Mixed-state models in image motion analysis: theory and applications

Tomas Crivelli, Bruno Cernuschi-Frias, Patrick Bouthemy, Jian-Feng Yao

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Mixed-state models in image motion analysis: theory and applications

Tomás Crivelli — Bruno Cernuschi Frías — Patrick Bouthemy — Jian-Feng Yao

N° 6335

October 2007

Thème COG

Rapport de recherche

ISSN 0249-6399 ISRN INRIA/RR--6335--FR+ENG

Mixed-state models in image motion analysis: theory and applications

Toms Crivelli[*] , Bruno Cernuschi Frías[†] , Patrick Bouthemy [‡] , Jian-Feng Yao [§]

Thème COG — Systèmes cognitifs
Projet VISTA

Rapport de recherche n° 6335 — October 2007 — 40 pages

Abstract: The aim of this work is to model the apparent motion in image sequences depicting textured motion patterns. We adopt the *mixed-state* Markov Random Fields (MRF) models recently introduced to represent the so-called motion textures. The approach consists in describing the spatial distribution of local *motion measurements* which exhibit values of two types: a discrete component related to the absence of motion and a continuous part for actual measurements. The former accounts for symbolic information that is beyond the null value of motion itself, providing crucial information on the dynamic content of the scene.

We propose several significant extensions and we give theoretical results regarding this model, which are of great importance for its application to motion analysis. In this context, dynamic content recognition applications are analyzed. We have defined a motion texture classification scheme, and a motion texture segmentation method exploiting this modeling. Results on real examples demonstrate the accuracy and efficiency of our method.

Key-words: Markov random fields, dynamic textures, mixed-state models, segmentation, classification

Joint collaboration between the Equipe Associé INRIA FIM, Fluides, Images et Mouvement , CONICET, and Facultad de Ingeniería, Universidad de Buenos Aires, Argentina, and VISTA, IRISA - INRIA, Rennes.

[*] Facultad de Ingeniería, Universidad de Buenos Aires.
[†] CONICET, Argentina
[‡] IRISA - INRIA
[§] IRMAR - Université de Rennes 1

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 Rennes Cedex (France)
Téléphone : +33 2 99 84 71 00 — Télécopie : +33 2 99 84 71 71

Modeles à états mixtes pour l'analyse du movement dans des images: théorie et applications.

Résumé : Le propos de ce travail est la modélisation de mouvement dans des séquences d'images qui montrent des patrons de texture de mouvement. Nous adoptons le modéle des Champs de Markov Aléatoires à états mixtes récemment introduit, pour modéliser ce que nous appelons textures de mouvement. Cette approche consiste à faire une description de la distribution spatiale des mesures de mouvement locales, qui montrent deux types de valeurs : une composante discréte reliée à l'absence de mouvement, et une partie continue que correspond aux mesures non nulles de mouvement. La premiére composante peut être interprétée comme une information symbolique qui va au-delà de la simple valeur nulle de vélocité, et qui nous donne une information essentielle pour l'analyse du contenu dynamique de la scéne. Ici nous proposons plusieurs extensions significatives et nous présentons plusieurs résultats théoriques sur ce modéle, qui sont de grande importance pour les applications d'analyse de mouvement. Dans ce contexte, la reconnaissance de contenu dynamique est analysée. Nous définissons un schéma de classification des textures de mouvement, et une méthode de segmentation des celles-ci en utilisant ces modéles. L'efficacité et la précision de notre méthode sont démontrées avec des résultats obtenus à partir d'exemples réels.

Mots-clés : Champs aleatoires de Markov, textures dynamiques, modéle à états mixtes, segmentation, classification.

Contents

1 Introduction

Dynamic content analysis from image sequences has become a fundamental subject of study in computer vision, with the objective of representing complex real situations in an efficient and accurate way. Meanwhile, the great amount of image data to be processed requires a modeling framework able to extract a compact representation of the information.

In the context of visual motion analysis, *motion textures* are closely related to *temporal* or *dynamic textures* [16, 32, 42], firstly introduced by Nelson and Polana [32]. Different from *activities* (walking, climbing, playing) and *events* (open a door, answer the phone), *temporal textures* show some type of stationarity or homogeneity, both in space and time. Mostly, they refer to dynamic video contents displayed by natural scene elements as rivers, sea-waves, smoke, moving foliage, fire, etc. Nevertheless, they also encompass any dynamic visual information that, from the observer point of view, can be classified as a texture with motion (Fig. 1). For example, consider a walking person. This *activity* can be analyzed as attached to an articulated motion; however a group of people or crowd walking, observed from a wide angle may show a repetitive motion pattern, more adequate to be considered as a temporal texture.

By definition, a motion texture is an instantaneous motion map obtained from a dynamic texture. This recently introduced term [4], in fact, does not make reference to a new type of dynamic phenomena, but indeed, to the type of information that is processed and modeled in these classes of video sequences.

When analyzing a complex scene, the three types of dynamic visual information (activities, events and temporal textures) may be present [36]. However, their dissimilar nature leads to considering substantially different approaches for each one in tasks as detection, segmentation, and recognition.

From motion detection, to optical flow estimation [1, 6, 12, 41], efforts has been devoted to extract reliable and representative motion quantities from a sequence of images. In the last few years, retrieving, indexing, recognition and classification of large quantities of video data, seem to have entailed an increasingly interest in dynamic content analysis. In this context, motion information have been pointed out and effectively used as a key feature for qualitative content characterization [8, 19, 20, 21, 27, 35, 37].

The aim of this work is to model the apparent motion in image sequences depicting textured motion patterns. We adopt the *mixed-state* Markov Random Fields (MRF) models recently introduced in [4] to represent the so-called motion textures. The approach consists in describing the spatial distribution of local *motion measurements* which exhibit values of two types: a discrete component related to the absence of motion and a continuous part for actual measurements. The former accounts for symbolic information that is beyond the null value of motion itself, providing crucial information on the dynamic content of the scene.

We propose several significant extensions to the first model introduced in [4] in order to capture more properties of the analyzed motion textures. First, we use an extended neighborhood for local conditional characteristics. Second, we consider a non-zero mean Gaussian distribution for the continuous part which allows us to express a stronger spatial correlation between continuous motion values.

A new approach for addressing the problem of the partition function calculation in Gibbs fields is also proposed. This allows us to appropriately handle

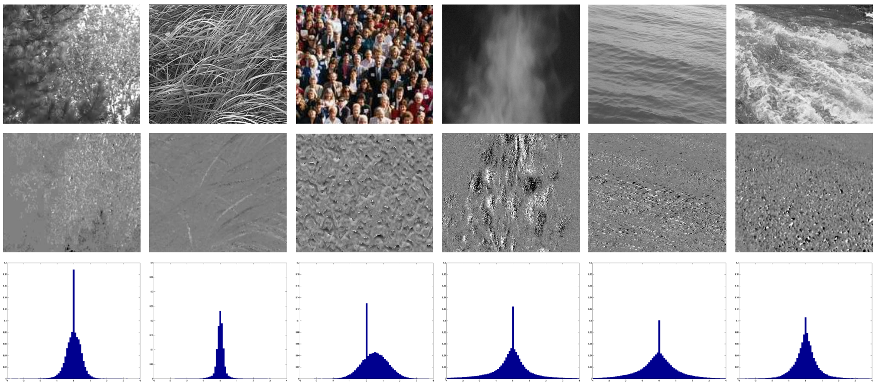

Figure 1: Top: sample images from dynamic textures. Middle: motion textures. Bottom: motion histograms

the normalization factors for the joint distributions, which are crucial in any decision-theory based method to dynamic content recognition, segmentation and classification.

The use of the proposed model in dynamic content recognition applications is analyzed. We have defined a motion texture classification scheme, analyzing the problem of computing a dissimilarity measure between mixed-state Markov random field models. We have also defined a motion texture segmentation method exploiting this modeling, where the problem is reduced to the computation of a label field that maximizes an energy function. We address this task by application of some recently introduced graph-cut based energy optimization methods. One important original aspect is that we do not assume conditional independence of the observations for each texture. Results on real examples will demonstrate the accuracy and efficiency of our method.

1.1 Related work

Although 2D spatial textures had been vastly analyzed in the computer vision literature [15, 18, 30], temporal or dynamic textures have received an increasing interest in the last few years. A first distinction between different approaches, lies in the type of image features extracted from the image sequence. Soatto *et al.* [16] proposed the use of ARMA models directly on image intensities with convincing results on dynamic texture synthesis. In [45] an improvement is proposed based on a control theory approach. In [43], linear dynamical systems (LDS) are also applied in combination with 2D translational models, that permits to deal with a non-static camera and/or moving dynamic textures. Other intensity-based dynamic texture models can be found in [10, 42].

Alternatively, there have been an increasing interest in the modeling of motion features extracted from dynamic textures instead of considering pixel-wise intensity values [4, 13, 37]. Particularly, normal flow is a very efficient and natural way of characterizing the local spatio-temporal dynamics of a dynamic texture [21]. A good survey on dynamic texture characterization can be found in [11].

Bouthemy *et al.* [4, 13] have recently formulated a novel statistical framework for modeling motion textures. This approach, exploits the mixed nature of motion information through the utilization of a new class of Markov random fields, namely, *mixed-state Markov random fields*.

Much efforts on dynamic texture characterization have been devoted to the recognition and classification of these types of image sequences. Recent results have shown that methods based on motion features give the highest recognition and classification rates for dynamic textures depicting natural scenes [21, 31, 33]. They are based on computing some motion statistics across the image sequence, and using them as class descriptors for the classification task. A different approach for dynamic texture recognition is proposed in [38], where a dissimilarity measure between linear dynamical models is utilized. Finally, the extension of the concept of dynamic textures to more complex scenes sets the necessity of considering more elaborated approaches for specific applications. See [46] as an interesting example.

Closely related to classification and recognition, the problem of segmentation of dynamic textures aims at determining and locating regions in the image that correspond to a same dynamic texture class. Up to our knowledge, existing segmentation methods of different classes of dynamic textures are based on the linear models proposed originally by Soatto *et al.* [16]. In [17], a level-set variational approach is proposed for minimizing a cost functional w.r.t. some geometric discrepancy measures between models and the boundary of the different regions. Vidal and Ravichandran [43] propose to model each dynamic texture as a linear system plus a 2-D translational motion component. This solves the problem of moving regions (or non-static camera) under rigid motion. Segmentation is achieved by a Principal Component Analysis (PCA) method, for subspace separation. Finally, in [10] a mixture generative model of several linear dynamic textures is proposed, which naturally leads to a segmentation approach.

2 Motion textures

Let $I_i(t)$ be a scalar function that represents the image intensity at image point $i \in S = \{1...N\}$ for time t where S denotes the image grid. We define a motion texture as the result of a motion feature extraction process applied on $I(t)$, for some given instant. The definition is general for any motion measure on the image sequence. This accounts for scalar motion measurements, and for vectorial observations as well.

Briefly, our approach consists in the following: once the motion measure is obtained, a sequence of intensity images is mapped to a sequence of *motion maps or fields*. These fields are in essence functions of image bidimensional location and then the problem is reduced to modeling a spatial distribution of motion values.

2.1 Local motion measurements

Obtaining reliable and, at the same time, easily computable motion information from image sequences is essential in our formulation, which aims at analyzing a large amount of data and without explosion of model dimension. We want to be clear in that the objective is not motion estimation by itself, but dynamic content analysis.

The Optical Flow Constraint [25] is a well known condition over intensity images from which velocity fields can be effectively estimated. Locally, it gives a valuable quantitative information about the spatio-temporal structure of the scene. However, the *aperture problem* allows us to measure only the component of the velocity of an image point in the direction of the spatial intensity gradient, i.e., *normal flow*, defined as $\mathbf{V}_i^{\perp}(t) = -\frac{\frac{\partial I_i(t)}{\partial t}}{\|\boldsymbol{\nabla} I_i(t)\|}\frac{\boldsymbol{\nabla} I_i(t)}{\|\boldsymbol{\nabla} I_i(t)\|}$ where $\nabla I_i(t)$ is the spatial intensity gradient at location i. We follow the approach of [19], for obtaining reliable scalar motion observations. It computes the magnitude of normal flow and applies a weighted average over a small window around an image location, to smooth out noisy measurements and enforce reliability. The result is a smoothed measure of local motion, which is always positive. However, in contrast to [19], we introduce a weighted vectorial average of normal flow in order to keep direction information:

$$\tilde{\mathbf{V}}_i^{\perp}(t) = \frac{\sum\limits_{j \in W} \mathbf{V}_j^{\perp}(t) \parallel \boldsymbol{\nabla} I_j(t) \parallel^2}{max(\sum\limits_{j \in W} \parallel \boldsymbol{\nabla} I_j(t) \parallel^2, \eta^2)}, \tag{1}$$

where η^2 is a constant related to noise, and W is a small window centered in location i. This average results in a local estimation of normal flow. The projection of this quantity over the intensity gradient direction gives rise to the following scalar motion observation:

$$v_i(t) \quad = \quad \tilde{\mathbf{V}}_i^{\perp} \cdot \frac{\boldsymbol{\nabla} I_i(t)}{\parallel \boldsymbol{\nabla} I_i(t) \parallel}, \tag{2}$$

with $v_i(t) \in (-\infty, +\infty)$. As already said, the definition of the measurement process determines to some extent, the type of motion textures that we will be dealing with.

2.2 Statistical properties of motion measurements

At this stage, it is important to emphasize a fundamental property of the resulting field. Real world observations can be represented as continuous random variables within a statistical setting. However, discrete (or symbolic) information is also of interest. Experiments obtained from computing the proposed motion quantities for several different dynamic textures have shown that, if we observe the corresponding motion histograms (Fig. 1), we note that the statistical distribution of the motion measurements has two elements: a discrete component at the null value $v_i = 0$, and a continuous distribution for the rest of the motion values.

The underlying discrete property of no-motion for a point in the image, is represented as a null observation. In fact, it is not the value by itself that matters, but the binary property of what we call *mobility*: the absence or presence of motion. Thus, the null motion value in this case, has a specific place in the sample space, and consequently, has to be modeled accordingly.

We call this type of fields *mixed-state random fields* as the corresponding random variables take their values in a mixed discrete-continuous space.

