Morphological Component Analysis for decomposing Dynamic Textures

Sloven Dubois, Renaud Péteri and Michel Ménard

Abstract The research context of this work is dynamic texture analysis and characterization. A dynamic texture can be described as a time-varying phenomenon with a certain repetitiveness in both space and time.

Many dynamic textures can be modeled as a large scale propagating wavefront and local oscillating phenomena.

The Morphological Component Analysis 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 Morphological Component Analysis framework, which greatly reduce the computation time when applied on videos. These strategies are studied with different criteria. Finally, tests on real image sequences illustrate the efficiency of the proposed method.

1 Introduction

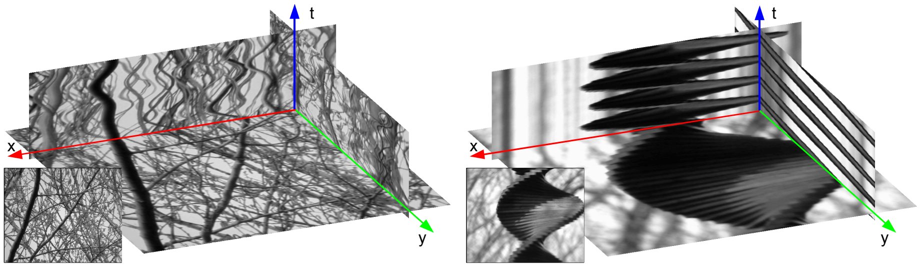
The study of dynamic textures, or temporal textures, is a recent research topic in the field of video processing. A dynamic texture can be described as a time-varying phenomenon with a certain repetitiveness in both space and

Sloven Dubois
Laboratoire Mathématiques, Image et Applications (MIA) et Laboratoire Informatique, Image et Interaction (L3i), Avenue Michel Crépeau, 17042 La Rochelle, France, e-mail: `sloven.dubois01@univ-lr.fr`

Renaud Péteri
Laboratoire Mathématiques, Image et Applications (MIA), Avenue Michel Crépeau, 17042 La Rochelle, France, e-mail: `renaud.peteri@univ-lr.fr`

Michel Ménard
Laboratoire Informatique, Image et Interaction (L3i), Avenue Michel Crépeau, 17042 La Rochelle, France, e-mail: `michel.menard@univ-lr.fr`

time. A flag in the wind, ripples at the surface of water, smoke or an escalator are all examples of dynamic textures. Figure 1 shows other examples of dynamic textures.



(a) (b)

Fig. 1 2D+T slices of two dynamic textures. Here, a dynamic texture is seen as a data cube and is cut at pixel $O(x, y, t)$, giving three planes $(\overrightarrow{x}O\overrightarrow{y})$, $(\overrightarrow{x}O\overrightarrow{t})$ and $(\overrightarrow{y}O\overrightarrow{t})$.

Without being exhaustive, the study of DT is an active topic, with many research areas such as synthesis [5], characterization [11, 12, 14] or segmentation [6, 7].

The context of our works 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 [8].

Giving a proper definition of dynamic textures is a notoriously difficult problem. Dynamic textures are often defined as phenomena varying in both space and time with a certain spatio-temporal repetitiveness. They cannot only be considered as a simple extension of static textures to the time domain, but as a more complex phenomenon resulting from several dynamics. Each dynamic texture has its own characteristics, such as stationarity, regularity, repetitiveness, propagation speed, ...

These characteristics are more or less difficult to extract depending on the complexity of the considered dynamic texture. For instance on figure 2.(a) showing an image sequence of sea waves, two motions can be observed: the high-frequency motion of small waves (2), carried by the overall motion of the internal wave (1). It gets more complex when the two phenomena overlap with each other (3). This statement can also be made about an image sequence of trees on figure 2.(b).

Many dynamic textures can be decomposed into one or several local oscillating motions carried by far range waves. In order to better characterize these two sets of components, it is necessary to extract them separately.

In this article, the Morphological Component Analysis is used for decom-

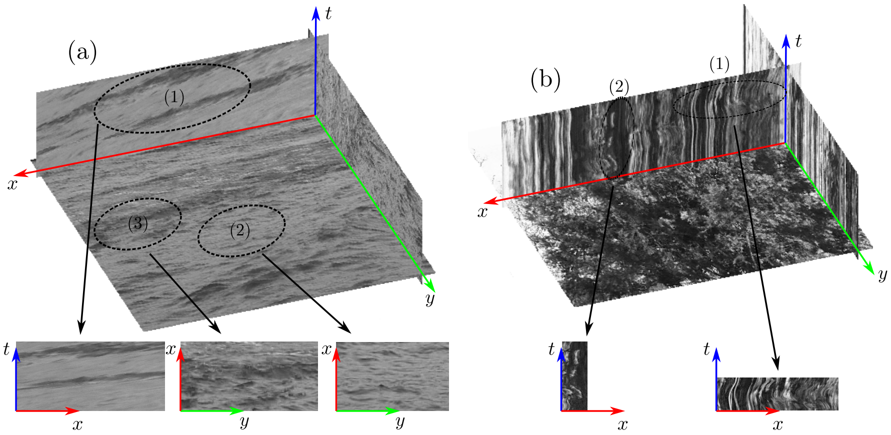


Fig. 2 2D+T slices of two dynamic textures: local oscillating phenomenon (2) and long range propagating wave (1) and a mixture of both of them (3)

posing and analyzing image sequences of natural scenes. To our knowledge, the only existing work using Morphological Component Analysis and video is recent and focuses on the inpainting of a cartoon sequence [17].

