2 GHz dual core CPU.

Results for the gradient constancy based segmentation method are not shown because they did not produce any meaningful segmentation. It seems that the gradient constancy flow is generally more accurate than the brightness constancy flow, but this benefit is independent of the type of dynamic texture on which it is calculated.

For the real-time method described in Sect. 4, the OpenCV implementation of the Horn-Schunck algorithm was used (<http://opencvlibrary.sourceforge.net/CvReference>). We tested the method on the same sequences (see Table 1). We ran the method on a three level Gaussian pyramid executing 100 iterations on each level and using the smoothness parameter $\lambda = 10^{-3}$. The segmentation threshold was set to 3 for optical flow residual values in $[0, 255]$. The size of the median filter window used in the post-processing stage was 15×15 pixels. The method reached a processing speed of 20 video frames per second on a computer with a 2 GHz dual core CPU (in the optimized version we employed multiple cores by using OpenMP compiler directives). Figure 6 demonstrates that the results are still pretty good. However, the method is not as robust as the level set scheme, because the threshold may need specific tuning for each sequence (and sometimes even for changes within a sequence). The sensitivity of the method to variation in the value of the threshold is demonstrated in Fig. 7.

Figure 8 shows three sequences as segmented by the three methods. The color constancy algorithm is a little better for the cases of fire and smoke, while the brightness conservation algorithm seems to be better in the sequences with water. The real time segmentation is usually noisier (with the noise subdued by using a 15×15 median filter).

6 Conclusions and Discussion

Analysing dynamic textures is important for a wide set of applications, spanning from motion detection to activity recognition, from background modeling to video in-

Fig. 3 Dynamic texture detection results obtained with the brightness conservation method. The segmentation is marked with a *white curve*. The samples are taken from the DynTex database (Péteri et al. 2006). Most sequences were shot with a moving camera, for a short description see Table 1. For more results see Fazekas et al. (2007). The segmentation of dynamic textures is very accurate

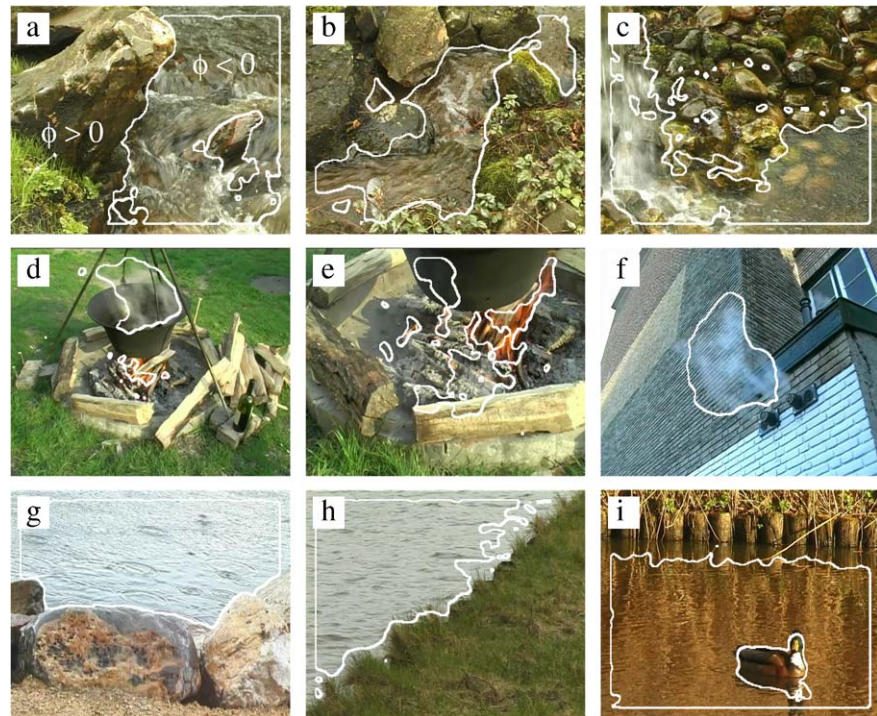
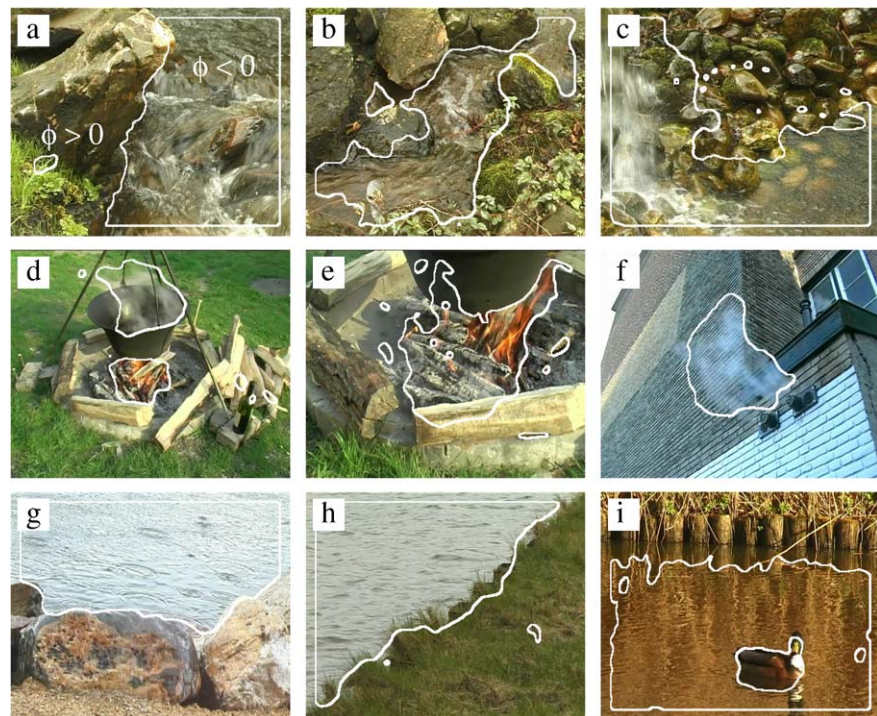