3 Mixed-state probability framework

3.1 A first approximation

The key observation made in the previous section about the statistical properties of motion measurements, settles the necessity for an adequate representation of the associated random variables. In a first approach, observing the histograms, the problem can be formulated as a mixture of two densities, where for the moment we can assume that one of them is a narrow Gaussian centered at the zero value. We denote the other component of the mixture as $p^a(x)$. Thus, we could assume the following mixture density function:

$$p(x) = \rho \mathcal{N}_{[0,\sigma_0]}(x) + (1-\rho)p^a(x), \tag{3}$$

where $\rho \in [0, 1]$ is the weight of the mixture. Here, the variance σ_0 is made arbitrarily small, and moreover, fixed to a convenient value. Then, we can assume that the zero value comes from $\mathcal{N}(0, \sigma_0)$ and the rest of the real values, from $p^a(x)$. With this approximation of the distribution of the null values (discrete values) considered as samples from a very narrow Gaussian density, the whole formulation corresponds to a classical multi-modal continuous model.

3.2 Mixed-state model as a limit case

In the limit case, when $\sigma_0 \to 0$, the corresponding Gaussian distribution approaches a Dirac delta functional, $\delta_0(x)$. Effectively, it should be interpreted as the limit in distribution of the sequence of zero-mean Gaussian pdf's with variance σ_0 going to zero. Let us note that any other sequence of functions $g_n(x)$ can be alternatively used, as long as $g_n(0) \to \infty$, with $n \to \infty$ and $\int_{-\infty}^{\infty} g_n(x) = 1$. Then, we can equivalently write,

$$p(x) = \rho\delta_0(x) + (1-\rho)p^a(x). \tag{4}$$

This density is well-defined and corresponds to a random variable that has a discrete value with probability mass concentrated at zero. In fact, it is easy to observe that $P(x = 0) = \rho$. This formulation has already been proposed in [13] and [14] for motion texture modeling.

3.3 Measure theoretic approach

From a probability theory point of view, it is more natural to redefine the probability space (the measurable space) w.r.t. a new probability measure, avoiding to deal with a functional or distribution as the Dirac delta, that may complicate the strict definition of the corresponding density, allowing also to generalize the case of a discrete real value (e.g., $x = 0$) to a generic symbolic value or abstract label, that may lie on an arbitrary label set.

In this section, we outline the theoretical framework attached to mixed-state random variables. Let us define $\mathcal{M} = \{r\} \cup R^*$ where $R^* = R \setminus \{r\}$, with r a possible "discrete" value, sometimes called *ground* value. A random variable X defined on this space, called *mixed-state variable*, is constructed as follows: with probability $\rho \in (0,1)$, set $X = r$, and with probability $= 1-\rho$, X is continuously distributed in R^*.

Consequently, the distribution function of X can be expressed as a monotone increasing function with a "step jump" at $X = r$. In order to compute the probability density function of the mixed-state variable X, $\mathcal{M}$ is equipped with a "mixed" reference measure:

$$m(dx) = \nu_r(dx) + \lambda(dx), \tag{5}$$

where ν_r is the discrete measure for the value r and λ the Lebesgue measure on R^*. Such a measure has already been used in [7, 39] for simultaneous fuzzy-hard image segmentation. Let us define the indicator function of the reference value r as $\mathbf{1}_r(x)$ and its complementary function $\mathbf{1}^*_r(x) = \mathbf{1}_{\{r\}^c}(x) = 1 - \mathbf{1}_r(x)$. Then, the above random variable X has the following density function w.r.t. $m(dx)$:

$$p(x) \quad = \quad \rho\mathbf{1}_r(x) + (1-\rho)\mathbf{1}^*_r(x)p^a(x), \tag{6}$$

where $p^a(x)$ is a continuous pdf from an absolutely continuous distribution w.r.t. λ, defined on R.

It should be clear that the models described in equations (3),(4) and (5) are totally equivalent, for the case of $r = 0$ (or any other discrete real value), and that in practice, there is no advantage on choosing one of them. However, from a theoretical point of view, it is easier to handle the probability densities, given w.r.t. a mixed measure, as there is no approximation to some limit case. These definitions can be extended easily to multi-dimensional mixed-state variables.

4 Mixed-state Markov models

The power of Markov models was primarily demonstrated in [2] with the introduction of the so-called *auto-models*. Based on this, Bouthemy *et al.* [4] have recently proposed a motion texture modeling framework, in the form of Mixed-state Markov Random Fields (MS-MRF).

As the seminal theorem of Hammersley-Clifford states, Markov random fields with an everywhere positive density function, are equivalent to nearest neighbor Gibbs distributions. The joint p.d.f. of the random variables that compose the field has the form, $p(\mathbf{X}) = \exp[Q(\mathbf{X})]/Z$, where $Q(\mathbf{X})$ is an energy function, and Z is called the partition function or normalizing factor of the distribution.

Let $S = \{1....N\}$ be a lattice of points or image locations such that $\mathbf{X} = \{x_i\}_{i\in S}$. Define $\mathbf{X}_A$ as the subset of random variables restricted to $A \subset S$, i.e., $\mathbf{X}_A = \{x_i\}_{i\in A}$. It is a well-known result that the energy $Q(\mathbf{X})$ can be expressed as a sum of potential functions, $Q(\mathbf{X}) = \sum_{\mathcal{C}\subset S} V_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}})$, defined over all subsets $\mathcal{C}$ of the lattice space S [2]. The advantage lies in specifying non-null potentials only for specific subsets $\mathcal{C}$ of some desired structure. Let $i, j \in S$ be two locations or sites of the field $\mathbf{X}$. It is easy to show that the two random variables x_i, x_j of the lattice are conditionally independent, if all $\{\mathcal{C} : i, j \in \mathcal{C}\}$ have associated potentials $V_{\mathcal{C}} = 0$. Any $\mathcal{C}$ for which $V_{\mathcal{C}} \neq 0$ is called a *clique*. The non-trivial potentials define, as a consequence, a neighborhood structure $\mathcal{N}_i$, which is defined as $\mathcal{N}_i = \{j \in S \mid \exists\, \mathcal{C} : i, j \in \mathcal{C} \,\wedge\, V_{\mathcal{C}} \neq 0\}$. The equivalence of this neighborhood description of Markov random fields and undirected graphs is straightforward.

Let us stress that the expansion of the energy function $Q(\mathbf{X})$ is not unique. However, if we define the *canonical potentials* [28] w.r.t a reference value r, as the set of functions $V_{\mathcal{C}}$ such that if for any $i \in \mathcal{C}$, $x_i = r$, then $V_{\mathcal{C}} = 0$, $Q(\mathbf{X})$ can be uniquely expressed as following:

$$Q(\mathbf{X}) = \log \frac{p(\mathbf{X})}{p(\mathbf{r})} = \sum_{\mathcal{C} \subset S} V_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}), \tag{7}$$

where now $V_{\mathcal{C}}$ are canonical potentials. See appendix A for a simple proof. Here, $\mathbf{r}$ is the uniform configuration equaling r everywhere and we have $Q(\mathbf{r}) = 0$. Equivalently, for any $i \in S$, we can write,

$$\log \frac{p(x_i \mid \mathbf{X}_{S \setminus i})}{p(r \mid \mathbf{X}_{S \setminus i})} = \sum_{\{i\} \subset \mathcal{C} \subset S} V_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}), \tag{8}$$

for which the sum is done for all subsets that contain the site i. Then, and up to a normalizing factor, if we know the shape of the global energy we can write the conditional distributions.

Equations (7) and (8), are the basis for the equivalence between Markov random fields and Gibbs distributions. Indeed, they allow us to start from conditional characteristics, keeping consistency with a joint or global model.

A Markov random field model for mixed-state variables is then defined, where the local conditional mixed-state densities w.r.t. the measure m, are [4]:

$$p(x_i \mid \mathbf{X}_{\mathcal{N}_i}) \quad = \quad \rho(\mathbf{X}_{\mathcal{N}_i})\mathbf{1}_r(x_i) + (1 - \rho(\mathbf{X}_{\mathcal{N}_i}))\mathbf{1}^*_r(x_i)p^a(x_i \mid \mathbf{X}_{\mathcal{N}_i}), \tag{9}$$

where $\rho(\mathbf{X}_{\mathcal{N}_i}) = P(x_i = r \mid \mathbf{X}_{\mathcal{N}_i})$ is now a function and $\mathbf{X}_{\mathcal{N}_i}$ is the subset of $\mathbf{X}$ restricted to a neighborhood of locations $\mathcal{N}_i$. As already said, this defines the non-null potentials. We want to be clear in that p^a is a continuous function, w.r.t. x_i and $\mathbf{X}_{\mathcal{N}_i}$, that defines itself a continuous Markov random field. In this article, we will not consider more general situations where p^a could be any arbitrary density function, possibly with discontinuities.

4.1 Mixed-state auto-models

In the context of MS-MRFs, the arising questions are: can we arbitrarily choose conditional pdf's in equation (9) and, which is the general form for the joint distribution of the field, that responds to such a formulation? For a certain family of conditional distributions, the answer is known and we show here how to apply this result to mixed-state models.

As J. Besag describes in his pioneering paper [2], when the conditional probability densities that define the local characteristics of a Markov random field, belong to a one-parameter exponential family, and assuming that the corresponding global Gibbs energy depends only on cliques that contain no more than two sites, i.e. *auto-models*, the expression for the parameter is known as an affine function of a sufficient statistic of the neighbors. In the case of d-parameter auto-models, this result has been extended in [4] and [9] through the following theorem:

Theorem 1. *Assume that the Markov random field probability density $p(\mathbf{X})$ and its energy function $Q(\mathbf{X})$ satisfy the following conditions:*

A- The potential functions are at most of second order, i.e.

$$Q(\mathbf{X}) = \sum_{i \in S} V_i(x_i) + \sum_{<i,j> \in S} V_{ij}(x_i, x_j).$$

B- The local conditional characteristics belong to the d-parameter exponential family

$$\log p\,(x_i \mid \mathbf{X}_{\mathcal{N}_i}) = \mathbf{\Theta}_i^T(\mathbf{X}_{\mathcal{N}_i})\mathbf{S}_i(x_i) + C_i(x_i) + D_i(\mathbf{X}_{\mathcal{N}_i}),$$

with $\mathbf{S}_i(x_i) \in R^d$, $\mathbf{\Theta}_i(\mathbf{X}_{\mathcal{N}_i}) \in R^d$, $C_i(x_i)$ and $D_i(\mathbf{X}_{\mathcal{N}_i}) \in R$, and the normalization conditions $C_i(r) = 0$ and $\mathbf{S}_i(r) = \mathbf{0}$.

C- The family of sufficient statistics $\{\mathbf{S}_i(x_i)\}_{i \in S}$ is regular in the sense that

$$\forall i \in S, \;\; Span\{\mathbf{S}(x_i)\} = R^d.$$

Then, the corresponding parameter vector is given by,

$$\mathbf{\Theta}_i(\mathbf{X}_{\mathcal{N}_i}) = \boldsymbol{\alpha}_i + \sum_{j \in \mathcal{N}_i} \boldsymbol{\beta}_{ij}\mathbf{S}_j(x_j), \tag{10}$$

with $\boldsymbol{\beta}_{ij} \in \mathbb{R}^{d \times d}$ and $\boldsymbol{\alpha}_i = [\alpha_1 .... \alpha_d]^T \in R^d$. Moreover, the canonical (w.r.t a reference r) potential functions take the form,

$$\begin{aligned} V_i(x_i) &= \boldsymbol{\alpha}_i^T \cdot \mathbf{S}_i(x_i) + C_i(x_i), \quad (11) \\ V_{ij}(x_i, x_j) &= \mathbf{S}_i(x_i)^T \boldsymbol{\beta}_{ij} \mathbf{S}_j(x_j). \quad (12) \end{aligned}$$

For a complete proof see [24]. Consequently, $\mathbf{\Theta}_i(\mathbf{X}_{\mathcal{N}_i})$ is a function of the neighbors of a particular location i, where the conditional dependence between sites cannot be arbitrary under the mentioned hypotheses, having a particular shape as seen in (10).

In order to express equation (9) in an exponential form, note that the conditional density is a sum of excluding terms, allowing us to express the logarithm of the left-hand side as the sum of the logarithms of the two terms on the right-hand side. Some calculations and rearrangements yield:

$$\log p(x_i \mid \mathbf{X}_{\mathcal{N}_i}) \;\; = \;\; \log \rho_i + \mathbf{1}_r^*(x_i) \log \left[p^a(x_i \mid \mathbf{X}_{\mathcal{N}_i}) \frac{(1-\rho_i)}{\rho_i} \right], \tag{13}$$

where $\rho_i \equiv \rho(\mathbf{X}_{\mathcal{N}_i})$ for simplicity of notation. Then, it is easy to show that if $p^a(x_i \mid \mathbf{X}_{\mathcal{N}_i})$ belongs to some exponential family defined by functions $\tilde{\mathbf{\Theta}}_i, \tilde{\mathbf{S}}_i, \tilde{C}_i, \tilde{D}_i$ substituting for $\mathbf{\Theta}_i, \mathbf{S}_i, C_i, D_i$ respectively, the mixed-state conditional density also belongs to the exponential family with:

$$
\begin{aligned}
\mathbf{\Theta}_i^T(\mathbf{X}_{\mathcal{N}_i}) &= \left[\tilde{D}_i(\mathbf{X}_{\mathcal{N}_i}) + \log\left[\frac{(1-\rho_i)}{\rho_i}\right] \ , \ \tilde{\mathbf{\Theta}}_i^T(\mathbf{X}_{\mathcal{N}_i})\right], && (14)\\
\mathbf{S}_i^T(x_i) &= \left[\mathbf{1}_r^*(x_i) \ , \ \tilde{\mathbf{S}}_i^T(x_i)\right], && (15)\\
C_i(x_i) &= \tilde{C}_i(x_i)\mathbf{1}_r^*(x_i), && (16)\\
D_i(\mathbf{X}_{\mathcal{N}_i}) &= \log\rho_i. && (17)
\end{aligned}
$$

In summary, if p^a belongs to an exponential family with q parameters, the mixed-state conditional density is also an exponential family distribution with $q+1$ dimensions.

