Pattern Recognition Letters 32 (2011) 1982–1989

Contents lists available at SciVerse ScienceDirect

Pattern Recognition Letters

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

A method for mixed states texture segmentation with simultaneous parameter estimation

A. Mailing *, B. Cernuschi-Frías

Engineering Faculty of the University of Buenos Aires, Argentina
Argentinian Mathematics Institute of CONICET, Argentina

ARTICLE INFO

Article history:
Received 31 December 2010
Available online 12 August 2011
Communicated by G. Sanniti di Baja

Keywords:
Motion textures
Segmentation
Expectation Maximization
Pseudo-likelihood
Markov Random Fields

ABSTRACT

In this work a method for mixed-state model motion texture segmentation and parameter estimation is presented. We use the Expectation Maximization algorithm for mixture parameter estimation, introducing the Gibbs distribution for moving points, excluding zero discrete component associated with no motion regions. We use then the *a posteriori* probabilities to generate an alternative field to segment the textures according to its statistical parameters.

1. Introduction

Image processing is commonly intended to obtain information that allows to take a decision without any human intervention, even when dealing with misinterpretations inherent to automated *understanding* or *classification*. When we apply a process to a real world image or an image sequence it is not always wanted that automated systems behave as human beings do, but perhaps take advantage of its precision and integration capabilities to make it behave better. We are particularly interested in motion textures obtained from certain sequences of images, which depict not only spatial but also time-spatial information. Motion textures are characterized by the presence of diffused motion distributed over a portion of the image. Here the pixel motion value represents a *motion interaction measure* of an interacting *physical particle*. This differs from the usual understanding of motion in images where each pixel has associated the velocity measure of the motion of rigid objects (*e.g.,* car, ball, *etc.*). The motion textures are also related to dynamic textures (Doretto et al., 2003a). For a brief survey about classifications of different kinds of motion see Chetverikov and Peteri (2005).

Owing to the textured characteristic of the data we use a probabilistic framework. It is then, the first aim of this work to provide a probabilistic model that characterizes the data with its particular complexity and being simple enough to avoid inefficient and expensive computations. The behaviour of the motion textures is related to a particular type of random variables taking continuous and discrete values, *i.e.* a mixed-state random variable. This can be seen as a mixture of a probability density function and a discrete probability mass. This mixed-state nature is present in motion textures due to the presence of repetitive values *e.g.,* the motion of the background or large objects. However, the task of distinguishing between the different properties of these motion textures when they are diffused, as well as the estimation of the several parameters involved by the model, is still considerably difficult. The motion textures are a representation of the interaction between particles, making the Markov Random Fields (MRF) a reasonable choice for that model (Chandler, 1987). A complete theoretical analysis about the use of a MRF for mixed-states random variables (*Mixed States-Markov Random Fields*) can be found in (Cernuschi-Fías, 2007 and Crivelli et al., 2007). Some related works on mixed-state texture segmentation can be found in (Crivelli et al., 2006b and Crivelli et al., 2006a) (see below). In the recent work (Crivelli et al., 2010) the conditional KullbackLeibler divergence between mixed-states distributions was used to introduce a motion texture tracking strategy. See also Crivelli et al. (2007, 2010) for additional applications of this model.

The problem of pixel classification previously mentioned, is in image processing commonly referred to as *segmentation*. More precisely, when we talk about *image segmentation* we refer to the partitioning of the image into different regions generally associated to different objects. The segmentation of textures (not necessarily obtained from motion) is in general more complicated (Doretto et al.,

* Corresponding author. Address: IAM-CONICET, Saavedra 15 3er piso, (1083) Ciudad de Buenos Aires, Argentina. Tel./fax: +54 11 4954 6781.
E-mail addresses: amailing@fi.uba.ar, agustin.mailing@gmail.com (A. Mailing), bcf@ieee.org (B. Cernuschi-Frías).

0167-8655/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2011.07.022

2003b) than segmentation of rigid objects images because the common characteristics we look for in each texture are given as a statistical relation between points. Moreover, different motion textures could not be necessarily associated to different objects, and motion textures do not have to be neither connected nor piecewise continuous. In addition the same 'object' may show very different local motion characteristics at different places. Reciprocally, two different objects may have similar local motion characteristics.

The problem of *motion texture* segmentation has been dealt, in the context of mixed-states random processes, from different approaches. Different techniques (Crivelli et al., 2007) have also been applied for segmentation. In (Crivelli et al., 2006a,b) a parametric mixed model (auto-model Bouthemy et al., 2005) is implemented and the segmentation is achieved as a *maximum a posteriori* (MAP) estimation for the *label field* of the sites. Whereas in (Crivelli et al., 2006b) the segmentation was obtained using a *graph-cut* energy optimization method. In (Crivelli et al., 2006a) it was carried out by *simulated annealing*. Mixed states random fields and motion textures have also been used (Crivelli et al., 2008) for recognition of dynamic video and background reconstruction.

Here, we use a parametric probabilistic model to represent the interactions between the sites in the textures. When we try to carry out the segmentation through the identification of the different patterns, based on the parameters of the model, we need to previously estimate those parameters. However, to do a proper estimation of the parameters, we should know to which class each site belongs, meaning that we need to previously perform the segmentation. Consequently, we fall in a circular problem.

We propose to use the iterative procedure Expectation Maximization (EM) to deal with the problem of simultaneous segmentation and parameter estimation. This algorithm was developed by Dempser et al. (1977) and has been used for dealing with a variety of problems in topics including: imaging (Fossati et al., 2008), medical imaging (Raheja et al., 1999), language processing (Wen et al., 2007), machine learning (Bailey and Elkan, 1995), *etc.*

The strength of the EM relies on the assumption that the samples are drawn according to a marginal of the joint distribution of the *samples* and the *hidden variables*. Thus, it can be proved (Neal and Hinton, 1998) that the algorithm aims to minimize the cost of 'explaining' the samples set with a wrong hidden variables distribution.