Fig. 4 Dynamic texture detection result obtained with color constancy method. The segmentation is marked with a *white curve*. The samples are taken from the DynTex database (Péteri et al. 2006). Most sequences were shot with a moving camera, for a short description see Table 1. The segmentation of dynamic textures is very accurate



dexing, and from video surveillance to multimedia databases. Certain video regions—identified as *strong* dynamic texture—possess inherent dynamics which cannot be completely modeled by motion estimation based on *brightness constancy*. This indicates that a more general description is needed. In this paper, three optical flow estimation methods based on alternative assumptions (gradient constancy,

color constancy and brightness conservation) are evaluated. Experimental tests demonstrate that the methods best suited for strong dynamic textures is the one based on the brightness conservation assumption. Gradient constancy is usually better at estimating flow than brightness constancy, but the improvement in estimation is not more pronounced in strong dynamic texture regions.

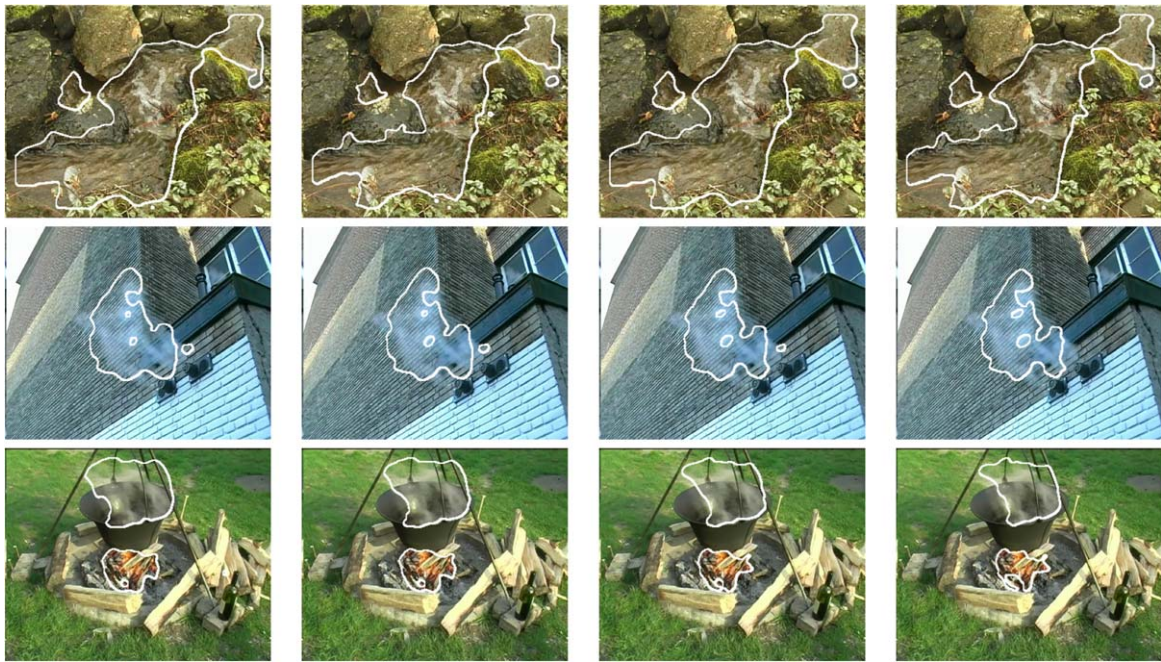
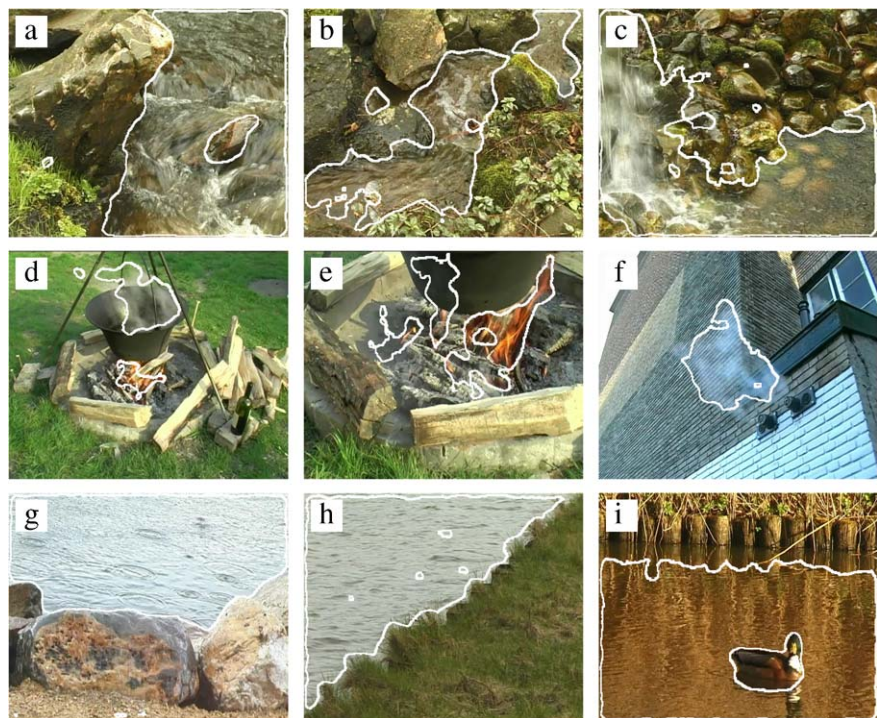


Fig. 5 Sensitivity of the brightness conservation segmentation to the parameter ρ . The parameters are fixed in all runs, except for ρ whose values are 0, 10^{-5} , 5×10^{-5} and 10^{-4} from left to right

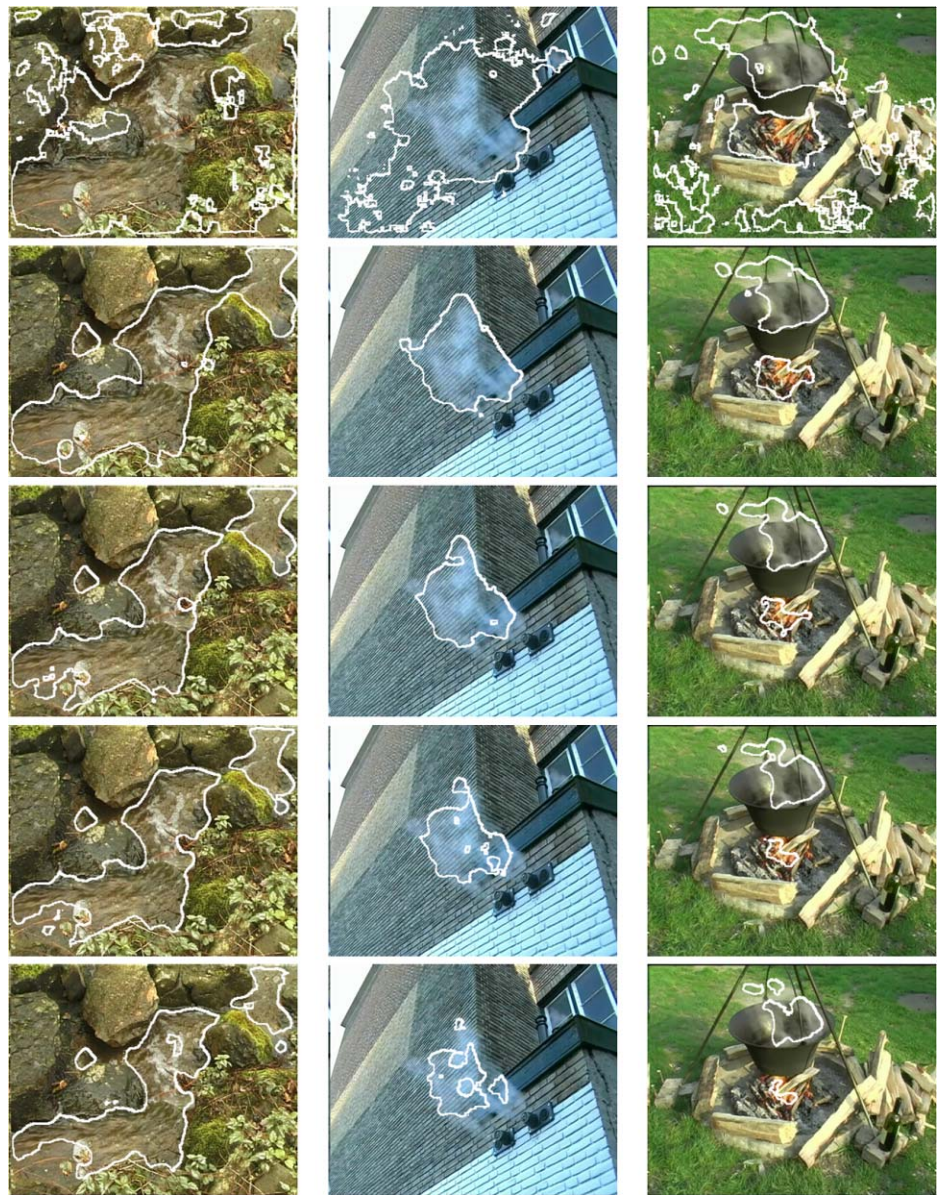
Fig. 6 Dynamic texture detection results obtained with the real time method. The segmentation is marked with a white curve. Most sequences were shot with a moving camera, for a short description see Table 1. The results are reasonable, although some small regions are misclassified