Note that the matrices $\boldsymbol{\beta}_{ij}$ define the pairwise interaction between neighboring points. A required condition for a well-defined Markov random field arises from the symmetry condition, $V_{ij}(x_i, x_j) = V_{ji}(x_j, x_i)$, as the potential functions are defined over non-ordered pairs. From equation (12), this implies that $\boldsymbol{\beta}_{ij} = \boldsymbol{\beta}_{ji}^T$. For the case of a mixed-state auto-model, assuming that p^a defines itself a purely continuous auto-model, more constrains on these matrices appear from the symmetry condition, imposing at the same time, a constraint on the shape of the global energy function. We thus have the following corollary:

Corollary 1. *For a mixed-state auto-model where p^a is a continuous function, the energy $Q(\mathbf{X})$ is decomposed into a purely discrete and a purely continuous component, i.e. $Q(\mathbf{X}) = Q^d(\mathbf{X}) + Q^c(\mathbf{X})$.*

Proof. First, define,

$$\boldsymbol{\beta}_{ij} = \begin{pmatrix} d_{ij} & \mathbf{Y}_{ij} \\ \mathbf{Y}'_{ij} & \widetilde{\boldsymbol{\beta}}_{ij} \end{pmatrix}, \qquad \boldsymbol{\alpha}_i = [\alpha_i^1 \quad \widetilde{\boldsymbol{\alpha}}_i^T]^T, \tag{18}$$

where $\mathbf{Y}_{ij}$ is the first row of $\boldsymbol{\beta}_{ij}$ minus the first element, d_{ij}, and $\mathbf{Y}'_{ij}$ is the first column of $\boldsymbol{\beta}_{ij}$ minus the first element. $\widetilde{\boldsymbol{\beta}}_{ij}$ is the lower-right $(d-1)\times(d-1)$ submatrix of $\boldsymbol{\beta}_{ij}$ and equivalently we define the vector $\widetilde{\boldsymbol{\alpha}}_i$. From equations (10), (14) and (15):

$$\tilde{\mathbf{\Theta}}_i^T(\mathbf{X}_{\mathcal{N}_i}) = \widetilde{\boldsymbol{\alpha}}_i + \mathbf{Y}'_{ij}\mathbf{1}_j^*(x_j) + \widetilde{\boldsymbol{\beta}}_{ij}\tilde{\mathbf{S}}_j(x_j). \tag{19}$$

As we have assumed that p^a is a continuous function that, in fact, defines a continuous MRF, so it is $\tilde{\mathbf{\Theta}}_i^T(\mathbf{X}_{\mathcal{N}_i})$. This implies that $\mathbf{Y}'_{ij} = \mathbf{Y}_{ji} = \mathbf{0}$ and analogously $\mathbf{Y}'_{ji} = \mathbf{Y}_{ij} = \mathbf{0}$. Consequently, from equations (12) and (15) we can write:

$$
\begin{aligned}
V_{ij}(x_i, x_j) &= d_{ij}\mathbf{1}_r^*(x_i)\mathbf{1}_r^*(x_j) + \tilde{\mathbf{S}}_i^T(x_i)\widetilde{\boldsymbol{\beta}}_{ij}\tilde{\mathbf{S}}_j^T(x_j),\\
V_{ij}(x_i, x_j) &= V_{ij}^d(x_i, x_j) + V_{ij}^c(x_i, x_j), && (20)
\end{aligned}
$$

where $V_{ij}^d(x_i, x_j)$ and $V_{ij}^c(x_i, x_j)$ are respectively the discrete and the continuous part of the potential. The same decomposition applies trivially to $V_i(x_i)$ and thus, we can arrive to the following expression:

$$Q(\mathbf{X}) = Q^d(\mathbf{X}) + Q^c(\mathbf{X}), \tag{21}$$

where $Q^d(\mathbf{X}) = \sum_{i \in S} V_i^d(x_i) + \sum_{<i,j>} V_{ij}^d(x_i, x_j)$ corresponds to a purely binary discrete potential, and $Q^c(\mathbf{X}) = \sum_{i \in S} V_i^c(x_i) + \sum_{<i,j>} V_{ij}^c(x_i, x_j)$ defines a continuous MRF on $\mathbf{X}$. This completes the proof □

Finally, $p(\mathbf{X}) = Z^{-1} \exp\left[Q^d(\mathbf{X}) + Q^c(\mathbf{X})\right]$. It is worthy to note that this decomposition result does not mean that the discrete and continuous parts are decoupled neither they are independent. It should also be clear that the joint distribution $p(\mathbf{X})$ is not the product of two pdf's of different nature. In fact, the normalization constant or partition function, Z, is, at the same time, a function of the continuous part and the discrete part.

In the next section, we design a mixed-state auto-model for motion texture modeling and segmentation exploiting these results.

5 A motion texture model

As mentioned before, the definition of a motion texture is related to the measurement process. The scalar motion measure proposed here gives rise to a motion field with mixed-states, where the discrete property corresponds to the absence/presence of motion. This corresponds to setting $r = 0$, as the discrete symbolic value.

Existing approaches to describing dynamic textures as spatio-temporal processes, and more specifically, the proposed generative models formulated within a statistical setting, are mostly based on the assumption of certain stationarity and homogeneity of such visual phenomena. Linear dynamic textures [10, 16, 45] are based on the hypothesis of second-order stationary processes. The learning of the model is equivalent to the problem of estimating a set of time-invariant matrices that describe the evolution of intensity over time, and thus, several frames of the sequence are needed.

Alternatively, dynamic texture recognition and classification approaches aim at efficiently retrieving some representative features from the image sequences, assuming, for the sake of estimation, that they are stationary in time and space. These quantities are then used as characteristic descriptors of different dynamic contents.

As already said, our approach of modeling the instantaneous motion maps associated to dynamic textures, is reduced to modeling a spatial field of mixed-state values. In general terms, the proposed conditional models could be defined by a different set of parameters for each location of the image (see equation 10). This would give rise to a motion texture model with a number of parameters proportional to the image size. Unfortunately, such high-dimensional representation of the associated dynamic information is unfeasible in practice and does not constitutes a compact description of motion textures. Moreover, an increasingly number of frames would be necessary for the estimation process. This conspires against a formulation oriented to efficient content recognition and retrieval. However, if needed, our framework could deal with spatially non-stationary motion textures.

Henceforth, we will assume that the extracted motion fields can be considered as realizations of an homogeneous spatial model. Indeed, the visual information attached to a dynamic texture is mostly displayed from spatially homogeneous motion regions, and moreover, mostly associated to statistically homogeneous textured intensity patterns. The spatial homogeneity hypothesis will allow us to formulate a compact motion texture model, with relatively few parameters, that can be estimated from a single motion map. Of course, the characterization of a whole dynamic texture sequence with this model will be based on the implicit assumption of temporal stationarity. Nevertheless, it is worthy to note that no such hypothesis is needed in the theoretical framework and the experimental results presented herein.

Three more aspects related to the proposed model need to be defined. It is well known that a Markov random field model is completely specified by the set of potential functions. Consequently, we will first assume that the mixed-state Gibbs energy will be composed by at most second order potentials. Pairwise interaction is vastly used in Markov random fields and is a good trade-off between simplicity and representativeness. Secondly, the potentials associated to $Q^c(\mathbf{X})$, i.e., the continuous part of the model, are chosen to correspond to a Gaussian distribution, which by definition is a continuous auto-model. In summary, we will adopt the mixed-state auto-model defined previously. Finally, we will suppose that for the second order clique $\mathcal{C} = \{i, j\}$, the corresponding potential will be non-null, only if i and j are second-order nearest neighbors. That is, from the definiton of neighborhood, $\mathcal{N}_i$ is the set of the 8-nearest neighbors for location i:

$$\mathcal{N}_i = \{i_E, i_W, i_N, i_S, i_{NW}, i_{SE}, i_{NE}, i_{SW}\}, \tag{22}$$

where, for example, i_E is the east neighbor of i in the image grid, i_{NW} the north-west neighbor, etc. Moreover, a necessary condition in order to define an homogeneous and, in fact, stationary spatial process, is that the parameters related to symmetric neighbors (E-W, N-S, NW-SE, NE-SW) must be the same. This is a consequence of the symmetry of the potentials.

In the following sections, we completely characterize a Gaussian mixed-state homogeneous model. This implies: defining the parameters for the discrete and continuous part, obtaining the local conditional densities, analyzing the admisibility of the joint distributions and adressing the problem of partition function calculation.

5.1 Gaussian mixed-state model

In general, we define a Gaussian mixed-state Markov random field as the one that has a Gibbs probability density given by:

$$p(\mathbf{X}) = Z^{-1} \exp\left[Q^d(\mathbf{X}) + 2\mathbf{X}^T\mathbf{B}\boldsymbol{\mu} - \mathbf{X}^T\mathbf{B}\mathbf{X}\right], \tag{23}$$

where $\mathbf{B}$ is a $N \times N$ (usually) sparse matrix and $\boldsymbol{\mu}$ is the mean vector, for which all its components are equal, according to the assumption of homogeneity.

Thus, the continuous part of the corresponding conditional mixed-state density follows a Gaussian law with mean $m_i \equiv m(\mathbf{X}_{\mathcal{N}_i})$ and variance $\sigma_i^2 \equiv \sigma^2(\mathbf{X}_{\mathcal{N}_i})$. The local characteristics are then expressed as:

$$p(x_i \mid \mathbf{X}_{\mathcal{N}_i}) \quad = \quad \rho_i \mathbf{1}_0(x_i) + (1-\rho_i)\mathbf{1}_0^*(x_i)\frac{1}{\sqrt{2\pi}\sigma_i}e^{-\frac{(x_i-m_i)^2}{2\sigma_i^2}}, \tag{24}$$

where the indicator function is now given w.r.t the null value. In exponential form, they are defined by:

$$\begin{aligned}
\boldsymbol{\Theta}_i^T(\mathbf{X}_{\mathcal{N}_i}) &= [\theta_{1,i}\ ,\ \theta_{2,i}\ ,\ \theta_{3,i}]\,, \\
&= \left[-\frac{m_i^2}{2\sigma_i^2} + \log\frac{1}{\sigma_i\sqrt{2\pi}} + \log\frac{1-\rho_i}{\rho_i}\ ,\ \frac{1}{2\sigma_i^2}\ ,\ \frac{m_i}{2\sigma_i^2}\right], \\
\mathbf{S}_i^T(x_i) &= \left[\mathbf{1}_0^*(x_i)\ ,\ -x_i^2\ ,\ x_i\right], \\
C_i(x_i) &= 0, \\
D_i(\mathbf{X}_{\mathcal{N}_i}) &= \log\rho_i. \qquad (25)
\end{aligned}$$

The parametrization of the conditional distribution in terms of $\boldsymbol{\Theta}_i$ allows us to express the dependence of a point on its neighbors through equation (10). Moreover, the parameters of the original parametrization, ρ_i, m_i, σ_i, are also functions of the neighborhood and can be obtained easily from the first line of equations (25), resulting in the following expressions:

$$\rho_i = \frac{(\sigma_i\sqrt{2\pi})^{-1}}{(\sigma_i\sqrt{2\pi})^{-1} + e^{\theta_{1,i} + \frac{m_i^2}{2\sigma_i^2}}}, \qquad \sigma_i^2 = \frac{1}{2\theta_{2,i}}, \qquad m_i = \frac{\theta_{3,i}}{2\theta_{2,i}}. \qquad (26)$$

From equations (7) and (8), we see how to obtain the potential functions from the conditional density expansion. Then, comparing (23) and (24), it is clear that σ_i^2 cannot be a function of the neighbors and thus must be a constant, i.e., $\sigma_i^2 = \sigma^2(\mathbf{X}_{\mathcal{N}_i}) = \sigma^2$; equivalently m_i must necessarily be linear w.r.t the neighbors. Now, having in mind equation (20), for the homogeneous mixed-state Gaussian model let us define:

$$\boldsymbol{\beta}_{ij} = \begin{pmatrix} d_{ij} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & h_{ij} \end{pmatrix}, \qquad \boldsymbol{\alpha} = \begin{bmatrix} a & b & c \end{bmatrix}^T. \qquad (27)$$

Here, homogeneity was applied to set $\boldsymbol{\alpha}_i = \boldsymbol{\alpha}$. From equation (10) we effectively obtain:

$$\begin{aligned}
\theta_{1,i} &= a + \sum_{j\in\mathcal{N}_i} d_{ij}\mathbf{1}_0^*(x_j), \\
\theta_{2,i} &= b, \\
\theta_{3,i} &= c + \sum_{j\in\mathcal{N}_i} h_{ij}x_j, \qquad (28)
\end{aligned}$$

and from (26):

$$m(\mathbf{X}_{\mathcal{N}_i}) = \frac{c}{2b} + \sum_{j\in\mathcal{N}_i}\frac{h_{i,j}}{2b}x_j. \qquad (29)$$

For the discrete component of the energy, according to the expression of the potentials in equations (11),(12) and (20), we have,

$$Q^d(\mathbf{X}) \quad = \quad \sum_{i \in S} a\mathbf{1}^*(x_i) + \sum_{\langle i,j \rangle : j \in \mathcal{N}_i} d_{ij}\mathbf{1}^*(x_i)\mathbf{1}^*(x_j). \tag{30}$$

Finally, note that the set of parameters involved in equations (28) completely describes the potential functions associated to the discrete and continuous Gibbs energies. As mentioned before, the homogeneity of the model implies the symmetry of the parameters; thus, for the 8-point neighborhood, we have 4 interacting directions: vertical (V), horizontal (H), diagonal (D)and anti-diagonal (AD). Then, $\boldsymbol{\beta}_{ij} = \boldsymbol{\beta}_k$ with $k \in \{H, V, D, AD\}$ and an homogeneous Gaussian mixed-state model is defined by:

$$\boldsymbol{\phi} = \{a, b, c, d_H, h_H, d_V, h_V, d_D, h_D, d_{AD}, h_{AD}\}, \tag{31}$$

that is a set of 11 parameters.

Another aspect, related to spatial interaction, is considered in the definition of the model. As pointed out by J. Besag [2], several one-parameter auto-models necessarily imply, in order to be well-defined, *spatial competition* between neighboring sites. In our case, this would mean that motion values tend to be locally "unaligned" following a physical term. However, the type of motion textures that we want to study show some local motion smoothness, mostly associated to *cooperative* schemes. Then, this condition is explicitly imposed in the model, resulting in a constraint on the parameters. Formally, in a mixed-state cooperative model, the conditional mean of the continuous component for a site has to be an increasing function of its neigbors. Following equation (29), this implies that $h_{ij} > 0$.