In this way, together with the proposed probabilistic model for the motion textures and the EM algorithm we present a method that achieves the segmentation and the parameter estimation of these textures, simultaneously. Whereas the previously cited works (Crivelli et al., 2006a,b, 2007, 2010) deal with complex models, here we use a simpler model with very intuitive parameters that allows to focus on the problem of segmentation. In this way, we replace the computing complexity of assuming non-conditional-independence by adding an agglomerating field to the model. This makes easier the task of introducing any available *a priori* knowledge about the dispersion of the texture, one may have. In (Liang et al., 1994, Zhang et al., 2001) similar approaches were presented using EM for segmentation of MRF image models, where the assumption of piecewise contiguity plays a fundamental role. It should be strengthened that piecewise contiguity is not usually valid for motion textures and that the method presented here does not require that condition.

2. Motion measurement texture

A video sequence can be thought as a frame sequence indexed in time and space. The motion is given by the intensity changes at each site from one frame to the other. If we measure the intensity variation between frames at each site and build a map with this information obtain the motion map. This map shows at given position the influence, not only from the motion of the seen objects, but also from the intensity gradient at that site and its corresponding variation in time. Therefore, there are several different ways to build that motion map with different characteristics. Here, we follow (Crivelli et al., 2006a,b) and also the related previous work (Bouthemy et al., 2005), where the vectorial expression for the normal flow as a local motion measurement is considered. Let $I_i(\boldsymbol{u}_i, t)$ is the scalar intensity at the point indexed by i with coordinates $\boldsymbol{u}_i$ at a time t. The temporal variation of the intensity for this pixel must be null according to Horn and Schunck (1981). Then,

$$\frac{\mathrm{d}I_i(\boldsymbol{u}_i,t)}{\mathrm{d}t} = \frac{\partial I_i(\boldsymbol{u}_i,t)}{\partial t} + \left\langle \nabla_{\boldsymbol{u}_i} I_i(\boldsymbol{u}_i,t), \frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t} \right\rangle = 0, \tag{1}$$

where $\nabla_{\boldsymbol{u}_i} I_i(\boldsymbol{u}_i,t)$ and $\frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}$ are the spatial intensity gradient and the vector velocity at the pixel $\boldsymbol{u}_i$ respectively, and where $\langle \cdot, \cdot \rangle$ denotes the inner product. From the previous equation, the velocity projection in the direction of the intensity gradient, is given equivalently by,

$$\boldsymbol{x}_i = \left\langle \frac{\mathrm{d}\boldsymbol{u}_i}{\mathrm{d}t}, \boldsymbol{J}_i \right\rangle \boldsymbol{J}_i = -\frac{\partial I_i(\boldsymbol{u}_i,t)}{\partial t}\boldsymbol{J}_i, \tag{2}$$

where the normalized intensity gradient has been defined as,

$$\boldsymbol{J}_i = \frac{\nabla_{\boldsymbol{u}_i} I_i(\boldsymbol{u}_i,t)}{\|\nabla_{\boldsymbol{u}_i} I_i(\boldsymbol{u}_i,t)\|}. \tag{3}$$

Taking a weighted average over a small set of points $\mathcal{W}_i$ around the pixel $\boldsymbol{u}_i$ to avoid noisy measurements, we define

$$\tilde{\boldsymbol{x}}_i(\mathcal{W}_i,t) \doteq \frac{\sum_{\boldsymbol{u}_j \in \mathcal{W}_i} \boldsymbol{x}_j \|\nabla_{\boldsymbol{u}_j} I_j(\boldsymbol{u}_j,t)\|^2}{\sum_{\boldsymbol{u}_j \in \mathcal{W}_i} \|\nabla_{\boldsymbol{u}_j} I_j(\boldsymbol{u}_j,t)\|^2}. \tag{4}$$

The weights are given by the squared norm of the gradient at each point, since larger gradients usually give us more information than smaller ones. Additionally, a constant in the denominator is added in practice to avoid null gradients (Crivelli et al., 2006a). Therefore, as the averaged *observable* velocity is in the direction of the intensity gradient, its expression can be written as,

$$\tilde{x}_i^o(\mathcal{W}_i,t) = \langle \tilde{\boldsymbol{x}}_i(\mathcal{W}_i,t), \boldsymbol{J}_i \rangle. \tag{5}$$

As we said before, we are interested in a class of motion characteristic of large sets of interacting particles, where the motion is homogeneous both in time and space. Such is the case of trees or grass blown by wind (sample sequence I, Fig. 2(a)), steam, smoke, water running (sample sequence II, Fig. 2(b)), crowds of people, *etc.*

This motion variety usually does not show a one-to-one correspondence between pixels and the 'physical particles', moreover, we commonly have the interaction of several particles represented (averaged) by a single pixel.

The motion measure in a pixel can be as diverse as the factors having influence on it, such as the gradient direction, how the motion is projected on it or the motion of its neighbors. Hence, the moving points behave as a random variable with a continuous distribution showing dependence on its neighbors. On the other hand, all the *no moving points* Fig. 1 have null motion value and therefore they can be associated to a discrete distribution.

2.1. A Gibbs distribution model for the moving points

Once the motion texture has been obtained using (5) from the sequence of frames, the resulting field (*i.e.* the sample set) has to be segmented without supervision. With this in mind, a probabilistic

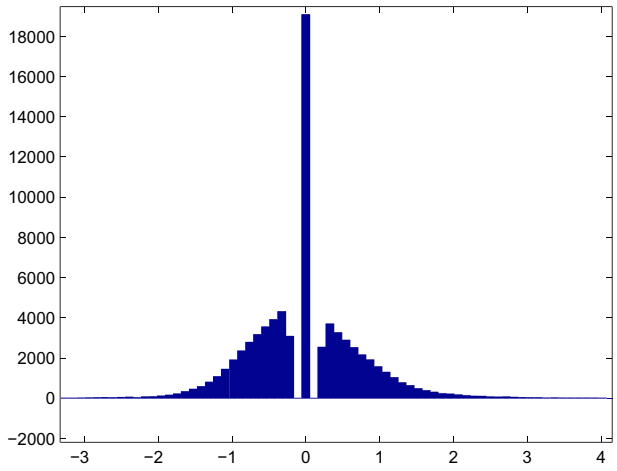


Fig. 1. Typical histogram for the pixel value on motion textures.

model for the resulting texture is introduced to relate the segmentation of the texture with the estimation of its parameters.