In section 2, the Morphological Component Analysis is briefly described. The dictionaries selected in the Morphological Component Analysis, adapted to dynamic textures, are presented.

A key issue is the computation time of Morphological Component Analysis that is related to the thresholding strategy. We propose in section 3 two new adaptive thresholding strategies that reduced the computation time by a factor of four compared to the original algorithm. To evaluate these new strategies, different criteria are proposed.

Results on real sequences of dynamic textures are presented in section 4 and future prospects are finally discussed.

2 Decomposing a dynamic texture using Morphological Component Analysis

In order to perform correct characterization, it is necessary to well understand the nature of dynamic textures. In many works, a dynamic textures is often described as a time-varying phenomenon with a certain repetitiveness in both space and time. According to researches on synthesis [10] and observations made on a large dynamic textures database [13], a dynamic texture can be modelled as a sum of local oscillations carried by longer range waves.

Recent works for decomposing images and videos [15, 4, 1] seem relevant for extracting these components. We have chosen the Morphological Component Analysis because of the richness of the available dictionary, which is crucial, considering the complexity of dynamic textures.

2.1 Morphological Component Analysis approach

The Morphological Component Analysis (MCA) approach finds an acceptable solution to the inverse problem of decomposing a signal onto a given vectorial basis, *i.e.* to extract components $(y_i)_{i=1,\ldots,N}$ 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:

$$\forall i = 1, \ldots, N, \quad y_i = \Phi_i \alpha_i \quad \text{and} \quad \alpha_i = \Phi_i^T y_i \tag{1}$$

where α_i are the projection coefficients of y_i on basis Φ_i.

In this way, the obtained dictionary is composed of atoms built by associating several transforms $\Phi = [\Phi_1, \ldots, \Phi_N]$ such that, 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:

$$\forall i, j \neq i \quad \|\Phi_i^T y_i\|_0 < \|\Phi_j^T y_i\|_0 \tag{2}$$

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

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

Solving (2) implies to find a solution to the equation : $y = \Phi\alpha$. Starck *et al.* propose a solution for it in [15] and [16] by finding morphological components $(y_i)_{i=1,\ldots,N}$ with the following optimization problem:

$$\min_{y_1,\ldots,y_N} \sum_{i=1}^{N} \left\|\Phi_i^T y_i\right\|_p^p \quad \text{such that} \quad \left\| y - \sum_{i=1}^{N} y_i \right\|_2 \leqslant \sigma \tag{3}$$

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

This optimization problem (3) 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 $r_i^{(k)} = y - \sum_{j \neq i} y_j^{(k-1)}$:

$$\alpha_i^{(k)} = \delta_{\lambda^{(k)}} \left(\Phi_i^T \left(r_i^{(k)} \right) \right) \tag{4}$$

$\delta_{\lambda^{(k)}}$ 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, the main steps of which are presented in algorithm 1.

Algorithm 1 Morphological Component Analysis

```
Task : Decompose a nD signal in dictionary $\Phi$.
Parameters :

  • The signal $y$ to be decomposed
  • The dictionary $\Phi = [\Phi_1, \ldots, \Phi_K]$
  • The thresholding strategy strategy
  • The stopping condition $\sigma$

Initialization :
// Components to be estimated are set to 0
for $i = 1$ to $N$ do
    $\tilde{y}_i^{(0)} = 0$
end for
// Initialization of $\lambda$
$\lambda^{(1)}$ = lambda_initialization(strategy)
// Initialization of the iteration number
$k = 1$
Main loop :
while $\left\| y - \sum_{j=1}^{N} \tilde{y}_j^{(k-1)} \right\|_2 \leqslant \sigma$ 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 $\Phi_i$
        $\tilde{\alpha}_i^{(k)} = \Phi_i^T \left( \tilde{r}_i^{(k)} \right)$
        // 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 $\lambda$
    $\lambda^{(k+1)}$ = update($\lambda^{(k)}$,strategy)
    // Iterate
    $k = k + 1$
end while
```

2.2 Choice of the dictionary

A crucial point in the Morphological Component Analysis approach is the dictionary definition. An unsuitable choice of transformations will lead to non sparse and irrelevant decompositions of the different dynamical phenomena present in the sequence.

As mentioned previously, we model dynamic texture as a sum of local oscillations carried by long range waves. It is therefore necessary to associate each component with the most representative basis.

In [7], the authors show that the curvelet transform [3] is relevant for extracting non-local phenomena propagating temporally. It thus seems particularly interesting to model long range waves present in a dynamic texture.

The second part of the model is composed of locally oscillating phenomena that will be extracted using the local cosine transform.

The dictionary that we use in the Morphological Component Analysis algorithm is then composed of the 2D+T curvelet transform Φ_1 and the 2D+T local cosine transform Φ_2.

2.3 Thresholding strategy

The purpose of this work is the decomposition of natural dynamic textures, therefore our experiments have been conducted on sequences from the DynTex database [13], a large database of dynamic textures. The processed sequences have a duration of 5 seconds (128 images) and a size of 648 by 540 pixels [1]. On volumes of such a size, the computation time is non negligible, as some transforms require several minutes.

