



Modèle de Textures Dynamiques et décomposition par l'approche Morphological Component Analysis

Rapport Interne

Sloven Dubois, Renaud Péteri, Michel Ménard

Laboratoire de **M**athématiques
Image et **A**pplications

Laboratoire **I**nformatique
I'Image et **I**'Interaction

Université de La Rochelle
Avenue Michel Crépeau, 17042 La Rochelle Cedex
FRANCE

2009

Résumé

Cette rapport apporte plusieurs contributions au domaine de l'analyse et de caractérisation de textures dynamiques.

La première porte sur la formalisation d'une texture dynamique. Celle-ci se compose d'une onde se propageant à grande échelle et de phénomènes locaux présentant, par exemple, des oscillations. L'algorithme des Morphological Component Analysis (MCA) est utilisé pour retrouver, à l'aide de plusieurs dictionnaires, ces composantes.

La deuxième contribution porte sur la définition d'une nouvelle stratégie de seuillage adaptative pour les MCA qui permet de diminuer fortement les temps de calcul dans le cas de séquences naturelles.

Table des matières

Introduction	2
Un modèle de texture dynamique	3
Décomposition de textures dynamiques	4
Morphological Component Analysis	5
Choix du dictionnaire	5
Stratégie de seuillage	7
Stratégie de seuillage adaptative	8
Résultats	9
Conclusion et perspectives	9

Introduction

Les textures dynamiques sont un thème de recherche récent dans le domaine de l'analyse de séquences d'images. Elles se définissent comme un phénomène variant dans le temps et possédant une certaine répétitivité à la fois spatiale et temporelle. Un drapeau dans le vent, les risées à la surface de l'eau, de la fumée, ou un escalator sont autant de textures dynamiques présentes dans des vidéos. Elles ne se présentent donc pas uniquement comme une extension des textures statiques au domaine temporel, mais comme un phénomène plus complexe résultant de plusieurs dynamiques. Sans être exhaustif, leur étude est un sujet de recherche actif comportant de nombreuses applications comme la synthèse [15, 9], la segmentation [1, 6], la caractérisation [9, 12, 16].

Le contexte de nos travaux se situe dans le cadre de l'indexation de textures dynamiques pour la recherche automatique dans des bases vidéos [8].

Chaque texture dynamique possède ses propres caractéristiques, comme sa stationnarité, sa répétitivité, sa vitesse de propagation, ... En regardant une séquence vidéo de la surface d'un lac (cf. Figure 1.(b)) deux mouvements peuvent être observés : un mouvement haute fréquence (les petites vagues) porté par un mouvement d'ensemble (l'onde interne).

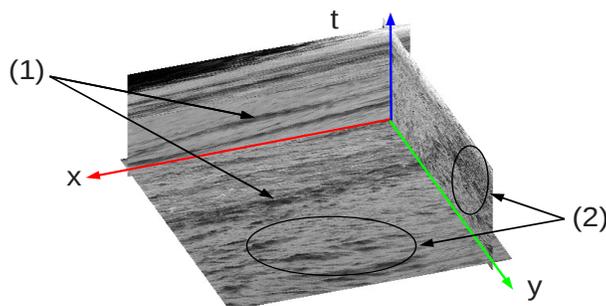


FIG. 1 – Sections 2D+T de textures dynamiques : on y observe plusieurs ondes porteuses (1) et des phénomènes locaux oscillants (2).

De nombreuses textures dynamiques se décomposent ainsi, en une ou plusieurs onde(s) porteuse(s) additionnée(s) d'un ou plusieurs phénomène(s) localisé(s). Afin de caractériser au mieux ces deux ensembles de composantes, il est nécessaire de les extraire de la séquence séparément.