Let $\widetilde{\mathcal{X}}$ be the random field (namely the *motion texture*) resulting after applying (5) to the sequence of images, and let $\mathcal{X} = \{\mathcal{X}_i\}_{i=1\cdots n}$ be the subset of $\widetilde{\mathcal{X}}$ containing the sites that have non-null motion (*moving points*). Now, as usual (Besag, 1974) we assume a locally dependent model for the conditional distribution of the *moving points*. Therefore, the conditional density function for a *moving point* depends only on its neighborhood,

$$p_{\mathcal{X}_i|\mathcal{X}\setminus\{\mathcal{X}_i\}}(\boldsymbol{x}_i) = p_{\mathcal{X}_i|\mathcal{N}_i}(\boldsymbol{x}_i) \quad \forall \boldsymbol{x}_i \in \mathcal{X}, \tag{6}$$

where $\mathcal{N}_i = \{\mathcal{X}_j\}_{j\in\mathcal{N}_i}$ denotes the subset of $\mathcal{X}$ whose points are neighbors of the moving point $\mathcal{X}_i$.

As the purpose is to use a simple model, the estimation of the parameters for the moving points is done separately from the *no moving points*. Otherwise, the strong influence of the discrete mass (*no moving points*) would introduce a *very strong bias*, resulting in undesirable results, see Appendix. By avoiding the discrete component the resulting sample set has as many 'spatial holes' as *no moving* points there are in the original texture. Lets see how the spatial information excluded will be filled and how to deal with the *moving points* neighboring the 'holes'. For the sake of clarity, we first explain the adopted model.

Here we use a Gibbs distributions to write (6) taking in account the interaction between moving points in the texture,

$$p_{\mathcal{X}|\mathcal{N}}(x_i|\mathcal{N}_i;\psi) = \frac{1}{Z_i}\cdot e^{-U_i(x_i|\{x_j\}_{j\in\mathcal{N}_i},\psi)}, \tag{7}$$

where the normalizing factor Z_i can be written as,

$$Z_i = \sum_{x_i} e^{-U_i\left(x_i|\{x_j\}_{j\in\mathcal{N}_i},\psi\right)}. \tag{8}$$

The functions U_i and Z_i are usually called energy function and partition function, respectively (Chandler, 1987).

As in (Crivelli et al., 2006a,b) it seems reasonable to suppose Gaussianity for conditional densities. Then, we simplify the analysis by writing,

$$U_i(x_i) = \sum_{m=1}^{M} f_m^2(x_i, \boldsymbol{x}_m(\mathcal{N}_i))\cdot g_m(\boldsymbol{x}_m(\mathcal{N}_i)), \tag{9}$$

where f_m and g_m are two functions related to the neighbors of x_i, and some parameters ψ to be described below, and f_m is also a function of x_i. Note that m is indexing the M possible vectors $\{\boldsymbol{x}_m(\mathcal{N}_i)\}_{m=1\cdots M}$ whose elements are neighbors of x_i. Here, for the sake of simplicity, g_m will be a positive function and f_m will be a linear function of x_i and its neighbors depicted by the vector $\boldsymbol{x}_m(\mathcal{N}_i)$.

We suppose that there are more than one texture class, meaning that each point can follow different distributions. This property is denoted by $x_i \in \Omega_j$ where Ω_j is a motion texture class. Then from (7) and (9), assuming functional independence between classes for the parameters ψ_j, we can write the conditional density for x_i, given that belongs to the texture class Ω_j,

$$p_{\mathcal{X}|\mathcal{N}}(x_i|\mathcal{N}_i, x_i\in\Omega_j,\psi_j) = \prod_{m=1}^{M} \frac{e^{-\frac{1}{2}f_m^2\left(x_i,\boldsymbol{x}_m(\mathcal{N}_i),\psi_j\right)g_m\left(\boldsymbol{x}_m(\mathcal{N}_i),\psi_j\right)}}{\sqrt{\frac{2\pi}{g_m\left(\boldsymbol{x}_m(\mathcal{N}_i),\psi_j\right)}}}. \tag{10}$$

Here, we propose the following specific model because it is accurate enough for the data we are dealing with, and it is simple and flexible with respect to we need to do,

$$p_{\mathcal{X}|\mathcal{N}}(x_i|\mathcal{N}_i, x_i\in\Omega_j,\psi_j) = \prod_{m\in M_i}\sqrt{\frac{g_{m,j}}{2\pi}}e^{-\frac{1}{2}\left(x_i-\boldsymbol{x}_m^T\cdot\boldsymbol{a}_{m,j}\right)^2 g_{m,j}}. \tag{11}$$

Thus, each vector $\boldsymbol{a}_{m,j}$ relates the influence of each *neighbor-group* on the mean of the distribution for x_i, depending on the *neighbor-group* index m and the texture class j. Since $g_{m,j}$ is considered here as a constant for each pair m, j it can be placed into the parameter set ψ_j as well as $\boldsymbol{a}_{m,j}$.