Let function $T()$ measures 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 platforms[2] have been used for the chosen dictionary, giving the computation time presented in table 1.

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

Table 1 Computation time required for performing an analysis or a synthesis with the chosen dictionary on 2 different hardware configurations.

[1] *i.e.* more than 44 million voxels

[2] **Platform 1** : Processor 32 bits, 2.4GHz, 4GB of RAM
Platform 2 : Processor 64 bits, 3.2GHz, 24GB of RAM

A recent work [2] 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 seconds sequence is given by: $100*(T(\Phi_1^T)+T(\Phi_1)+T(\Phi_2^T)+T(\Phi_2))$, which corresponds to 21 *hours* on platform 1, and around 11 *hours* on platform 2.

If we extend this result to the entire DynTex database, 605 days of calculation are required on a standard computer. This computation time can be reduced to 309 days on a dedicated server.

Recently, Bobin *et al.* [2] have proposed a thresholding strategy 'Mean of Max' (MoMS) that enables to obtain similar results but with fewer iterations (50 in average instead of 100). It requires a computation time of approximately 15 hours 45 (respectively 6 hours on platform 2) for a 5 seconds video, resulting in approximately 453 days (respectively 232 days) for the whole database.

For indexing the whole DynTex database, the computation time of the MoMS is still acceptable, since it is always possible to divide the workload between 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.

3 Two new thresholding strategies

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 3 shows two different evolutions of $\lambda^{(k)}$ corresponding to two strategies (S1) and (S2). Evolution of $\lambda^{(k)}$ is slower in case (S1) than in case (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 (LTS) leads to the optimum $\lambda^{(k)}$ for 100 iterations [2]. 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 suboptimal. A good strategy for computing $\lambda^{(k)}$ should lead to a slope greater than 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

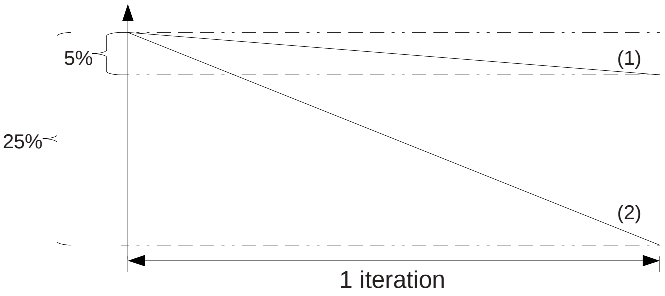


Fig. 3 Two thresholding strategies leading to different evolutions of the threshold value during one iteration of the main loop.

strategy tends to reduce too drastically this slope, or even almost cancel it.

3.1 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:

$$\lambda^{(k+1)} = \min\left(\frac{1}{2}(m_1+m_2), \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{100}\right) \tag{5}$$

where:
$m_1 = \max\limits_{\forall i} \left\| \Phi_i^T r^{(k)} \right\|_\infty$
$m_2 = \max\limits_{\forall j, j \neq i_0} \left\| \Phi_j^T r^{(k)} \right\|_\infty$ with $i_0 = \operatorname*{argmax}\limits_{\forall i} \left\| \Phi_i^T r^{(k)} \right\|_\infty$
$r^{(k)} = y - \sum_{j=1}^{K} \tilde{y}_j^{(k)}$ being the total residual

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 $\lambda^{(k+1)} = \lambda^{(k)} - \dfrac{\lambda^{(1)} - \lambda_{\min}}{100}$. 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.

3.2 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 4 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, [16] 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 4). 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 set to a large value.

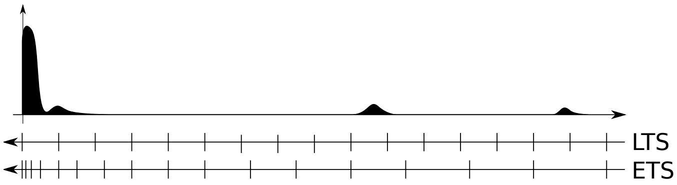


Fig. 4 Threshold intervals for the two strategies LTS and ETS on an illustrative distribution.

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:

$$\lambda^{(k+1)} = \min\left(\frac{1}{2}(m_1+m_2), \lambda^{(k)} * \left(\lambda^{(1)} - \lambda_{\min}\right)^{-\frac{1}{99}}\right) \tag{6}$$

In other words, when MoMS leads to values of $\lambda^{(k+1)}$ evolving too slowly, $\lambda^{(k+1)}$ follows the ETS strategy, $\lambda^{(k+1)} = \lambda^{(k)} * \left(\lambda^{(1)} - \lambda_{\min}\right)^{-\frac{1}{99}}$. 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.

3.3 Evaluation

In a first step the criterion of the gain in computing time is studied, then in a second step, several criteria for evaluating the decomposition quality are presented.

3.3.1 Computation time consideration

In both cases, the number of iterations required for the decomposition of an image sequence of DynTex decreases sharply. Indeed, it takes an average of 12 iterations for ATSLc and of 17 iterations for ATSEc to achieve decomposition.

The gain in terms of computing time is not proportional to the number of iterations removed. Indeed, as in the case of MoMS strategy, ATSLc and ATSEc strategies require an additional projection on all bases to calculate m_1 and m_2.