Based on the observation that the residual of brightness conservation based flow is significantly smaller than the residual of classical optical flow in regions of strong dynamic texture, a segmentation method was presented. The proposed approach is based solely on motion analysis and on separating regions obeying different motion assumptions.

The only restriction placed on the boundary between regions is that its length will be minimal. The segmentation method can be applied in challenging cases when other visual cues (color and geometry) are misleading or simply useless. Such situations arise, for example, in case of smoke and shallow water, which are usually transparent, and tex-

Fig. 7 Sensitivity of the real-time segmentation to the threshold parameter. The parameters are fixed in all runs, except for the threshold parameter whose values are 1, 2, 3, 4 and 5 from *top to bottom*



tured patterns behind them spoil the detection and recognition efforts. Tests of the segmentation scheme were carried on several video sequences recorded with a moving camera showing water, steam, smoke, and fire in a large variety of natural scenes (Fazekas et al. 2007). The methods based on the brightness conservation and color constancy assumption proved to be effective in detecting and segmenting such complex visual patterns.

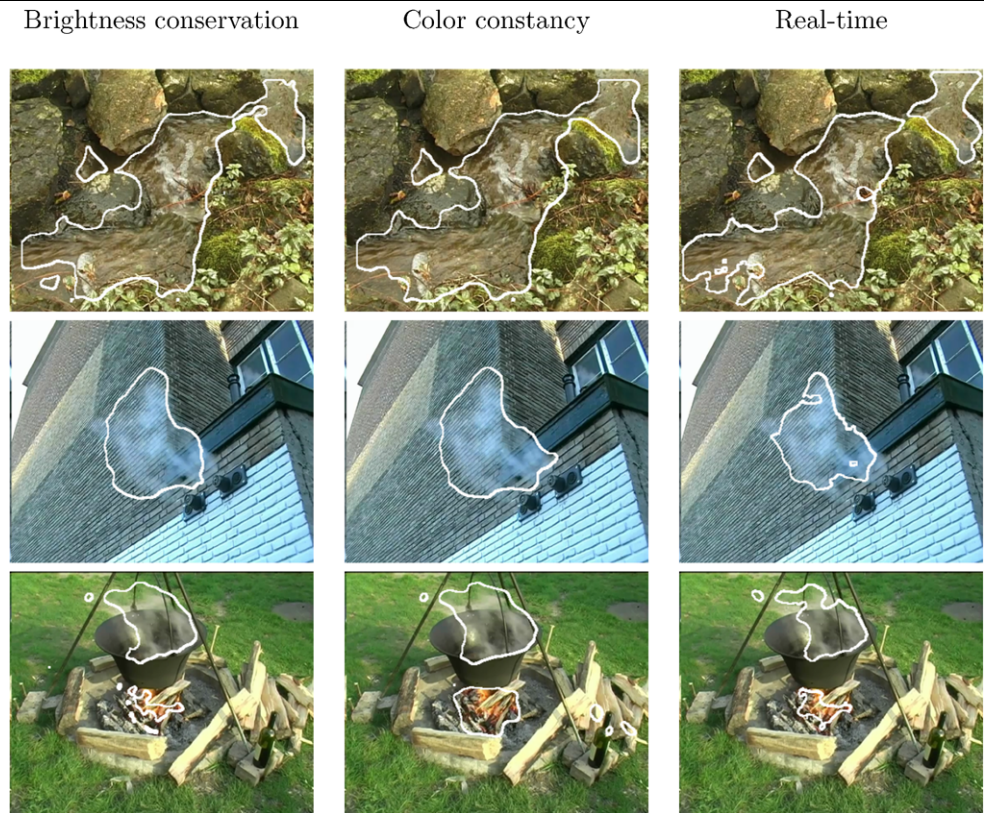
In its complete form, the algorithm was implemented as a variational level set method. Real-time performance, with reasonable accuracy, was obtained by a simplified algorithm, using thresholding of the motion-compensated residual of the Horn-Schunck flow. Even though the reliance on a threshold value limits the robustness of this method, it could