A pseudo-likelihood approximation is used to estimate the parameters of the model as the interaction between the neighbors prevents us from writing the true likelihood in a convenient manner. It would be also possible to separate the points and neighbors into two disjoint sets and then write the true likelihood as the product of the conditional probabilities, however this would result in a loss of precision on the estimation owing to the loss of sample data. A discussion about this problem can be found in (Besag, 1974, 1975, where it is proposed to write the pseudo-likelihood for the texture under the assumption of conditionally independent samples. Under this assumption we can now write the pseudo-likelihood approximation of the likelihood for any subset of points belonging to the class Ω_j as,

$$p(A|\mathcal{N}_A, A\subset\Omega_j;\boldsymbol{\psi}_j) = \prod_{x_i\in A} p_{\mathcal{X}|\mathcal{N}}(x_i|\mathcal{N}_i, x_i\in\Omega_j;\psi_j). \tag{12}$$

Hence, considering $x_i \in \Omega_j$ as a hidden variable,

we would be able to apply the EM algorithm for segmentation and parameter estimation if we were able to assume piecewise contiguity (similar to Liang et al., 1994). However, as we are not taking into account the *no moving points* for the estimation, wide empty spaces between little moving regions are allowed.

Even when it can be thought that this represents a minor problem, the disintegration of the regions may result in a poor estimation of the parameters. Then, using small neighborhoods (with a few parameters) results insufficient to distinguish between similar textures. On the other hand, by using a strategy that increases the neighborhood size as much as necessary, the result becomes texture-dependent, because for more spaced moving regions bigger neighborhoods would be necessary. We solve this problem by introducing a new field $\mathcal{Z}_j|\{x_i\in\Omega_{j,i}\}_{\forall x_i\in\widetilde{\mathcal{X}}}$ representing the smoothed or diffused *a posteriori* probability for the region belonging to each texture class. Therefore propose the following model,

$$p_{\mathcal{X}\mathcal{Z}|\mathcal{N}}(x_i,z_i|\mathcal{N}_i, x_i\in\Omega_j) = p_{\mathcal{X}|\mathcal{N}}(x_i|\mathcal{N}_i, x_i\in\Omega_j)\cdot p_{\mathcal{Z}}(z_i|x_i\in\Omega_j), \tag{13}$$

where $p_{\mathcal{X}|\mathcal{N}}$ is as in (11) and $p_{\mathcal{Z}}$ follows a beta distribution, that is,

$$p_{\mathcal{Z}}(z_i|x_i\in\Omega_j) = \frac{z_i^{\alpha-1}\cdot(1-z_i)^{\beta-1}}{B(\alpha,\beta)}. \tag{14}$$

The parameter β must be equal to one and α grater than one in order to make $p_{\mathcal{Z}}$ an increasing function of z_i given $x_i \in \Omega_j$. We then obtain a reasonable model where, given that $x_i \in \Omega_j$, the variable $\mathcal{Z}$

is more probably to be close to one. Thus $\mathcal{Z} = \{\mathcal{Z}_i = [Z_{i,1} \cdots Z_{i,c}]^T\}$ is a vector field where each $\mathcal{Z}_i$ has c components for c motion texture classes, and the sum over the components is one.

3. Using Expectation Maximization for segmentation and parameter estimation

As said before, we propose to use the EM algorithm (Dempser et al., 1977; Xu and Jordan, 1996) to segment and estimate the parameters for (10) assuming c classes of textures. Given a sample set, the EM algorithm assumes that it follows a marginal distribution of a joint distribution of the samples and hidden variables. Where the hidden (or unknown) variables are related to the samples that 'cannot be sampled'. As can be seen in (Neal and Hinton, 1998), the EM algorithm can be thought as a method to minimize the cost of 'understanding' the sample set with a wrong assumption over the distribution of the hidden random variables, *i.e.* the cross-entropy of the joint *samples-hidden variables*. In this work we want to 'understand' the texture sample set given by model, finding as well as possible, the class assignation for the sites (the segmentation).

Therefore, EM seems a reasonable method for improving iteratively the estimation of the parameters of the model, and at the same time improve the segmentation.

The class estimation problem (or segmentation) over the texture sample set, can be seen as a mixture distribution with parameter $p_{x\in\Omega_c}$ for x. With this in mind, from the previous section, the parameters of the model are,

$$\boldsymbol{\psi} = \left\{ \{\boldsymbol{a}_{m,1}, \ldots, \boldsymbol{a}_{m,c}; g_{m,1}, \ldots, g_{m,c}\}_{m=\{1\cdots M\}}; p_{x\in\Omega_1}, \ldots, p_{x\in\Omega_c}; \alpha_1, \ldots, \alpha_c \right\} \tag{15}$$

From the standard EM theory we obtain the following expression to maximize with respect to $\boldsymbol{\psi}$ for the mixture distribution parameter estimation (Duda et al., 2000),

$$Q(\boldsymbol{\psi}, \boldsymbol{\phi}) = \sum_{k=1}^{N} \sum_{j=1}^{c} p(x_k \in \Omega_j | x_k, z_k; \phi_j) \cdot \ln\left(p_{\mathcal{X}\mathcal{Z}|\mathcal{N}}(x_k, z_k | \mathcal{N}_k, x_k \in \Omega_j, \phi_j; \psi_j) \cdot p_{x\in\Omega_j} \right), \tag{16}$$

and the additional constraint,

$$\sum_{j=1}^{c} p_{x\in\Omega_j} = 1 \qquad \forall x \in \mathcal{X}. \tag{17}$$

The parameter set ϕ is fixed from the previous EM iteration.