The computing time for a sequence of images corresponds to the relationship : (Number of iterations)$*(2 * T(\Phi_1) + T(\Phi_1^T) + 2 * T(\Phi_2^T) + T(\Phi_2))$. The average performance of the two strategies, ATSLc and ATSEc, is estimated in table 2 on the whole DynTex database. The computation time strategies of the literature are also presented.

	LTS	ETS	MoMS	ATSLc	ATSEc	
Platform 1	≈ 21h	≈ 21h	≈ 15h45	≈ 3h45	≈ 5h20	Estimated time for a video
Platform 2	≈ 10h45	≈ 10h45	≈ 8h	≈ 1h55	≈ 2h45	
Platform 1	≈ 605d	≈ 605d	≈ 453d	≈ 108d	≈ 154d	Estimated time for the whole
Platform 2	≈ 309d	≈ 309d	≈ 232d	≈ 56d	≈ 79d	DynTex database

Table 2 Estimated computation times required to perform the decomposition using the MCA algorithm for different thresholding strategies. This is the average time estimated on the DynTex database for an image sequence. Moreover, the number of days for complete decomposition of the database is indicated, for both platforms.

In the case of the use of platform 2, about 2 hours are needed for the decomposition of an image sequence with a size of $720 \times 576 \times 128$ voxels using the ATSLc strategy. It is a reduction by a factor of 5 of the computing time compared to the original LTS and ETS strategies.

If this result is extended to the whole of DynTex database, it takes about 60 days to perform a decomposition using ATSLc and ATSEc strategies. Moreover, this time can be divided by the number of cores used. In our case, the decomposition of the whole image sequences of dynamic textures, *i.e.* about 700, was carried out in a week.

In terms of computation time, the gain achieved by using ATSLc and AT-

SEc strategies seems very promising. An evaluation of the quality of the results is now presented.

3.3.2 ℓ_0-norm criteria

At first, the quality of the decomposition using ATSLc and ATSEc strategies through the ℓ_0-norm, as compared to LTS and ETS original strategies, is studied.

The number of coefficients selected by the MCA algorithm over the iterations k after the thresholding step is:

$$N_{coef}^{(k)} = \sum_{i=1}^{N} \left\| \alpha_i^{(k)} \right\|_0 \tag{7}$$

This indicator is used to evaluate the quality of the thresholding strategy. Indeed, for a poorly suited strategy, the number of coefficients selected will be irregular during iterations. This phenomenon is observed through a non-regular growth curve. In other words, during iterations, the more $N_{coef}^{(k)}$ evolves regularly, the more the thresholding strategy can be considered as being good.

On plots of figure 5, the evolution of $N_{coef}^{(k)}$ was computed for LTS, ETS, ATSLc and ATSEc strategies for four different sequences; they are displayed here in logarithmic scale. Observations made on these four videos can apply to all the sequences that we have tested (about 200 image sequences):

- The ATSLc and ATSEc strategies have similar behavior during the first iterations and diverge thereafter. During these first iterations, the two strategies choose the maximum slope that is $\frac{1}{2}(m_1 + m_2)$, then each of them behaves differently according to their respective corrections.
- For strategy LTS, most coefficients are selected in the latest iterations of the algorithm (the last 20 approximately). This is observed through the rapid growth curve $N_{coef}^{(k)}$ in the last steps. Thus, this strategy can lead to carrying out many iterations without selecting coefficients, then to a selection that is too fast at the end.
- By observing the growth of $N_{coef}^{(k)}$ for the ATSLc and ATSEc strategies, we can observe that the ATSEc is more regular. Indeed, the number of coefficients selected by the ATSLc strategy increases rapidly in the last iterations, contrary to ATSEc.

Now let us observe, for each basis Φ_i, the error on the choice of coefficients after the decomposition process. To this aim, the selected coefficients for two different threshold strategies are studied and the differences counted.

This criterion is formalized as follows:

$$\forall i, \quad \zeta_i = \frac{1}{M} \left\| \Gamma\left(\alpha_i^{(S1)}\right) - \Gamma\left(\alpha_i^{(S2)}\right) \right\|_0 \tag{8}$$