be successfully used in applications which require a very short response time.

The functional (21) used to calculate the brightness conservation flow includes explicitly the brightness values of the processed images. Optical flow methods usually include only derivatives of the image brightness and are, therefore, invariant to the numerical value of black and white. The brightness conservation assumption as stated in (20) violates this principle. Future work is needed to unveil the consequences of this property. Another issue that deserves further study is the incorporation of sources and sinks of brightness, for example, the point from which smoke is originating.

In the methods presented, a post-processing operator was used to provide temporal smoothness of the segmentation result. The level set methods could be improved by formu-

Fig. 8 Comparison of the segmentation methods on three sequences



lating the segmentation functional with a three-dimensional spatio-temporal smoothness term, thus accounting for temporal correlations. The results can be further refined by using statistical data taking into consideration characteristics of specific dynamic textures. The statistics of the calculated flows could be useful in such a scheme.

The linear approximation of the flow functional (21) negatively affects the accuracy of the method (see e.g. Papenberg et al. 2006). Reformulating the flow estimation as a non-linear method could significantly improve the results. The method can be further enhanced by using Sobolev active contours (Sundaramoorthi et al. 2007) and additional features such as image intensity data (Caselles et al. 1997). The concept of segmenting video based on the applicability of different motion models to different regions can be expanded to other motion models. For example, the parametric models presented by Nir et al. (2008) could be used in a similar manner and one could, for example, separate background motion and foreground activities.

Acknowledgements This research was supported partly by MUSCLE: Multimedia Understanding through Semantics, Computation and Learning, a European Network of Excellence funded by the EC 6th Framework IST Program. At Tel Aviv University this research was supported also by the A.M.N. Foundation.

Appendix: Level Set Discretization

This appendix provides details on the numerical approximation and discretization methods used in the implementation of our algorithm. We denote

$$h \triangleq \Delta x = \Delta y.$$

Discrete spatial indices are denoted by subscripts, and the temporal index (iteration count) by a superscript, as in

$$\phi_{i,j}^n \triangleq \phi(n\Delta t, i\Delta x, j\Delta y).$$

The numerical approximations of the Heaviside function and its derivative are

$$H_{\phi_0}(\phi) = \frac{1}{2} \left[1 + \frac{2}{\pi} \arctan \left(\frac{\phi}{\phi_0} \right) \right],$$

$$\delta_{\phi_0}(x) = H'_{\phi_0}(x) = \frac{1}{\pi} \frac{\phi_0}{\phi_0^2 + x^2}.$$

Given the warping of an image by an optical flow field, the image values on the resampling grid are obtained by bilinear interpolation.

We wish to numerically solve (50):

$$\frac{\partial \phi}{\partial t} = v\delta(\phi)\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) - \delta(\phi)[R^2(u, v) - \gamma \tilde{R}^2(\tilde{u}, \tilde{v}) - \rho].$$