En Section 2, nous présentons un modèle formel de textures dynamiques. En Section 3, après avoir rappelé l'intérêt de la décomposition pour la caractérisation de textures et l'indexation de bases vidéos, l'algorithme des Morphological Component Analysis (MCA) est décrit. Les dictionnaires utilisés et adaptés aux séquences naturelles sont présentés en Section 3.1. Les temps de calcul actuels des MCA sont fortement liés à la stratégie de seuillage de l'algorithme. Nous proposons pour les textures naturelles une stratégie de seuillage adaptative. Les temps de calcul, sur ce type de textures, sont réduits en moyenne

d'un facteur quatre par rapport à l'algorithme de seuillage original. Nous présentons des résultats sur des séquences synthétisées afin de valider notre approche, puis sur des vidéos réelles issues de la base DynTex [11].

Un modèle de texture dynamique

Selon des recherches en synthèse [10] et des observations faites sur la base de données DynTex [11], dans une séquence vidéo, plusieurs textures dynamiques peuvent être présentes, avec des supports spatio-temporels différents qui peuvent donc se superposer. La composante texture d'une séquence vidéo T_V se définit alors comme la somme de N textures dynamiques Υ_i , chacune de support spatio-temporel Ω_i :

$$T_V = \sum_i \Upsilon_i \quad (1)$$

Certaines textures dynamiques présentant des transparences, comme la fumée, l'eau, ... les supports Ω_i peuvent s'intersecter.

Nous proposons d'écrire une texture dynamique Υ_i comme la composition d'une onde se propageant à grande échelle et de phénomènes localement oscillants :

$$\Upsilon_i = \sum_{\mathbf{x} \in \Omega_i} [\mathcal{P} + \mathcal{L}] \quad (2)$$

où $\mathbf{x} = (x, y, t)^T$ représente les coordonnées d'un voxel dans un cube vidéo T_V . \mathcal{P} et \mathcal{L} sont deux fonctions représentant respectivement l'onde porteuse et les phénomènes locaux d'une texture dynamique (par soucis de simplicité, nous ne considérons ici qu'une seule onde se propageant par texture dynamique).

L'onde porteuse \mathcal{P} est le phénomène le plus complexe. Sa forme dépend bien sûr de la séquence. Par exemple, pour la séquence présentée sur la figure 1, nous considérons une somme de fonctions cosinus, d'amplitude A_{p_i} , de pulsation ω_{p_i} et de déphasage propres ψ_{p_i} à chacune d'elle :

$$\mathcal{P} = \sum_{p_i \in \mathcal{P}} A_{p_i}(\mathbf{x}) e^{j(\omega_{p_i} \mathbf{x} + \psi_{p_i})} \quad (3)$$

Les phénomènes locaux sont des phénomènes présentant des oscillations. Ils se distinguent de ceux constituant l'onde porteuse par leur comportement purement local :

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

avec pour un phénomène local ℓ particulier, $\mathcal{N}_G(\mu_\ell, \Sigma_\ell)$ un noyau gaussien associé décrivant sa localisation, $A_{\ell_k^\ell}$, $\omega_{\ell_k^\ell}$ et $\psi_{\ell_k^\ell}$ l'amplitude, la pulsation et le

déphasage associés à ℓ .

La figure 2 représente une séquence vidéo synthétique obtenue à l'aide du modèle ci-dessus.

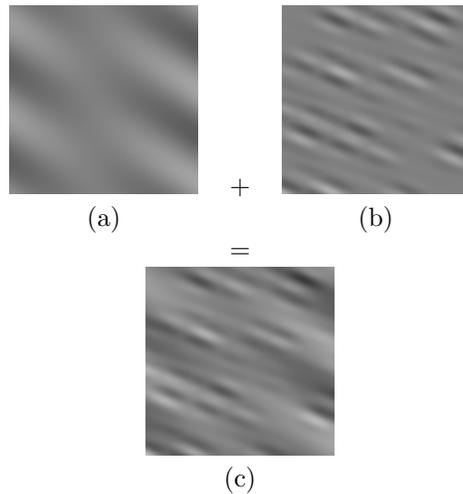


FIG. 2 – Texture dynamique synthétique obtenue à l'aide du modèle décrit précédemment (2). (a) Image de la vidéo représentant l'onde porteuse. (b) phénomènes localement oscillants. (c) texture dynamique composée de (a) et (b).