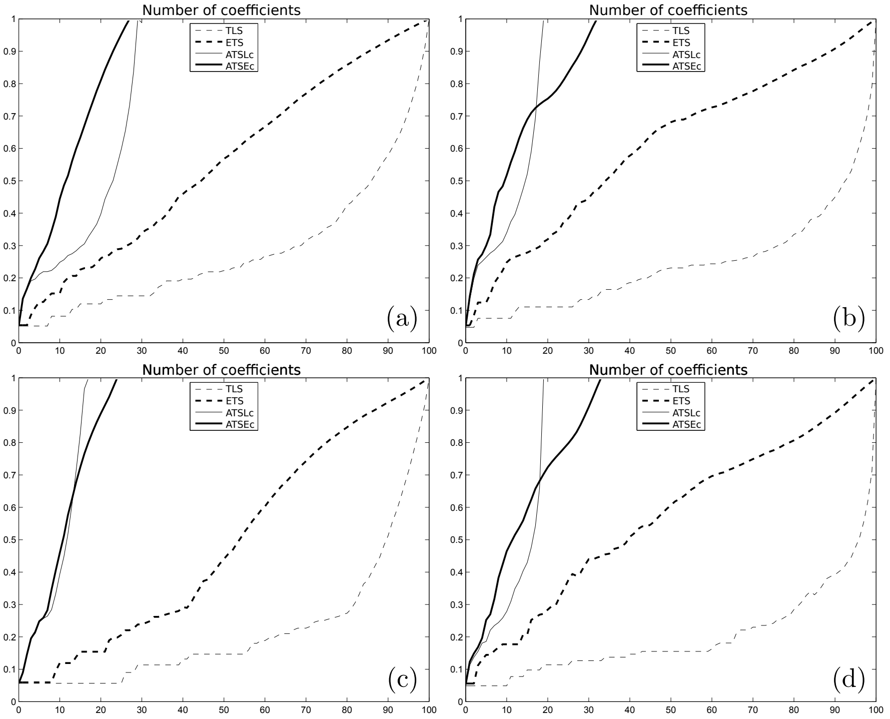


Fig. 5 *Study of LTS, ETS, ATSLc and ATSEc strategies. The plots represent the number of thresholded coefficients (logarithmic scale) during the iterations on four different image sequences.*

whereby $S1$ and $S2$ are the two strategies in competition, $\alpha_i^{(S1)}$ (respectively $\alpha_i^{(S2)}$) coefficients of the basis Φ_i selected by the MCA algorithm using S1 strategy (respectively S2), M the total number of coefficients α_i, and $\Gamma(A)$ the operator setting the value to 1 if the coefficients are different from 0.

Table 3 presents the means and standard deviations of the errors on the coefficients poorly selected between two competing strategies (for example LTS and ATSLc) for each basis of the dictionary. These data were obtained using 200 videos. ζ_1 (respectively ζ_2) represents the error on the component of the 2D+T curvelet transform (respectively the component of the 2D+T local cosine transform).

We note, as for criterion (8), that:

- Both ATSLc and ATSEc strategies are approaching the solutions obtained by LTS and ETS strategies. Indeed, the greatest average error is 16.66%. However, this is the comparison between ATSEc and LTS strategies, which is not the easiest to compare. Even if the final components for the two

	LTS				ETS			
	ζ_1		ζ_2		ζ_1		ζ_2	
	μ	σ	μ	σ	μ	σ	μ	σ
ATSLc	6.01%	3.12	0.24%	0.16	15.93%	11.12	1.66%	1.48
ATSEc	16.66%	11.56	0.90%	0.70	2.46%	1.67	0.98%	1.03

Table 3 Means and standard deviations of the error on the coefficients poorly selected of S1 strategy (for example ATSLc) compared to a reference S2 strategy (for example LTS) for each basis of the dictionary (ζ_1 for the 2D+T curvelet transform and ζ_2 for the 2D+T local cosine transform).

strategies are similar (this will be discussed later), the position of selected coefficients may differ.

- The adaptive thresholding strategy best approaching original strategies is the ATSEc strategy. This is observed through the means and standard deviations of the errors: ζ_1 and ζ_2 which are weak.
- As observed using criterion (7), ATSLc strategy tends to select quickly many coefficients which can lead to decomposition errors. This is observed by an average error higher than in the case of the use of ATSEc. This strategy implied many more iterations to divide the coefficients around the origin, thus leading to a small error. These statements can be made for other criteria.

3.3.3 ℓ_2-norm criteria

After studying the influence of strategies on the selection of coefficients, the ℓ_2-norm reconstruction error of the algorithm over the iterations is computed. It is formalized by:

$$\xi_T^{(k)} = \left\| y - \sum_{i=1}^{N} \tilde{y}_i^{(k)} \right\|_2 \tag{9}$$

As for criterion (7), the regularity of the plot reflects the performance of the thresholding strategy: the more rapidly and steadily the reconstruction error decreases over the iterations, the more the selected coefficients are relevant and representative of the original signal. On plots of figure 6, the evolution of $\xi_T^{(k)}$ was calculated for the LTS, ETS, ATSLc and ATSEc strategies for four image sequences. As for criterion (8), the results obtained on these four videos are representative of all tests (200 videos).

Several observations can be made:

- in the case of the LTS strategy, the most representative coefficients of the original signal are selected during the latest iteration of the algorithm. This is observed through the rapid decrease of the plot $\xi_T^{(k)}$.

