

Decomposition of dynamic textures using Morphological Component Analysis

Sloven Dubois^{1,2}, Renaud Péteri¹ and Michel Ménard²

{sloven.dubois01, renaud.peteri, michel.menard}@univ-lr.fr

¹ *Laboratoire de Mathématiques Image et Applications*
La Rochelle, France

² *Laboratoire Informatique Image et Interaction*
La Rochelle, France

Abstract—The research context of this work is dynamic texture analysis and characterization. Many dynamic textures can be modeled as large scale propagating wavefronts and local oscillating phenomena. After introducing a formal model for dynamic textures, the Morphological Component Analysis (MCA) approach with a well chosen dictionary is used to retrieve the components of dynamic textures. We define two new strategies for adaptive thresholding in the MCA framework, which greatly reduce the computation time when applied on videos. Tests on real image sequences illustrate the efficiency of the proposed method. An application to global motion estimation is proposed and future prospects are finally exposed.

Index Terms—Dynamic Textures, Spatio-Temporal Decompositions, Morphological Component Analysis

I. INTRODUCTION

A. Context

A recent theme in image sequence analysis is the extension of static textures to the temporal domain, referred as dynamic textures. A flag flapping in the wind, ripples at the surface of water, fire, waving trees, smoke or an escalator are all examples of dynamic textures that can be present in real scenes. Other examples are shown in Figure 1.

On Figure 1, each image sequence is viewed as a 3D data cube where cuts enable to observe motions occurring at different spatio-temporal scales.

Dynamic textures are a research topic of highly growing interest. The number of publications on dynamic textures in major computer vision conferences has risen sharply in recent years. Dynamic or temporal textures were introduced by the pioneer works of Nelson and Polana [1], [2], where a first dynamic texture definition was given. Different methods were thereafter proposed for dynamic texture characterization with a steep increase since 2003. This growing interest can be explained by both the democratization of video acquisition and processing systems, and by a large field of potential applications. Among them, one can mention:

- video indexing [3], [4]: the goal is to perform elaborate queries associating features of semantic nature. For example, one can search for videos of turbulent water

(“turbulent” being a temporal characteristic), or a fire, a calm lake, a tree waving in the wind, *etc.*

- video surveillance [5]: in some image sequences, dynamic texture is an important characteristic of the scene. To detect an accident or a risky behaviour in traffic, to supervise and characterize the motion of a crowd, to detect forest fires or smoke are all examples where a robust description of dynamic textures is necessary.
- spatio-temporal segmentation of image sequences [6]: being able to segment a video sequence with respect to dynamic textures can enrich the comprehension of a scene. It can enable to detect a perturbation in a given dynamic texture (presence of a boat on a lake for instance), to help building video summaries (apparition at time t of a given dynamic texture), or to better compress videos according to their texture content.
- dynamic background subtraction: in such applications, the background can be composed of dynamic textures, such as moving trees, and a precise characterization of dynamic textures can improve the efficiency of background subtraction algorithms.
- tracking [7]: being able to track dynamic textures in image sequences could enable to follow and analyze the evolution of phenomena such fluid flows or vortices, fire.
- videos synthesis [8], [9], [10]: realistic dynamic texture synthesis is necessary for video games, animations or video inpainting.

Giving a proper definition of dynamic textures is a notoriously difficult problem. Dynamic textures are often described as phenomena varying in both space and time with a certain spatio-temporal repetitivity. They can not only be considered as a simple extension of static textures to the time domain, but as a more complex phenomenon resulting from several dynamics. Yet, around 75% of major publications do not specify their definition of dynamic textures. For a better understanding of these complex phenomena, our first contribution is an original taxonomy of dynamic textures.

B. Taxonomy

Figure 2 shows the proposed taxonomy of dynamic textures. Different observations can be made:

- videos containing dynamic textures are not all of the

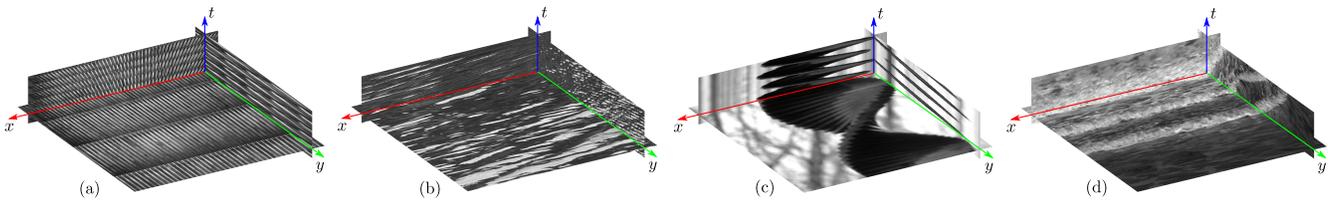


Figure 1. 2D+T sections of different dynamic textures. An escalator (a), ripples at the surface water (b), an endless spiral (c) and a wave motion with sea foam (d) can be observed.

same nature. They can be natural (natural processes), artificial (created by humans) or synthetic (generated by a computer).

- an image sequence may contain static and/or dynamic texture components (the dynamic texture component of an image sequence contains at least one dynamic texture). For example, a bridge, rocks or ivy are static textured patterns. Water and trees have a motion that generates two different dynamic textures.
- a dynamic texture is induced by three factors:
 - a textured pattern (rigid or deformable).
 - a motion, generated by a force on the textured pattern or by the camera (translation, zoom). This motion can be deterministic or stochastic, and the force generating this motion can be internal (motor of an escalator), or external (the wind blowing a windmill).
 - changes in the acquisition conditions (light reflection and illumination, *etc.*). These variations induce an apparent change of the texture and hence create a dynamic texture.

A windmill with a rotational motion, an escalator going upward, car traffic, *etc.* are examples of dynamic textures generated by a rigid textured pattern with deterministic motion. A fish shoal, a colony of ants, *etc.* are discrete texture elements but with a stochastic motion.

Dynamic textures generated by deformable textured patterns with stochastic motions are for instance a waterfall with eddies, an anemone tossed by the current, *etc.* Trees or flowers waving in the wind, a waterfall without eddies, *etc.* are examples of dynamic textures composed of deformable textured patterns with deterministic motion. Depending on the observation scale, some of these phenomena can however be seen as rigid patterns animated by a deterministic motion.

- a dynamic texture is composed of visually relevant modes [11]. For instance on Figure 3.a showing an image sequence of sea waves, two motions (called modes) can be observed: the high-frequency motion of small waves (cf. Figure 3.a.2), carried by the overall motion of the internal wave (cf. Figure 3.a.1). The process gets more complex

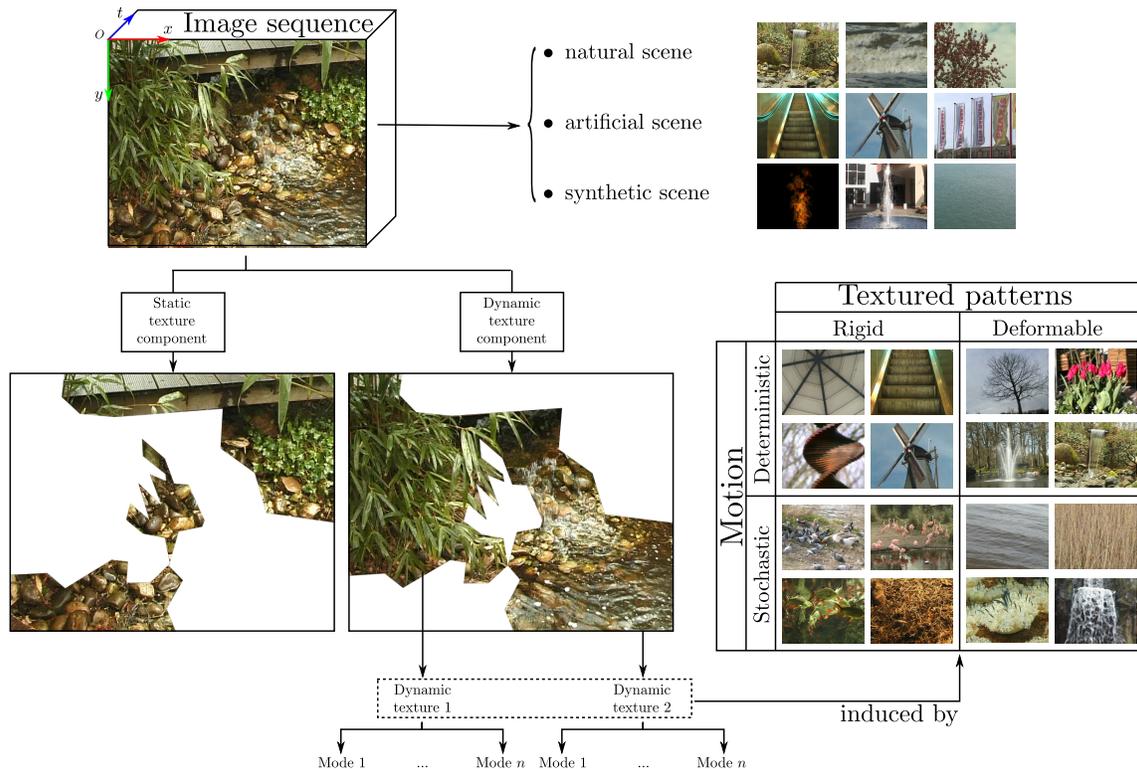


Figure 2. Taxonomy of dynamic textures.