Décomposition de textures dynamiques

Les résultats de synthèse, dont un est présenté sur la figure 2, obtenus à l'aide des équations (2), (3) et (4) témoignent de la pertinence du modèle pour la représentation de certaines textures naturelles, comme par exemple les ondes à la surface de l'eau, ou le mouvement d'un drapeau dans le vent.

Le problème de l'identification des coefficients et des paramètres du modèle pour la synthèse d'une texture donnée est un problème difficile. Cependant, les résultats expérimentaux obtenus valident l'hypothèse d'une superposition de composantes linéaires. Les approches de décomposition d'images de la littérature [2, 5, 13] semblent donc pertinentes pour l'extraction de ces composantes.

Nous avons choisi l'approche Morphological Component Analysis (MCA) de part la richesse des bibliothèques d'analyse qu'elle permet d'utiliser. La diversité et l'intégration aisée de ces dernières, la souplesse de l'algorithme sont importantes au regard de la complexité des textures naturelles.

Morphological Component Analysis

Soit un signal y que l'on décrit comme une superposition linéaire de N composantes morphologiques perturbé par un bruit ε :

$$y = \sum_{i=1}^N y_i + \varepsilon \quad (5)$$

L'approche *MCA* permet de trouver une solution acceptable au problème inverse de décomposition dans des bases, et donc d'extraire les composantes $(y_i)_{i=1,\dots,N}$ à partir de l'observation dégradée y selon un critère de parcimonie. L'approche *MCA* suppose que chaque composante y_i est représentée de manière creuse dans une base associée Φ_i .

$$\forall i = 1, \dots, N, \quad y_i = \Phi_i \alpha_i \quad (6)$$

Ainsi, le dictionnaire représentant l'ensemble des bases peut être construit en associant plusieurs transformées $\Phi = [\Phi_1, \dots, \Phi_N]$ telles que pour chaque i , y_i est bien représenté dans Φ_i et n'est pas, ou alors très peu représenté dans Φ_j , ($j \neq i$). Ceci se traduit par :

$$\forall i, j \neq i \quad \|\Phi_i^T y_i\|_0 < \|\Phi_j^T y_i\|_0 \quad (7)$$

Le choix des bases est bien sûr primordial. Chaque transformée va posséder ses propres caractéristiques et permettre l'extraction de tels ou tels phénomènes.

Résoudre l'équation (7) revient à trouver une solution au système suivant : $y = \Phi \alpha$. Starck et al. proposent dans [13] et [14] de résoudre celui-ci, et de trouver les composantes morphologiques $(y_i)_{i=1,\dots,N}$ en résolvant le problème d'optimisation suivant :

$$\min_{y_1, \dots, y_N} \sum_{i=1}^N \|\Phi_i^T y_i\|_p^p \quad \text{tel que} \quad \left\| y - \sum_{i=1}^N y_i \right\|_2 \leq \sigma \quad (8)$$

où le terme $\|\Phi_i^T y_i\|_p^p$ pénalise la non parcimonie (pour $0 \leq p \leq 1$, nous avons l'évaluation de la parcimonie la plus intéressante). σ représente l'écart-type du bruit.

Ce problème d'optimisation est résolu par un algorithme itératif de seuillage dans les différentes bases. Les étapes principales de celui-ci sont décrites dans l'algorithme 1.

Choix du dictionnaire

Le point crucial dans l'approche *MCA* est la définition du dictionnaire. Un choix non adapté des transformations par rapport à la dynamique des phénomènes présents dans la séquence est préjudiciable quant à la qualité du résultat : décomposition non pertinente, norme ℓ_0 importante, coefficients non

Algorithme 1 Morphological Component Analysis

Tache : Décomposition d'un signal nD sur un dictionnaire Φ .