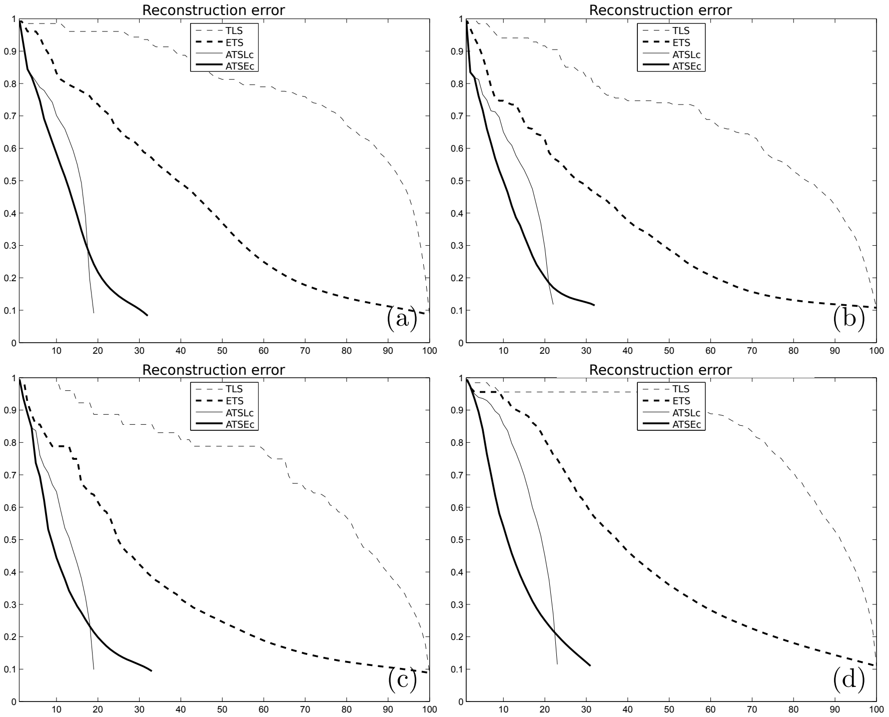


Fig. 6 *Study of LTS, ETS, ATSLc and ATSEc strategies. The plots represent the reconstruction error in ℓ_2-norm during the iterations.*

- the ETS strategy selects many relevant coefficients during the first iterations: a rapid decline of $\xi_T^{(k)}$ is observed. Then it decreases for the remaining coefficients to be allocated. This phenomenon can be explained by the fact that the size ranges of selected coefficients decrease gradually, as a result of the iterations of the MCA algorithm.
- in the case of the two adaptive strategies, ATSLc and ATSEc, the reconstruction error decreases very quickly. Both strategies regularly select pertinent coefficients and quickly approach the original signal.
- the adaptive ATSEc strategy seems more efficient than ATSLc. Indeed, a sudden drop in the reconstruction error is observed in the case of ATSLc. This is because the MCA algorithm selects many relevant coefficients within few iterations. However, the coefficient allocation can be problematic, unlike the ATSEc strategy for which the allocation is made on a larger number of iterations.

The previous study focused on the ℓ_2-norm error between the reconstruction and the original image sequence. However, we observe that the last iteration of the algorithm (cf. plots on figure 6) is identical for all strategies. This is explained by the construction of different strategies: each led the threshold $\lambda^{(k)}$ to a minimum value λ_{min} determined by the user.

Criterion (9) thus makes it possible to observe the performance of strategies over the iterations of the algorithm, it does not make it possible to evaluate the quality of the different components extracted.

For this purpose, a criterion that compares the morphological components obtained by the competing strategies is proposed:

$$\forall i, \quad \xi_i = \frac{1}{\xi_{max}} \left\| \tilde{y}_i^{(S1)} - \tilde{y}_i^{(S2)} \right\|_2 \tag{10}$$

where $\tilde{y}_i^{(S1)}$ (respectively $\tilde{y}_i^{(S2)}$) is the morphological component i estimated with the $S1$ strategy (respectively $S2$). ξ_{max} is computed for each component $\tilde{y}_i^{(S2)}$ as the biggest possible mistake we can commit.

Table 4 presents the means and standard deviations of errors in ℓ_2-norm on each base of the dictionary with respect to the chosen strategies. These results were obtained from 200 image sequences. ξ_1 (respectively ξ_2 and ξ_T) represents the error on the component of the 2D+T curvelet transform (respectively the component of the 2D+T local cosine transform and the reconstruction).

	LTS					
	ξ_T		ξ_1		ξ_2	
	μ	σ	μ	σ	μ	σ
ATSLc	0.27%	0.10	1.56%	1.08	1.56%	0.92
ATSEc	0.48%	0.20	2.30%	1.82	2.18%	1.50

	ETS					
	ξ_T		ξ_1		ξ_2	
	μ	σ	μ	σ	μ	σ
ATSLc	0.39%	0.15	2.67%	1.90	2.43%	1.58
ATSEc	0.38%	0.17	1.55%	1.51	1.50%	1.22

Table 4 Means and standard deviations of the error in ℓ_2-norm between two strategies $S1$ (for example ATSLc) and S2 (for example LTS) for each base of dictionary (ξ_1 for the 2D+T curvelet transform, ξ_2 for the 2D+T local cosine transform) and its reconstruction ξ_T.

Observations are summarized here:

- the maximum mean error between reconstructions obtained from two different strategies is only 0.48%. This indicates that, whatever the strategy used, the reconstruction will be almost identical.
- both adaptive strategies ATSLc and ATSEc, fairly approach the decompositions obtained using the original strategies. Indeed, for each component,

the maximum average error observed between two different strategies is only 2.67%.

Considering the different criteria studied and the gain in terms of computation time, we can establish that the proposed strategies prove to be relevant for the processing of dynamic texture sequences. In the next section, we present the decomposition results of dynamic textures using the MCA algorithm using the adaptive thresholding strategy with exponential correction.

4 Experiments

4.1 Conditions

Results obtained using ATSLc and ATSEc strategies are promising and satisfactory. In this section, three of them are described in details[3].

Decompositions of dynamic textures presented here were obtained using a dictionary composed of the 2D+T curvelet transform and the 2D+T local cosine transform.

The curvelet transform was calculated using 5 scales of decomposition, and 4 angular subdivisions at each change of scale. The local cosine transform was performed on windows of size $32 \times 32 \times 32$ voxels. These decompositions were calculated using the adaptive thresholding strategy with exponential correction. As in many experiments [9], we use $\lambda_{min} = \tau\sigma$ with $\tau = 3$.