For the sake of completeness, we now write the expressions of the matrix $\mathbf{B} = \{B^{ij}\}$ and the mean vector $\boldsymbol{\mu} = \{\mu\}$. From equation (24), we can obtain the expressions for the first and second order cliques of the continuous part, and comparing with (23), $B^{ii} = \frac{1}{2\sigma^2} = b$, $B^{ij} = -\frac{h_{ij}}{2}$ if $j \in \mathcal{N}_i$ and $B^{ij} = 0$ otherwise. Equivalently, $c = 2\mu(b - \sum_{j \in \mathcal{N}_i} \frac{h_{ij}}{2})$ and then, $\mu = \frac{c}{2}(b - \sum_{j \in \mathcal{N}_i} \frac{h_{ij}}{2})^{-1}$, for any i.

As expected, the expression of μ corresponds to the mean for a point, restricted to the non-null values, that is, the mean of the continuous part of the MRF. Effectively, knowing that in general $E[X] = E_Y\left[E_{X|Y}[X \mid Y]\right]$,

$$\begin{aligned} E[x_i \mid x_i \neq 0] \quad &= \quad E_{\mathbf{X}_{\mathcal{N}_i}}\left[E_{x_i|\mathbf{X}_{\mathcal{N}_i}}[x_i \mid \mathbf{X}_{\mathcal{N}_i}, x_i \neq 0, x_j \neq 0, j \in \mathcal{N}_i]\right] \\ &= \quad E_{\mathbf{X}_{\mathcal{N}_i}}[m(\mathbf{X}_{\mathcal{N}_i}) \mid x_j \neq 0, j \in \mathcal{N}_i], \end{aligned} \tag{32}$$

and replacing by equation (29), putting $E[x_i \mid x_i \neq 0] = \mu$, and assuming an homogeneous model,

$$\mu = \frac{c}{2b} + \sum_{j \in \mathcal{N}_i} \frac{h_{i,j}}{2b}\mu. \tag{33}$$

We thus obtain the same result as before.

5.2 Admissibility of the Gibbs distribution

A required condition for the joint model is that it must satisfy $\int_\Omega e^{Q(\mathbf{X})} m(d\mathbf{X}) < \infty$ in order to be well-defined. From equation (23) we observe that the asymptotic properties of $Q(\mathbf{X})$ are determined by $Q^c(\mathbf{X})$, which is of the form $Q^c(\mathbf{X}) = -\mathbf{y}^T(\mathbf{X})\mathbf{B}\mathbf{y}(\mathbf{X})$, with $\mathbf{y}(\mathbf{X})$ growing linearly. Hence, the condition for integrability reduces to ensure that $\mathbf{B}$ is positive definite. This matrix is of a particular form: all rows add up to the same value, as the model is homogeneous; the elements of the diagonal are all equal to $b = \frac{1}{2\sigma^2}$ and the matrix is symmetric. Then, we have the following theorem:

Proposition 1. *For a cooperative homogeneous Gaussian mixed state model as defined in (24), the associated joint distribution is well defined if and only if* $b > \sum_j \frac{h_{ij}}{2}$.

Proof. We can first show that the required condition is sufficient by applying the theorem of Gersgorin disks [26] that ensures that all the eigenvalues of the matrix $\mathbf{B}$ are positive if $b > \sum_j \mid \frac{h_{ij}}{2} \mid$, where we can take out the modulus since $h_{ij} > 0$. The necessary part is demonstrated observing that the vector $\mathbf{X} = [111...1]^T$ is an eigenvector of $\mathbf{B}$, so $\mathbf{BX} = \lambda\mathbf{X}$ where $\lambda = b - \sum_j \frac{h_{ij}}{2}$ which necessarily has to be positive. □

5.3 Parameter Estimation

In order to estimate the parameters of the motion-texture model from motion measurements, we adopt the pseudo-likelihood maximization criterion [2]. Therefore, we search the set of parameters $\hat{\boldsymbol{\phi}}$ that maximizes the function $L(\boldsymbol{\phi}) = \sum_{i \in S} \log p(x_i \mid \mathbf{X}_{\mathcal{N}_i}, \boldsymbol{\phi})$. We use a gradient descent technique for the optimization as the derivatives of L w.r.t $\boldsymbol{\phi}$ are known in closed form.

5.4 Analysis: parameters of the model

Here, we analyze the influence and role of each of the parameters of the proposed model in the structure and shape of the associated spatial random fields. In order to illustrate the expected behaviour from the model equations, we generate synthetic fields using a Gibbs sampler [22].

First we rewrite equation (24) as:

$$\begin{aligned} p(x_i \mid \mathbf{X}_{\mathcal{N}_i}) &= \rho_i . e^{\theta_{1,i} \mathbf{1}_0^*(x_i)} . e^{-\theta_{2,i} x_i^2} . e^{\theta_{3,i} x_i} \\ &= \frac{(\sigma_i \sqrt{2\pi})^{-1}}{(\sigma_i\sqrt{2\pi})^{-1} + e^{\theta_{1,i} + \frac{m_i^2}{2\sigma_i^2}}} \left[e^{\theta_{1,i} \mathbf{1}_0^*(x_i)} . e^{-\theta_{2,i} x_i^2} . e^{\theta_{3,i} x_i} \right] \qquad (34) \end{aligned}$$

With this expression in mind, several conclusions will be extracted and confirmed by experiments

The value of $\theta_2 = b = \frac{1}{2\sigma}$ determines the variance of the continuous part of the motion distribution: when this parameter increases the distribution tends to concentrate around $x = 0$. In fact, with this model the variance doesn't depend on the neighbours so is a global parameter for the whole texture.

In some way it only determines a sort of scale of the modeled data. So in the following we set $b = 1$.

An increase on the value of θ_1 decreases the probability of null samples: observing equation (34) we note that with increasing θ_1 the value of e^{θ_1} increases and the first term tends to zero. Only when in the second term, $\mathbf{1}^*(x_i) \neq 0$ (which means $x_i \neq 0$), the whole distribution doesn't approaches zero, tending to the continuous distribution.

An increase of the magnitude of θ_1 but with negative sign, increases the probability of null samples: in equation (34) the first term tends to 1, and the second tends to zero when $x_i \neq 0$. With $x_i = 0$ the second term is equally 1 and the third and fourth terms too; so the result is that the probability of being null approaches one.

Increasing d_H, d_V, d_D and/or d_{AD} increases the probability of non null samples: in the following it is shown binary images where white pixels correspond to $x \neq 0$ and black pixels to $x = 0$.

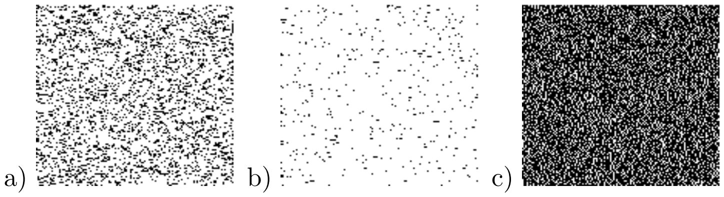

Figure 2: Simulated Gaussian mixed-state spatial fields. Binary (null-nonnull) map: a) $\phi = [0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, b) $\phi = [0\ 1\ 0\ 2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$, c) $\phi = [0\ 1\ 0\ \text{-}2\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$

We observe in the figure 2 a) and b) that with one of the parameters d_k we control the density of non-null samples. It is a consequence of the expression (28) where a positive value of, for example, d_H increases θ_1 when the horizontal neighbours are different of zero. In fact this tends to align pixels with $x \neq 0$ as will be clarified in following simulations: for a point with its neighbours in motion the probability of itself being in motion too, is high. Also note that, with this configuration when the neighbours of a point are null the effect of the particular value of d_k dissappears.

Decreasing d_k increases the probability of null samples: this is the counterpart of the last conclusion. In this case a negative parameter decreases the value of θ_1 when the neighbours are non-null increasing the probability of beeing zero (figure 2 c)). Is a kind of disaligment in the corresponding direction.

Combining a positive and negative d_k results in a oriented texture of particular characteristics: for example, increasing d_H tends to align points with motion in the horizontal direction. At the same time decreasing d_V makes a point tend to zero when its vertical neighbours are not null. Actually, the only neighbours that produce an effect are those for which $x_j \neq 0$: the dependency in the horizontal direction results in aligment of the point values. These points are at the same time, vertical neighbours with $x_j \neq 0$ of some points, so these tend to zero through the effect of the parameter d_V.

In figure 3 a) we observe this effect: alternating lines of motion-no motion appeared because of the mentioned aligment. White lines are a consequence

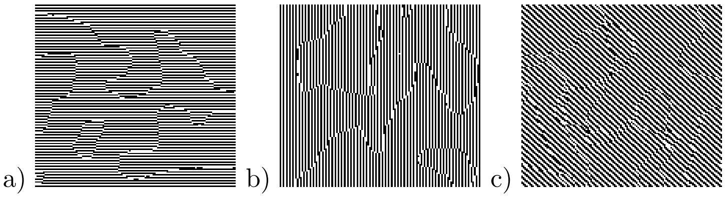

Figure 3: Simulated Gaussian mixed-state spatial fields. Binary (null-nonnull) map: a) ϕ = [0 1 0 5 0 -5 0 0 0 0 0], b) ϕ = [0 1 0 -5 0 5 0 0 0 0 0], c) ϕ = [0 1 0 0 0 0 0 5 0 -5 0]

of the alignment of horizontal neighbours; black lines, of the disalignment of vertical neighbours. The image in b) is the inverse: negative d_H and positive d_V.

In the case of the diagonal neighbours the effect is the same (figure 3 c)): we observe that the aligment results in a diagonal direction.

The orientation of the texture can be defined through a combination of diagonal and lateral coefficients: in this case, we get a texture that is somehow between the diagonal and horizontal orientation (figure 4). In fact the orientation on the binary pattern is manifested through longer segments of non-null (or null) values in one direction or another. Note that small changes in the parameters result in big changes due to the exponential behavior of ρ_i.

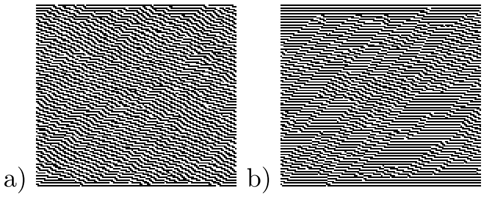

Figure 4: Simulated Gaussian mixed-state spatial fields. Binary (null-nonnull) map: a) ϕ = [0 1 0 5 0 -5 0 4.6 0 -4.6 0], b) ϕ = [0 1 0 5 0 -5 0 5 0 -5 0]

The parameter 'a' works as a null-value "probability offset" : observing (28) we note that if we use a negative value of 'a' we get the following: when the neighbours are null, $\theta_1 = a$ so it results in a negative value also. This leads to increasing the probability of also being null. Choosing, for example, parameters d_H and d_V both positive we obtain high probability of a point beeing non null when more neighbours are non null.

In the figure 5 a) we confirm this: the point tends to form groups or regions over the texture with no orientation because $d_H = d_V$. If we use different values of d_k we observe the orientation effect as we can see for the textures in figures 5 b) and c).

An homogenuous oriented texture can be obtained controlling the mean through h_k. In figure 6 a) we can see that $d_H = 1$ and $d_V = -1$ which defines the orientation. Then with $h_H = 1$ the horizontal neighbours have distributions with a mean depending on the horizontal continuous values. The binary pattern also is oriented. In c) and d) we see a diagonal texture

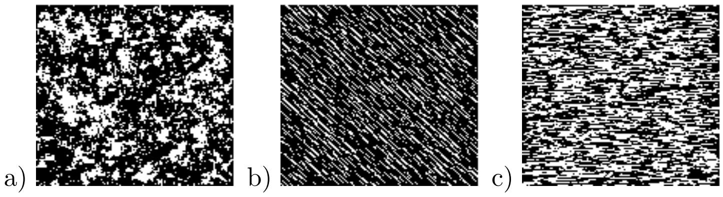

Figure 5: Simulated Gaussian mixed-state spatial fields. Binary (null-nonnull) map: a) ϕ = [-2.9 1 0 1.5 0 1.5 0 0 0 0 0], b) ϕ = [-3.2 1 0 0 0 0 0 3 0 -1 0], c) ϕ = [-2.9 1 0 3 0 0 0 0 0 0 0]

where $d_D = d_{AD}$ and the orientation in the binary pattern is not so clear, but it can be observed in the continuos values due again to the effect of h_D

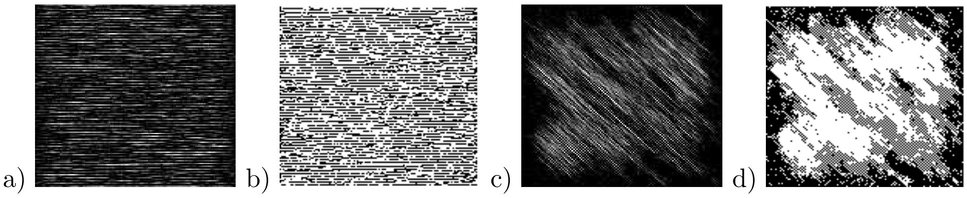

Figure 6: a) Continuous field and b) binary field with ϕ = [-0.2 1 0 1 1 -1 0 0 0 0 0]. c) Random field and d) binary field with ϕ = [-2.8 1 0 0 0 0 0 1 1 1 0]

Then, the result is an oriented textured but not only capturing a binary pattern, also beeing dependent on the specific continuous values, which can be noted in the highly correlated field.