We adopt the notations (Vese and Chan 2002)

$$C_1 = \frac{1}{\sqrt{\left(\frac{\phi_{i+1,j}^n - \phi_{i,j}^n}{h}\right)^2 + \left(\frac{\phi_{i,j+1}^n - \phi_{i,j}^n}{2h}\right)^2}},$$

$$C_2 = \frac{1}{\sqrt{\left(\frac{\phi_{i,j}^n - \phi_{i-1,j}^n}{h}\right)^2 + \left(\frac{\phi_{i-1,j+1}^n - \phi_{i-1,j}^n}{2h}\right)^2}},$$

$$C_3 = \frac{1}{\sqrt{\left(\frac{\phi_{i+1,j}^n - \phi_{i-1,j}^n}{2h}\right)^2 + \left(\frac{\phi_{i,j+1}^n - \phi_{i,j}^n}{h}\right)^2}},$$

$$C_4 = \frac{1}{\sqrt{\left(\frac{\phi_{i+1,j-1}^n - \phi_{i-1,j-1}^n}{2h}\right)^2 + \left(\frac{\phi_{i,j}^n - \phi_{i,j-1}^n}{h}\right)^2}},$$

and

$$m = \frac{\Delta t}{h^2} \delta_{\phi_0}(\phi_{i,j}) \nu,$$

$$C = 1 + m(C_1 + C_2 + C_3 + C_4).$$

The discretized residual terms are denoted by

$$R_{i,j}^n = I_{i,j}^n + u_{i,j}^n I_{x,i,j}^n + v_{i,j}^n I_{y,i,j}^n,$$

$$\begin{aligned} \tilde{R}_{i,j}^n &= I_{i,j}^n + \tilde{u}_{i,j}^n I_{x,i,j}^n + \tilde{v}_{i,j}^n I_{y,i,j}^n \\ &+ I_{i,j} \frac{\tilde{u}_{i+1,j}^n - \tilde{u}_{i-1,j}^n}{2h} + I_{i,j} \frac{\tilde{v}_{i,j+1}^n - \tilde{v}_{i,j-1}^n}{2h}. \end{aligned}$$

The discrete evolution equation for ϕ finally takes the form

$$\begin{aligned} \phi_{i,j}^{n+1} &= \frac{1}{C} [\phi_{i,j}^n + m(C_1 \phi_{i+1,j}^n + C_2 \phi_{i-1,j}^n \\ &+ C_3 \phi_{i,j+1}^n + C_4 \phi_{i,j-1}^n) \\ &- \Delta t \delta_{\phi_0}(\phi_{i,j}^n) \cdot (R_{i,j}^n - \gamma \tilde{R}_{i,j}^n - \rho)]. \end{aligned}$$

References

- Amiaz, T., & Kiryati, N. (2006). Piecewise-smooth dense optical flow via level sets. *International Journal of Computer Vision*, 68(2), 111–124.
- Amiaz, T., Fazekas, S., Chetverikov, D., & Kiryati, N. (2007). Detecting regions of dynamic texture. In *LNCS: Vol. 4485. Proc. SSVM 2007* (pp. 848–859). Berlin: Springer.
- Anandan, P. (1989). A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3), 283–310.
- Béréziat, D., Herlin, I., & Younes, L. (2000). A generalized optical flow constraint and its physical interpretation. In *Proc. conf. comp. vision pattern rec.* (pp. 487–492).
- Black, M. J., & Anandan, P. (1996). The robust estimation of multiple motions: parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63(1), 75–104.
- Boutheimy, P., & Fablet, R. (1998). Motion characterization from temporal co-occurrences of local motion-based measures for video indexing. In *Proc. int. conf. pattern recognition* (Vol. 1, pp. 905–908).