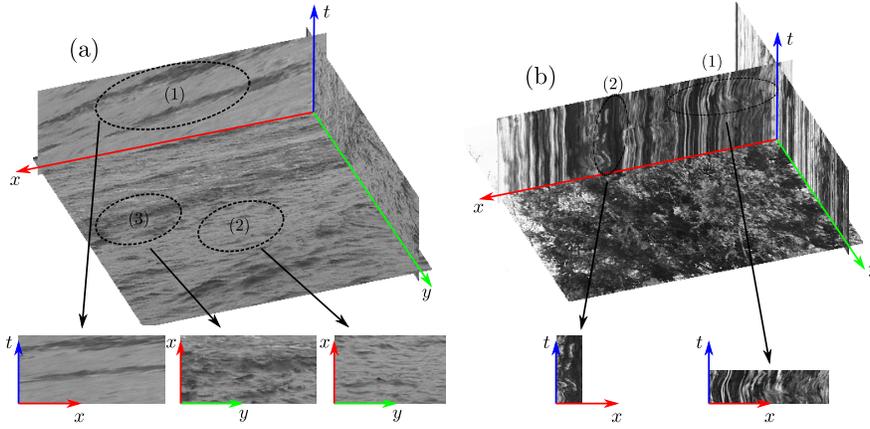


Figure 3. 2D+T slices of two dynamic textures. One can observe several wavefronts (1), local oscillating phenomena (2) and a mixture of both of them (3).

when the two phenomena overlap with each other (cf. Figure 3.a.3). These two modes can also be observed on the image sequence of waving trees on Figure 3.b.

- each dynamic texture has its own characteristics, such as stationarity, regularity, repetitivity, propagation speed, *etc.* These characteristics are more or less difficult to extract depending on the complexity of the considered dynamic texture.

All these considerations lead to the following definition of dynamic textures:

A natural, artificial or synthetic image sequence may contain a static texture component and/or a dynamic texture component. This latest one is composed of at least one dynamic texture.

A dynamic texture is a textured pattern that can be rigid or deformable. This pattern has a motion induced by a force which can be internal, external or created by camera motions. This motion can be deterministic or stochastic. Dynamic textures are composed of modes, which may overlap, characterized by repetitive spatial and temporal phenomena.

C. Outline of the article

The context of our work is the characterization and the analysis of these dynamic textures, with the aim of being able to automatically retrieve video scenes with given dynamic textures [4]. In the context of dynamic textures characterization, works can be classified according to the following taxonomy: methods based on optical flow [12], [13], [14], which have been the most popular, methods computing geometric properties in the spatio-temporal volume [15], [16], methods based on spatio-temporal filtering [17], and methods that compute spatio-temporal transforms [3], [4].

As mentioned previously, a dynamic texture is composed of different motions occurring at different spatio-temporal scales: for instance on Figure 3.b, a low spatio-temporal motion of a tree's trunk and high spatio-temporal motions from its branches and foliage can be observed. To efficiently characterize dynamic textures implies being able to extract this spatio-temporal behaviour. A natural tool for multiscale analysis is the wavelet transform. In image processing,

the wavelet transform has been successfully applied for characterizing static textures [18]. For instance, Gabor wavelets have been used for computing texture features in the MPEG-7 norm [19]. A natural idea is to extend these multiscale decompositions to the time domain in order to characterize dynamic textures.

Due to their complexity and variability, finding relevant decompositions into simpler components could highly help the understanding and feature extraction of dynamic textures. Existing image decomposition approaches seem to be promising for extracting these components [20], [21], [22].

The main contributions of this article is to present a new method for analyzing dynamic textures using the Morphological Component Analysis approach (MCA) in order to extract the underlying dynamical components. The method is coherent with the proposed dynamic texture model adapted to natural image sequences.

Section II introduces a new formal model for dynamic textures, based on observations of the DynTex database [11] and models used in video synthesis [23]. The relevancy of such a model is discussed compared to image sequences from DynTex.

After stressing the interest of decomposition methods for texture characterization and video indexing (Section III-A), the Morphological Component Analysis approach (MCA) is described. Based on the proposed dynamic texture model, we define dictionaries used in the MCA for separating the different observed phenomena (Section III-B). Properties of the iterative projection scheme are then analyzed. Two new thresholding strategies are presented in Section III-D in order to reduce the computation time currently prohibitive for such approaches.

In Section IV-B, a comparison between the different proposed thresholding strategies is performed. Our approach is validated on several image sequences. Finally, decomposition results are used in Section V for the estimation of global motion in an image sequence, highlighting both the practical and theoretical interest of the proposed approach.

II. A MODEL OF DYNAMIC TEXTURE

As mentioned previously, a dynamic texture is often described as a time varying phenomenon with a certain repetitivity in both space and time; this definition remains unprecise and ambiguous. In the previous section, a taxonomy of dynamic textures has been proposed for better understanding these phenomena. In this section, inspired by works on video synthesis, a model for several kinds of dynamic textures is presented. Our purpose is to model and understand the different components of a dynamic texture. It is not intended to outperform existing methods for synthesizing dynamic textures [8], [10].

Figure 4 shows four image sequences from the DynTex database, where several dynamic textures occur on the same video. These textures have different spatio-temporal supports. Some dynamic textures are even totally or partially transparent, like smoke or water. Consequently, their spatio-temporal support can overlap (see Figure 4).

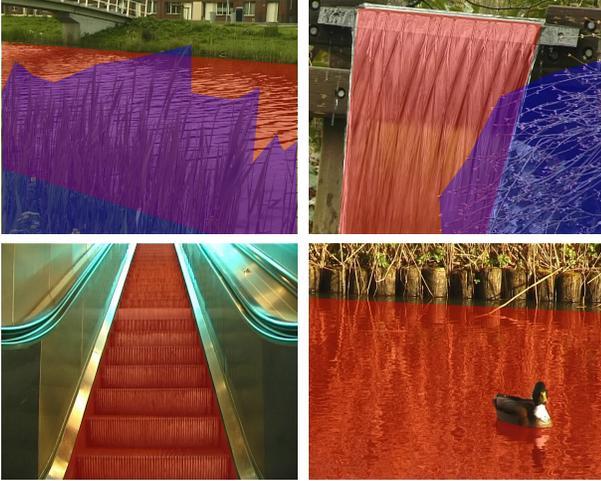


Figure 4. Examples of dynamic textures with different supports. Blue and red colors correspond to different spatio-temporal supports; the purple color traduces the overlapping zone of these two supports.

For a given video, the dynamic texture component T_V can be defined as a sum of $N \in \mathbb{N}^*$ dynamic textures Υ_i , each of them with spatio-temporal support Ω_i :

$$T_V(\mathbf{x}) = \sum_{i=1}^N \Upsilon_i^{\Omega_i}(\mathbf{x}) \quad (1)$$

where $\mathbf{x} = (x, y, t)^T$ represents the coordinates of a voxel in the video cube.

As observed on Figure 3, a dynamic texture Υ_i can be modeled as the superposition of large scale wavefronts and local oscillating phenomena. It can thus be defined as:

$$\forall i, \Upsilon_i^{\Omega_i}(\mathbf{x}) = \mathcal{P}_i(\mathbf{x}) + \mathcal{L}_i(\mathbf{x}) \quad (2)$$

where \mathcal{P}_i and \mathcal{L}_i are two functions describing respectively the wavefront and local phenomena composing a dynamic texture Υ_i (for the sake of simplicity, we will consider only one propagating wave for each dynamic texture).

Formalization (2) is well adapted for the following dynamic textures:

- deformable textured patterns with stochastic or deterministic motion, such as fluid flows (lake, sea, water stream, *etc*), oscillations generated by wind (grass, trees, flag, *etc*), smoke propagation, *etc*.
- rigid textured patterns with deterministic motion such as an escalator, a windmill, *etc*.
- discrete textures with stochastic motion such as fish shoal, insect swarm, *etc*.

The carrying wave \mathcal{P}_i is the most complex phenomenon, and depends on the considered image sequence. It is characterized by its propagating speed, its direction and its degree of stationarity. If one refers to works on video synthesis [23] and by observing the DynTex database, the wavefront \mathcal{P}_i of a given dynamic texture can be formalized as a sum of cosine functions, with amplitude $A_{p_n} \in \mathbb{R}^{+*}$, angular frequency $\omega_{p_n} \in \mathbb{R}^3$ and phase $\psi_{p_n} \in \mathbb{R}$:

$$\mathcal{P}_i(\mathbf{x}) = \sum_{p_n \in \mathcal{P}_i} A_{p_n}(\mathbf{x}) \operatorname{Re} \left(e^{j(\omega_{p_n} \cdot \mathbf{x} + \psi_{p_n})} \right) \quad (3)$$

Functions \mathcal{P}_i propagate texture information given by local oscillating phenomena.