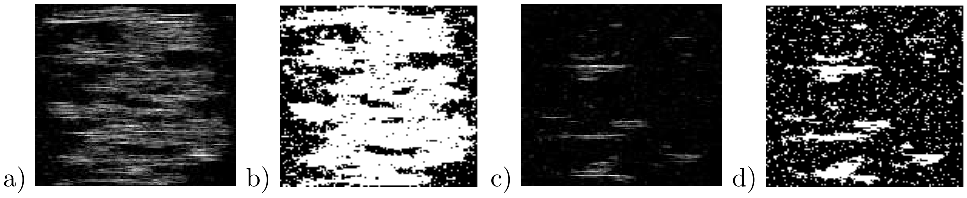

Figure 7: a) Continuous field and b) binary field with ϕ = [-2.8 1 0 1 1 1 0 0 0 0 0]. c) Random field and d) binary field with ϕ = [-2.9 1 0 1 1 1 0 0 0 0 0]

In figures 7, 8 and 9, we present more simulations with different combinations of the model parameters illustrating how we can control the density and orientation of the binary pattern and also the orientation of the continuous values. Note how the presence of the discrete component also allows to generate fields that present regions or blobs of highly correlated values separated by large regions of null values (figures 8 and 9). Some comments have to be done regarding these last results. The generated spatial fields are not evidently homogeneous, but they were generated using an iterative process (Gibbs sampler) from an homogeneous model, with some limited number of repetitions. If we leave the

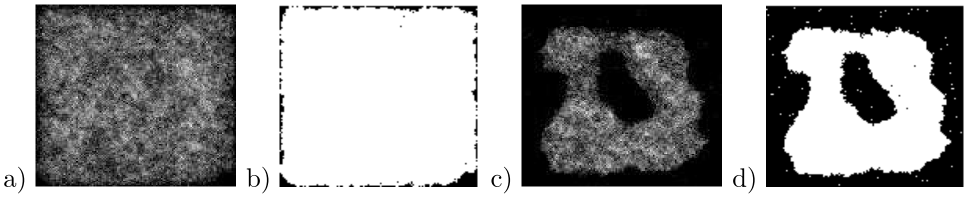

Figure 8: a) Continuous field and b) binary field with ϕ = [-4.5 1 0 1 0.25 1 0.25 1 0.25 1 0.25]. c) Random field and d) binary field with ϕ = [-4.8 1 0 1 0.25 1 0.25 1 0.25 1 0.25]

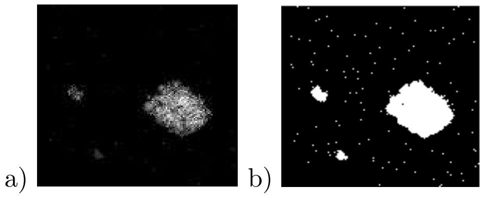

Figure 9: a) Continuous field and b) binary field with ϕ = [-4.9 1 0 1 0.25 1 0.25 1 0.25 1 0.25]

Table 1: Motion texture parameters

Texture	a	b	c	d_H	h_H	d_V	h_V	d_D	h_D	d_{AD}	h_{AD}
Trees	-2.753	5.632	0.041	1.055	2.905	1.087	1.709	0.352	2e-19	0.411	0.962
Leaves	-2.503	2.213	0.009	0.757	0.382	0.971	0.163	0.214	-9e-20	0.297	0.173
Grass	-3.574	2.594	0.003	1.114	1.077	1.115	1.307	0.583	0.183	0.511	3e-19
Sea	-2.038	3.235	-0.026	1.552	3.203	0.363	1e-19	0.068	1e-19	-0.035	2-19

Gibbs sampler to continue, the simulated field would result, for example, in a mostly null map. However, if we estimate the parameters from fields as those in figures 8 and 9, the values are close to the ones used in the simulation, showing that is valid to characterize them with the proposed model. The idea here was to show what type of different properties of the fields can be captured by the parameters.

5.5 Analysis: parameter estimation of real motion textures

We estimated the model for several real scenes of natural textures, obtaining the corresponding parameters. Then we utilized those parameters to synthesize new textures in order to compare how the model can capture the properties of the motion textures. A Gibbs sampler was used to obtain the simulated textures.

The results are shown in figures 10, 11, 12 and 13 . Each figure represents a different real scene. In b) we present the real residual motion map; c) is the real binary pattern of motion-no motion (white equals a moving point); d) and e) are respectively, the simulated motion and simulated binary pattern. We include the values of the estimated coefficients in Table 1.

We can see at first that the four textures can be differentiated in the sense that, when we generate the synthesized textures, the configurations are visually different. Moreover, the artificial patterns are quite similar to the originals and specially for the binary (motion-no motion) configuration.

In the Tree scene (figure 10) we have good results for both binary and continuous values. For the Leaves texture in figure 10, the model apparently estimates well the binary pattern, of course not capturing the shape of the leaves, but is good with respect to the density of moving points. The synthesized residual motion values are not so similar to the original, which doesn't mean that the set of parameters can not differentiate this texture. In the case of the Grass sequence (figure 12), which is in fact a more complex texture, we can see that in the real scene there are some sort of blobs that can be somehow captured by the model. Evidently this texture is not very homogeneous. Finally for the Sea scene, note that the parameters related to horizontal neighbors are clearly greater than those related to other directions showing how the model is capable of capturing the orientation of the field.

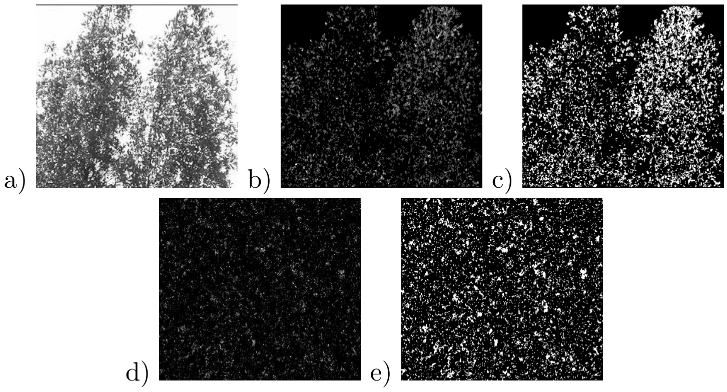

Figure 10: Tree scene: a) Original image, b) real residual motion, c) real motion-no motion map, d) synthesized residual motion, e) synthesized motion-no motion map

6 A simmilarity measure between mixed-state models

6.1 An expression for general Gibbs random fields

One of the key aspects of a model oriented to dynamic content recognition, as the one proposed here, is the ability to define a way of computing some simmilarity measure between models, in order to embed it in a decision-theory-based application. In this context, the Kullback-Leibler divergence is a well-known distance (more precisely, a pseudo-distance) between statistical models. Here, we present a result for general Gibbs distributions and obtain an expression for the case of mixed-state models, that allows us to compute this quantity.

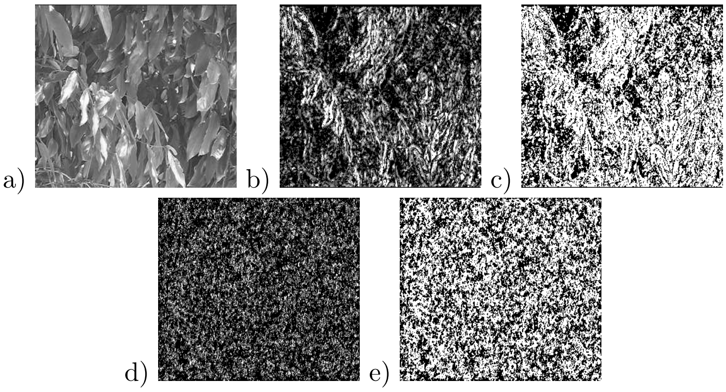

Figure 11: Leaves scene: a) Original image, b) real residual motion, c) real motion-no motion map, d) synthesized residual motion, e) synthesized motion-no motion map

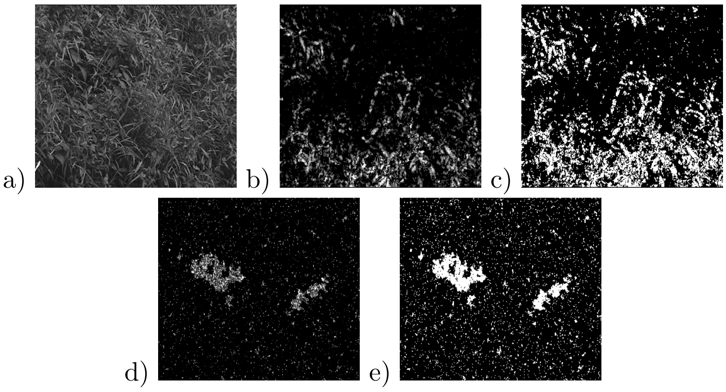

Figure 12: Grass scene: a) Original image, b) real residual motion, c) real motion-no motion map, d) synthesized residual motion, e) synthesized motion-no motion map

Let $p_1(\mathbf{X})$ and $p_2(\mathbf{X})$ be two probability densities with $\mathbf{X} \in \Omega$, given w.r.t some measure $m(d\mathbf{X})$. Then the Kullback-Leibler divergence from p_1 to p_2 is defined as

$$KL(p_1(\mathbf{X}) \mid p_2(\mathbf{X})) = \int_\Omega \log \frac{p_1(\mathbf{X})}{p_2(\mathbf{X})} p_1(\mathbf{X}) m(d\mathbf{X}), \tag{35}$$

which is independent of the measure m. Define the symmetrized KL divergence as:

$$d_{KL}(p_1(\mathbf{X}), p_2(\mathbf{X})) = \frac{1}{2} \left[KL(p_1(\mathbf{X}) \mid p_2(\mathbf{X})) + KL(p_2(\mathbf{X}) \mid p_1(\mathbf{X})) \right]. \tag{36}$$

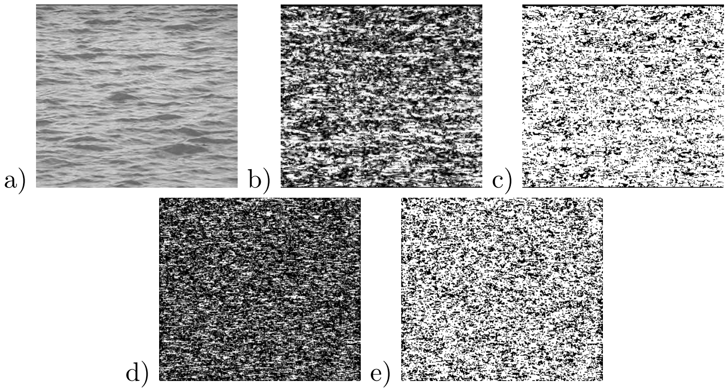

Figure 13: Sea scene: a) Original image, b) real residual motion, c) real motion-no motion map, d) synthesized residual motion, e) synthesized motion-no motion map

Now, assume that $p_1(\mathbf{X})$ and $p_2(\mathbf{X})$ are Markov random fields, i.e. $p_1(\mathbf{X}) = Z_1^{-1} \exp Q_1(\mathbf{X})$ and $p_2(\mathbf{X}) = Z_2^{-1} \exp Q_2(\mathbf{X})$. Define $\Delta Q(\mathbf{X}) = Q_2(\mathbf{X}) - Q_1(\mathbf{X})$. Then,

$$\begin{aligned} \log \frac{p_1(\mathbf{X})}{p_2(\mathbf{X})} &= -\Delta Q(\mathbf{X}) + \log \frac{Z_2}{Z_1}, \\ \log \frac{p_2(\mathbf{X})}{p_1(\mathbf{X})} &= \Delta Q(\mathbf{X}) - \log \frac{Z_2}{Z_1}. \end{aligned} \tag{37}$$

Thus,

$$\begin{aligned} d_{KL}(p_1(\mathbf{X}), p_2(\mathbf{X})) &= \frac{1}{2}\left(E_{p_1}\left[\log \frac{p_1(\mathbf{X})}{p_2(\mathbf{X})}\right] + E_{p_2}\left[\log \frac{p_2(\mathbf{X})}{p_1(\mathbf{X})}\right]\right) \\ d_{KL}(p_1(\mathbf{X}), p_2(\mathbf{X})) &= \frac{1}{2}\left(E_{p_1}\left[-\Delta Q(\mathbf{X}) + \log \frac{Z_2}{Z_1}\right] + E_{p_2}\left[\Delta Q(\mathbf{X}) - \log \frac{Z_2}{Z_1}\right]\right) \\ d_{KL}(p_1(\mathbf{X}), p_2(\mathbf{X})) &= \frac{1}{2}\left(E_{p_2}\left[\Delta Q(\mathbf{X})\right] - E_{p_1}\left[\Delta Q(\mathbf{X})\right]\right) \end{aligned} \tag{38}$$

We observe from this general equation, that we do not need to have knowledge of the partition functions of the Gibbs distributions which simplifies enormously the handling of this expression.

6.2 A result for Gaussian MS-MRF

Let $p_1(\mathbf{X})$ and $p_2(\mathbf{X})$ be two Gaussian MS-MRF. Then,

$$\begin{aligned} Q_1(\mathbf{X}) = \sum_i V_i^{(1)}(x_i) + \sum_{<i,j>} V_{i,j}^{(1)}(x_i, x_j) \\ Q_2(\mathbf{X}) = \sum_i V_i^{(2)}(x_i) + \sum_{<i,j>} V_{i,j}^{(2)}(x_i, x_j) \end{aligned} \tag{39}$$

where

$$\begin{aligned} V_i^{(k)}(x_i) &= \boldsymbol{\alpha}^{(k)T}\mathbf{S}(x_i) = a^{(k)}\mathbf{1}^*(x_i) - b^{(k)}x_i^2 + c^{(k)}x_i \\ V_{i,j}^{(k)}(x_i, x_j) &= \mathbf{S}^T(x_i)\boldsymbol{\beta}_j^{(k)}\mathbf{S}(x_j) = d_j^{(k)}\mathbf{1}^*(x_i)\mathbf{1}^*(x_j) + h_j^{(k)}x_ix_j \end{aligned} \qquad (40)$$

and $\mathbf{S}(x_i) = \begin{bmatrix} \mathbf{1}^*(x_i) & -x_i^2 & x_i \end{bmatrix}^T$, $\boldsymbol{\alpha}^{(k)} = [a^{(k)}, b^{(k)}, c^{(k)}]^T$, $\boldsymbol{\beta}_j^{(k)} = \begin{bmatrix} d_j^{(k)} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & h_j^{(k)} \end{bmatrix}$.