Therefore, by maximizing (16) using Lagrange multipliers we obtain closed expressions for both $\boldsymbol{a}_m$, g_m and also probability $p_{x\in\Omega_j}$ for a site i to belong to a class labeled j. This results in,

$$\boldsymbol{a}_{m,i} = \frac{\sum_{k=1}^{N} p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi}) \cdot x_k \cdot \boldsymbol{x}_{m,k}}{\sum_{k=1}^{N} p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi}) \cdot \boldsymbol{x}_{m,k} \boldsymbol{x}_{m,k}^T},$$

$$g_{m,i} = \frac{\sum_{k=1}^{N} p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi})}{\sum_{k=1}^{N} p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi}) f_m^2(x_k, \boldsymbol{a}_i)},$$

$$\alpha_i = \max\left\{ \frac{-\sum_{k=1}^{N} p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi})}{\sum_{k=1}^{N} p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi}) \ln(z_k)}, 1 \right\},$$

$$p_{x\in\Omega_i} = \frac{1}{N} \sum_{k=1}^{N} p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi}),$$

where $p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi})$ is calculated using (13). Here, $\boldsymbol{x}_{m,k} = \boldsymbol{x}_m(\mathcal{N}_k)$ in order to simplify the notation. It should be noted that there is a constraint over α_i for the purpose of making beta an increasing function over the interval $z_k \in [0,1]$, this is $\alpha \geqslant 1$. For faster convergence we can constrain α_i to be greater than two, obtaining a beta function which is increasing and also concave.

After each EM iteration new parameters estimation are obtained. With these parameters we can calculate the new *a posteriori* probabilities for each site and for each class. Then, the $\mathcal{Z}$ field is obtained from the smoothing of the estimation of this *a posteriori* probabilities, behaving as a cohesive parameter for each algorithm iteration. This procedure allows to calculate the segmentation over the whole set of samples $\widetilde{\mathcal{X}}$, as it is explained in the next section. Note that although a more complex model could be used for $\mathcal{Z}$, as the no-moving points may not have statistical properties to take advantage, it would be unlikely to gain benefits in the estimation process. Moreover, the computation of $\mathcal{Z}$ on each step would be considerably more complex.

3.1. Segmentation and estimation problem

As has been noticed, the idea behind applying the EM algorithm is not only to achieve the parameter estimation but also to obtain the *a posteriori* probabilities, *i.e.* $p(x_k \in \Omega_i | x_k, z_k; \boldsymbol{\phi})$. These can be interpreted as a measure of the degree of membership to the ith motion textures class for the kth site. With this in mind we can use the *a posteriori* probabilities to make a *pre-segmentation* of the motion texture, generating a statistical similarity map from which we can obtain information for the segmentation step.

In conventional *intensity textures* we usually have several large almost-stationary connected regions, for example, the still images Fig. 2(a) and (b). However, when we deal motion textures we have a very dispersed texture among a large quantity of zeroes (no moving points). This can be noticed in Fig. 3(a) and (b) and is highlighted in Figs. 4(b) and 5(b). This means that we have an important amount of moving points having no-moving neighbors. In practice this would disturb the model (11) behavior and would

(a) Two trees being blown by wind(Test Sequence I). (b) A river running and bushes moved by wind(Test Sequence II).

Fig. 2. Images from two different video sequences used for the tests.

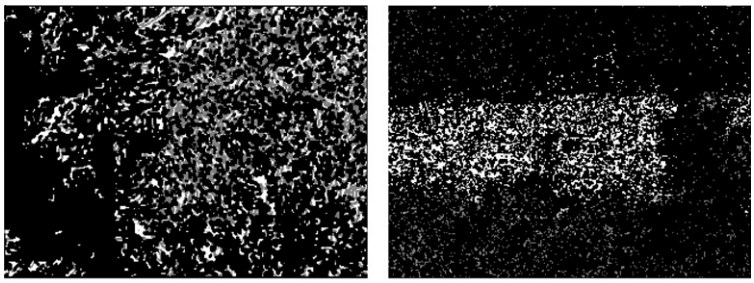
also ill-pose many single-step agglomeration terms that usually work well with intensity textures segmentation This happens as the *no moving* points belonging to the neighborhood of a site have very different statistics to the rest of the *moving* points (see Appendix). We handled this problem by removing this points from the estimation step. In practice, when the *moving points* are very dispersed and an eight neighbors scheme is used, too many points are lost in the assumption of a set without *no-moving points* nor *no-moving neighbors*. Then, accepting points that have *no-moving points* as, at most half of its neighbors, shows satisfactory results in most of the cases.

As the whole texture needs to be segmented, the segmentation cannot be accomplished from $p(x_k \in \Omega_i|\boldsymbol{\phi})$ as it would be expected, since it is not defined on the *no-moving points*. Instead, it is obtained by thresholding the resulting (smooth) $\mathcal{Z}$ field. This can be seen in Figs. 4(a) and 5(a).

4. Experimental results

In practice noisy measurements are obtained if some truncations are not applied in the calculation of (5) (see Fig. 1). This may produce an unexpected behaviour of the EM algorithm, applied to the model (11), when an independent term is added to f_m, *i.e.*, $\boldsymbol{x}_m = [1, x_m^1, \ldots, x_m^l]$. Then, we assume no global motion is present and so $\boldsymbol{x}_m = [x_m^1, \ldots, x_m^l]$. This does not imply an important loss of generality as there exist many techniques for its estimation and removal, see (Wang and Wang, 1997) for example.

For the initialization of the EM algorithm $p(x_k \in \Omega_i|x_k)$ was initialized randomly according to a zero-one uniform distribution, and normalized to meet $\sum_i p(x_k \in \Omega_i|x_k) = 1$. This choice may result in a slow speed of convergence of the algorithm, however, the initialization of ϕ requires previous knowledge of the parameters, owing to the wide possible values depending on the texture and



(a) The *a posteriori* for test seq.(I) (b) The *a posteriori* for test seq.(II)

Fig. 3. The *a posteriori* probability map for 20 iterations of the algorithm. In (a) the *Left Tree*(*I*) pixels are classified as white, the *Right Tree*(*I*) as gray, and *no motion* as black. In (b) the *River*, the *Bushes* and *no motion* pixels are classified as white, gray and black, respectively.

(a) Final segmentation

(b) Expected segmentation. The large zones without any motion are actually unclassifiable from their motion characteristics. They are marked in gray colour.

Fig. 4. Final segmentation for the sequence 'two trees being blown by the wind' (Fig. 2(a), done exclusively from its motion texture field and without any supervision.)