Local phenomena \mathcal{L}_i differ from the carrying wave by being purely local. The spatio-temporal support of these phenomena is given by a spatio-temporal gaussian kernel. The choice of a gaussian kernel is made because of its optimum time/frequency tradeoff and for computation time considerations. Local phenomena \mathcal{L}_i are given by the following expression:

$$\mathcal{L}_i(\mathbf{x}) = \sum_{\ell \in \mathcal{L}_i} \mathcal{N}_G(\mu_\ell, \Sigma_\ell)(\mathbf{x}) \sum_{\ell_k \in \mathcal{L}_i} A_{\ell_k}(\mathbf{x}) \operatorname{Re} \left(e^{j(\omega_{\ell_k} \cdot \mathbf{x} + \psi_{\ell_k})} \right) \quad (4)$$

with for a given local phenomenon ℓ , $\mathcal{N}_G(\mu_\ell, \Sigma_\ell)$ is a gaussian kernel localizing phenomenon ℓ . $A_{\ell_k} \in \mathbb{R}^{+*}$, $\omega_{\ell_k} \in \mathbb{R}^3$ and $\psi_{\ell_k} \in \mathbb{R}$ represent respectively the amplitude, the angular frequency and the phase associated to ℓ .

Figure 5 shows the different generated components using the proposed model: the wavefront (5.a.), local oscillations (5.b.) and their superposition (5.c.). These results show the relevance of our model for representing some natural textures, for instance waves at the surface of water, the movement of a flag blowing in the wind *etc*

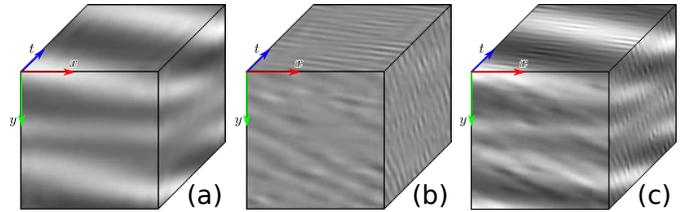


Figure 5. A synthetic dynamic texture generated by the model of equation (2). (a) Video of the wavefront, (b) locally oscillating phenomena. (c) dynamic texture generated by the sum of (a) and (b).

Analyzing this class of dynamic textures results in decomposing them into local oscillating phenomena and non

local wavefronts. When dealing with natural images, these decompositions need adapted transforms. These two aspects constitute our methodological framework for analyzing natural dynamic textures.

III. DECOMPOSING DYNAMIC TEXTURES

Identifying the parameters and coefficients of the proposed model is a difficult task if one wants to synthesize a given texture. Yet, obtained experimental results validate the hypothesis of a superposition of linear components. Existing image decomposition approaches [20], [21], [22] then seem to be relevant for extracting these components.

Considering the richness of the available analysis dictionary, the Morphological Component Analysis approach (MCA) has been chosen. The diversity and flexibility of the MCA framework are important points regarding the complexity of dynamic textures.

Usually used for spatial decompositions, the MCA is extended in this work to the temporal dimension. In the following sections, 2D+T multiscale transforms means truly 3D multiscale transforms with two spatial variables and one temporal variable. With such 2D+T transforms, spatio-temporal correlations in a video can be extracted, contrary to successive 1D and 2D transforms.

A. Morphological Component Analysis

The MCA approach allows to find an acceptable solution to the inverse problem of decomposing a signal onto a given vectorial basis, *i.e.* to extract components $(y_i)_{i=1,\dots,N}$ from a degraded observation y according to a sparsity constraint. This is obviously an ill-posed inverse problem. The MCA approach assumes that each component y_i can be represented sparsely in the associated basis Φ_i :

$$\forall i = 1, \dots, N, \quad y_i = \Phi_i \alpha_i \quad (5)$$

In this way, the obtained dictionary is composed of atoms built by associating several transforms $\Phi = [\Phi_1, \dots, \Phi_N]$ such as, for each i , y_i is well represented (sparse) in Φ_i and is not, or at least not as well represented in Φ_j ($j \neq i$).

This induces that:

$$\forall i, j \neq i \quad \|\Phi_i^T y_i\|_0 < \|\Phi_j^T y_i\|_0 \quad (6)$$

$\|\dots\|_0$ being the pseudo-norm ℓ_0 (number of non-zero coefficients).

The choice of the basis is of course primordial. Each transform possesses its own characteristics and will be adapted for extracting a particular phenomenon. This choice will be discussed in the next section.

Solving equation (6) implies to find a solution to the equation : $y = \Phi \alpha$. Starck *et al.* propose a solution for it in [24] and [22] by finding morphological components $(y_i)_{i=1,\dots,N}$ with the following optimisation problem:

$$\min_{y_1, \dots, y_N} \sum_{i=1}^N \|\Phi_i^T y_i\|_p^p \quad \text{such that} \quad \left\| y - \sum_{i=1}^N y_i \right\|_2 \leq \sigma \quad (7)$$

where $\|\Phi_i^T y_i\|_p^p$ penalizes non-sparse solutions (usually $0 \leq p \leq 1$). σ is the noise standard deviation.

This optimization problem (7) is not easy to solve. If all components y_j except the i^{th} are fixed till iteration $k-1$, it is however proved that the solution $\alpha_i^{(k)}$ is given by hard thresholding the marginal residual $r_i^{(k)} = y - \sum_{j \neq i} y_j^{(k-1)}$:

$$\alpha_i^{(k)} = \delta_{\lambda^{(k)}} \left(\Phi_i^T \left(r_i^{(k)} \right) \right) \quad (8)$$

$\delta_{\lambda^{(k)}}$ being the thresholding function for threshold $\lambda^{(k)}$ at step k . These marginal residuals r_i are by construction likely to contain missing informations of y_i . This idea induces an iterative algorithm for thresholding the marginal residuals for which main steps are presented in Algorithm 1.

Algorithm 1 Morphological Component Analysis

Task : Decompose a nD signal in dictionary Φ .

Parameters :

- The signal y to decompose
- The dictionary $\Phi = [\Phi_1, \dots, \Phi_K]$
- The thresholding strategy `strategy`
- The stopping condition σ

Initialization :

// Components to estimate are set to 0

for $i = 1$ to N **do**

$$\tilde{y}_i^{(0)} = 0$$

end for

// Initialization of λ

$$\lambda^{(1)} = \text{lambda_initialization}(\text{strategy})$$

// Initialization of the iteration number

$$k = 1$$

Main loop :

while $\left\| y - \sum_{j=1}^N \tilde{y}_j^{(k-1)} \right\|_2 \leq \sigma$ **do**

// For each component

for $i = 1$ to N **do**

// Compute the marginal residual

$$\tilde{r}_i^{(k)} = y - \sum_{j \neq i} \tilde{y}_j^{(k-1)}$$

// Projection of $\tilde{r}_i^{(k)}$ on basis Φ_i

$$\tilde{\alpha}_i^{(k)} = \Phi_i^T \left(\tilde{r}_i^{(k)} \right)$$

// Hard thresholding of $\tilde{\alpha}_i^{(k)}$

$$\alpha_i^{(k)} = \delta_{\lambda^{(k)}} \tilde{\alpha}_i^{(k)}$$

// New estimation of \tilde{y}_i

$$\tilde{y}_i^{(k)} = \Phi_i \left(\alpha_i^{(k)} \right)$$

end for

// Update of threshold λ

$$\lambda^{(k+1)} = \text{update}(\lambda^{(k)}, \text{strategy})$$

// Iterate

$$k = k + 1$$

end while

B. Choice of the dictionary

The crucial point in the MCA approach is the choice of the dictionary. Transforms not adapted to the dynamics of

phenomena present in the image sequence will deteriorate the results quality, leading to unsuitable decompositions, large values of pseudo-norm ℓ_0 and unrepresentative coefficients. As seen in Section II, a dynamic texture can be decomposed into two distinct phenomena. It is therefore necessary to associate to each of them the most representative basis.

In [25], [26], the authors show that the 2D+T curvelet transform [27] brings a relevant discrimination for non local phenomena propagating temporally. It seems particularly interesting to model long range wavefronts present in a dynamic texture.

A dynamic texture often presents locally oscillating phenomena. Therefore, the second base of the dictionary is built from a local transform adapted to oscillations: the 2D+T local cosine transform.

The MCA dictionary Φ is hence composed of the 2D+T curvelet transform Φ_1 and of the 2D+T local cosine transform Φ_2 .

C. Thresholding strategy

The purpose of this article is the decomposition of natural dynamic textures, therefore our experiments have been conducted on sequences from the DynTex database. The processed sequences have a duration of 5 seconds (128 images) and a size of 648 by 540 pixels¹. On volumes of such a size, the computation time is non negligible, as some transforms require several minutes.

Let function $T()$ measuring the execution time of a transform Φ_i during one cycle of the algorithm (analysis via Φ_i and synthesis via Φ_i^T). Two different platforms² have been used for the chosen dictionary, giving the computation time presented in table I.

	Platform 1 (32 bits)	Platform 2 (64 bits)
$T(\Phi_1) \approx T(\Phi_1^T)$	≈ 259 seconds	≈ 109 seconds
$T(\Phi_2) \approx T(\Phi_2^T)$	≈ 120 seconds	≈ 85 seconds

Table I
COMPUTATION TIME REQUIRED FOR PERFORMING AN ANALYZE OR A SYNTHESIS WITH THE CHOSEN DICTIONARY ON 2 DIFFERENT HARDWARE CONFIGURATIONS.