Paramètres :

- Le signal y à décomposer
- Le dictionnaire $\Phi = [\Phi_1, \dots, \Phi_K]$
- La stratégie de seuillage **stratégie**
- La condition d'arrêt σ

Initialisation :

// Mise à zéros des composantes à estimer

pour $i = 1$ à N **faire**

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

fin pour

// Initialisation de λ

$\lambda^{(1)} = \text{initialisation_lambda}(\text{stratégie})$

// Initialisation du nombre d'itérations

$k = 1$

Boucle principale :

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

// Pour chaque composante

pour $i = 1$ à N **faire**

// Calcul du résidu marginal

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

// Projection du résidu marginal dans la base Φ_i

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

// Estimation de la composante \tilde{y}_i

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

fin pour

// Mise à jour du seuil λ

$\lambda^{(k+1)} = \text{mise_à_jour}(\lambda^{(k)}, \text{stratégie})$

// Progression d'une itération

$k = k + 1$

fin tant que

représentatifs.

Le modèle de texture dynamique (cf. équation 2) suppose deux phénomènes distincts. Il est donc nécessaire d'associer à chacun d'eux la base la plus représentative.

Dans [7], les auteurs montrent que la transformée en curvelets [4] apporte une discrimination pertinente sur des phénomènes non locaux se propageant temporellement. Elle semble donc particulièrement intéressante pour modéliser les ondes porteuses présentes dans une texture dynamique.

La deuxième partie du modèle repose sur des phénomènes localement oscillants. Par conséquent, la deuxième base du dictionnaire est construite à partir d'une transformée locale adaptée aux oscillations. Nous avons proposé pour cette étude la transformée en cosinus qui semble la mieux adaptée à leur capture.

Le dictionnaire Φ de décomposition des MCA est donc composé de la transformée en curvelets Φ_1 et de la transformée en cosinus Φ_2 .

Stratégie de seuillage

DynTex est une base de séquences d'images présentant de nombreuses textures dynamiques, elle nous semblait donc appropriée pour cette étude. Les séquences analysées ont une durée de 5 secondes. La taille des images est de 648 sur 540. Les temps de calcul des algorithmes de décomposition représentent une des principales difficultés pour l'analyse de séquences (indexation et recherche). Certaines transformées nécessitent en effet plusieurs minutes de temps de calcul. Soit la fonction $T()$ mesurant le temps d'exécution d'une transformée Φ durant un cycle de l'algorithme (obtention des coefficients via Φ , ou synthèse/reconstruction via Φ^T), nous avons obtenu pour notre dictionnaire :

$$\begin{aligned} T(\Phi_1) &\approx T(\Phi_1^T) \approx 240 \text{ secondes} \\ T(\Phi_2) &\approx T(\Phi_2^T) \approx 120 \text{ secondes} \end{aligned}$$

Dans la littérature [3], les auteurs s'accordent à dire qu'une centaine d'itérations est nécessaire à l'algorithme des MCA pour établir une bonne séparation des différentes composantes lorsqu'une stratégie de seuillage linéaire est utilisée, SSL. Pour le dictionnaire choisi dans cette étude, ceci représente un temps de calcul pour une séquence de : $100 * (T(\Phi_1^T) + T(\Phi_1) + T(\Phi_2^T) + T(\Phi_2)) = 20h$ (pour une vidéo de 5 secondes).

Si nous étendons ce résultat à l'ensemble des séquences de la base de données DynTex, et toujours pour une durée de séquences de 5 secondes, nous obtenons environ 583 jours de calcul pour effectuer correctement la décomposition.

Dernièrement, Bobin *et al.* ont proposé une stratégie de seuillage 'Mean of Max', SMoM [3] permettant d'obtenir des résultats équivalents mais avec un nombre d'itérations moindre (25 en moyenne au lieu de 100). Ceci représente un temps de calcul d'environ 7 heures 30 pour une de nos séquences vidéo, conduisant à environ 219 jours pour l'ensemble de la base.