(a) Final segmentation

(b) Expected segmentation. The large zones without any motion are marked in gray colour.

Fig. 5. Final segmentation (a) for the sequence 'river running and bushes moved by the wind' (Fig. 2(b). It is done exclusively from its motion texture field and without any supervision.)

Table 1
Parameters $\boldsymbol{a}_{j,m}$ for the sequences I and II.

Class	a_1	a_2	a_3	a_4
Left Tree (I)	1.0244	0.9936	1.0853	0.9482
Right Tree (I)	0.6780	0.6736	0.6428	0.5982
River (II)	0.3497	0.3320	0.7883	0.7994
Bushes (II)	0.4727	0.4598	0.5585	0.5301
Class	a_5	a_6	a_7	a_8
Left Tree (I)	0.8903	1.0323	1.0078	0.9256
Right Tree (I)	0.4682	0.5045	0.5411	0.5052
River (II)	0.3229	0.3269	0.3153	0.3417
Bushes (II)	0.2358	0.2714	0.3741	0.3657

Table 2
Parameters $g_{j,m}$ for the sequences I and II.

Class	g_1	g_2	g_3	g_4
Left Tree (I)	32.5273	33.0725	28.4778	26.9319
Right Tree (I)	2.9366	2.8623	2.6966	2.5016
River (II)	1.3120	1.3010	2.7194	2.7698
Bushes (II)	3.9437	3.8928	4.3543	4.2087
Class	g_5	g_6	g_7	g_8
Left Tree (I)	8.3550	8.7467	10.2006	10.1727
Right Tree (I)	2.0646	2.1481	2.2278	2.1600
River (II)	1.2838	1.2835	1.2790	1.2935
Bushes (II)	3.2847	3.3337	3.5282	3.5225

Table 3
Priors and α for the sequences I (estimated α) and II (fixed α)

Class	α	$p_{x\in\Omega}$
Left Tree (I)	1.3245	0.4093
Right Tree (I)	1.3469	0.5907
River (II)	6	0.6743
Bushes (II)	6	0.3257

model. Moreover, a wrong initialization of ϕ usually becomes into undesirable results.

As can be seen in Fig. 3(a) and (b), if the segmentation is performed directly from the *a posteriori* probability map, it would arise the problem of the missing information of the no-moving points. If an agglomerating field is not included, reaching satisfactory results just from the parameter estimation would turn out to be almost impossible.

Here we used an eight neighbors scheme numbered from one to eight as: north, south, west, east, south–east, north–west, south–west and north-east respectively. Then, from the estimated parameters and the segmentation the following can be observed. On *River*(*II*) from Tables 1 and 2 it can be noticed that the points are more correlated (higher $a_{j,m}$ and $g_{j,m}$) to its horizontal neighbors. On *Left Tree*(*I*) higher values of the first four components of $\boldsymbol{g}$ indicate that an even motion is being observed. Some erroneous class assignment can be observed on the segmentation, in the top-left quadrant of Fig. 4(a) and in the top-right quadrant of Fig. 5(a), due to the fact that not much information available in the regions with not much motion (highlighted in Figs. 4(b) and 5(b)).

To make a comparison, we can see that similar segmentation results are achieved in (Crivelli et al., 2006b) for the sequence (I), were the segmentation is obtained using a different model (mixed-state auto-model) and the MAP optimization is done through a graphical method (graph-cut). It should be noticed that the model used here is much simpler than the one used in (Crivelli et al., 2006b).

The method shows a weakness when a texture is too dispersed *i.e.*, has large zero regions with very little neighborhoods with motion. In this case the best experimental results have been achieved forcing α to be between 5 and 7 (see Table 3). For these values of α, the beta distribution is concave enough to make the better samples (those samples which have more certainty) to have more weight during the estimations. Then the "class frontier" on $\mathcal{Z}$ between regions is more defined and convergence is faster, see Fig. 3(b) and Fig. 5(a).

However, fixing the parameter α has to be done with care because too much sharpened beta distributions may cause the *first guess* to be strong enough to avoid proper *a priori* probabilities estimation. For the first iteration it seems convenient to make α close to one in order to let the $\mathcal{X}$ field lead the EM to early convergence. For some motion textures, having primarily zeroes with little motion regions, the resolution used for the measurements becomes important owing to the significance of the neighborhood information related to each moving point.

In the example shown in Fig. 6, a texture was generated synthetically using a Metropolis sampler (Geman et al., 1984; Metropolis et al., 1953). The parameters that maximize the likelihood for each texture are given in Table 4(a). For the synthesis and the estimation, a four neighbors model (N–S–E–W) was used. Two classes of textures were generated and combined (see Fig. 6(a)) to form the 200 × 400 two-class texture Fig. 6(b). Then, near 23000 (≈30%) of

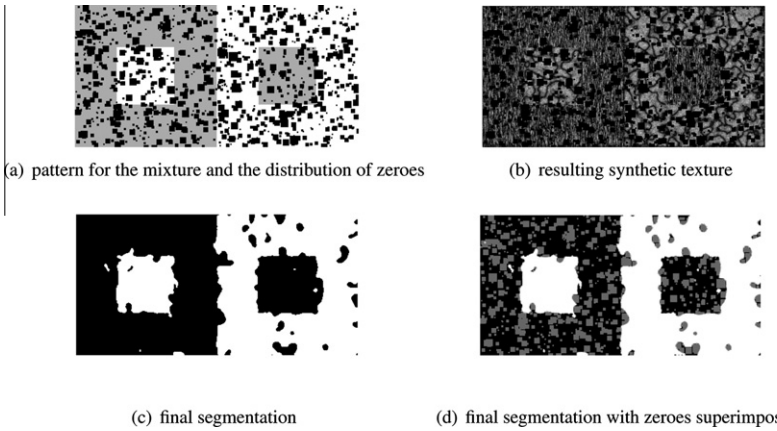


Fig. 6. A synthetic two-class texture of 200 × 400 pixels generated according to the model given in (7) with parameters given in Table 4(a). About 30% of the samples are zero, distributed randomly as small blobs. In (a), (b) and (c) are shown the pattern used for the mixture, the resulting texture and the resulting segmentation, respectively. In (d) the zeroes are superimposed to the segmentation to show their relation with the misclassified sites.