A recent work [28] has shown that a hundred of iterations is necessary to establish a good separation of the different components when a linear thresholding strategy (LTS) is used. In our case, the total computational time for a 5 second sequence is given by: $100 * (T(\Phi_1^T) + T(\Phi_1) + T(\Phi_2^T) + T(\Phi_2))$, which represents 21 hours on platform 1, and around 10 hours on platform 2.

If we extend this result to the entire DynTex database (about 700 videos), $700 * 21 \text{ hours} = 612 \text{ days}$ of calculation are required on a standard computer. This computation time can be reduced to 291 days on a dedicated server.

Recently Bobin *et al.* [28] have proposed an adaptive thresholding strategy 'Mean of Max' (MoMS) that enables

to obtain similar results but with fewer iterations (50 in average instead of 100). It represents a computation time of approximately 10 hours 30 (respectively 5 hours on platform 2) for a 5 second video, resulting in approximately 306 days (respectively 145 days) for the whole database.

As our aim is dynamic texture characterization for indexing the DynTex database, the computation time of the MoMS is still acceptable, since it is always possible to divide the workload on several processors. In the case where one searches for a particular texture using a query sequence, these calculations are acceptable only on sequences with limited duration and low resolution. We propose to reduce these limitations by introducing two new thresholding strategies.

D. Two new adaptive thresholding strategies for improving computation time

Results of the decomposition using the MCA algorithm strongly depend on the evolution of the threshold $\lambda^{(k)}$ in one iteration of the main loop. Figure 6 shows two different evolutions of $\lambda^{(k)}$ corresponding to two fictive examples of evolution strategies (S1) and (S2). Evolution of $\lambda^{(k)}$ is slower in case (S1) than in (S2). In this example, evolution (S1), respectively (S2), leads to select 5% of the coefficients, respectively 25%, in the two bases. If we consider that evolution (S1) gives here an optimal threshold, a failure to control the value of $\lambda^{(k)}$ (case S2) will lead to a rapid allocation of too many coefficients in the two bases, degrading the final decomposition. The linear thresholding strategy

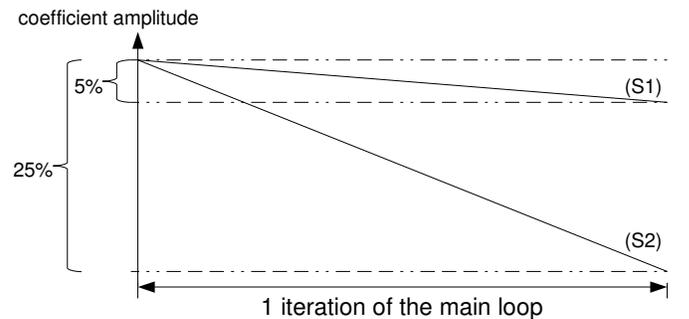


Figure 6. Two thresholding strategies leading to different evolutions of the threshold value during one iteration of the main loop.

(LTS) leads to the optimum $\lambda^{(k)}$ for 100 iterations [28]. For a large number of natural textures, this number of iterations can be greatly reduced, depending on the texture properties. LTS is then no longer optimum. However, the threshold evolution using LTS can be considered as a minimum slope below which the evolution of $\lambda^{(k)}$ is sub-optimal. A good strategy for computing $\lambda^{(k)}$ should lead to a slope greater or equal to the one obtained using LTS.

The 'Mean of Max' strategy (MoMS) is interesting as it can adaptively change the evolution of $\lambda^{(k)}$. However, on natural texture sequences, this strategy tends to reduce too drastically this slope, or even almost cancel it.

¹ie more than 44 million voxels

²Platform 1 : Processor 32 bits 2.4GHz, 4Go of RAM

Platform 2 : Processor 64 bits 3.2GHz, 24Go of RAM

a) *Adaptive thresholding strategy with linear correction (ATSLc)*: we propose to combine these two strategies into a new so-called adaptive thresholding strategy with linear correction (ATSLc), which defines $\lambda^{(k+1)}$ as the minimum value of $\lambda^{(k+1)}$ calculated using strategies LTS and MoMS.

The $\lambda^{(k)}$ update using ATSLc is formalized as follows:

$$\lambda^{(k+1)} = \min \left(\frac{1}{2}(m_1 + m_2), \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{N_{max}} \right) \quad (9)$$

with:

$$m_1 = \max_{\forall i} \|\Phi_i^T r^{(k)}\|_{\infty}$$

$$m_2 = \max_{\forall j, j \neq i_0} \|\Phi_j^T r^{(k)}\|_{\infty} \quad \text{with } i_0 = \operatorname{argmax}_{\forall i} \|\Phi_i^T r^{(k)}\|_{\infty}$$

$$r^{(k)} = y - \sum_{j=1}^K \tilde{y}_j^{(k)} \quad \text{being the total residual}$$

N_{max} being the total number of iterations.

Using this strategy, we are sure to change the value of $\lambda^{(k+1)}$ corresponding to the steepest slope. In other words, when MoMS leads to values of $\lambda^{(k+1)}$ evolving slowly, $\lambda^{(k+1)}$ follows the LTS $\lambda^{(k+1)} = \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{N_{max}}$. Otherwise, $\lambda^{(k+1)}$ follows the MoMS, $\lambda^{(k+1)} = \frac{1}{2}(m_1 + m_2)$, reducing the number of main loops in Algorithm 1.

b) *Adaptive thresholding strategy with exponential correction (ATSEc)*: in some cases, the distribution of the coefficients is concentrated around the origin. This phenomenon can occur for several reasons, for instance an unsuitable choice of the decomposition bases leading to a similar non-sparse representation in the different bases.

In these cases, the LTS is no longer optimum. Close to the origin, the threshold range will indeed be too large compared to the number of coefficients to select in this interval. Figure 7 shows that about 80% of these coefficients are contained in the last interval: they will be all assigned at once, leading to unsuitable decompositions.

To overcome this problem, [22] use a thresholding strategy with exponential decay, ETS. This strategy enables to threshold on a large range of coefficients at the first iterations of the algorithm, and on small intervals at the last iterations (see Figure 7). This strategy leads to a better assignment of the coefficients when concentrated around this origin. However, as for the LTS strategy, the number of iterations has to be fixed to a large value.

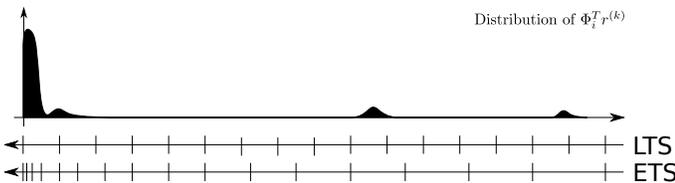


Figure 7. Threshold intervals for the two strategies LTS and ETS on an illustrative distribution.

Similarly to the ATSLc, we propose a second thresholding strategy combining the ETS and the MoMS approaches, so-called adaptive thresholding strategy with exponential correction, ATSEc.

It can be formalized as follows:

$$\lambda^{(k+1)} = \min \left(\frac{1}{2}(m_1 + m_2), \lambda^{(k)} * \left(\lambda^{(1)} - \lambda_{\min} \right)^{-\frac{1}{N_{max}-1}} \right) \quad (10)$$

In other words, when MoMS leads to values of $\lambda^{(k+1)}$ evolving too slowly, $\lambda^{(k+1)}$ follows the ETS strategy, $\lambda^{(k+1)} = \lambda^{(k)} * \left(\lambda^{(1)} - \lambda_{\min} \right)^{-\frac{1}{N_{max}-1}}$. Otherwise, $\lambda^{(k+1)}$ follows the MoMS strategy, $\lambda^{(k+1)} = \frac{1}{2}(m_1 + m_2)$, which also enables to decrease the iteration number in the main loop of Algorithm 1, while significantly decreasing the slope close to the origin.

IV. RESULTS

A. Computation time considerations

The new proposed thresholding strategies ATSLc and ATSEc have been implemented in the Morphological Component Analysis framework and extended to image sequences.

In both cases, the number of needed iterations for decomposing a video greatly diminishes. In average, 12 iterations are required for the ATSLc strategy and 17 iterations for the ATSEc strategy using $N_{max} = 100$. The computation time gain is however not proportional to the number of removed iterations: as seen in Section III-C, for each iteration and each basis, an analysis and a synthesis are required to perform the decomposition. For the ATS, ATSLc and ATSEc strategies, an additional projection on each basis is need to compute m_1 and m_2 .

The computation time for a given image sequence is then: (Number of iterations) $*$ ($T(\Phi_1^T) + 2 * T(\Phi_1) + T(\Phi_2^T) + 2 * T(\Phi_2)$). Performances of the two strategies ATSLc and ATSEc are given in table II. On a 64-bit platform, only 2 hours are needed to perform the complete decomposition of a video. Finally 60 days are required for the whole database using ATSLc and ATSEc strategies when using a single processor computer.

This computation time can moreover be divided by the number of cores of the server. With the Platform 2 of Table I, the decomposition of the whole DynTex database, ie about 700 videos, has been computed in one week.