Pour l'indexation d'une base comme DynTex, les temps de calcul de la stratégie SMoM restent acceptables, puisqu'il est toujours possible de répartir

la charge de calcul sur plusieurs unités. Dans le cadre de la recherche d'une texture particulière à l'aide d'une séquence requête, ces calculs ne peuvent se faire actuellement qu'à partir de séquences de durée limitée et de résolution faible. Un des objectifs de ce travail est donc de diminuer ces contraintes en proposant une nouvelle stratégie de seuillage.

Stratégie de seuillage adaptative

La qualité des résultats de la décomposition d'un signal, à l'aide de l'algorithme des MCA, dépend fortement de l'évolution du seuil $\lambda^{(k)}$ au cours d'une itération (boucle `for`). Nous montrons sur la figure 3 deux évolutions différentes de $\lambda^{(k)}$ correspondant à deux stratégies (1) et (2) appliquées dans une boucle `for` dans l'algorithme MCA. L'évolution de $\lambda^{(k)}$ est plus lente dans le cas (1) que dans celui de (2). Dans cet exemple, l'évolution (1), respectivement (2), conduit à répartir 5% des coefficients, respectivement 25%, sur les deux bases. Si on considère que l'évolution (1) est dans cet exemple optimale en terme de seuillage, une évolution non maîtrisée de la valeur de $\lambda^{(k)}$ (cas 2) amènera à répartir trop rapidement un grand nombre de coefficients dans les bases, dégradant ainsi la décomposition.

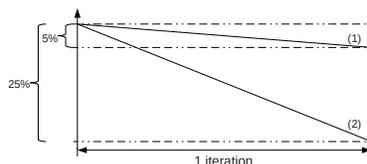


FIG. 3 – Deux stratégies de seuillage conduisant à des évolutions différentes de la valeur de seuil durant une itération.

La stratégie de seuillage linéaire, SSL, conduit à une évolution optimale de $\lambda^{(k)}$ lorsqu'une centaine d'itérations est fixée [3]. Dans un grand nombre de textures naturelles, nous avons constaté que le nombre d'itérations peut être fortement diminué, et dépend de la texture elle-même, SSL n'est alors plus optimum. Cependant, l'évolution selon SSL peut être considérée comme une pente minimum en dessous de laquelle l'évolution de $\lambda^{(k)}$ sera sous-optimum. Une bonne stratégie pour le calcul de $\lambda^{(k)}$ doit donc conduire à une pente supérieure ou égale à celle engendrée par SSL.

La stratégie 'Mean of Max', SMoM, est très intéressante car elle permet de modifier l'évolution de $\lambda^{(k)}$ lorsque cela s'avère nécessaire. Sur les séquences de textures naturelles, cette stratégie a cependant souvent tendance à diminuer fortement la pente, voire presque à l'annuler.

Nous proposons de rassembler ces deux stratégies en une nouvelle dite adaptative, SA, qui définit $\lambda^{(k)}$ comme le minimum des valeurs de $\lambda^{(k)}$ calculées par

les stratégies SSL et SMoM. SA se formalise ainsi :

$$\lambda^{(k)} = \min \left(\frac{1}{2}(m_1 + m_2), \lambda^{(k)} - \frac{\lambda^{(1)} - \lambda_{\min}}{100} \right) \quad (9)$$

avec :

$$\begin{aligned} m_1 &= \max_{\forall i} \|\Phi_i^T r^{(k)}\|_{\infty} \\ m_2 &= \max_{\forall j, j \neq i} \|\Phi_j^T r^{(k)}\|_{\infty} \\ r^{(k)} &= y - \sum_{j=1}^K \tilde{y}_j^{(k)} \text{ résidu total} \end{aligned}$$

Ainsi, avec cette stratégie, nous sommes assurés de modifier la valeur de $\lambda^{(k)}$ selon la plus grande pente.

Autrement dit, lorsque SMoM conduit à des valeurs de $\lambda^{(k)}$ évoluant faiblement, les valeurs de $\lambda^{(k)}$ suivent la stratégie SSL, $\frac{1}{2}(m_1 + m_2)$. Sinon, $\lambda^{(k)}$ suit la stratégie SMoM, et permet ainsi de diminuer le nombre de boucle `for` dans l'algorithme.