- Boykov, Y. Y., & Jolly, M.-P. (2001). Interactive graph cuts for optimal boundary & region segmentation of objects in n - d images. In *8th int. conf. on computer vision* (Vol. 01, pp. 105–112).
- Brox, T., Bruhn, A., Papenberger, N., & Weickert, J. (2004). High accuracy optical flow estimation based on a theory for warping. In *LNCS: Vol. 3024. Proc. European conf. on computer vision* (pp. 25–36). Berlin: Springer.
- Brox, T., Bruhn, A., & Weickert, J. (2006). Variational motion segmentation with level sets. In *LNCS: Vol. 3951. Proc. European conf. on computer vision* (pp. 471–483). Berlin: Springer.
- Bruce, V., Green, P. R., & Georgeson, M. (1996). *Visual perception*. Psychology Press.
- Burnham, K., & Anderson, D. (1998). *Model selection and inference—a practical information-theoretic approach*. Berlin: Springer.
- Caselles, V., Kimmel, R., & Sapiro, G. (1997). Geodesic active contours. *International Journal of Computer Vision*, 22(1), 61–79.
- Chan, T. F., & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2), 266–277.
- Chetverikov, D., & Péteri, R. (2005). A brief survey of dynamic texture description and recognition. In *4th int. conf. on computer recognition systems* (pp. 17–26).
- Corpetti, T., Mémin, E., & Pérez, P. (2000). Adaptation of standard optic methods to fluid motion. In *Int. symposium on flow visualization* (pp. 1–10).
- Cremers, D., & Soatto, S. (2004). Motion competition: A variational approach to piecewise parametric motion segmentation. *International Journal of Computer Vision*, 62(3), 249–265.
- Cuzol, A., & Mémin, E. (2005). Vortex and source particles for fluid motion estimation. In *LNCS: Vol. 3459. Proc. scale-space 2005* (pp. 254–266). Berlin: Springer.
- Cuzol, A., Hellier, P., & Mémin, E. (2007). A low dimensional fluid motion estimator. *International Journal of Computer Vision*, 75(3), 329–349.
- Dervieux, A., & Thomasset, F. (1979). A finite element method for the simulation of rayleigh-Taylor instability. In *Lecture notes in mathematics* (Vol. 771, pp. 145–158).
- Doretto, G., Chiuso, A., Soatto, S., & Wu, Y. N. (2003a). Dynamic textures. *International Journal of Computer Vision*, 51, 91–109.
- Doretto, G., Cremers, D., Favaro, P., & Soatto, S. (2003b). Dynamic texture segmentation. In *Ninth int. conf. on computer vision* (p. 1236).
- Doretto, G., Jones, E., & Soatto, S. (2004). Spatially homogeneous dynamic textures. In *LNCS: Vol. 3022. Proc. European conf. on computer vision* (pp. 591–602). Berlin: Springer.
- Fablet, R., & Boutheimy, P. (2003). Motion recognition using non-parametric image motion models estimated from temporal and multiscale co-occurrence statistics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25, 1619–1624.
- Fazekas, S., Amiaz, T., Chetverikov, D., & Kiryati, N. (2007). *Dynamic texture detection and segmentation*. <http://vision.sztaki.hu/~fazekas/dtsegm>.
- Fazekas, S., & Chetverikov, D. (2005). Normal versus complete flow in dynamic texture recognition: A comparative study. In *Int. workshop on texture analysis and synthesis* (pp. 37–42).
- Fazekas, S., & Chetverikov, D. (2007). Analysis and performance evaluation of optical flow features for dynamic texture recognition. *Signal Processing: Image Communication*, 22, 680–691. Special issue on Content-Based Multimedia Indexing.
- Fujita, K., & Nayar, S. (2003). Recognition of dynamic textures using impulse responses of state variables. In *Int. workshop on texture analysis and synthesis* (pp. 31–36).
- Galun, M., Apartsin, A., & Basri, R. (2005). Multiscale segmentation by combining motion and intensity cues. In *Proc. conf. comp. vision pattern rec.* (Vol. 1, pp. 256–263).
- Golland, P., & Bruckstein, A. M. (1997). Motion from color. *Computer Vision and Image Understanding*, 68(3), 346–362.