	Platform 1 (32 bits)	Platform 2 (64 bits)
ATSLc	\approx 3h47	\approx 1h56
ATSEc	\approx 5h22	\approx 2h44

Table II
AVERAGE COMPUTATION TIME NEEDED TO PERFORM THE MCA DECOMPOSITION OF A VIDEO FROM DYNTEX USING ATSLC AND ATSEc STRATEGIES, FOR TWO HARDWARE CONFIGURATIONS.

The use of ATSLc and ATSEc strategies seems very interesting for reducing the computation time. The decomposition quality using the new adaptive thresholding strategies is analyzed in the following section.

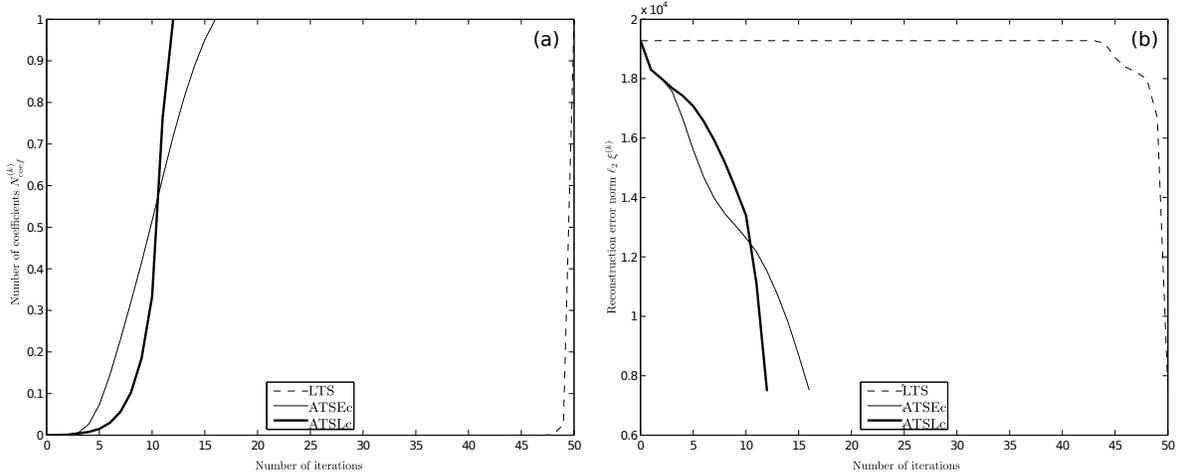


Figure 8. Thresholding strategies LTS, ATSLc and ATSEc. On the left, the number of thresholded coefficients during the iteration process $N_{coef}^{(k)}$. On the right, the reconstruction error norm ℓ_2 with respect to the number of iterations $\xi^{(k)}$.

B. Comparison of the thresholding strategies

Decomposition results of ATSLc and ATSEc strategies are compared using two quantitative criteria. The linear thresholding strategy (LTS) [24], [22] will be used as a reference for comparison.

The first criterion is the number of coefficients selected by the MCA algorithm during each iteration k :

$$\forall k, N_{coef}^{(k)} = \sum_{i=1}^N \left\| \alpha_i^{(k)} \right\|_0 \quad (11)$$

where $\|\dots\|_0$ is the pseudo-norm ℓ_0 (number of non-zero coefficients).

The thresholding strategy can be considered as successful if $N_{coef}^{(k)}$ grows steadily during the iteration process. Indeed, an unsuitable thresholding strategy will irregularly allocate the coefficients during the iteration process, and will be represented by a non-regularly growing function.

The second criterion is the reconstruction error norm ℓ_2 after k iterations of the algorithm:

$$\forall k, \xi^{(k)} = \left\| y - \sum_{i=1}^N \hat{y}_i^{(k)} \right\|_2 \quad (12)$$

Similarly to the first criterion, the curve regularity indicates the performance of the thresholding strategy. The more the reconstruction error quickly decreases during the iterations, the more relevant the selected coefficient are.

Figure 8 shows the evolution of $N_{coef}^{(k)}$ and $\xi^{(k)}$ with respect to the number of iterations for LTS, ATSLc and ATSEc strategies applied on several videos. For obtaining this curves, a mean of curves on 20 image sequences is made.

Several observations can be made:

- Using the LTS strategy, most of the coefficients are selected during the latest iterations of the algorithm (about 80%). This can be observed both on the rapid growth of curve $N_{coef}^{(k)}$, and on the rapid decay of curve $\xi^{(k)}$ during the final iterations.

- Both strategies ATSLc and ATSEc have a similar behaviour at the first iterations and diverge thereafter. During the first iterations, the two strategies choose the maximum slope $\frac{1}{2}(m_1 + m_2)$, but thereafter differ in their respective corrections.
- One can notice that the reconstruction error decreases almost uniformly for the two adaptive strategies. Therefore both strategies constantly select appropriate coefficients. If a strategy selects unappropriated coefficients, the reconstruction error will indeed stay constant or could even grow.
- During the iteration process, both adaptive strategies spend more time sorting coefficients than the LTS strategy. It takes about 5 iterations to the LTS strategy to classify approximately 80% of the information, whereas adaptive strategies spend at least 10 iterations.

Considering the gain in computation time and the performance reached, ATSLc and ATSEc strategies appear to be relevant and promising for processing image sequences with natural textures.

C. Comparison between static and dynamic MCA decomposition

To highlight the temporal influence in our approach, Figure 9 presents a comparison between static MCA decomposition (computed frame by frame) and our 2D+T MCA decomposition. The difference between these methods is the chosen dictionary. Indeed, for static MCA decomposition, 2D curvelet transform and 2D local cosine transform are used rather than 2D+T multiscale transforms. The other parameters (number of scales, LDCT window size, convergence criterion, *etc*) are identical for both approaches.

Many observations (highlighted areas in Figure 9) can be made on this comparison:

- (a) with 2D MCA decomposition, the local phenomena are present in the geometric component, contrary to 2D+T decomposition which only captures the lake surface mo-

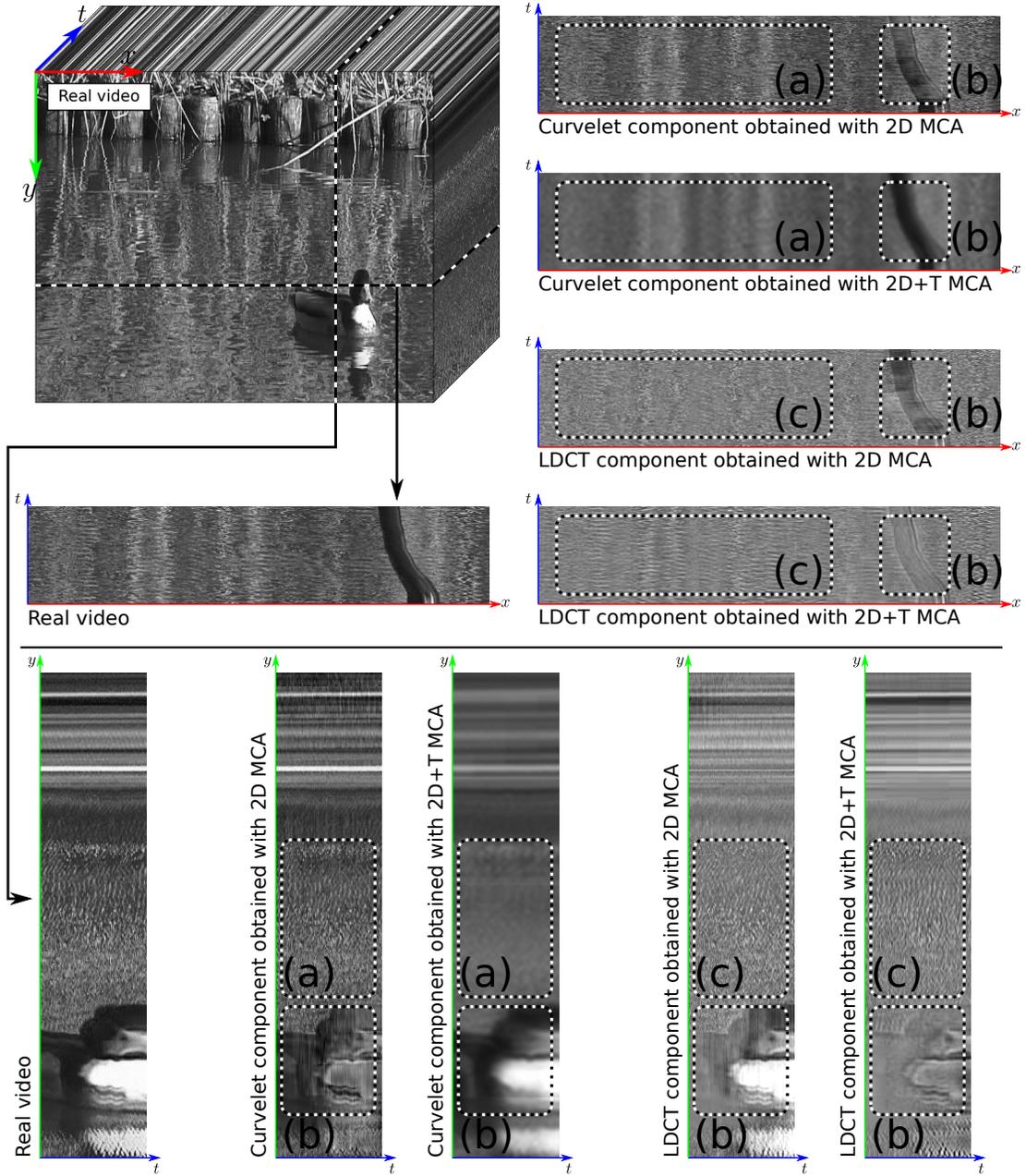


Figure 9. Comparison between static MCA decomposition and our 2D+T MCA decomposition. Spatio-temporal cuts (xt and yt) enable to emphasize the temporal aspect of our 2D+T decomposition compared to 2D decomposition.

tion (corresponding to the wavefront for this dynamic texture).