Résultats

L'application de SA permet un gain en temps de calcul important. En effet, en moyenne, pour une vidéo, le temps de calcul est d'environ 4 heures 30 (au lieu de 7h30), ce qui conduit à 131 jours pour l'ensemble de la base DynTex [11].

La figure 4.a montre le résultat d'une vidéo d'eau générée par notre modèle de textures dynamiques (voir équation 2). Après la décomposition à l'aide de l'algorithme des MCA et notre stratégie de seuillage, nous obtenons bien une onde porteuse (fig. 4.b) et des oscillations locales (fig.4.c). Ce résultat nous renforce dans le choix du dictionnaire composé de la transformée en curvelets et de la transformée en cosinus.

Appliqué à une séquence réelle de DynTex [11] d'une séquence d'eau coulant dans un canal (fig. 5.a), notre algorithme est capable de séparer les fronts de propagation du canal (fig. 5.b) des oscillations locales (fig. 5.c).

Conclusion et perspectives

Cet article formalise un nouveau modèle de texture dynamique. Celui-ci repose sur une onde se propageant à grande échelle additionnée de phénomènes locaux présentant des oscillations. Nous proposons d'utiliser les approches de décomposition afin de retrouver ces différentes composantes. L'algorithme MCA est adapté, mais souffre d'un temps de calcul important. Nous proposons dans ce cadre une nouvelle stratégie de seuillage qui conduit à un gain important en temps de calcul. Ceci permet de relâcher quelque peu les contraintes de faible résolution et de durée lors de requêtes se présentant sous la forme d'une

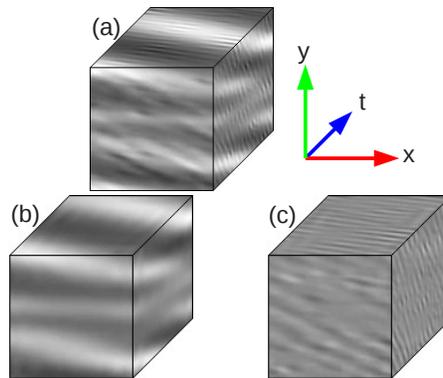


FIG. 4 – Résultat de la décomposition des MCA sur une vidéo de synthèse. (a) image de la vidéo de synthèse originale, (b) et (c) résultats de la décomposition à l'aide des MCAs, respectivement sur les bases curvelets et DCT

séquence.

D'autres stratégies de seuillage sont à l'étude afin d'améliorer encore les temps de calcul.

Il est nécessaire notamment de développer des stratégies prenant mieux en compte notre modèle et les caractéristiques des textures naturelles.