- Hildreth, E. C. (1987). The analysis of visual motion: From computational theory to neural mechanisms. *Annual Review of Neuroscience*, 10, 477–533.
- Horn, B. K. P. (1986). *Robot vision*. New York: McGraw-Hill.
- Horn, B. K. P., & Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17(1-3), 185–203.
- Lu, Z., Xie, W., Pei, J., & Huang, J. (2005). Dynamic texture recognition by spatiotemporal multiresolution histograms. In *Proc. of the IEEE workshop on motion and video computing (WACV/MOTION)* (pp. 241–246).
- Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *DARPA image understanding workshop* (pp. 121–130).
- Mileva, Y., Bruhn, A., & Weickert, J. (2007). Illumination-robust variational optical flow with photometric invariants. In *LNCS: Vol. 4713. DAGM-Symposium* (pp. 152–162). Berlin: Springer.
- Mumford, D., & Shah, J. (1989). Optimal approximation by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42, 577–685.
- Murray, D. W., & Buxton, B. F. (1987). Scene segmentation from visual motion using global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(2), 220–228.
- Nelson, R. C., & Polana, R. (1992). Qualitative recognition of motion using temporal texture. *CVGIP: Image Understanding*, 56, 78–89.
- Nir, T., Bruckstein, A. M., & Kimmel, R. (2008). Over-parameterized variational optical flow. *International Journal of Computer Vision*, 76(2), 205–216.
- Ohta, N. (1989). Optical flow detection by color images. In *IEEE int. conf. on image processing* (pp. 801–805).
- Osher, S., & Sethian, J. A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79, 12–49.
- Otsu, N. (1979). A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9, 62–66.
- Otsuka, K., Horikoshi, T., Suzuki, S., & Fujii, M. (1998). Feature extraction of temporal texture based on spatiotemporal motion trajectory. In *Proc. int. conf. pattern recognition* (Vol. 2, pp. 1047–1051).
- Papenberg, N., Bruhn, A., Brox, T., Didas, S., & Weickert, J. (2006). Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2), 141–158.
- Paragios, N., & Deriche, R. (2005). Geodesic active regions and level set methods for motion estimation and tracking. *Computer Vision and Image Understanding*, 97(3), 259–282.
- Peh, C. H., & Cheong, L.-F. (2002). Synergizing spatial and temporal texture. *IEEE Transactions on Image Processing*, 11, 1179–1191.
- Péteri, R., & Chetverikov, D. (2005). Dynamic texture recognition using normal flow and texture regularity. In *LNCS* (Vol. 3523, pp. 223–230). Berlin: Springer.
- Péteri, R., Huskies, M., & Fazekas, S. (2006). *DynTex: A comprehensive database of Dynamic Textures*. <http://www.cwi.nl/projects/dyntex>.
- Saisan, P., Doretto, G., Wu, Y. N., & Soatto, S. (2001). Dynamic texture recognition. In *Proc. conf. comp. vision pattern rec.* (Vol. 2, pp. 58–63). Kauai, Hawaii.
- Schnörr, C. (1984). Segmentation of visual motion by minimizing convex non-quadratic functionals. In *Proc. int. conf. pattern recognition* (pp. 661–663).
- Schoenemann, T., & Cremers, D. (2006). Near real-time motion segmentation using graph cuts. In *LNCS* (Vol. 4174, pp. 455–464). Berlin: Springer.
- Schunck, B. G. (1984). The motion constraints equation for optical flow. In *Proc. int. conf. pattern recognition* (Vol. 1, pp. 20–22).
- Shi, J., & Malik, J. (1998). Motion segmentation and tracking using normalized cuts. In *Sixth int. conf. on computer vision* (pp. 1154–1160).
- Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), 888–905.
- Smith, J., Lin, C.-Y., & Naphade, M. (2002). Video texture indexing using spatiotemporal wavelets. In *Proc. int. conf. on image processing* (Vol. 2, pp. 437–440).
- Soatto, S., Doretto, G., & Wu, Y. (2001). Dynamic textures. In *Eighth int. conf. on computer vision* (Vol. 2, pp. 439–446).
- Song, S., & Leahy, R. M. (1991). Computation of 3d velocity fields from 3d cine images of a beating heart. *IEEE Transactions on Medical Imaging*, 1, 462–472.
- Sundaramoorthi, G., Yezzi, A., Mennucci, A. C., & Sapiro, G. (2007). New possibilities with Sobolev active contours. In *LNCS: Vol. 4485. Proc. SSVM 2007* (pp. 153–164). Berlin: Springer.
- Szummer, M., & Picard, R. (1996). Temporal texture modeling. In *Proc. int. conf. image processing* (Vol. 3, pp. 823–826).
- Todorovic, D. (1996). A gem from the past: Pleikart Stumpf's anticipation of the aperture problem, Reichardt detectors, and perceived motion loss at equiluminance. *Perception*, 25, 1235–1242.
- Uras, S., Girosi, F., Verri, A., & Torre, V. (1988). A computational approach to motion perception. *Biological Cybernetics*, 60, 79–97.
- Vese, L. A., & Chan, T. F. (2002). A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3), 271–293.
- Wang, J. Y. A., & Adelson, E. H. (1994). Representing moving images with layers. *IEEE Transactions on Image Processing*, 3(5), 625–638.
- Wildes, R. P., & Bergen, J. R. (2000). Qualitative spatiotemporal analysis using an oriented energy representation. In *LNCS: Vol. 1843. Proc. European conf. on computer vision* (pp. 768–784). Berlin: Springer.
- Wu, P., Ro, Y. M., Won, C. S., & Choi, Y. (2001). Texture descriptors in MPEG-7. In *LNCS* (Vol. 2124, pp. 21–28). Berlin: Springer.
- Yuan, L., Weng, F., Liu, C., & Shum, H.-Y. (2004). Synthesizing dynamic texture with closed-loop linear dynamic system. In *LNCS: Vol. 3022. Proc. European conf. on computer vision* (pp. 603–616). Berlin: Springer.
- Zheng, H., & Blostein, S. D. (1995). Motion-based object segmentation and estimation using the MDL principle. *IEEE Transactions on Image Processing*, 4(9), 1223–1235.
- Zhong, J., & Scarlarioff, S. (2002). *Temporal texture recognition model using 3D features* (Technical report). MIT Media Lab Perceptual Computing.