- (b) the geometric component retrieved using 2D+T curvelet transform better captures the structure of objects. Indeed, for 2D MCA decomposition, the duck is extracted both in the geometric component and the texture component. This phenomenon is not observed with 2D+T decomposition.
- (c) the extracted behavior is more temporally consistent with 2D+T MCA decomposition compared to 2D decomposition. Indeed, in this last decomposition, there is no temporal coherence in the texture component.

In the 2D approach, as the temporal information is not taken into account, the decomposition can not extract the spatio-temporal behavior of dynamic textures.

D. Dynamic texture decomposition

Results obtained using strategies ATSLc and ATSEc are promising and satisfactory. In this section, three of them are described in details³.

The first video shows a duck drifting slowly in a canal (Figure 10). Reflections of trees in the rippling water and

³These three videos and other results are visible at: http://mia.univ-larochelle.fr/demos/dynamic_textures/

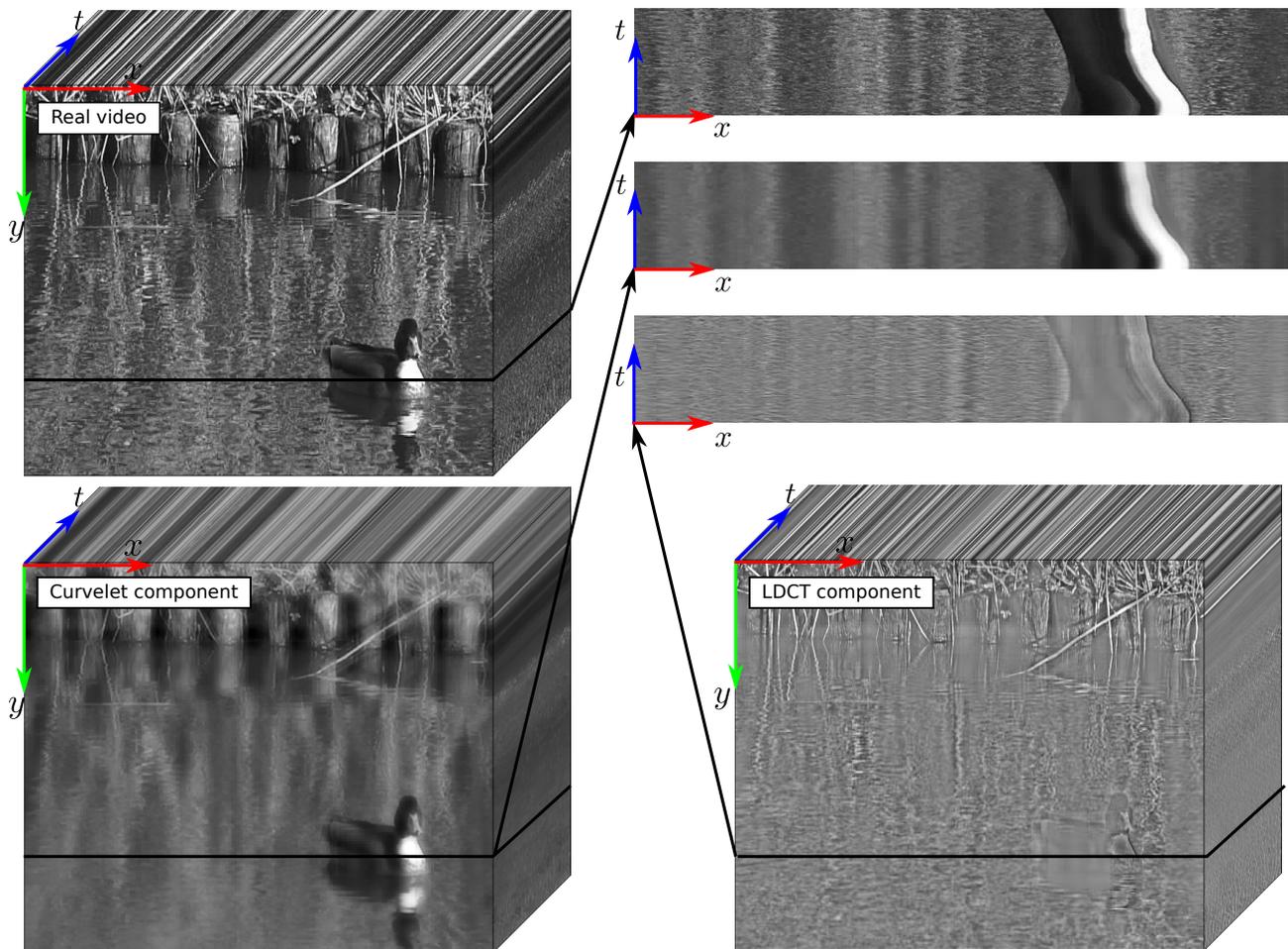


Figure 10. Results of the MCA decomposition on a video using the ATSLc strategy. spatio-temporal cuts xt enable to emphasize the temporal aspect of the decomposition. The geometric component is retrieved using the 2D+T curvelet transform while the texture component is obtained by the 2D+T local cosine transform.

a static texture background are also observable. Figure 10 shows the decomposition results obtained on this video using the MCA algorithm with the ATSLc strategy.

The geometric component is retrieved using the 2D+T curvelet transform while the texture component is obtained by the 2D+T local cosine transform: ripples, which are local phenomena, are well captured by the texture component, whereas reflections on the water surface are retrieved in the geometric component.

Spatio-temporal cuts along a xt plane enable to visualize the obtained decomposition. They show that the different objects in the scene (the duck, reflections of the trees on water, etc) are correctly considered as geometry. Reflections of the trees are not present in the texture component anymore. One can also observe that oscillations have been strongly attenuated in the geometric component. Cuts also show that spatio-temporal local phenomena (lake ripples) are well extract in the texture component.

This decomposition can bring us information that was not discernible on the original image sequence. For example, the texture of the duck's plumage under its neck can be observed

in the texture component, and is not visible in the original video.

The next image sequence represents a fountain (Figure 11). This fountain is composed of a jet, which once expelled, creates ripples at the water surface. Results of the 2D+T decomposition are shown on Figure 11.

The two obtained components seem relevant: in the geometric part, the central column of the fountain and the bell shape caused by the jet are visible, whereas almost absent in the texture component.

One can also notice that the entire area in front of the water jet is free of ripples, observable contrariwise in the other component. These observations are also well noticeable on areas represented as surfaces where the geometric part is composed of a slight wave free of high frequency oscillations. Wavefront and local phenomena are not distinguishable on the surface representation of the original video, but are clearly visible after 2D+T MCA decomposition.