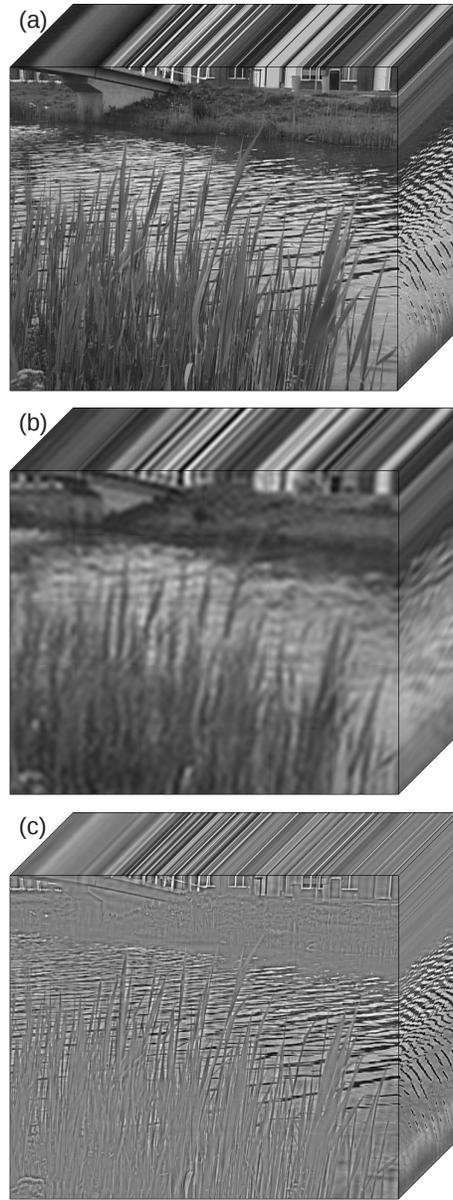


FIG. 5 – Résultat de la décomposition des MCA sur une vidéo réelle. (a) image de la vidéo originale, (b) et (c) résultats de la décomposition à l'aide des MCAs, respectivement sur les bases curvelets et DCT

Bibliographie

- [1] T. Amiaz, S. Fazekas, D. Chetverikov, and N. Kiryati. Detecting regions of dynamic texture. In L. N. in Computer Science, editor, *1st International Conference on Scale Space and Variational Methods in Computer Vision (SSVM'07)*, volume 4485, pages 848–859, 2007.
- [2] J. Aujol, , and A. Chambolle. Dual norms and image decomposition models. *Int. J. Comput. Vision*, 63(1) :85–104, 2005.
- [3] 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*, IEEE transactions on image processing, pages 2675–2681. Institute of Electrical and Electronics Engineers, 2007.
- [4] E. Candès, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. Technical report, California Institute of Technology, Mar. 2006.
- [5] A. Chan and N. Vasconcelos. Mixtures of dynamic textures. In *Proceedings of Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 1, pages 641–647, 2005.
- [6] G. Doretto, D. Cremers, P. Favaro, and S. Soatto. Dynamic texture segmentation. In *Proceedings of Ninth IEEE International Conference on Computer Vision (ICCV'03)*, volume 2, pages 1236–1242, 2003.
- [7] S. Dubois, R. Péteri, and M. Ménard. A 3d discrete curvelet based method for segmenting dynamic textures. In *16th International Conference on Image Processing (ICIP'09)*, pages 1373–1376, Cairo, Egypt, November 2009.
- [8] S. Dubois, R. Péteri, and M. Ménard. A comparison of wavelet based spatio-temporal decomposition methods for dynamic texture recognition. In *4th Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA '09)*, volume 5524, pages 314–321, Pova de Varzim, Portugal, 2009.
- [9] J. Filip, M. Haindl, and D. Chetverikov. Fast synthesis of dynamic colour textures. In *Proceedings of the 18th IAPR Int. Conf. on Pattern Recognition (ICPR'06)*, pages 25–28, Hong Kong, 2006.
- [10] M. Finch. *GPU Gems : Programming Techniques, Tips, and Tricks for Real-Time Graphics, Chap.1*. Randima Fernando.
- [11] R. Péteri, M. Huiskes, and S. Fazekas. Dyntex : A comprehensive database of dynamic textures. <http://www.cwi.nl/projects/dyntex/>.
- [12] P. Saisan, G. Doretto, Y. N. Wu, and S. Soatto. Dynamic texture recognition. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'01)*, volume 2, pages 58–63, Kauai, Hawaii, December 2001.
- [13] J.-L. Starck, M. Elad, and D. Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE Transactions on Image Processing*, 14 :1570–1582, 2004.
- [14] J.-L. Starck, M. Elad, and D. Donoho. Redundant multiscale transforms and their application for morphological component analysis. *Advances in Imaging and Electron Physics*, 132, 2004.

- [15] M. Szummer and R. W. Picard. Temporal texture modeling. In *Proceedings of IEEE International Conference on Image Processing (ICIP'96)*, volume 3, pages 823–826, 1996.
- [16] G. Zhao and M. Pietikainen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence journal (TPAMI'07)*, 6(29) :915–928, 2007.