Table 4
In (a) the ML parameters for each textures class are given, and in (b) the parameters estimated simultaneously with the segmentation are shown in Fig. 6(c).

	a_1	a_2	g_1	g_2	p
(a) Parameters of each texture class					
Class_1	0.49	0.12	2.23	0.27	0.5
Class_1	0.49	0.11	1.97	0.40	0.54
(b) Resulting estimated parameters					
Class_2	0.5	0.5	3.18	3.18	0.5
Class_2	0.50	0.50	1.85	1.78	0.45

Table A.5
Discrete component in a simple example. A $N = 10000$ sample vector is generated according to parameters form (a) and then about 500 samples are set as zero. In (b) the parameters used in Table A.6 are shown.

	Class_1	Class_2
(a) Parameters used in sample generation N = 1e + 4		
μ	−1	1
σ	0.5	0.2
a	0.2	0.2
p	0.5	0.5
$x_k \to 0$	≈500	
(b) Some parameters that increases the likelihood		
μ	0	0
σ	**1e−20**	1.2
a	0.2	0.2
p	0.4	0.6

Table A.6
Discrete component in a simple example.

Parameters used	Log-likelihood original sample	Log-likelihood modified sample
Table A.5(a)	−209	−1200
Table A.5(b)	−6000	+15000

the pixels were set to zero. In order to be more realistic the zeroes were introduced as 1000 blobs of random size, in such a way that the average blob size is 5×5 pixels. The estimation converged in approximately 20 iterations reaching the segmentation shown in Fig. 6(c) with parameters given in Table 4(b). Some differences appear in the g_k parameters but it is not large if they are seen in terms of the standard deviation (*i.e.* for the class_2 $\sqrt{\frac{1}{g_1}} = 0.56$ vs. 0.73). In (d) the zeroes are superimposed to the final segmentation to show the effect of large zero regions. As the algorithm does not have a global view of the texture shape, some regions with large quantity of zeroes, mainly in the boundaries between textures, may be misclassified as they may belong to any of the classes.

5. Conclusion

We have explained briefly the motion texture measurement and its mixed-state nature. Then, we proceeded to suggest a model that allows the parameter estimation with simultaneous segmentation through using EM-based algorithm. Because a classic approach (12) does not work owing to the mixed nature of the textures, we have introduced a new field(13) taking advantage of the EM functional structure. We also proposed an easy way for a problem-independent initialization of the algorithm.

Acknowledgements

The authors thank the useful suggestions given by the two anonymous reviewers and would like to thank to the Universidad de Buenos Aires, CONICET and ANPCyT, Ministerio de Ciencia y Tecnica, Argentina for the financial support.

Appendix A. The discrete component effect in a simple example

Here, the effect of a discrete component in the sample distribution is shown for the ML estimation of a simple model. A one-dimensional *causal* model is used for this illustrative example, avoiding the need to estimate the likelihood, since it can be calculated exactly for any given sample set.

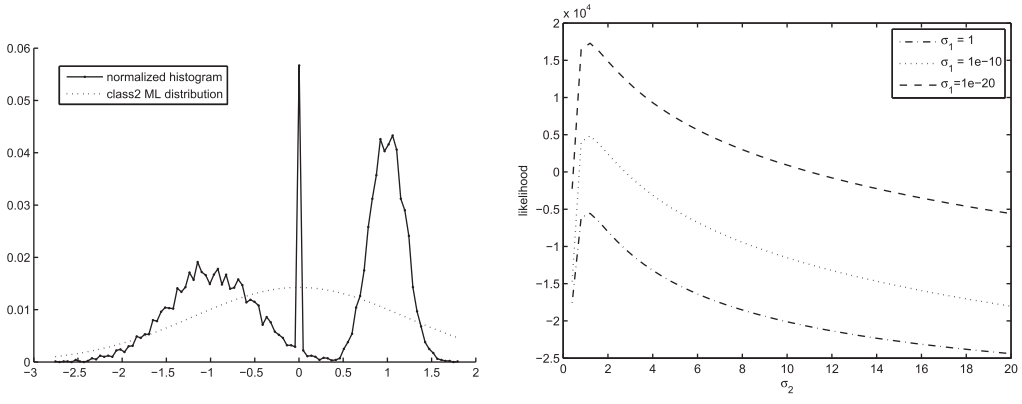
The following distribution gives the dependence of each sample from the previous sample,

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \{a, \mu, \sigma\}) = \frac{e^{-\frac{1}{2}\left(\frac{x_k - a x_{k-1} - \mu}{\sigma}\right)^2}}{\sqrt{2\pi}\sigma}. \tag{A.1}$$

Then the mixture for the typical two-classes problem is,

$$p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) = p_1 p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \psi_1) + (1 - p_1) p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}, \psi_2), \tag{A.2}$$

where the parameter vectors, $\psi_j = \{a_j, \mu_j, \sigma_j\}$, are as usual. Then, assuming $p(\boldsymbol{x}_0)$ known, the likelihood for the sample set can be written as,



(a) $\boldsymbol{x}_k - a\boldsymbol{x}_{k-1}$ histogram (b) likelihood vs σ_2 for different values of σ_1

Fig. A.7. A sample vector of $N = 10000$ points is generated according to the causal model (A.2) with parameters given in Table A.5(a). About 500 zeroes are introduced randomly as bursts. In Fig. A.7(a) the modified histogram is shown and the class_2 ML-estimated has been overlapped.

$$p(X) = p(\boldsymbol{x}_0)\prod_{k=1}^{N} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \tag{A.3}$$

In this way, though different, the model can be thought as a simpler case of the model proposed in Section 2.1.