The last video shows an escalator (Figure 12). This

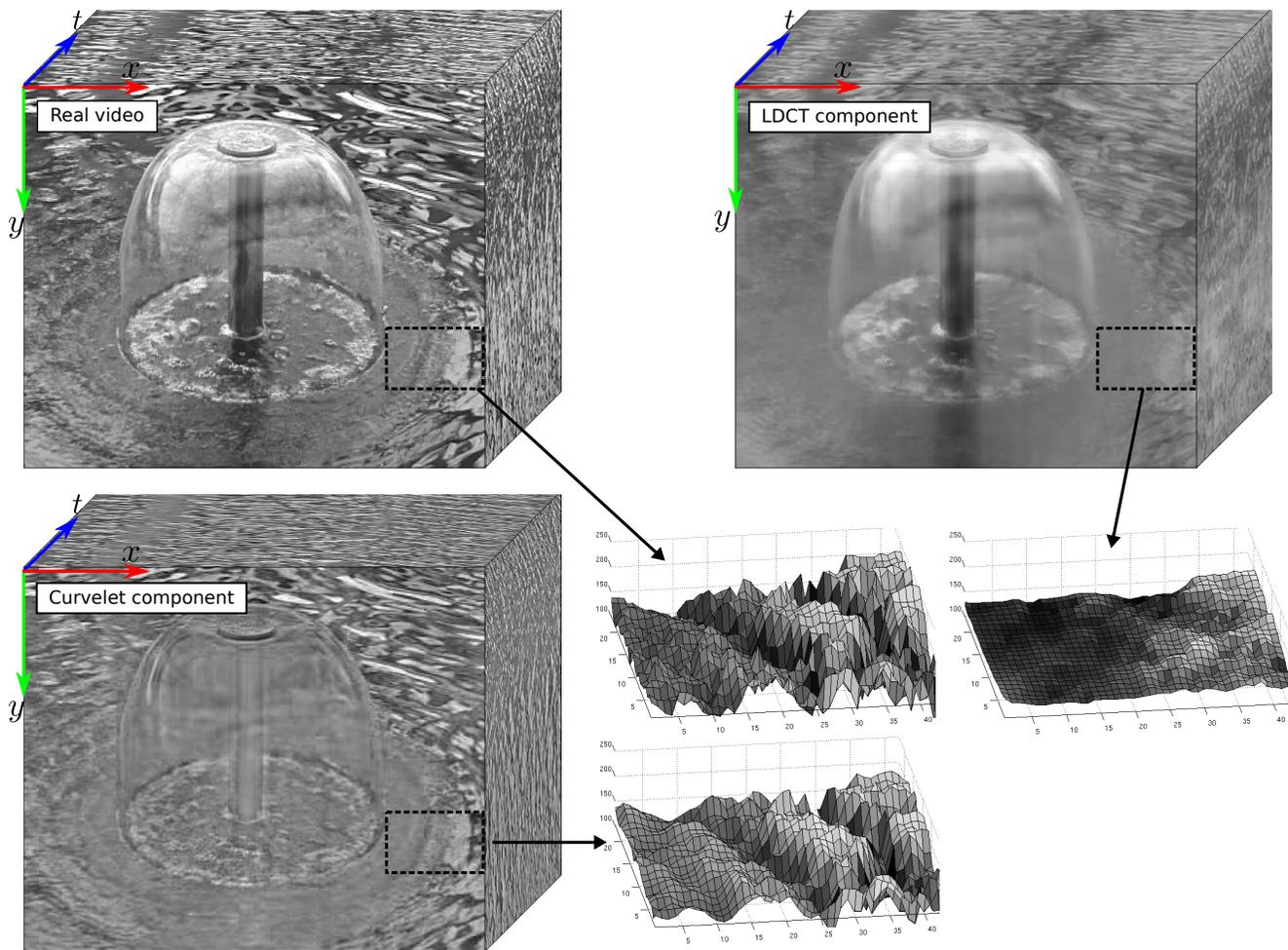


Figure 11. Decomposition results of a video using the MCA algorithm and the ATSLc strategy. Regions of interest are plotted as surfaces in order to better visualize the algorithm behaviour. The geometric component is retrieved using the 2D+T curvelet transform while the texture component is obtained by the 2D+T local cosine transform.

example is interesting because it is a dynamic texture that illustrates our model definition. An escalator is indeed composed of a long range wavefront (the steps) and more local phenomena (the step streaks). Figure 12 displays the results obtained using the 2D+T MCA algorithm.

After our 2D+T MCA decomposition, the long range wavefront and local phenomena are well separated. Spatio-temporal cuts permit a better visualization of the decomposition results. One can also observe that the step motion is better observable on the YT plane of the geometrical component than on the original sequence. In addition, most of the step streaks are captured in the texture component.

All the results presented above show that our decomposition based on the MCA algorithm extended to the temporal dimension can extract many different phenomena present in complex scenes containing dynamic textures. These findings are observed on other videos of the DynTex database.

The next section presents a concrete application of the 2D+T MCA decomposition for computing the global motion in an image sequence.

V. APPLICATION OF DYNAMIC TEXTURE DECOMPOSITION TO OPTICAL FLOW ESTIMATION

Decomposing a dynamic texture into a geometrical component and a texture component brings a better visual understanding of the different phenomena. In order to push forward the analysis, our decomposition method is applied to the detection of the principal motion of a dynamic texture. Motion in the video is estimated using the classical Horn and Schunk algorithm [29]. The optical flow is computed both on the original video and on the geometrical component.

The global motion estimation is performed on a video of the sea on which waves and foam can be observed. Results are presented on Figure 13. Two visualization systems are used to characterize the estimated optical flow:

- a map of the vector field where the color (respectively the saturation) indicates the direction (respectively the norm) of the optical flow.
- the orientation homogeneity of the motion vector field, introduced in [14]. The orientation homogeneity reflects the flow homogeneity of the overall motion compared to its mean orientation.

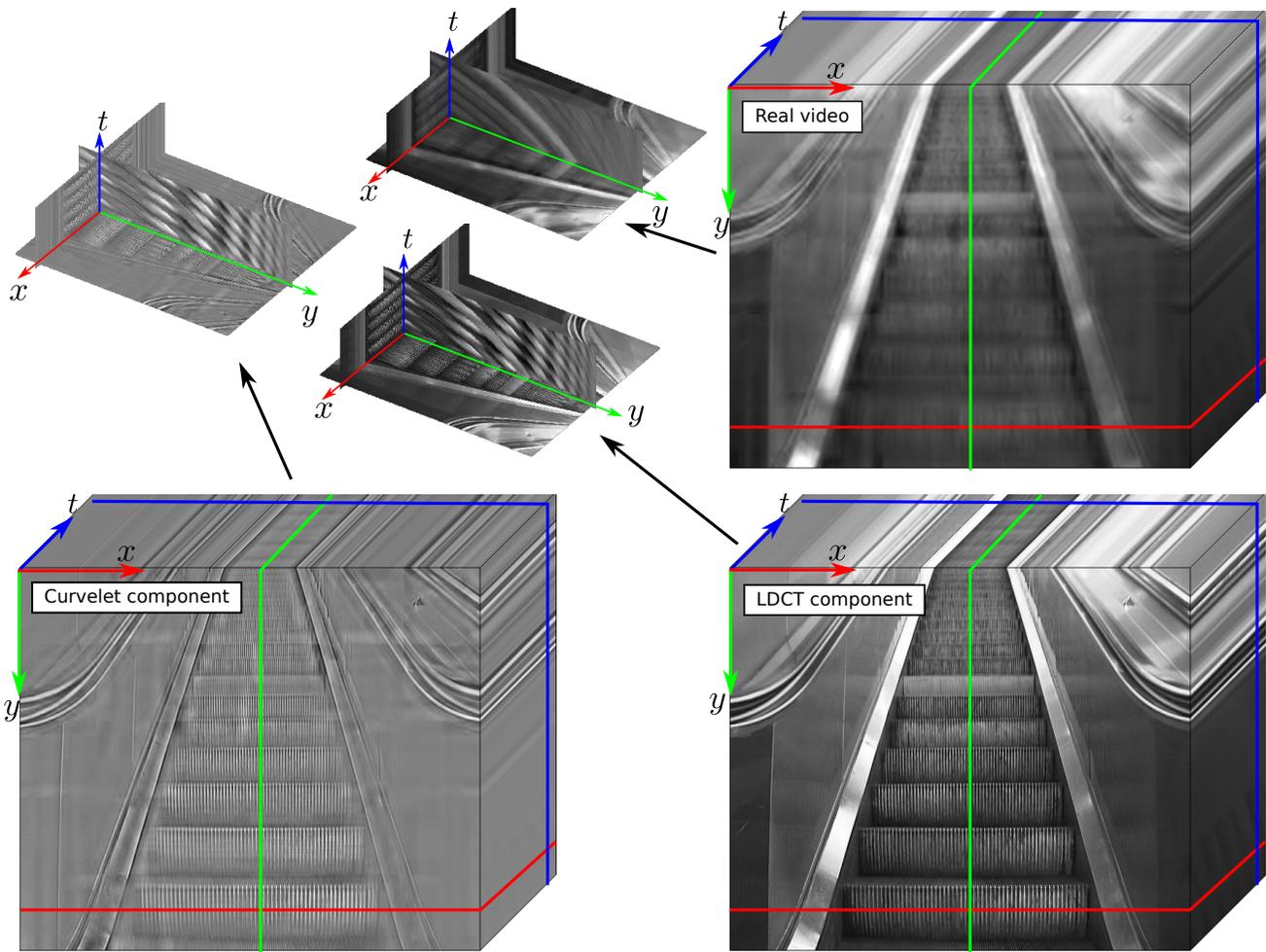
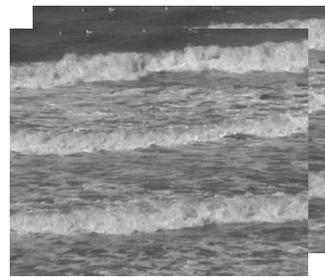
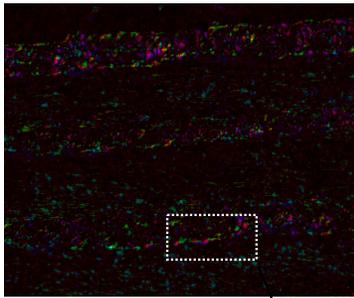


Figure 12. Results of the MCA decomposition on a video using the ATSLc strategy. Spatio-temporal cuts are presented in order to better visualize the decomposition results. The geometric component is retrieved using the 2D+T curvelet transform while the texture component is obtained by the 2D+T local cosine transform.