4.2 Results

The first video shows a duck drifting slowly in a canal (Figure 7). Reflections of trees in the rippling water and a static texture background are also observable. Figure 7 shows the decomposition results obtained on this video using the MCA algorithm with the ATSEc strategy.

The geometric component is retrieved using the curvelet transform while the texture component is obtained by the 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.

Spatiotemporal cuts along a xt plane make it possible to visualize the obtained decomposition. They show that the different objects in the scene

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

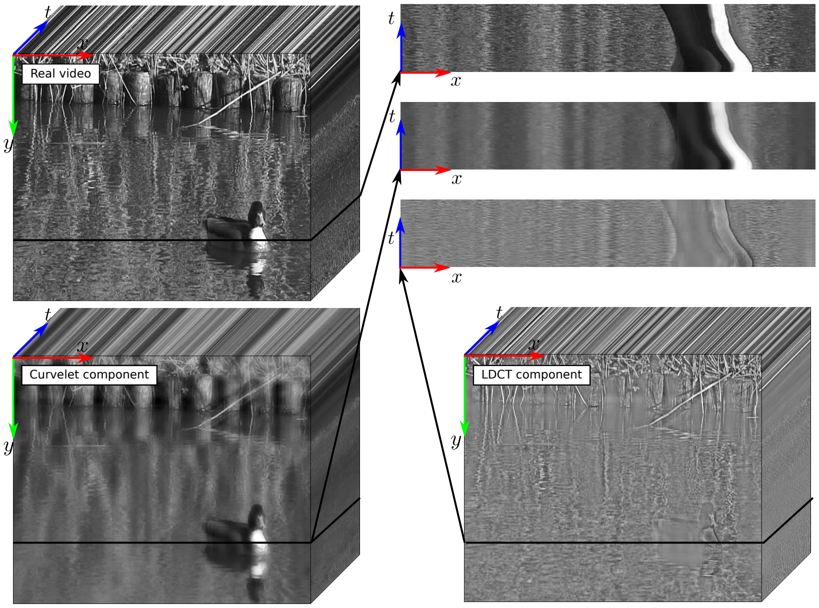


Fig. 7 Results of the MCA decomposition on a video using the ATSEc strategy. Spatio-temporal cuts xt enable to emphasize the temporal aspect of the decomposition.

(the duck, reflections of the trees on water, . . .) 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.

This decomposition can bring us information that was not discernable 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 second sequence presented is the surface of the sea near a beach. It consists of a carrier wave (the waves roll) and local phenomena (the foam). Figure 8 shows the decomposition results.

The part corresponding to the curvelet transform consists of the roller waves while the local foam and small ripples are captured by the local cosine tranform. On areas represented as surfaces, the texture component does not contain the rolls of the waves, completely present in the geometry. The separation between the waves roll and the foam is clearly observable.

The next image sequence represents a fountain (Figure 9). 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 9.

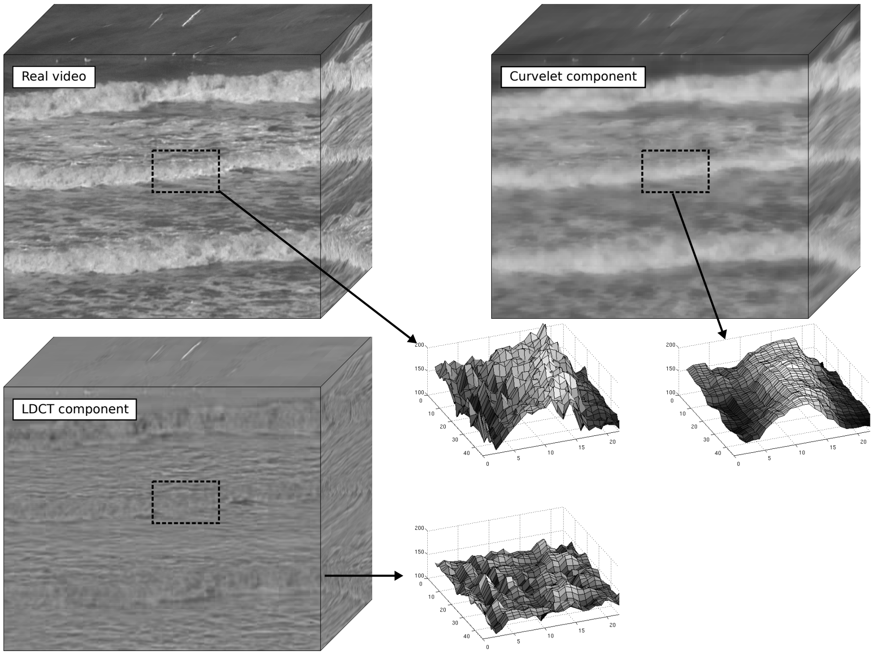


Fig. 8 *Decomposition results of a video using the MCA algorithm and the ATSEc strategy. Regions of Interest are plotted as surfaces in order to better visualize the algorithm behaviour.*

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 ripples.