Next, a sample set is drawn according to (A.2) using the parameters given in Table A.5(a), and then, some samples (we used about 5% but could be less) are changed to zero. This is done randomly, in such a way that some of them appear consecutively. So that, some little neighborhoods of zero-valued samples play the role of no-moving points in the textures, as they usually do not appear isolated. This example, is intended to give an insight of the problem that the discrete component causes in the ML-estimation. With this in mind, given that K samples are zero-valued and $\mu_1 = 0$, it is easy to show that the log-likelihood becomes, (see Table A.6)

$$\begin{aligned} l(\psi_1,\psi_2) \sim & \sum_{x_k,x_{k-1}\neq 0} \log p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) \\ & + \sum_{x_k=x_{k-1}=0} \log\left(\frac{p_1}{\sqrt{2\pi}\sigma_1} + (1-p_1)p(0|0,\psi_2)\right) \qquad \text{(A.4)} \\ \approx & \sum_{x_k,x_{k-1}\neq 0} \log p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) - K\log\sigma_1, \qquad \text{(A.5)} \end{aligned}$$

for σ_1 sufficiently small. Thus, by choosing ψ_2 such that the summation of $\log p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1})$ is not too small, we can make $-K\log\sigma_1$ to be as large as we want (see Fig. A.7(b)).

References

Bailey, T.L., Elkan, C., 1995. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. In: Machine Learning, pp. 51–80.

Besag, J., 1974. Spatial interaction and the statistical analysis of lattice systems. J. Roy. Statist. Soc. 36, 192–225.

Besag, J., 1975. Statistical analysis of non-lattice data. The Statistician 24 (3), 179–195.

Bouthemy, P., Hardouin, C., Piriou, G., Yao, J., 2005. Auto-models with mixed states and analysis of motion textures, Technical Report 1682, IRISA.

Cernuschi-Fías, B., July 2007. Mixed states Markov random fields with symbolic labels and multidimensional real values, Research Report 6255, INRIA.

Chandler, D., 1987. Introduction to Modern Statistical Mechanics. Oxford University Press, USA.

Chetverikov, D., Peteri, R., 2005. A Brief Survey of Dynamic Texture Description And Recognition. In: In Soft Computing, S.A. (Ed.), Fourth Internat. Conf. Comput. Recognit. Syst. (CORES'05), pp. 17–26.

Crivelli, T., Cernuschi-Frfas, B., Bouthemy, P., Yao, J.-F., 2008. Recognition of dynamic video contents based on motion texture statistical models. In: VISAPP. Vol. 1, pp. 283–289.

Crivelli, T., Cernuschi-Frías, B., Bouthemy, P., Yao, J.-F., October 2006a. Mixed-state Markov random fields for motion texture modeling and segmentation. In: ICIP IEEE Internat. Conf. Image Process.

Crivelli, T., Cernuschi-Frías, B., Bouthemy, P., Yao, J.-F., August 2006b. Segmentation of motion textures using mixed-state Markov random fields. In: Mathematics of Data/Image Pattern Recognition, Compression, and Encryption with Applications,Vol. 6315-63150J, SPIE.

Crivelli, T., Cernuschi-Frías, B., Bouthemy, P., Yao, J.-F., September 2007. Mixed-state models in image motion analysis: Theory and applications, Research Report 6335, INRIA.

Crivelli, T., Cernuschi-Frias, B., Bouthemy, P., Yao, J.-F., 2010. Mixed-state causal modeling for statistical KL-based motion texture tracking. Pattern Recognit. Lett. 31 (14), 2286–2294.

Dempser, A., Laird, N., Rubin, D., 1977. Maximum likelihood form incomplete data via the EM algorithm. J. Roy. Statist. Soc..

Doretto, G., Chiuso, A., Wu, Y., Soatto, S., 2003a. Dynamic texture. Int. J. Comp. Vision 2 (51), 91–109.

Doretto, G., Cremers, D., Favano, P., Soatto, S., 2003b. Dynamic texture segmentation. In: Proc. 9th Internat. Conf. Comput. Vision, ICCV 03, p. 12361242.

Duda, Hart, Stork, 2000. Pattern Recognit., 2nd ed. Wiley Interscience.

Fossati, A., Arnaud, E., Horaud, R.P., Fua, P., June 2008. Tracking articulated bodies using generalized expectation maximization. In: Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (CVPR workshop, NORDIA'08).

Geman, S., Geman, D., 1984. Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. IEEE Trans. Pattern Anal. Mach. Intell. PAMI-6 (6), 721–741.

Horn, B., Schunck, B.G., 1981. Determining optical flow. Artificial Intell. 17, 185–203.

Liang, Z., MacFall, J., Harrington, D., 1994. Parameter estimation and tissue segmentation from multispectral MR images. IEEE Trans. Medical Imag. 13 (3), 441–449.

Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E., 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21 (6), 1087–1092.

Neal, R., Hinton, G., 1998. A new view of the EM algorithm that justifies incremental, sparse and other variants. In: Jordan, M.I. (Ed.), Learning in Graphical Models. Kluwer Academic Publishers., pp. 355–368.

Raheja, A., Doniere, T., Dhawan, A., 1999. Multiresolution expectation maximization reconstruction algorithm for positron emission tomography using wavelet processing. IEEE Trans. Nuclear Sci. 46 (3), 594–602.

Wang, D., Wang, L., 1997. Global motion parameters estimation using a fast and robust algorithm 7 (5), 823–826.

Wen, J., Li, Z., Zhang, L., Hu, X., Chen, H., 2007. A new method of cluster-based topic language model for genomic ir. ainaw 1, 301–306.

Xu, L., Jordan, M.I., 1996. On convergence properties of the EM algorithm for gaussian mixtures. Neural Comput. 8 (1), 129–151.

Zhang, Y., Brady, M., Smith, S., 2001. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans. Medical Imag. 20 (1), 45–57.