Here, j is an index that stands for one of the n (usually 8) neighbors of site i. Then,

$$\Delta Q(\mathbf{X}) = Q_2(\mathbf{X}) - Q_1(\mathbf{X}) = \sum_i \Delta\boldsymbol{\alpha}\mathbf{S}(x_i) + \sum_{<i,j>} \mathbf{S}(x_i)\Delta\boldsymbol{\beta}_j\mathbf{S}(x_j) \qquad (41)$$

where $\Delta\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(2)} - \boldsymbol{\alpha}^{(1)}$ and $\Delta\boldsymbol{\beta}_j = \boldsymbol{\beta}_j^{(2)} - \boldsymbol{\beta}_j^{(1)}$. From this definition:

$$\begin{aligned} E_{p_1}[\Delta Q(\mathbf{X})] &= \sum_i \Delta\boldsymbol{\alpha} E_{p_1}[\mathbf{S}(x_i)] + \sum_{<i,j>} E_{p_1}\left[\mathbf{S}(x_i)\Delta\boldsymbol{\beta}_j\mathbf{S}(x_j)\right] \\ &= \sum_i \Delta a E_{p_1}[\mathbf{1}^*(x_i)] - \Delta b E_{p_1}\left[x_i^2\right] + \Delta c E_{p_1}[x_i] \\ &+ \sum_{<i,j>} \Delta d_j E_{p_1}[\mathbf{1}^*(x_i)\mathbf{1}^*(x_j)] + \Delta h_j E_{p_1}[x_ix_j] \end{aligned} \qquad (42)$$

As we have an homogeneous model:

$$\begin{aligned} E_{p_1}[\Delta Q(\mathbf{X})] &= N\Delta a E_{p_1}[\mathbf{1}^*(x_i)] - N\Delta b E_{p_1}\left[x_i^2\right] + N\Delta c E_{p_1}[x_i] \\ &+ \frac{N}{2}\sum_j \Delta d_j E_{p_1}[\mathbf{1}^*(x_i)\mathbf{1}^*(x_j)] + \Delta h_j E_{p_1}[x_ix_j] \end{aligned} \qquad (43)$$

The same result applies to $E_{p_2}[\Delta Q(\mathbf{X})]$ and then it is straightforward to compute equation (38). In a practical application, the idea is to use the parameters of the two models that we want to compare, to generate synthetic fields using a Gibbs sampler from which we can estimate the involved expectations.

7 Partition function calculation

In this section, we propose a new approach for solving the problem of partition function calculation for Gibbs distributions. This issue is crucial as it will enable to properly handle the optimization of the global energy function which will be defined for the motion texture segmentation problem.

In the case of image processing applications, it is usual to analyze distributions of different data classes in a decision-theory framework, as it occurs in detection, segmentation and classification problems. Thus, it is fundamental to accurately know the partition function in order to allow the comparison of different models.

For a general Gibbs distribution, the expression for the normalizing factor is:

$$Z = \int_{\Omega} e^{Q(\mathbf{X})} m(\mathbf{X}), \tag{44}$$

where $\mathbf{X} = [x_1....x_N]^T$ is the vector of random variables that form the field and Ω is the sample space. Define, $\mathbf{X}_A$ with $A \subset S$, a subset of random variables restricted to the set of sites A. Let $\Delta Q(\mathbf{X}_A)$ be a variation on the energy function, not necessarily small, due to an arbitrary variation in the field and being a function only of $\mathbf{X}_A$. Then we can write the resulting partition function, Z', as a function of the former one, Z:

$$\begin{aligned} Z' &= \int_{\Omega} e^{Q'(\mathbf{X})} m(\mathbf{X}) = \int_{\Omega} e^{Q(\mathbf{X})+\Delta Q(\mathbf{X}_A)} m(\mathbf{X}), \\ &= \int_{\Omega} Zp(\mathbf{X}) e^{\Delta Q(\mathbf{X}_A)} m(\mathbf{X}), \\ &= ZE_{\mathbf{X}_A}\left[e^{\Delta Q(\mathbf{X}_A)}\right], \end{aligned} \tag{45}$$

where the expectation operator is applied w.r.t. the marginal probability distribution of $\mathbf{X}_A$. Conceptually, the approach consists in estimating the unknown partition function from reference values which are either known in closed form or can be easily estimated.

7.1 A result for mixed-state models

In c-class segmentation applications, it is common to formulate the problem as an energy optimization one. For model-based segmentation of images, some information about the partition function for each class must be known. Available optimization methods are mostly based on iteratively computing energy changes as a result of adding or taking out points from a class. Defining ΔQ appropriately, we then have an expression for the change on the normalizing factor.

In the case of the mixed-state models we define here, taking out a point x_i from a class is equivalent to extracting the cliques corresponding to that point, and for the sake of obtaining the partition function, the integral must be calculated with respect to one less variable. This change in Z can be expressed setting:

$$e^{\Delta Q(\mathbf{X}_A)} = \mathbf{1}_0(x_i) e^{-V_i(x_i) - \sum_{j \in \mathcal{N}_i} V_{i,j}(x_i,x_j)}. \tag{46}$$

Here, $\mathbf{1}_0(x_i)$ allows integrating also with respect to x_i without changing the value of the integral. Then, knowing that $V_{i,j}(0, x_j) = 0$ and $V_i(0) = 0$, the exponent in equation (46) is null for $x_i = 0$, and consequently, $e^{\Delta Q(\mathbf{X}_A)} = e^{\Delta Q(x_i)} = \mathbf{1}_0(x_i)$. Finally:

$$\begin{aligned} Z' &= ZE_{x_i}\left[e^{\Delta Q(x_i)}\right] = ZE_{x_i}\left[\mathbf{1}_0(x_i)\right], \\ &= ZP(x_i = 0). \end{aligned} \tag{47}$$

Equivalently, we can calculate the change on the value of the partition function due to an extraction of an arbitrary subset T of points from the field, redefining equation (46) adequately, introducing the indicator function $\mathbf{1}_0(x_i)$ for each point. Following the same reasoning, we arrive to:

$$Z' = ZP(\mathbf{X}_T = \mathbf{0}). \tag{48}$$

We have come to a relation for the change on the partition function due to an extraction of a subset of points from a mixed-state field. Let us point out that if the sites would be independent, the partition function would take the form $Z = Z_0^N$ for homogeneous fields, where Z_0 is the partition function for a single point, taking the value $Z_0 = 1/P(x_i = 0)$, and N is the total number of points. This is consistent with equation (48).

8 Application: classification of motion textures

Up to this point, we have seen how the proposed model can deal with different mixed-state configurations, even in real situations, suggesting that it is somehow able to be applied in recognition and classification of motion textures. As stated in the previous sections, a motion texture mixed-state model is defined by 11 parameters. In this section, we adopt the Euclidean distance between parameter vectors to compute the difference between two motion textures. We choose not to use the expression for the Kullback-Leibler divergence obtained previously, in order to simplify and speed-up the classification process, knowing that for these experiments, results are similar in both cases. While this approach may seem sub-optimal, the promising results support its application.

The recognition performance of mixed-state motion texture models was tested with real sequences extracted from the DynTex dynamic texture database [34]. We first took motion textures where the homogeneity assumption was mostly valid and divided them in 10 different classes (Figure 14): steam, straw, traffic, water, candles, shower, flags, water-rocks, waves, fountain. A total of 30 different sequences were considered, and for each one, 5 pairs of consecutive images were selected at frames 1,20,40,60,80, for a total of 150 samples. Each motion texture class parameter set was learned from a single pair of images picked from only one of the sequences belonging to each type of motion texture. All sequences were composed by gray scale images with a resolution of 720x576 pixels, given at a rate of 25 frames per second. In order to reduce computation time, the original images were filtered and subsampled to a resolution of 180x144 pixels.

Having estimated the reference model parameters for each of the 10 training samples, we then estimated $\boldsymbol{\phi}$ for each test sample and computed the distance with each learned parameter vector. The recoginition was based on assigning the class of motion texture that was closer to the test sample.

In Table 2 we show the confusion matrix for the 10 motion texture classes. A correct recognition is considered when both, the test sample and the closest reference parameter vector, belong to the same class.

A promising overall classification rate of 89.29% was achieved. As for the confusion matrix, let us note that it is likely that waves are classified as water or water-rocks as they correspond to similar dynamic contents, straw is confused

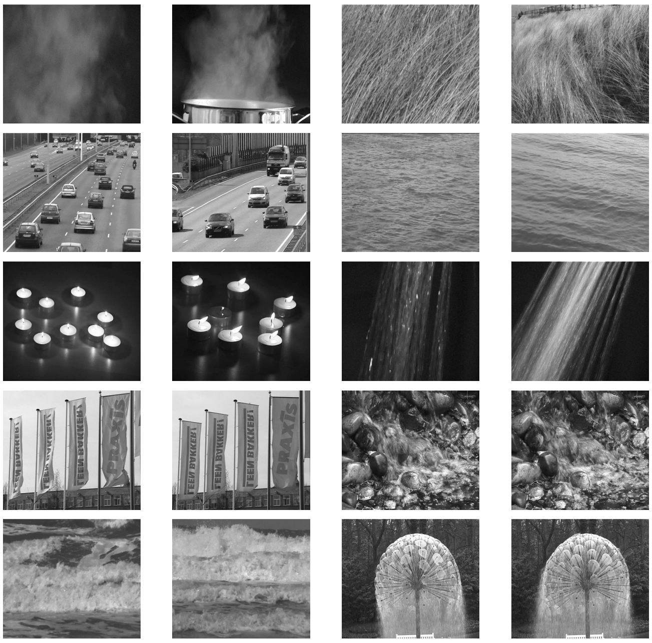

Figure 14: Sample images from the 10 motion texture classes used for the recognition experiments.

with shower, they have similar vertical orientation, and candles can be classified as traffic, as both classes show a motion pattern consisting of isolated blobs.

The non-symmetry of the confusion matrix is associated to the nature of the tested data set, where for some classes, the tested sequences have a closer resemblance to the training sample, while for others, there are notorious variations, that may lead to a misclassification.

Table 2: Motion texture class confusion matrix

	steam	straw	traffic	water	candles	shower	flags	water-rocks	waves	fountain
steam	100%	-	-	-	-	-	-	-	-	-
straw	-	95%	-	-	-	5%	-	-	-	-
traffic	-	-	80%	-	-	-	-	20%	-	-
water	-	-	-	100%	-	-	-	-	-	-
candles	-	-	10%	30%	60%	-	-	-	-	-
shower	20%	-	-	-	-	80%	-	-	-	-
flags	-	-	-	-	-	-	86.7%	13.3%	-	-
water-rocks	-	-	-	-	-	-	-	100%	-	-
waves	-	-	-	6.7%	-	-	-	26.6%	66.7%	-
fountain	-	-	-	-	-	-	-	-	-	100%

9 Application: motion texture segmentation

One important aspect that the model should fulfill for dynamic content analysis, is the ability of discrimination. In this context, the problem of segmentation, closely related to classification, aims at determining and locating regions in the image that correspond to a same motion texture or class.

Up to our knowledge, existing segmentation methods of different classes of dynamic textures are based on the linear models proposed originally by Soatto *et al.* [16]. In [17], a level-set variational approach is proposed for minimizing a cost functional w.r.t. some geometric discrepancy measures between models and the boundary of the different regions. The results are good, but this method is not able to capture temporal variations on the regions occupied by each of the dynamic textures, due to the temporal homogeneity assumption. Moreover, it tends to require a high computational cost for the minimization procedure. Vidal and Ravichandran [43] propose to model each dynamic texture as a linear system plus a 2-D translational motion component. This solves the problem of moving regions (or non-static camera) under rigid motion. Segmentation is achieved by a Principal Component Analysis (PCA) method, for subspace separation. The learning of the models is still closely attached to considering a sufficiently large time window. Finally, in [10] a mixture generative model of several linear dynamic textures is proposed, which naturally leads to a segmentation approach.

In our method, the representation of a motion texture with a relatively small set of parameters permits for a parsimonious characterization of the different parts of a scene consisting of more than one dynamic texture. Moreover, the lack of a temporal homogeneity assumption will allow us to deal with the problem cited in the previous paragraph, and to overcome some of the limitations of the existing segmentation approaches.

9.1 The method

As we deal with lattice systems, the problem is equivalent to assign a label to each point in the image grid, indicating that it belongs to a certain motion texture. Here, we follow a Bayesian approach for determining in an optimal way the distribution of the motion texture labels, related to the segmentation of the field, with the motion map as input data. Thus, we search for a label realization $\mathbf{l} = \{l_i\}$, where $l_i \in \{0, 1, ..., c-1\}$ is the class label value, that maximizes $p(\mathbf{l} \mid \mathbf{X}) \propto p(\mathbf{X} \mid \mathbf{l})p(\mathbf{l})$, where $\mathbf{X}$ represents the motion map including several motion textures. This corresponds to a MAP (maximum-a-posteriori) estimation of the label field $\mathbf{l}$.

If we suppose that the c different motion textures come from independent dynamic phenomena, given the label field, we can write:

$$p(\mathbf{X} \mid \mathbf{l}) = \prod_{k=0}^{c-1} p(\mathbf{Y}_k) = \prod_{k=0}^{c-1} \frac{e^{Q_k(\mathbf{Y}_k)}}{Z_k(\mathbf{l})}. \tag{49}$$

In the proposed method we do not assume conditional independence given the label field, within a motion texture. We introduce the following notation: we call Q_k the energy function corresponding to the texture class k; define $\mathbf{Y}_k = \{x_i : l_i = k\}$ as the vector of motion random variables that belong to texture

k, and that is a subset of $\mathbf{X}$; $Z_k(\mathbf{l})$ is the corresponding partition function, and depends on the distribution of a texture on the lattice, through the label field $\mathbf{l}$. That is, each texture k has its Gibbs distribution, that depends on the parameters $\boldsymbol{\phi}_k$ of that class k, and on the region that it occupies. This approach allows us to account for conditional dependence. The only approximation is that we do not formally account for interaction between different textures, but this concerns only few pixels of the image grid.