5 Conclusion and perspectives

This paper deals with the decomposition of dynamic textures in image sequences into different dynamical components. After considering the MCA algorithm and 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 calcula-

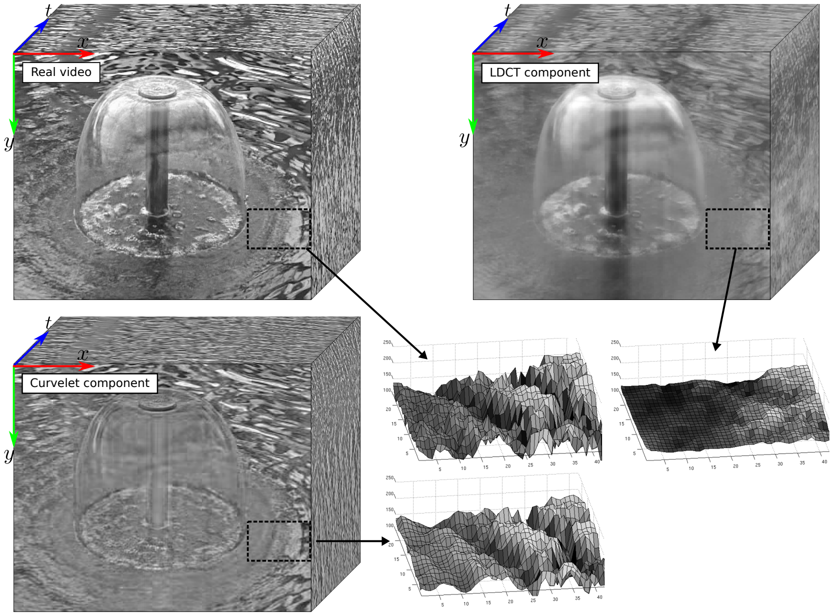


Fig. 9 Decomposition results of a video using the MCA algorithm and the ATSEc strategy. Regions of Interest are plotted as surfaces in order to better visualize the algorithm behaviour.

tions are reduced by about five times, with equivalent quality of results. In our research context of dynamic texture indexing, it can eliminate 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 make it possible to understand the different complex phenomena present in dynamic textures.

Other thresholding strategies are being studied to further improve this computation time. It is particularly necessary to develop strategies that better take into account our proposed model and the features of natural dynamic textures. The extracted components of dynamic textures can later be used as features for video retrieval applications.

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 overall movement,. . .) and some characterizing more local phenomena (speed, local vortex, . . .).

References

1. Aujol, J., Chambolle, A.: Dual norms and image decomposition models. Computer Vision **63**, 85–104 (2005)
2. Bobin, J., Starck, J., Fadili, J., Moudden, Y., Donoho, D.: Morphological component analysis : An adaptive thresholding strategy. IEEE Transactions on Image Processing pp. 2675–2681 (2007)
3. Candès, E., Demanet, L., Donoho, D., Ying, L.: Fast discrete curvelet transforms. Tech. rep., California Institute of Technology (2005)
4. Chan, T., Osher, S., Shen, J.: The digital tv filter and nonlinear denoising. IEEE Transactions on Image Processing **10**, 231–241 (2001)
5. Doretto, G., Chiuso, A., Wu, Y., Soatto, S.: Dynamic textures. International Journal of Computer Vision **51**, 91–109 (2003)
6. Doretto, G., Cremers, D., Favaro, P., Soatto, S.: Dynamic texture segmentation. In: IEEE International Conference on Computer Vision (ICCV 03), pp. 1236–1242. Beijing, China (2003)
7. Dubois, S., Péteri, R., Ménard, M.: A 3D discrete curvelet based method for segmenting dynamic textures. In: International Conference on Image Processing (ICIP 09), pp. 1373–1376. Cairo, Egypt (2009)
8. Dubois, S., Péteri, R., Ménard, M.: A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition. In: Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA 09), pp. 314–321. Povoa de Varzim, Portugal (2009)
9. Fadili, J., Starck, J., Elad, M., Donoho, D.: Mcalab: Reproducible research in signal and image decomposition and inpainting. IEEE Computing in Science and Engineering **12**, 44–63 (2010)
10. Finch, M.: GPU Gems: Programming Techniques, Tips, and Tricks for Real-Time Graphics, Chap.1. Randima Fernando (2004). DOI http://http.developer.nvidia.com/GPUGems/gpugems_part01.html. URL `http://http.developer.nvidia.com/GPUGems/gpugems_part01.html`
11. Nelson, R., Polana, R.: Qualitative recognition of motion using temporal texture. Computer Vision and Image Understanding **56**, 78–89 (1992)
12. Péteri, R., Chetverikov, D.: Qualitative characterization of dynamic textures for video retrieval. In: International Conference on Computer Vision and Graphics (ICCVG 04), pp. 33–38. Warsaw, Poland (2004)
13. Péteri, R., Fazekas, S., Huiskes, M.: Dyntex: A comprehensive database of dynamic textures. Pattern Recognition Letters **31**, 1627–1632 (2010)
14. Saisan, P., Doretto, G., Wu, Y., Soatto, S.: Dynamic texture recognition. In: Conference on Computer Vision and Pattern Recognition (CVPR 01), pp. 58–63. Kauai, USA (2001)
15. Starck, J., Elad, M., Donoho, D.: Redundant multiscale transforms and their application for morphological component analysis. Advances in Imaging and Electron Physics **132** (2004)
16. Starck, J., Elad, M., Donoho, D.: Image decomposition via the combination of sparse representations and a variational approach. IEEE Transactions on Image Processing **14**, 1570–1582 (2005)
17. Woiselle, A., Starck, J., Fadili, J.: Inpainting with 3d sparse transforms. In: 22ème édition du colloque GRETSI. Dijon, France (2009)