Optical flow estimation
on the original video



Optical flow estimation
on the geometric component

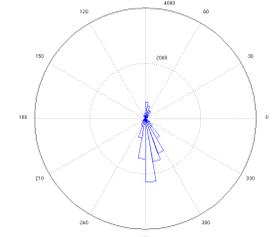
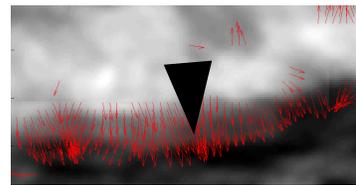
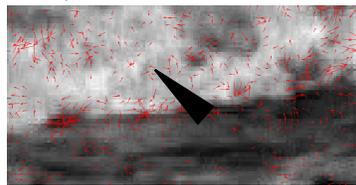
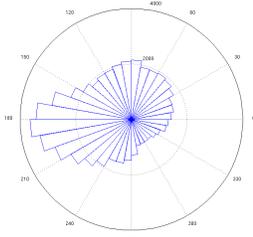
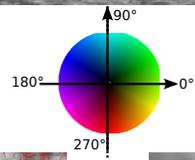
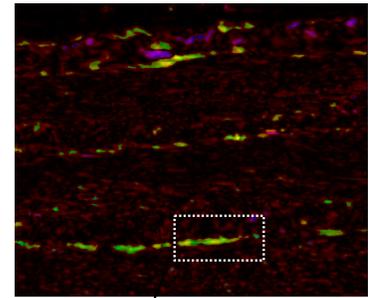


Figure 13. Results of the optical flow estimation using the Horn and Schunk method applied on the original video and on its geometric component.

Circular histograms of the motion direction distribution also give information on the estimated optical flow.

One can notice that all the local dynamic phenomena are detected in the original sequence. No color is prominent in the optical flow norm map as motion is mainly due to water foam which is turbulent and non directional.

Conversely, when the optical flow is computed on the geometric component only, a main motion direction can be distinguished, which is the one of the wavefront. Motion borders are well localized, contrary to motion estimation on the original sequence with a large regularization coefficient. Finally, the main motion direction can be observed on the circular histogram, and is much more isotropic than in the previous case.

This application emphasizes the interest of our 2D+T MCA decomposition method for understanding, interpreting and extracting dynamic textures.

VI. CONCLUSION AND PROSPECTS

A. Conclusion

This paper formalizes a new model of dynamic textures, based on the superimposition of large scale propagating wavefronts and local oscillating phenomena. Decomposition approaches enable to better understand these different components. Among them, the MCA approach is very appropriate, but suffers from a high computation time, which is an important issue in the context of video.

After considering the different possible dictionaries for dynamic texture analysis, we propose two new adaptive thresholding strategies: the ATSLc and the ATSEc. These two new thresholding strategies lead to a significant gain in computation time. Compared to the original strategy, the necessary calculations are reduced by about five times, with equivalent quality of results. In our research context of dynamic texture indexing, it enables to release the constraints of low resolution and duration on queries in a large video database.

Results on real videos from DynTex have been finally presented. These results confirm the relevance of the proposed model and enable to understand the different complex phenomena present in dynamic textures.

Finally, results of the MCA decomposition are used for estimating the global motion of a given video. Computing the optical flow on the geometric component appears to be much more relevant than a direct computation on the original sequence.

B. Prospects

The model developed in this article characterizes a certain class of dynamic textures. This model can be extended to other classes by possibly adding new dynamics (divergence, vortex, etc). These new parts of the model will be later extracted by adapted bases in the MCA algorithm.

In the context of video indexing, the different components obtained using the MCA algorithm can be used for extracting characteristic features: some related to the geometry of the dynamic texture (main motion direction, uniformity of the global movement, etc) and some characterizing more local

phenomena (speed, local vortex, etc).

An application of our 2D+T MCA decomposition on optical flow estimation has been presented. Many other applications can be considered: modification of dynamic texture dynamics (for instance to reverse the stream of a river), spatio-temporal segmentation, masking static objects on dynamic backgrounds (ie remove the duck from the video in Figure 10), etc.

REFERENCES

- [1] R. C. Nelson and R. Polana, "Qualitative recognition of motion using temporal texture," *CVGIP: Image Understanding*, vol. 56, no. 1, pp. 78–89, 1992.
- [2] R. Polana and R. Nelson, "Recognition of motion from temporal texture," in *Conference on Computer Vision and Pattern Recognition (CVPR '92)*, 1992.
- [3] J. R. Smith, C.-Y. Lin, and M. Naphade, "Video texture indexing using spatio-temporal wavelets," in *International Conference on Image Processing (ICIP'02)*, vol. II, 2002, pp. 437–440.
- [4] S. Dubois, R. Péteri, and M. Ménard, "A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition," in *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'09)*, vol. LNCS 5524, Povoia de Varzim, Portugal, 2009, pp. 314–321.
- [5] W. Phillips, M. Shah, and N. Lobo, "Flame recognition in video," *Pattern Recognition Letters*, vol. 23, pp. 319–327, 2002.
- [6] J. Li, L. Chen, and Y. Cai, "Dynamic texture segmentation using 3-d fourier transform," in *International Conference on Image and Graphics (ICIG 09)*, 2009, pp. 293–298.
- [7] R. Péteri, "Tracking dynamic textures using a particle filter driven by intrinsic motion information," *Machine Vision and Applications*, pp. 1–9, 2010.
- [8] G. Doretto, "Dynamic texture modeling," Master's thesis, University of California, Los Angeles, CA, June 2002.
- [9] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *International Journal of Computer Vision*, vol. 51, no. 2, pp. 91–109, February 2003.
- [10] A. Chan and N. Vasconcelos, "Modeling, clustering, and segmenting video with mixtures of dynamic textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 909–926, 2008.
- [11] R. Péteri, S. Fazekas, and M. J. Huiskes, "DynTex : a comprehensive database of dynamic textures," *Pattern Recognition Letters*, doi: 10.1016/j.patrec.2010.05.009. [Online]. Available: <http://projects.cwi.nl/dyntex/>
- [12] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto, "Dynamic texture recognition," in *Conference on Computer Vision and Pattern Recognition (CVPR'01)*, vol. 2, Kauai, Hawaii, December 2001, pp. 58–63.
- [13] R. Fablet and P. Bouthemy, "Motion recognition using spatio-temporal random walks in sequence of 2d motion-related measurements," in *International Conference on Image Processing (ICIP'01)*, 2001, pp. III: 652–655.
- [14] R. Péteri and D. Chetverikov, "Qualitative characterization of dynamic textures for video retrieval," in *International Conference on Computer Vision and Graphics (ICCVG'04)*, S. B. Heidelberg, Ed., vol. 32, 2004, pp. 33–38.
- [15] K. Otsuka, T. Horikoshi, S. Suzuki, and M. Fujii, "Feature extraction of temporal texture based on spatiotemporal motion trajectory," in *International Conference on Pattern Recognition (ICPR'98)*, vol. 2. Washington, DC, USA: IEEE Computer Society, 1998, p. 1047.
- [16] J. Zhong and S. Scarlaroff, "Temporal texture recognition model using 3d features," Jing Zhong and Stan Scarlaroff Department of Computer Science Boston University, Tech. Rep., 2002.
- [17] R. P. Wildes and J. R. Bergen, "Qualitative spatiotemporal analysis using an oriented energy representation," in *European Conference on Computer Vision-Part II (ECCV'00)*, vol. 1843/2000. London, UK: Springer-Verlag, 2000, pp. 768–784.
- [18] A. Mojsilovic, R. Mojsilović, M. V. Popovic, and D. M. Rackov, "On the selection of an optimal wavelet basis for texture characterization," *IEEE Transactions on Image Processing*, vol. 9, pp. 2043–2050, 2000.
- [19] P. Wu, Y. M. Ro, C. S. Won, and Y. Choi, "Texture descriptors in MPEG-7," in *International Conference on Computer Analysis of Images and Patterns (ICAIP'01)*. London, UK: Springer-Verlag, 2001, pp. 21–28.
- [20] T. F. Chan, S. Osher, and J. Shen, "The digital tv filter and nonlinear denoising," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 231–241, 2001.

- [21] J.-F. Aujol and A. Chambolle, "Dual norms and image decomposition models," *International Journal of Computer Vision*, vol. 63, no. 1, pp. 85–104, 2005.
- [22] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Transactions on Image Processing*, vol. 14, pp. 1570–1582, 2005.
- [23] M. Finch, *GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics, Chap.1*, R. Fernando, Ed. Randima Fernando, 2004. [Online]. Available: http://http.developer.nvidia.com/GPUGems/gpugems_part01.html
- [24] J.-L. Starck, M. Elad, and D. Donoho, "Redundant multiscale transforms and their application for morphological component analysis," *Advances in Imaging and Electron Physics*, vol. 132, 2004.
- [25] S. Dubois, R. Péteri, and M. Ménard, "A 3d discrete curvelet based method for segmenting dynamic textures," in *International Conference on Image Processing (ICIP'09)*, Cairo, Egypt, November 2009, pp. 1373–1376.
- [26] E. Candès and L. Demanet, "The curvelet representation of wave propagators is optimally sparse," *Communications on Pure and Applied Mathematics*, vol. 58, pp. 1472–1528, 2005.
- [27] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," California Institute of Technology, Tech. Rep., 2005.
- [28] J. Bobin, J.-L. Starck, J. Fadili, Y. Moudden, and D. Donoho, "Morphological component analysis : An adaptive thresholding strategy," in *IEEE transactions on image processing*. Institute of Electrical and Electronics Engineers, 2007, pp. 2675–2681.
- [29] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.