For the *a priori* information on the segmentation label field, $p(\mathbf{l})$, we introduce another 8 nearest-neighbour Markov random field that behaves as a regularization term for the labeling process. We have $p(\mathbf{l}) \propto exp[Q_S(\mathbf{l})]$ with:

$$Q_S(\mathbf{l}) = \sum_i \sum_{j \in \mathcal{N}_i} \gamma \mathbb{I}_0(l_i - l_j), \tag{50}$$

where $\mathbb{I}_0(z)$ is the null argument indicator function. $p(\mathbf{l})$ penalizes the differences of labeling between adjacent neighbors, smoothing the segmentation output. The complete formulation can be stated as maximizing the energy:

$$E(\mathbf{l}) = \sum_{k=0}^{c-1} Q_k(\mathbf{Y}_k) - \sum_{k=0}^{c-1} \log(Z_k(\mathbf{l})) + Q_S(\mathbf{l}). \tag{51}$$

9.2 Energy maximization methods

Maximizing equation (51) is a very challenging task, and addressing equivalent problems has been an area of increasing interest for image processing applications. Exhaustive search of the optimal discrete field (or labels) is impossible to handle and, at the same time, the non-convex shape of the energy functions, is prohibitive for using simple optimization techniques.

Different general methods for energy maximization have been proposed. Simulated annealing [22] is a general-purpose stochastic relaxation technique, suitable for virtually any shape of the energy function. While its simple implementation provides a good first-approach for estimating the label field, it is slow and not that efficient. Moreover, despite theoretic convergence is guaranteed, in practice, it is very difficult to reach a global maximum. For energy functions related to conditional models, i.e., those arising from Markov random fields, Iterative Conditional Modes (ICM) [3] is based on maximizing the conditional local density for a point w.r.t its label, and is faster and much less time consuming than simulated annealing. However, it relies on a strong assumption, that, given the label field, the observations are conditionally independent. This is not the case of the segmentation method proposed here (see equation 49) where we exploit the whole motion texture model for each class. The ICM assumption is valid when the knowledge of the label for a point is sufficient for writing its generative density.

9.3 Graph-cut based energy maximization

In the last years, a new family of energy optimization methods, based on *graph-cuts* has been proposed and developed with a growing impact in many applications [5, 23, 29, 40]. One of the first uses of this approach was described by Greig *et al.* [23] where a binary energy is reformulated as a minimum cut problem in capacitated networks.

This technique can be applied for assigning labels to points in the image grid, that is, for obtaining a segmentation of the field. In this framework, one seeks the labeling $\mathbf{l}$ that maximizes (or eventually, minimizes) energy functions of the type:

$$E(\mathbf{l}) = \sum_{i \in S} D_i(l_i) + \sum_{<i,j>} E^{ij}(l_i, l_j), \tag{52}$$

where $D_i(l_i)$ is a function that takes into account the energy term associated to assigning the label l_i to image point x_i and that is derived from the observed data. $E^{ij}(l_i, l_j)$ is a term that measures the cost of assigning labels l_i, l_j to neighboring points and is related to spatial smoothness.

The method relies on constructing a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, with vertices $\mathcal{V}$ and edges $\mathcal{A}$. The set of vertices include the images points and two special vertices, namely, *terminal vertices*. Then, we define a set $C \in \mathcal{A}$ of edges of the graph, which we refer as a *cut*, that splits $\mathcal{G}$ into two disjoint sets in such a way that each set contains one of the terminal vertices. The result is an induced graph $\mathcal{G}' = (\mathcal{V}, \mathcal{A} - C)$. From all the possible partitions of this kind, the minimum cut is defined as the one that has the smallest cost $|C|$, i.e., the sum of costs of all edges in C.

Finally, it is shown that there is a one-to-one correspondence between a partition of $\mathcal{G}$ and a complete binary labeling of the image points. Moreover, assigning appropriate edge weights, obtaining a minimum cut is equivalent to optimizing $E(\mathbf{l})$. Thus, the formulation reduces to computing a min-cut/max-flow problem. Note that this construction is only valid for binary labels.

Among the methods that exploit this equivalence, different approaches were employed for constructing appropriate graphs, many of them being problem-dependent and complex as well. However, in the last few years, the group of Zabih *et al.* [5, 29], have proposed a very efficient method for energy maximization based on graph cuts providing, at the same time, a general graph construction scheme. The method, called *expansion move* algorithm, is valid for multi-label situations, and is based on iteratively computing an expansion of the region occupied by a label class $l \in \{0, ..., c-1\}$ in the current labeling, in such a way that this change generates a decrease in the global cost function. The expansion algorithm cycles through all the labels and finds the optimal expansion by application of the graph-cut method. The algorithm terminates when there is no expansion move, for any label class, with lower energy. It is worthy to mention that this approach can be applied to multi-label situations, as one can define, for each iteration and each label l, an auxiliary label, namely not-l, and reduce the step to a binary expansion move.

In this paper, we propose to use this technique for efficiently addressing the problem of motion texture segmentation. In our case we have to rewrite equation (51) in the form of (52). Let us first consider the case of two classes ($c = 2$). Then we have:

$$E(\mathbf{l}) \quad = \quad Q_1(\mathbf{x}_1) + Q_2(\mathbf{x}_2) - \log\left(Z_1(\mathbf{l})\right) - \log\left(Z_2(\mathbf{l})\right) + Q_S(\mathbf{l}), \tag{53}$$

where for each class $k \in \{0, 1\}$,

$$Q_k = \sum_{i \in S_k} V_i^{(k)}(x_i) + \sum_{<i,j> \in S_k} V_{ij}^{(k)}(x_i, x_j). \tag{54}$$

$V_i^{(k)}$ and $V_{ij}^{(k)}$ are the corresponding potentials for each motion texture model and $S_k \in S$ is the subset of N_k points that belong to texture k. For the partition function, observe equation (47). Note that the probability $P(x_i = 0)$ is computed w.r.t. the marginal distribution for the site, before taking out the point. At the same time, we can assume that this marginal density remains approximately constant as we successively take points from a class, as long as the parameters that describe the remaining field do not vary too much. Of course, the extreme case of leaving only one site in the class, will violate this assumption. Finally, note in (48) that this approximation allows us to write, $P(\mathbf{X}_T = \mathbf{0}) \approx [P(x_i = 0)]^{N_T}$, where N_T is the number of extracted points. As $P(x_i = 0)$ is considered constant, we can assume that each point contributes equally to the energy change in each iterative step of the expansion move algorithm, and we then define:

$$\begin{aligned} D_i(l_i) &= V_i^{(l_i)}(x_i) + \log P(x_i = 0 \mid l_i), && (55) \\ E^{ij}(l_i, l_j) &= \mathbb{I}_0(l_i - l_j)\left[\gamma + V_{ij}^{(l_i)}(x_i, x_j)\right]. && (56) \end{aligned}$$

Thus, we have come to a formulation of the energy function that allows the application of graph cut algorithms for motion texture segmentation. One more aspect has to be considered, related to this approach. The type of energies that can be handled by these methods, are restricted to graph-representable functions [29]. That is, the energy that we want to maximize should be plausible of being asociated to the cuts on a graph. Consequently, *regularity* conditions arise, over the second order terms:

$$E^{ij}(0,0) + E^{ij}(1,1) \geq E^{ij}(0,1) + E^{ij}(1,0), \tag{57}$$

where 0 and 1 are the two binary labels. From equation (56), we observe that this condition reduces to

$$V_{ij}^{(0)}(x_i, x_j) + V_{ij}^{(1)}(x_i, x_j) \geq -2\gamma. \tag{58}$$

It is clear that, as the potential functions can eventually take arbitrary values, this condition will not be always satisfied. However, experiments on real motion texture segmentation, showed that this violation to the energy regularity, occurs in less than 5% of the points of the lattice, and consequently, the problem can be overcome with no impact in the final result. Moreover, note that the smoothing term coefficient γ acts also as a regularity factor for the energy terms, for which larger values decrease the possibility of a non-regular term. It is possible that more complex regularization schemes (non-constant) are required for ensuring the complete regularity of the cost function. We will address this problem in a future work.

9.4 Experiments on segmentation of motion textures

9.4.1 Initialization

The formulation of the segmentation problem we are dealing with, does not assume that the motion texture parameters for each class are known. Then, it is necessary to correctly estimate the intervening mixed-state motion texture models. As a simplification, we will assume that the number of classes is known.

As an initialization of the label field, we divide the motion map in square blocks of some fixed size, and for each block a set of motion-texture model parameters is estimated. Then, we apply a clustering technique to obtain a first splitting of blocks.

In this step, the selection of the distance between clusters is crucial. An appropriate distance is the symmetrized Kullback-Leibler (KL) divergence between probability densities. The ideal situation would be to calculate it between the joint probability distributions (the global distribution on each initial block) of the motion textures we want to segment. But it can be time consuming, specially for an initialization step that gives a rough segmentation.

Here, we adopt a simplified approach. We calculate the KL distance between marginal distributions $p(x_i)$ for single points. Assuming homogeneity and that the distribution for a pixel is a mixed-state Gaussian density, one can estimate easily its parameters (probability of null value, mean and variance of the Gaussian density). Having the parameters is also easy to obtain the divergence expression in closed form.

For the marginal distribution, let us recall that we have $p(x) = \rho\delta_0(x) + (1-\rho)\mathcal{N}(\mu,\sigma)$, where $\mathcal{N}(\mu,\sigma)$ represents a Gaussian density with mean μ and variance σ^2. This results in the following expression for the Kullback-Leibler divergence of a density p_1 with respect to a density p_2, due to the difference in model parameters:

$$\begin{aligned} KL(p_1(x)\|p_2(x)) &= \int_{-\infty}^{\infty} \log\left[\frac{p_1(x)}{p_2(x)}\right] p_1(x)dx, \\ &= \rho_1 \log\left[\frac{\rho_1}{\rho_2}\right] + (1-\rho_1)\left[\log\left[\frac{\sigma_2(1-\rho_1)}{\sigma_1(1-\rho_2)}\right] + \right. \\ &+ \left. \frac{1}{2}\left[\frac{\sigma_1^2}{\sigma_2^2} + \frac{(\mu_2-\mu_1)^2}{\sigma_2^2} - 1\right]\right]. \end{aligned} \quad (59)$$

In fact this expression is not strictly a distance as it is not symmetric, so that we use the symmetrized version $KL(p_1(x), p_2(x)) = \frac{1}{2}(KL(p_1(x)\|p_2(x)) + KL(p_2(x)\|p_1(x)))$.

A partition around medoids (PAM) [44] clustering algorithm is used. Similar to the k-means method, it allows operating over a dissimilarity matrix between samples, which we easily obtain for the set of blocks, from (59). In each iteration, a representative block (medoid) of each cluster is chosen as the one that minimizes its average distance w.r.t. the rest of the elements of the same class. Then, each block is classified according to the closest medoid. The algorithm converges when the representative block is not changed for any cluster. At this stage, it is easy to discard blocks that are likely to be unreliable representatives of the motion texture: we compute the diameter D of each cluster as the maximum distance between any two of its elements, and we keep only as valid

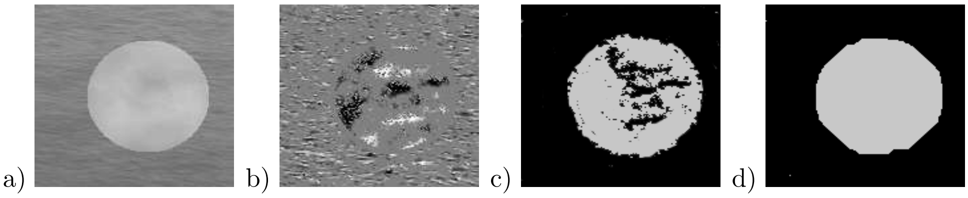

Figure 15: Ocean-steam sequence: a) image from the original sequence, b) motion map, c) result of the segmentation process using simulated annealing, d) result of the segmentation process using the graph-cut based method.

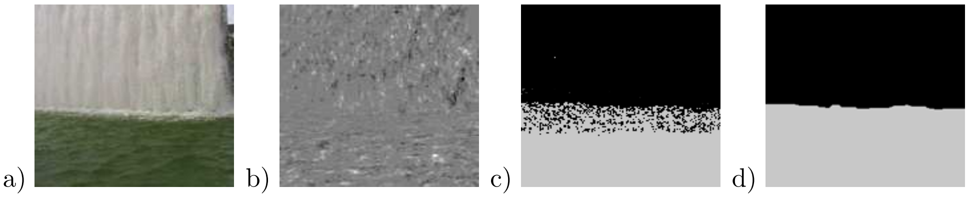

Figure 16: Fountain sequence: a) image from the original sequence, b) motion map, c) result of the segmentation process using simulated annealing, d) result of the segmentation process using the graph-cut based method.

blocks those that have a distance to the medoid lower than some fraction of D, typically 0.3.

Once we have a first segmentation of the field by clustering, we re-estimate the parameters for each final cluster using the valid blocks, in order to obtain a new set of motion-texture model parameters that represent each class. Then, the process of segmentation by energy minimization is started.

9.4.2 Segmentation of real motion textures

In this section we present results on segmentation of real motion textures. From a pair of consecutive images we compute a motion map to which the proposed method is applied.

In Fig. 15 we test the segmentation algorithm on a sequence consisting of two motion textures (c=2), where a circular region asssociated to smoke was superimposed to a sequence of waves. The images are of a size of 150x150 pixels and we obtain the motion map, as seen in Fig. 15b). For the initialization of the algorithm, blocks of 30x30 pixels where used. In Fig. 15c) and 15d) we observe the segmented regions for the cases of energy minimization using graph-cuts and simulated annealing, respectively.

Secondly, we present results for the fountain-water sequence (Fig. 16). This is a real scene with presence of two motion textures. Fig. 16b) shows that both regions display motion patterns with different orientations (vertical for the fountain, horizontal for the water) which can be differentiated by the model. The size of the images is 320x240 and we have used a block size of 40x40.

In Fig. 17, we show a scene consisting of two overlapping trees moved by the wind. This is a complex scene since the trees have not only similar intensity

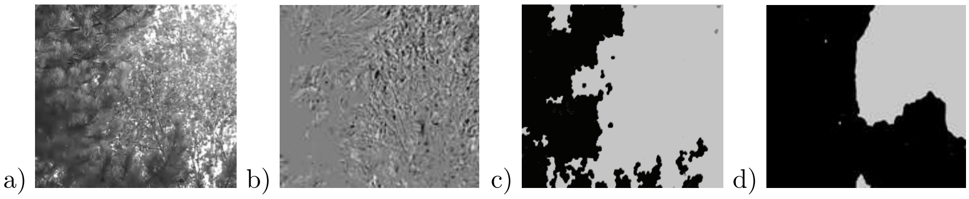

Figure 17: Tree sequence: a) image from the original sequence, b) motion map, c) result of the segmentation process using simulated annealing, d) result of the segmentation process using the graph-cut based method.

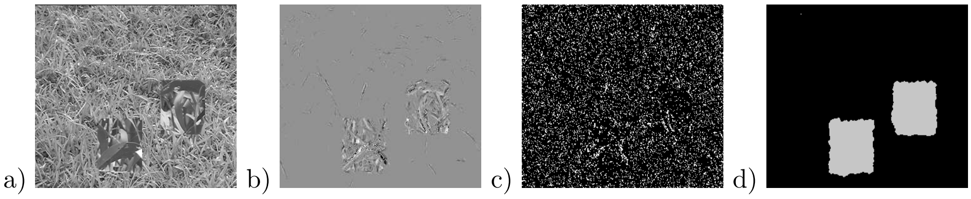

Figure 18: Grass-leaves sequence: a) image from the original sequence, b) motion map, c) result of the segmentation process using simulated annealing, d) result of the segmentation process using the graph-cut based method.

textures, but also similar motion textures (Fig. 17b)). Even for the human eye, it is difficult to separate both regions.

Finally, we display results for an artificially composed motion texture consisting of a sequence of grass, and two small regions of moving leaves (Fig. 18). The results for simulated annealing are poor, compared to the graph-cut method, for which the motion field is well segmented.

9.4.3 Segmentation of deformable dynamic textures

Now we show how the proposed algorithm can be applied to segmentation of motion textures, where the regions occupied by each class vary its shape over time. For this type of experiments, we have applied the graph-cut based algorithm, as it have shown an improved performance. In Fig. 19 we present a sequence consisting of a fountain that suddenly appears over a static background. Note that, in our model, the static background can be considered as a motion texture class for which the probability of null motion values is very high. Thus, it can be segmented in the same manner as before. We observe in the segmentation result that the algorithm can effectively capture the temporal variations of the fountain texture.

Next, in Fig. 20, a motion texture of water is combined with a motion texture of grass, over a region that changes its shape and size. Note that this is a challenging situation where also the intensity textures are very difficult to separate.

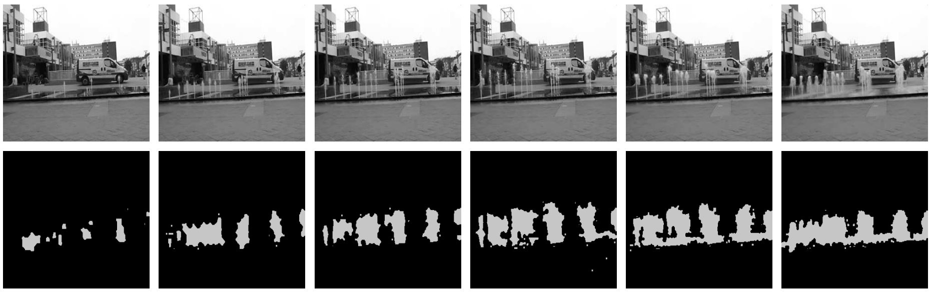

Figure 19: Segmentation of a sequence corresponding to a deformable motion texture of a fountain, using the graph-cut based method. The algorithm captures the varying shape of the segmented regions.

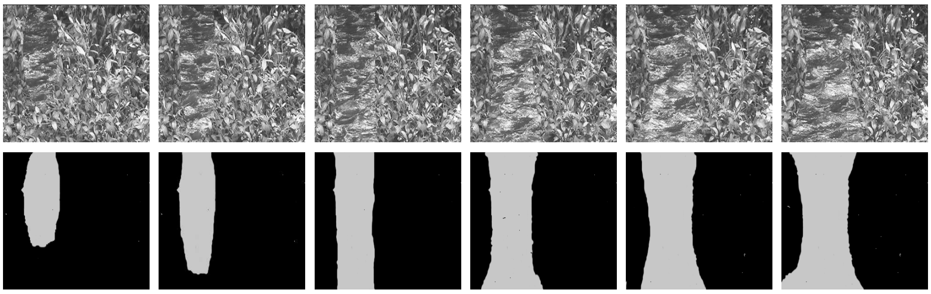

Figure 20: Segmentation of a sequence composed by two motion textures of water and grass, using the graph-cut based method. The area occupied by one of the classes changes over time.

10 Conclusions

We have proposed a new approach to dynamic texture modeling for motion-based content recognition applications . The contributions of this paper consist of:

1. A new mixed-state motion texture model that has shown to be a powerful non-linear representation for describing complex dynamic content with only a few parameters
2. Theoretical results on simmilarity measures for Gibbs distributions and mixed-state models
3. Results on classification of motion textures with the proposed model.
4. An original segmentation framework which does not assume conditional independence of the observations for each of the textures, and fully exploits the mixed-state motion texture model.
5. New results on the partition function calculation.
6. Segmentation results on deformable dynamic textures, which can not be handled by the existing previous approaches.

Currently, we are investigating the introduction of several discrete states, in particular symbolic abstract labels, for adressing problems as motion texture detection and modeling of high-level contextual information.

11 Appendix

A Uniqueness of the canonical decomposition of $Q(\mathbf{X})$

Let $p(\mathbf{X})$ be a function with $p(\mathbf{X}) > 0$ for all $\mathbf{X}$, and define $Q(\mathbf{X}) = \log \frac{p(\mathbf{X})}{p(\mathbf{r})}$. Then exists a unique decomposition,

$$Q(\mathbf{X}) = \sum_{\mathcal{C} \subset S} V_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}), \tag{60}$$

such that if for any $i \in \mathcal{C}$, $x_i = r$, then $V_{\mathcal{C}} = 0$. The uniqueness of this canonical representation of the energy function $Q(\mathbf{X})$ can be demonstrated supposing that exists another representation i.e.,

$$Q(\mathbf{X}) = \sum_{\mathcal{C} \subset S} H_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}). \tag{61}$$

Then, $\sum_{\mathcal{C} \subset S} V_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) - H_{\mathcal{C}}(\mathbf{X}_{\mathcal{C}}) = 0$. As this is true for every $\mathbf{X}$, we first choose $\mathbf{X} = (r, .., r, x_i, r, ..., r)$ and the only potentials that remain non-null are $V_i(x_i)$ and $H_i(x_i)$. Consequently, $V_i(x_i) - H_i(x_i) = 0$. We have then shown that the first order potentials in the decomposition are unique. Next, we choose $\mathbf{X} = (r, .., r, x_i, r, .x_j.., r)$ where all the sites except two of them, where forced to the value r. Following the previous reasoning, $V_{i,j}(x_i, x_j) - H_{i,j}(x_i, x_j) = 0$, and analogously, we show that the canonical potentials are unique.

References

[1] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *Int. J. Comput. Vision*, 12(1):43–77, 1994.

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B*, 36:192–236, 1974.

[3] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B*, 48(3):259–302, 1986.

[4] P. Bouthemy, C. Hardouin, G. Piriou, and J. Yao. Mixed-state auto-models and motion texture modeling. *Journal of Mathematical Imaging and Vision*, 25:387–402, 2006.

[5] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intellingence*, 20(12):1222–1239, Nov. 2001.

[6] A. Bruhn, J. Weickert, , C. Feddern, T. Kohlberger, and C. Schnorr. Variational optical flow computation in real time. *IEEE Trans. on Image Processing*, 14(5):608–615, 2005.

[7] H. Caillol, W. Pieczynski, and A. Hillon. Estimation of fuzzy gaussian mixture and unsupervised statistical image segmentation. *IEEE Trans. on Image Processing*, 6(3):425–440, 1997.

[8] C. Cedras. and M. Shah. Motion-based recognition: A survey. *Image and Video Computing*, 13(2):129–155, March 1995.

[9] B. Cernuschi-Frias. Mixed-states markov random fields with symbolic labels and multidimensional real values. *Rapport de Recherche INRIA.*, 2007. Http://hal.inria.fr/docs/00/16/59/37/PDF/RR-6255.pdf.

[10] A. Chan and N. Vasconcelos. Mixtures of dynamic textures. In *Proceedings of the 10th IEEE Int. Conf. on Computer Vision, ICCV'05, Beijing*, pages 641–647, Oct 2005.

[11] D. Chetverikov and R. Peteri. A brief survey of dynamic texture description and recognition. In *Proc. 4th Int. Conf. on Computer Recognition Systems CORES'05*, pages 17–26. Springer Advances in Soft Computing., 2005.

[12] T. Corpetti, E. Memin, and P. Perez. Dense estimation of fluid flows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):365–380, March 2002.

[13] T. Crivelli, B. Cernuschi, P. Bouthemy, and J. Yao. Segmentation of motion textures using mixed-state markov random fields. In *Proceedings of SPIE*, volume 6315, 63150J, Aug. 2006.

[14] T. Crivelli, B. Cernuschi, P. Bouthemy, and J. Yao. Mixed-state Markov random fields for motion texture modeling and segmentation. In *Proceedings of IEEE International Conference on Image Processing, ICIP'06*, pages 1857–1860, Atlanta, 2006.

[15] G. Cross and A. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(1):25–39, January 1983.

[16] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *Intl. Journal of Comp. Vision*, 51(2):91–109, 2003.

[17] G. Doretto, D. Cremers, P. Favano, and S. Soatto. Dynamic texture segmentation. In *Proc. of 9th Int. Conf. on Computer Vision. ICCV'03, Nice*, pages 1236–1242, Oct 2003.

[18] I. Elfadel and R. Picard. Gibbs random fields, cooccurrences, and texture modeling. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(1):24–37, 1994.

[19] R. Fablet and P. Bouthemy. Motion recognition using non-parametric image motion models estimated from temporal and multiscale co-ocurrence statistics. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(12):1619–1624, Dec 2003.

[20] R. Fablet, P. Bouthemy, and P. Perez. Non-parametric motion characterization using causal probabilistic models for video indexing and retrieval. *IEEE Trans. on Image Processing*, 11(4):393–407, April 2002.

[21] S. Fazekas and D. Chetverikov. Normal versus complete flow in dynamic texture recognition: a comparative study. In *Texture 2005: 4th international workshop on texture analysis and synthesis. ICCV'05*, pages 37–42, Beijing, 2005.

[22] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[23] D. Greig, B. Porteus, and H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B*, 51(2):271–279, 1989.

[24] C. Hardouin and J. Yao. Multi-parameter auto-models and their application. Preprint. *Biometrika*, 2007.

[25] B. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, Aug. 1981.

[26] R. Horn and C. Johnson. *Matrix analysis*. Cambridge University Press, New York, NY, USA, 1986.

[27] J. Hsieh, S. Yu, and Y. Chen. Motion-based video retrieval by trajectory matching. *IEEE Trans. Circuits and Systems for Video Technology.*, 16(3):396–409, 2006.

[28] R. Kindermann and J. Snell. *Markov random fields and their applications*. American Mathematical Society, Providence, Rhode Island, 1980.

[29] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, Feb. 2004.

[30] S. Krishnamachari and R. Chellappa. Multiresolution Gauss-Markov random field models for texture segmentation. *IEEE Transactions on Image Processing*, 6(2):251–267, 1997.

[31] Z. Lu, W. Xie, J. Pei, and J. Huang. Dynamic texture recognition by spatio-temporal multiresolution histograms. In *IEEE Workshop on Motion and Video Computing (WACV/MOTION)*, pages 241–246, 2005.

[32] R. C. Nelson and R. Polana. Qualitative recognition of motion using temporal texture. *CVGIP: Image Underst.*, 56(1):78–89, 1992.

[33] R. Peteri and D. Chetverikov. Dynamic texture recognition using normal flow and texture regularity. In *Proc. of IbPRIA*, pages 223–230, Estoril, 2005.

[34] R. Peteri, M. Huiskes, and S. Fazekas. Dyntex: A comprehensive database of dynamic textures. http://www.cwi.nl/projects/dyntex/index.html.

[35] G. Piriou, P. Bouthemy, and J.-F. Yao. Recognition of dynamic video contents with global probabilistic models of visual motion. *IEEE Trans. on Image Processing*, 15(11):3417–3430, 2006.

[36] R. B. Polana. *Temporal texture and activity recognition*. PhD thesis, Rochester, NY, USA, 1994.

[37] A. Rahman and M. Murshed. Real-time temporal texture characterisation using block based motion co-occurrence statistics. In *Proc. of the 11th IEEE Int. Conf. on Image Processing, ICIP'04*, pages 1593–1596, 2004.

[38] P. Saisan, G. Doretto, Y. Wu, and S. Soatto. Dynamic texture recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition. CVPR'01*, pages 58–63, Hawaii, 2001.

[39] F. Salzenstein and W. Pieczynski. Parameter estimation in hidden fuzzy markov random fields and image segmentation. *Graph. Models Image Process.*, 59(4):205–220, 1997.

[40] M. Shah and J. Xiao. Motion layer extraction in the presence of occlusion using graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(10):1644–1659, October 2005.

[41] C. Stiller and J. Konrad. Estimating motion in image sequences. *IEEE Signal Processing Magazine*, 16(4):70–91, July 1999.

[42] M. Szummer and R. Picard. Temporal texture modelling. In *Proc. of the 3rd IEEE Int. Conf. on Image Processing, ICIP'96*, pages 823–826, Sept. 1995.

[43] R. Vidal and A. Ravichandran. Optical flow estimation and segmentation of multiple moving dynamic textures. In *Proc. of CVPR'05*, volume 2, pages 516–521, San Diego, 2005.

[44] R. Xu and D. Wunsch. Survey of clustering algorithms. *IEEE Trans. on Neural Networks*, 16(3):645–678, May 2005.

[45] L. Yuan, F. Wen, C. Liu, and H. Shum. Synthesizing dynamic textures with closed-loop linear dynamic systems. In *Proc. of the 8th European Conf. on Computer Vision, ECCV'04*, volume LNCS 3022, pages 603–616, Prague, 2004.

[46] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):915–927, June 2007.

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes
4, rue Jacques Monod - 91893 ORSAY Cedex (France)
Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)
Unité de recherche INRIA Sophia Antipolis : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399